Vignesh Sampath

Deep Learning Algorithms in Industry 4.0; Application of Surface defect inspection for quality control

Director/es

Maurtua Ormaetxea, Iñaki
Aguilar Martín, Juan José

ISSN 2254-7606

Prensas de la Universidad
Universidad Zaragoza

Servicio de Publicaciones

ISSN 2254-7606

Tesis Doctoral

DEEP LEARNING ALGORITHMS IN INDUSTRY 4.0; APPLICATION OF SURFACE DEFECT INSPECTION FOR QUALITY CONTROL

Autor

Vignesh Sampath

Director/es

Maurtua Ormaetxea, Iñaki
Aguilar Martín, Juan José

UNIVERSIDAD DE ZARAGOZA
Escuela de Doctorado

Programa de Doctorado en Ingeniería de Diseño y Fabricación

2023

Universidad Zaragoza
1542

TESIS DOCTORAL

Deep Learning Algorithms in Industry 4.0; Application of Surface defect inspection for quality control

Vignesh Sampath

Zaragoza, 2023

1542

Universidad
Zaragoza

Deep Learning Algorithms in Industry 4.0; Application of Surface defect inspection for quality control

Vignesh Sampath

Dirigida por

Dr. Iñaki Maurtua

Dr. Juan José Aguilar Martín

Para la obtención del Título de Doctor

por la Universidad de Zaragoza

Zaragoza, Abril de 2023

Esta tesis se presenta como un compendio de las siguientes publicaciones:

- V. Sampath, I. Maurtua, J. J. Aguilar Martín, and A. Gutierrez, ‘‘A survey on generative adversarial networks for imbalance problems in computer vision tasks,’’ J. Big Data, vol. 8, no. 1, pp. 1–59, Dec. 2021.
- V. Sampath, I. Maurtua, J. J. A. Martín, A. Rivera, J. Molina and A. Gutierrez, "Attention Guided Multi-Task Learning for Surface defect identification," in IEEE Transactions on Industrial Informatics, doi: 10.1109/TII.2023.3234030.
- V. Sampath, I. Maurtua, J. J. Aguilar Martín, A. Iriondo, I. Lluvia, and G. Aizpurua, "Intraclass Image Augmentation for Defect Detection Using Generative Adversarial Neural Networks," Sensors, vol. 23, no. 4, p. 1861, Feb. 2023, doi: 10.3390/s23041861.
- T. Chen, V. Sampath, M.C. May, S. Shan, O.J. Jorg, J.J. Aguilar Martin, F. Stamer, G. Fantoni, G. Tosello, M. Calaon, "Machine Learning in Manufacturing towards Industry 4.0: From 'For Now' to 'Four-Know,'" Applied Sciences, vol. 13, no. 3, p. 1903, Feb. 2023, doi: 10.3390/app13031903.
- V. Sampath, I. Maurtua, J. J. Aguilar Martín, A. Iriondo, I. Lluvia and A. Rivera, "Vision Transformer based knowledge distillation for fasteners defect detection," 2022 International Conference on Electrical, Computer and Energy Technologies (ICECET), 2022, pp. 1-6, doi: 10.1109/ICECET55527.2022.9872566.

Parte del trabajo realizado en esta tesis forma parte de los proyectos:

- This research work was undertaken in the context of DIGIMAN4.0 project ("Digital Manufacturing Technologies for Zero- defect", https ://www.digim an4-0.mek.dtu.dk/). DIGIMAN4.0 is a European Training Network supported by Horizon 2020, the EU Framework Programme for Research and Innovation (Project ID: 814225).
- This research was also partly supported by "5R- Red Cervera de Tecnologías robóticas en fabricación inteligente", contract number CER20211007, under "Centros Tecnológicos de Excelencia Cervera" programme funded by "The Centre for the Development of Industrial Technology (CDTI)".

D. Iñaki Maurtua, Doctor por la Universidad del País Vasco y Responsable de la Unidad de Sistemas Inteligentes y Autónomos del centro tecnológico TEKNIKER.

INFORMA:

Que la tesis titulada "Deep Learning Algorithms in Industry 4.0; Application of Surface defect inspection for quality control", elaborada por D. Vignesh Sampath, ha sido realizada bajo mi codirección, se ajusta al proyecto de tesis inicialmente presentado y cumple los requisitos exigidos por la legislación vigente para optar al grado de Doctor por la Universidad de Zaragoza. Una vez finalizada, autorizo su presentación en la modalidad de compendio de publicaciones para ser evaluada por el tribunal correspondiente.

Zaragoza, a 30 de Marzo de 2023,

Fdo. D. Iñaki Maurtua, Bilbao

D. Juan José Aguilar Martín, Doctor por la Universidad de Zaragoza y Profesor Titular del departamento de Ingeniería de Diseño y Fabricación de la Universidad de Zaragoza.

INFORMA:

Que la tesis titulada "Deep Learning Algorithms in Industry 4.0; Application of Surface defect inspection for quality control", elaborada por D. Vignesh sampath, ha sido realizada bajo mi dirección, se ajusta al proyecto de tesis inicialmente presentado y cumple los requisitos exigidos por la legislación vigente para optar al grado de Doctor por la Universidad de Zaragoza. Una vez finalizada, autorizo su presentación en la modalidad de compendio de publicaciones para ser evaluada por el tribunal correspondiente.

Zaragoza, a 28 de Marzo de 2023,

Fdo. D. Juan José Aguilar Martín

Statement of originality

The work contained in this thesis has not been previously submitted for a degree or diploma at any other higher education institution. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due references are made.

Vignesh Sampath

March of 2023

Abstract

Ferromagnetic parts are widely used in various industries such as automotive, aerospace, and machinery. The production of these parts involves several processes, including casting, forging, and machining. During these processes, defects can occur in the surface or near-surface of the parts. These defects can compromise the integrity of the parts, leading to potential failures in their performance. Therefore, it is crucial to detect and identify these defects before the parts are put into use. Magnetic particle inspection (MPI) is a non-destructive testing method widely used in the industry to detect surface and near-surface defects in ferromagnetic parts. In this method, magnetic particles are applied to the surface of the part, and a magnetic field is applied to the part. The magnetic particles accumulate around the defects, creating a visible indication of their presence. Qualified operators perform visual inspection of the parts to identify the defects. However, manual inspection by qualified operators can be time-consuming and error-prone. Moreover, the identification of defects can be subjective and dependent on the operator's experience and expertise. Therefore, there is a need to develop an automated method for defect identification in ferromagnetic parts based on the magnetic particle technique.

In this context, this PhD study aims to investigate and develop a deep learning based method for automatic defect identification in ferromagnetic parts based on the magnetic particle technique. Our proposed system for detecting defects in fasteners is based on the application of convolutional neural networks (CNNs). CNNs are a type of deep learning architecture that are particularly effective in image recognition tasks. They are able to learn complex features and patterns in images without the need for manual feature engineering or preprocessing. This makes them ideal for automating the detection of defects in manufacturing settings. The system is designed to process raw images of fasteners and output the presence and locations of defects. This is achieved through a training process where the CNN is fed a large number of labeled images of fasteners, some with defects and some without. The network learns to recognize the patterns and features associated with the presence of defects and is able to generalize this learning to new images of fasteners. CNNs are a powerful tool for computer vision and machine learning, but they face several challenges. One of these challenges is overfitting, where the model is too closely fit

to the training data and performs poorly on new data. Regularization techniques can be used to address this challenge. Additionally, limited data, imbalanced classes, or inconsistent labels can make training and evaluation difficult. There is also a growing demand for CNNs to be explainable, especially in applications such as manufacturing defect detection. Finally, there are often constraints on the size and complexity of CNN models, particularly when they need to be deployed on devices with limited resources or integrated into robotic systems.

The starting point of this PhD study is to design a reliable image acquisition system that combines both frame and line scan cameras to capture the head and shank portions of rotating fasteners. This methodology captures high-resolution images of both sections, allowing for detailed analysis of surface finish and dimensional tolerances, which can help identify potential defects. The use of both cameras at high speeds provides for efficient image acquisition while improving the accuracy and repeatability of the image analysis. The proposed methodology represents a significant advancement in the field of fastener inspection and quality control.

The second step employs a data-centric approach through the use of data augmentation and GAN-based synthetic images to expand the size and diversity of the training dataset. Combining the data-centric approach with the model-centric approach using multi-task learning improves the performance of the defect detection model by allowing it to learn from multiple sources of information and to generalize better to new tasks. The proposed multi-task learning model can handle multiple tasks simultaneously, making it more efficient and interpretable.

Finally, explainable AI techniques are used to make the defect detection model more interpretable and explainable, with GradCAM generating the most interpretable and explainable heatmaps. The combination of knowledge distillation, transfer learning, and fine-tuning is also used to improve the speed and accuracy of the model. Overall, the proposed methodology combines multiple techniques and approaches to improve the efficiency and effectiveness of the defect detection process, with potential applications in various industries.

Acknowledgement

Firstly, I would like to express my heartfelt gratitude and acknowledgement to my advisors, Dr. Iñaki Maurtua and Prof. Juan José Aguilar Martín, for their unwavering support throughout my PhD study and research. Their guidance, encouragement, and expertise in the field of industry 4.0 have been invaluable to me. They have provided me with the freedom to conduct independent research, and their belief in my abilities has been a great motivation throughout my PhD study. I am truly grateful for their mentorship and could not have asked for better advisors for my PhD study.

I would also like to extend my gratitude to Miguel Angel Pérez, Eneko Ugalde and Francisco Febrer, my managers at TEKNIKER, for their unwavering support and guidance throughout my research journey. Their advice, encouragement, and availability whenever I needed help were invaluable. Their contributions have been crucial to the success of this thesis, and I am honored to have had the opportunity to work under their leadership.

I am grateful to Jorge Molina, Aitor Gutierrez, Javier del Pozo, Gotzone Aizpurua at Tekniker and colleagues at Erreka fastening solutions for their invaluable contributions in helping me to acquire the image data for my research. Their extensive knowledge and expertise in the field of robotics have been crucial to the success of my project. Without their support and guidance, my research would not have been possible.

I would like to extend my sincere thanks to Ander Iriondo, Joseba Domingo, Andoni Rivera, Ander Ansuategi, Iker Lluvia, Brahim Ahmed, Xabi Sukia, Unai Mutilba, Jon Martin, Jon Lambarri, DIGIMAN and SAI colleagues for their continuous encouragement and the valuable knowledge and insights that they shared with me during my research journey. Their support and mentorship have played a significant role in shaping both my personal and professional growth. I am grateful for the time and effort that they have dedicated to helping me succeed and achieve my goals. It has been an honor to work with such exceptional colleagues, and their contributions have been essential to the success of my thesis.

I am also deeply grateful to my parents, Sampath and Savithri, for their unwavering love, support, and guidance throughout my life. Their encouragement, understanding, and emotional support have been a constant source of strength and motivation for me. I am grateful for their unshakable belief in me and for the sacrifices they have made to ensure my success. I want to express my heartfelt thanks to them for everything they have done for me, from the bottom of my heart.

Finally, I would like to express my heartfelt thanks to my other family members and friends who have supported me throughout this journey. I thank them for their faith in me, for the valuable feedback and discussions, and for the many moments of laughter and companionship that have helped me through this challenging and rewarding experience.

Contents

List of Figures

List of Tables

List of Abbreviations

AC	Alternating current
ACGAN	Auxiliary Classifier GAN
ASNT	American Society for Non-Destructive Testing
ASTM	American Society for Testing and Materials
BCE	Binary cross-entropy
BRIEF	Binary Robust Independent Elementary Features
cGAN	Conditional GAN
CNNs	Convolutional Neural Networks
D2GAN	Diverging Discriminator GAN
DC	Direct current
DCGAN	Deep Convolutional Generative Adversarial Networks
DSS	Data Security Standard
EFNDT	European Federation for Non-Destructive Testing
ET	Eddy current testing
FID	Fréchet Inception Distance
FN	False Negative
FOV	Field of view
FP	False Positive
FPN	Feature Pyramid Network
GAN	Generative Adversarial Neural Network
GAP	Global Average Pooling
GBP	Guided Backpropagation
GCP	Google Cloud Platform
GCS	Google Cloud Storage
GDPR	General Data Protection Regulation
GRAD CAM	Gradient-weighted Class Activation Mapping
GRAN	Generative Recurrent Adversarial Networks
info-GAN	Information Maximizing GAN
IoT	Internet of Things
IoU	Intersection-over-Union
IS	Inception Score
KD	Knowledge Distillation
KRL	KUKA Robot Language
LAPGAN	Laplacian Pyramid GANs
LPI	Liquid penetrant Inspection
MADGAN	Multi-scale Attention-based Discriminator GAN
mAP	Mean Average Precision
ML	Machine learning
MLP	Multi-Layer Perceptron
MPI	Magnetic particle Inspection
NDT	Nondestructive testing
OEMs	Original Equipment Manufacturers

PCA	Principal Component Analysis
PCI	Payment Card Industry
Pro-GAN	Progressive Growing of GANs
RD	Research and Development
RT	Radiographic Testing
SCGAN	Self-Conditioned GAN
SE	Squeeze and excite
SIFT	Scale Invariant Feature Transform
SinGAN	Single Image Non-Parametric GAN
SURF	Speeded-Up Robust Features
SVD	Singular value decomposition
TF-records	TensorFlow records
TN	True Negative
TP	True Positive
UT	Ultrasonic Testing
VACGAN	Variational Autoencoder GAN

1. INTRODUCTION

Introduction 1

The field of manufacturing has seen a significant shift towards Industry 4.0 [1], with the integration of advanced technologies such as the Internet of Things (IoT) [2], big data analytics [3], and machine learning (ML) [4] into the manufacturing process. Among these technologies, ML has been recognized as a key enabler of Industry 4.0, with the potential to improve manufacturing efficiency, quality, and flexibility. One area in which ML can be particularly beneficial is in the inspection and quality control of fasteners [5]. Fasteners are an essential component in many manufacturing processes, and their quality and integrity are critical for ensuring the performance and safety of the final product. However, automating the inspection process of fasteners using ML can be a challenging task, particularly when dealing with high-speed production processes and small, intricate components.

To tackle this challenge in this field of study, we present a solution methodology that integrates multiple approaches to develop a strong computer vision application for inspection and quality control of fasteners. Our proposed methodology involves the following steps:

- **Image Acquisition System:** The first step in our proposed methodology is to design a reliable image acquisition system for acquiring images from the fastener production process. We used a combination of both frame and line scan cameras in an image acquisition system to capture the head and shank portion of rotating fasteners respectively. This combination offers several advantages, including the ability to capture high-resolution images at high speeds, improved accuracy and repeatability, and the potential for improved efficiency and effectiveness in fastener inspection processes.
- **Data centric deep learning approach:** In this step, we used a data-centric approach to expand the size and diversity of the training dataset acquired in step 1. This is achieved through a combination of traditional data augmentation and Generative Adversarial Neural Network (GAN) based synthetic images. Data augmentation is used to artificially expand the size of the training dataset by generating new, synthesized samples that are variations of the original samples. This can help to improve the generalization ability of the model and reduce overfitting. In addition, GANs are used to generate synthetic images

of defects, which greatly expand the size and diversity of the training dataset, resulting in improved accuracy of the model.

- **Model centric deep learning approach:** Our proposed model-centric approach involves the use of deep learning models to analyze the images acquired in step 1 and 2. Our proposed model-centric approach focuses on utilizing models and incorporates the technique of multi-task learning, which trains a single model to perform multiple related tasks simultaneously. This can improve the performance of the defect detection model by allowing it to learn from multiple sources of information and generalize better to new tasks. Additionally, a multi-task learning model can be more interpretable than a single-task model, as it can provide insights into the shared features that are important for multiple tasks. This approach allows the model to learn from multiple sources of information and to generalize better to new tasks. Additionally, we employed attention mechanisms to leverage the most relevant features for the inspection task, resulting in improved accuracy and efficiency.
- **Explainable deep learning algorithms:** We utilized Explainable deep learning algorithms to analyze and understand the predictions provided by deep learning models. This can help quality inspectors to understand the reasoning behind the predictions and make sure that the deep learning model is focusing on the correct predictors for the task at hand.
- **Model compression:** We employed a combination of knowledge distillation and post-quantization methods to reduce the size and improve the speed of real-time inference for deep learning-based defect detection models on edge devices. This allows for faster and more efficient detection of defects in products.

With the integration of all these steps, we aim to develop a strong computer vision application for inspecting and controlling the quality of fasteners, thereby enhancing the efficiency and efficacy of fastener inspection procedures in the manufacturing sector. The proposed methodology is a significant advancement in the area of deep learning-based surface inspection and quality control, and has the potential to significantly enhance the quality and safety of manufactured products.

The remainder of this chapter focuses on the current state of the art techniques for fastener manufacturing in real-time industrial scenarios, as well as the methods used for quality inspection. This includes both manual quality inspections using destructive and non-destructive testing methods, as well as automated vision-based inspections utilizing traditional machine vision and deep learning-based approaches. The justification for this thesis is then thoroughly explained.

1.1 State of the art

Traditional factories are being transformed by the Industry 4.0 revolution into intelligent manufacturing environments with a high degree of autonomy and automation [6]. An increasing amount of flexibility is needed during the manufacturing processes, including the quality check. The need for flexibility is a result of the extensive range of product variations driven by consumer preferences and the dynamic variation in yearly demand [7]. The IOT is viewed as a key technology for Industry 4.0, which is a term used to describe the new industrial revolution that involves the digitalization of smart factories [8].

The quality control process is still based on manual operations in the manufacturing of fastener components. The term "fastener" refers to a broad range of screws, bolts, nuts, and other kinds of tools used to join and secure components together. The Fasteners are crucial for connecting two or more components together when designing structural systems. They are ubiquitous component of our world, used in everything from household appliances to airplanes. Among the numerous industries that depend on them are automotive manufacturing, aerospace, furniture, household appliance, building and construction, security, military and defense, electronics, and wind turbines. Fasteners come in many different varieties and are used in a wide range of commercial, industrial, and consumer product situations. The market for industrial fasteners was estimated at USD 88.43 billion in 2021 [9], and from 2022 to 2030, it is anticipated to rise at a compound annual growth rate of 4.5% [9]. The market is anticipated to be driven by the increased demand for industrial fasteners in the automotive and aerospace industries, as well as by the expanding population, significant investments in the construction industry, and rising construction spending.

To survive in a competitive environment, the fasteners manufacturing companies need to produce quality products with high productivity and low operation costs [10]. The quality of the fastener is an important factor in determining consumer satisfaction. To satisfy customers, ensuring the quality of manufactured products has always been essential [11]. As a result, many manufactures tend to evaluate the quality of their products from the standpoint of the customer rather than from their own. Massive product recalls involving high-end automobiles, electronic equipment, or household appliances serve as ominous reminders of the potential consequences of a lack of quality. Therefore, manufacturing firms create standardized procedures to guarantee the quality of their manufactured products in an effort to remain competitive [12]. These procedures are primarily incorporated into the production process. For instance, it might use a variety of inspection strategies or random testing using measurements and visual inspection.

The most fundamental and widely recognized method of inspection for finished goods in manufacturing quality control and asset maintenance is visual inspection. Visual inspection of the finished goods is carried out, once production is finished and all manufacturing processes are complete. To find any flaws before it gets into the hands of the clients, it is crucial to introduce inspections at this stage.The affected items should be removed if a defect is found at this stage. To stop the error from happening again, it is critical to determine its root cause. It will make it possible to keep low quality products off the market. It avoids the need for product recalls in the event that a problem is discovered too late. Furthermore, it prevents the loss of money and reputation that could result from a consumer complaining.

While there are some benefits to manual human inspection, particularly its high degree of flexibility to the many different types and sizes of inspected objects, it also has numerous significant drawbacks.The following are some of the main concerns that could arise when conducting manual inspection:

1. Human fatigue causes manual inspection quality to decline over time.
2. Humans are often highly sensitive to small defects.
3. High cost of quality inspectors' training.
4. It takes a lot of time and money to prepare final reports.
5. Since humans might make mistakes, manual testing cannot be expected to be more accurate.
6. Because humans can only focus on one or two verification points during manual testing, the scope of the test case is very limited.

Automated inspection has proven to be the finest solution for industries to rely on in order to get over these challenges. Greater workflow efficiency is made possible by automated quality assurance. By reducing costs associated with wasted materials and raising overall corporate productivity, it improves the bottom line. Additionally, it raises the standard and dependability of the complete inspection procedure. Compared to a manual human visual inspection, it speeds up the quality assurance and inspection procedure.

1.1.1 Manufacturing process of fasteners

Fasteners can be made in a broad variety of sizes and shapes, but the fundamental manufacturing method usually remains the same. Steel wire is first cold forged into the desired shape, then heat treated to increase strength and surface treated to increase durability, and then packaged for shipment [13]. For more sophisticated fasteners designs, the production process may need a few extra processes. In general, cold forging, hot forging, or machining are the three basic methods used to create fasteners. One approach may be superior to the others depending on the desired fastener type and the metal used to manufacture it. Figure 1.1 Illustrates the fastener manufacturing process.

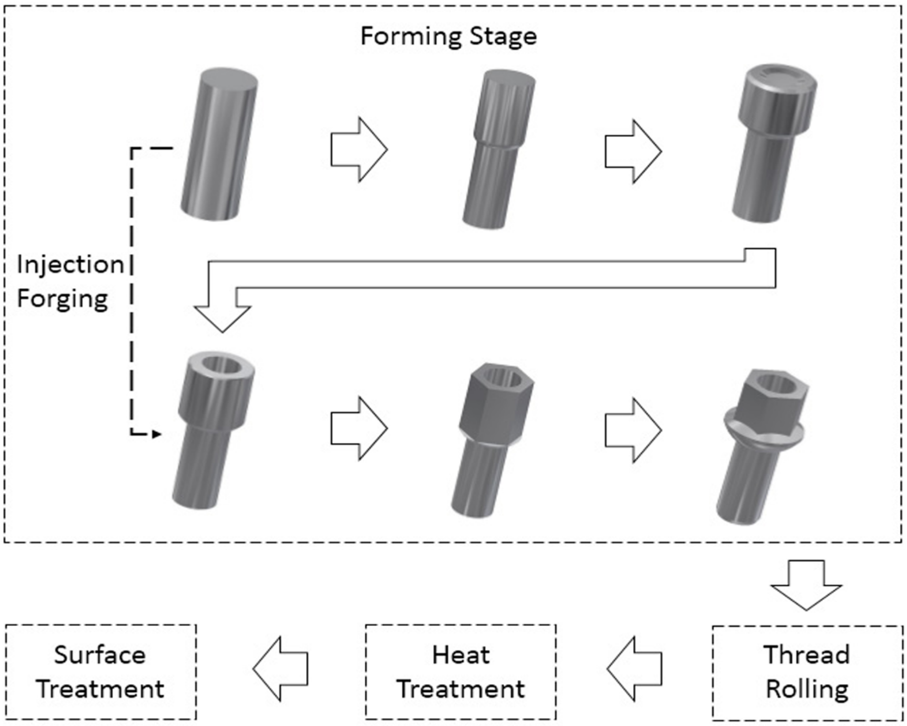


Fig. 1.1: Illustration of the fastener manufacturing process [13]

Cold forging: A variety of precision products, including torsion bars, brake components, engine valves, and fasteners, are produced using the cold forging method, also known as cold forming [14]. Extreme pressure is used in the procedure to accurately and permanently deform metal while it is still at room temperature. Large coils of wire produced by extrusion serve as the raw material, which is usually carbon or stainless steel. To correct any existing wire curvatures, the coils are subsequently put through a straightening operation while being mounted on uncoiling equipment. The wire is drawn into the cold forging process via a feeding device, also known as a feeder. The wire is aimed at the cut-off mechanism to be divided into distinct pieces known as blanks.

Additional turning or drilling may be required for more intricate bolt designs that cannot be contoured through cold forging alone. During turning, steel is cut away as the fastener is spinning at high speed to create the desired shape and design. Making holes through the bolt can be achieved by drilling.

Hot forging: Fasteners of large diameters, starting with a thread size of approximately M36 or greater, and lengths of approximately 300 mm or more, are typically produced by hot forging. In order to make the bar stock more malleable, it is heated to high temperatures before being fed into a forging press. The temperature of the process is determined by the bar material, geometry, and tolerances. Even complex shapes and high degrees of forming can be produced using this method [15]. Fasteners consisting of titanium alloys and nickel-based alloys are also most often produced using the hot forging method.

Thread forming: The thread is typically created in a thread rolling machine, where the pieces are placed between two flat dies, one of which is fixed and the other rotates, or between three rotating cylindrical dies. The surfaces of the dies include grooves that match the intended thread to be created [16]. Thread rolling is a cold forming method: It produces homogeneous, smooth, and exact exterior threads without affecting the integrity of the microstructure. As a result, it enhances a fastener's mechanical properties. Thread cutting can also be used to create a thread. Although cut threads can be produced to almost any specification, many manufacturers choose rolling threads instead because they are frequently smoother and more durable when handled.

Heat treatment: Fasteners are frequently subjected to heat treatments that alter their microstructure and, in turn, their physical characteristics, such as strength and ductility. Typically, the threading procedure is applied before heat treatment. The process steps rely on the fasteners' metallurgical properties. According to their carbon content, steel fasteners, for instance, are heated to a specified temperature and maintained there for a specific time [17]. The pieces are then quenched in water or oil to increase their toughness and strength. To produce better ductility with fewer microstructure distortions, the pieces are then reheated at a lower temperature. For instance, a heat treatment line for steel fasteners includes stations for cleaning, degreasing, hardening, quenching, cleaning, annealing, and dyeing. These lines are typically mesh belt furnaces where fasteners go through the various stages at a specific pace.

Surface treatment: Addition to heat tratment, in some cases, specific surface treatments can also be necessary. The choice of surface treatment process is determined by the application of fasteners and the specifications of the customer [15]. To enhance the characteristics of fasteners, for instance, specific coatings may be used. For example, self-drilling and tapping screws both use case-hardening.The screws are heated and held for a predetermined amount of time in a carbon-rich environment. As the carbon seeps through the surface, the amount of carbon in the area rises. The fasteners are then quenched, which causes them to harden. As a result, these fasteners have a very hard exterior while maintaining a ductile inside. Corrosion resistance is often the main issue with fasteners, thus an electrolytically applied zinc-plated coating is a typical remedy. In this procedure, a zinc-containing liquid is submerged around the fastener, and an electric current is then used to coat the fastener in zinc. The risk of hydrogen embrittlement is elevated during electrolytic treatment, though. Zinc flakes are a further choice that, despite costing more, provide even greater corrosion resistance.

Causes of Fastener Defects: Problems with faulty manufacturing process and material selection are equally likely to result in fastener failure [18]. Potential manufacturing errors include:

1. If the temperature rises over 700° C during heat treatment and the furnace's protective environment is insufficient, metal decarbonization may take place. Soft threads that could peel out may be the result of the metal decarbonization [19].
2. The fasteners should be tempered as soon as possible after being retrieved from the quench and before they get entirely cold during the quenching and drawing process [20]. Quenching crack and pre-mature failure could occur if this is not done.
3. It's crucial that the metal's grain flow lines form in the right direction during the head-forming process. Grain lines that move sharply in the direction of the head-to-shank junction struggle to maintain a healthy grain flow. Due to this, the fastener can be vulnerable to the head coming off during installation.
4. When torque is applied, threads that are rolled too closely to the head put the head under more stress. This can result in a failure from the head to the shank.
5. Internal hydrogen embrittlement happens when hydrogen, absorbed during the fastener plating process, becomes trapped in the steel and migrates to stress concentrations along grain boundaries. Under load, this might result in an abrupt catastrophic failure.
6. The majority of threaded fasteners are protected against rust and corrosion by plating or other protective coatings, and then they are tested for this resistance

using American Society for Testing and Materials (ASTM) B117 (salt spray fog test) [21]. The protective coating may deteriorate as a result of this process.

1.1.2 Quality Inspection of fasteners

The most crucial step in the production of fasteners is quality control [22]. By comparing a sample of the output to the specification, quality control is a process for preserving standards in manufactured goods. For quality control, every aspect of the product is carefully examined to determine whether there was quality throughout the entire production process. In essence, the main goal of quality control is to identify any defects and properly correct them. The best test procedures must be used by the industries that manufacture fasteners to analyze the fastener and provide a quantitative analysis of the fastener with supporting data in the form of elaborate test results. Industry 4.0 has received a lot of attention from academic researchers and industry professionals in recent years [23].To remain competitive in the consumer market in this era of Industry 4.0, manufacturing industries must produce goods and services of the greatest quality [24].

Supplies with ever-higher quality standards that are as close as possible to the "zero defect" aim is the main area of convergence for consumers and suppliers [25]. The "zero defect" objective implies for fasteners manufacturers a continuous effort to improve production processes through a very careful management in the production phases, in the phases of handling and storage of the fasteners, in the phases of "manufacturing process" and finished product quality control.

In this context, Original Equipment Manufacturers (OEMs) are frequently expected to deliver products that are guaranteed in accordance with strict standards for both the performance and quality of each individual component as well as the overall conformance of the supply. International standards like ISO 3269:2002 and UNI EN ISO 16421:2005 [26], which describe the requirements for quality assurance systems for fasteners and the test procedures for lot acceptance, respectively, are available for manufacturers and distributors to use.

The better the fasteners are made, the better they will perform. The fasteners performance from assembly to the lifespan of the product that are installed in depends on a variety of parameters, including its geometry, material, heat treatment, finish, and others. The requirements for a fastener quality assurance system are outlined in International Standard ISO 16426, and they must be met by distributors and producers of fasteners [27]. These criteria aim to limit or stop the manufacturing of non-conforming fasteners in order to get as close to zero defects for the given attributes.

Fasteners of poorer quality can be produced due to a variety of production faults, including the use of poor-quality raw materials, improper plating or heat treatment, wrong machining, irresponsible handling, and more. To assure a high-quality finished product, testing and inspection methods are necessary because many flaws and imperfections in fasteners and the raw materials used to make fasteners are not visible to the naked eye. The testing schedule or frequency and the kinds of testing needed vary greatly from company to company and are dependent on factors including: business type, industries served, intended use of fasteners, applicable standards; and customer requirement.

Testing of raw materials and/or manufactured fasteners is a common practice in product development, production procedures, and purchasing choices for many different types of organizations, including material suppliers, fastener manufacturers, distributors, fabricators, and end users. For every company to grow strong sales, a good reputation, and long-term success, this crucial aspect of quality control measures whether or not a particular benchmark or standard has been met. Standardization exists for the majority of industrial fasteners in terms of both dimensions and tolerances as well as materials and qualities [28].

Testing is occasionally done on samples taken from each batch of fasteners bought to be used in the assembly of certain products. Fasteners that will experience heavy wear or could pose a safety risk are typically checked often. Only when a material source or a production process is changed, or when fasteners used in less demanding situations are subject to testing to ensure everything is in working order. Other companies could only do testing if a client demands that they include a certification with the raw materials or fasteners they are purchasing.

From product development and Research and Development (R&D) to fastener failure or disposal, testing can be done at any stage of the product life cycle. Testing can generally be classified as either **destructive** or **nondestructive testing (NDT)** [29]. NDT won't damage the test sample, rendering it unusable, whereas destructive testing will.

1.1.2.1 Destructive Testing

Destructive testing is a type of testing that examines the point at which a fastener fails. In order to learn how a fastener responds to pressure, inspectors subject the fastener they are testing to several destructive test procedures that will cause the fastener to distort or entirely destroy it. The physical characteristics of a fastener, such as its toughness, hardness, flexibility, and strength, can be determined through destructive testing techniques [30].

Testing Mechanical Properties: It is crucial to understand whether a material's mechanical properties are appropriate for the intended application of the fastener before using it to make fasteners. The various mechanical testing methods examine the qualities of manufactured fasteners or raw material specimens under compression and tension at various temperatures. These tests include hardness testing, hydrogen embrittlement, axial tensile, wedge tensile, proof-load, cone strip, stress rupture, yield strength, and more. Testing can assess how heat treatments affect a material's mechanical properties. Tensile, yield, and elongation tests can establish whether the desired improvements have been made after heat treating [30].

Testing Material Characteristics and Structure: Determining the composition of materials, identifying materials and locating impurities are all made possible by chemical analysis. Testing can be done to compare raw materials when choosing materials or to confirm a new material when changing suppliers. Additionally, there are circumstances in which testing samples of finished fasteners is important to confirm the composition, such as when a product is being replicated and it is uncertain what the original material was or when a failure needs to be looked into [31].

A detailed analysis of a material sample or fastener using microscopic techniques can also be provided by metallurgical evaluation in order to examine the structure and characteristics or find flaws. Information about coating and plating thickness, decarburization, micro-hardness, and micro-structure can be obtained via optical magnification.

Testing can also be used to determine how certain environmental factors, such as humidity and salt spray, affect a material.

1.1.2.2 Non-Destructive Testing

NDT is the process of testing for flaws or changes in characteristics in materials, components, or assemblies while preserving the serviceability of the part or system. To put it another way, the part is still usable once the inspection or test is finished [29].

The two main goals of NDT are as follows:

1. Quality Control – Before it is used in service

2. Maintenance and health monitoring - While it is in service

NDT techniques can be broadly divided into two categories: 1. surface NDT and 2. bulk NDT, depending on whether the flaw or defect is present on the component's surface or volume (bulk) [32]. The surface NDT approach covers a number of techniques, such as:

1. Visual-Optical Inspection
2. Liquid penetrant Inspection (LPI)
3. Magnetic particle Inspection (MPI)
4. Eddy current testing (ET)

The bulk NDT approach, on the other hand, includes techniques like Radiographic Testing (RT), Ultrasonic Testing (UT) and Acoustic emission testing.

Visual-Optical Inspection: One of the fundamental inspection techniques used before applying any NDT techniques is visual optical testing. Visual inspection involves visually inspecting a component's exterior surface in a well-lit environment in order to find defects [33]. Magnifying glasses and other optical tools can occasionally be used to improve visibility of the component's surface. If the component is tiny, a light microscope with modest magnification can be utilized to improve the visibility of the component surface. The inspection of sheet material surfaces is done using high-speed visual inspection with automated output, and machine vision techniques may make use of improved picture and pattern recognition algorithms. It is also possible to take photos of surfaces that are inaccessible from a distance, like those inside a radioactive environment. In order to examine quick events, high-speed camera can also be employed. If it is necessary to observe something that is below the surface or that lies beneath, it is preferable to utilize one of other NDT methods as visual optic has its own limitations in that it can only perform limited external inspection. Figure 1.2 illustrates the design of a visual inspection system for inspecting the manufactured fasteners.

Liquid Penetrant Inspection: LPI, commonly known as dye penetrant testing, operates according to the principle of surface energy and capillary action. Both ferrous and non-ferrous materials can be processed using this technique [34]. When liquid is applied to the surface of a solid, one of two things can happen depending on the interaction or surface energies: either the liquid will spread over the solid surface, or in other words, it will wet the surface, or it won't spread at all. Wetting of the surface depends on surface tension, interfacial tension, wetting contact angle, energy of adhesion (surface energy). The liquids employed as penetrants are often colored dyes, most frequently red. The contact angle between the surface of the

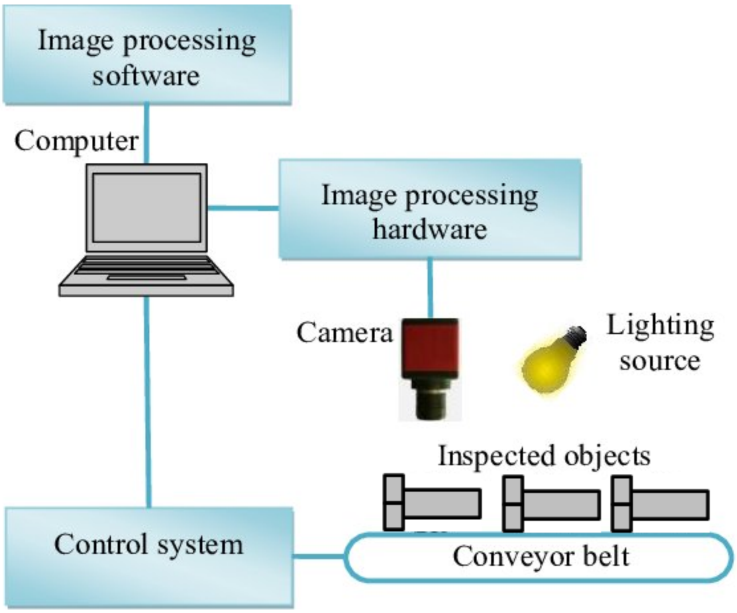


Fig. 1.2: The design of a visual inspection system [33]

component being tested and the liquid dye must be less than 90 degrees in order for this color dye to spread over the solid surface of the component being tested. The surface has to be clean for that to happen. Surface cleaning is therefore the first step in this procedure. Cleaning is required to remove all impurities from the surface of the component that will be tested. Grease, oil, or scale that may have built up on the surface could all be considered pollutants. All of these are likely to make the contact angle greater. A clean surface makes it more likely that the contact angle will be smaller than 90 degrees and that the liquid dye will spread across the solid surface. Due to its low viscosity, the liquid dye can penetrate flaws and cracks once it has spread throughout the surface. The third step is to carefully rinse the component surface with water following a period of dwell time of several minutes. Water fully eliminates the penetrant from the surface, yet it still remains in the crack. The test piece is now dried, and the fourth stage involves applying a developer to the surface. The developer, a fine-grained white powder, dissolves in a liquid. It covers the surface evenly, forming a coating. Once it has dried, the crack's penetrant is drawn out onto the surface. The surface clearly shows where the crack is located. Figure 1.3 illustrates the principle of the LPI technique.

Eddy current testing: ET uses electromagnetic induction as its basis to find defects in conductive materials. On the surface of a conductor, induced currents are created when the magnetic flux flowing through it changes. Eddy currents are those currents that flow in a direction that is opposite to the magnetic flux [35]. Eddy current generation is the first step in this testing process. A probe is utilized to produce

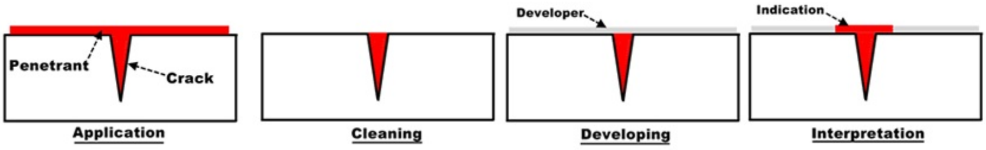


Fig. 1.3: Principle of the LPI technique [34]

the eddy currents necessary for an inspection. An electrical conductor that has been wound into a length inside the probe is the probe. In order to create an oscillating magnetic field, an alternating current is fed through the coil. A metal test piece is positioned close to the probe and its magnetic field. A circular flow of Eddy current will begin to move through the metal. Eddy currents flowing in the material will generate their own secondary magnetic field which will oppose the coil's primary magnetic field over there. Changes in metal thickness or defects like near surface cracking will interrupt or alter the amplitude and pattern of the eddy current and thus resulting in the magnetic field. This in turn affects movement of electrons in coil by varying the electrical impedance of the coil itself. Eddy current instrument plots changes in the impedance amplitude and phase angle, which can be used by a trained operator to identify changes in the test piece. This method is widely used in the aerospace industry and in other manufacturing and service environments that require inspection of thin metal for potential safety-related or quality related-problems. Figure 1.4 illustrates the principle of the ET technique.

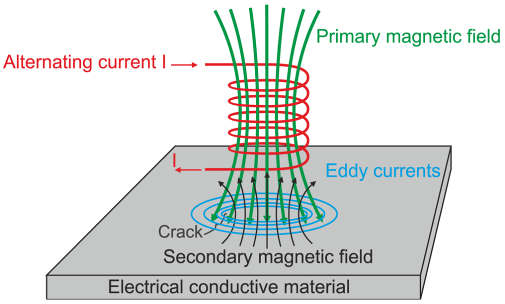


Fig. 1.4: Principle of the ET technique [35]

Radiographic Testing: RT technique is used to determine internal flaws in ferrous, non-ferrous metals and other materials [36]. In this method the components are exposed to short wavelength radiations in the form of x-rays where wavelength is less than 0.001 x 10^{-8} cm to about 40 x 10^{-8} cm. Gamma-rays whose wavelength is about 0.005 x 10^{-8} cm to about 3 x 10^{-8} cm from a suitable source such as an x-ray tube or cobalt-60 can also be used instead of x-rays. Gamma rays which are produced by Radio isotopes such as cobalt-60 can penetrate through specimen and can inspect better thickness than x-rays. Specimen is tested by placing between the source of radiation and film. The film is used as a recording medium in radiography testing. Both sides of the film base are covered by protective coating and emulsion coating. The base is made of polyester and provides a transparent medium. The emulsion is a suspension of silver salts in gelatin. The emulsion coating increases quantity of radiation absorbed and the protective coating is done to safeguard the film externally. If there is a void or defect in the part, large radiation passes through the defect areas and the film appears to be darker, as shown in Figure 1.5.

The primary benefit of radiography testing is that it is ideal for identifying flaws in thin sections and is applicable to many types of materials. The drawback of RT is that it is unsuitable for identifying flaws in the surface of specimen. Its inability to reveal the depth of a defect is another drawback. The main drawback of radiography test is that it puts people's health in danger.

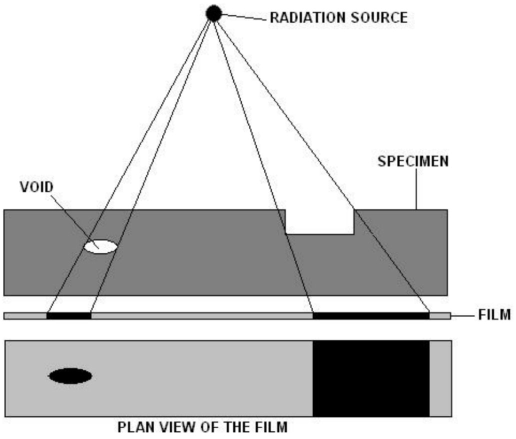


Fig. 1.5: Illustration of the RT Technique [36]

Ultrasonic Testing: High-frequency sound pulses are employed in UT to find internal flaws in the material [37]. This technique can test materials up to 30 feet in length and thickness. The components of an ultrasonic system include a probe with pulser and transducer, and display devices as shown in Figure 1.6. High voltage electric pulses utilized to operate the transducer are produced by the pulsar as part of its function. Afterward, the transducer produces high-frequency ultrasonic energy. Sound energy produced by transducer is directed on the specimen which is to be inspected for defects. If the sound wave approaches a crack in the specimen, it is reflected away from its path. As a result, the transducer transforms the reflected wave signal into an electrical signal that is displayed on the screen. The position, size, and kind of faults in the specimen are indicated by the degree of reflection and distortion.

The key benefit of this technique is that it may be completely automated and portable. Cast iron and other coarse-grained materials are difficult to evaluate using UT because of the limited sound transmission and significant signal noise.

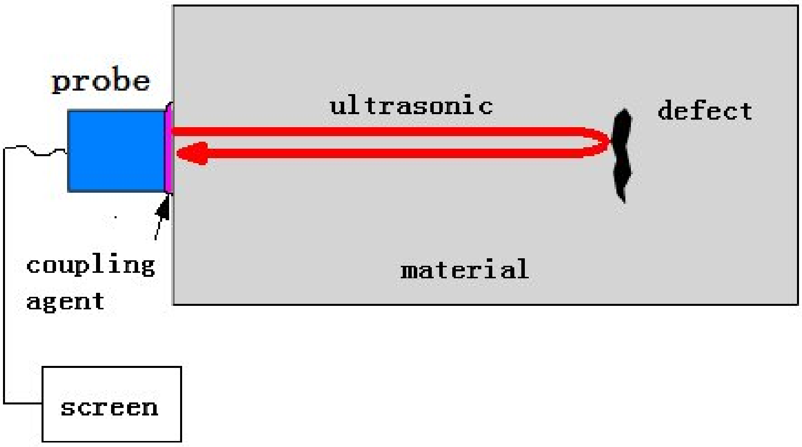


Fig. 1.6: Illustration of the UT Technique [37]

Acoustic emission testing: Acoustic emission testing is carried out by providing a localized external force to the component under test, such as a sudden mechanical load or a sharp change in temperature or pressure [38]. The resulting stress waves then cause tiny material displacements, or plastic deformation, on the surface of the component, which are monitored by sensors affixed to the component surface. These waves are short-lived, high frequency elastic waves. The collected data from many sensors can be analyzed to find discontinuities in the part.

Magnetic particle Inspection: MPI is a very sensitive non-destructive test method used to locate surface defects in ferromagnetic materials such as forgings, castings, weldments and machined or stamped parts [39]. The test pieces are cleaned before inspection with a solvent degreaser to remove all contaminations. After the cleaning solvent has evaporated, the parts to be inspected are brought into magnaflux horizontal bench units for inspection. The basic principle is to magnetize the parts to be inspected parallel to its surface. If the parts are free from defects the magnetic field lines run within the fastener and parallel to its surface. In case of magnetic inhomogeneity, for instance, near cracks, the magnetic field lines will locally leave the surface and a leakage field occurs. When a suspension of ferromagnetic particles is applied onto the test piece surface the magnetic particles will run off at defect free areas. In the places of leakage fields the magnetic particles are attracted and clustered together thus indicating the location of the defect. The surface defects can be visible under ultra violet light [40]. Figure 1.7 illustrates the principle of the magnetic particle testing technique.

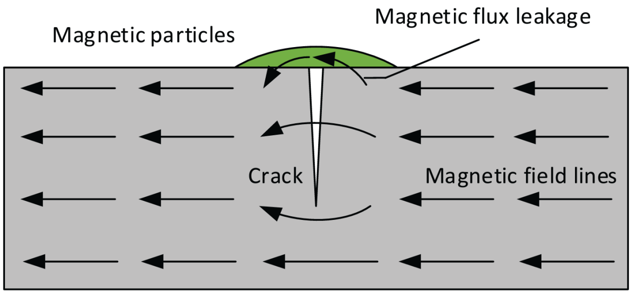


Fig. 1.7: Principle of the MPI technique [41]

1.1.3 Machine Vision based Quality Inspection

Machine Vision Inspection is an image processing-based technology that is used to automate inspection procedures in production lines in a variety of manufacturing industries. In essence, it enables the use of computer vision software, industrial cameras, and lighting to inspect and assess the quality of products. By releasing physical labor from time-consuming, difficult inspection chores and/or expanding the current quality control, the technology ensures product quality and increases overall efficiency. Machine vision inspection can also be utilized to offer production data and statistics, which can aid in production optimisation, by digitizing the inspection process and generating documenting data. It is widely and steadily being employed in manufacturing sectors due to its versatility and efficiency. Machine vision inspection for surface defect detection is include two categories: i) traditional approaches; ii) deep learning-based approaches.

1.1.3.1 Traditional approaches

The traditional method for defect detection uses well-established computer vision techniques such feature descriptors (Scale Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), Binary Robust Independent Elementary Features(BRIEF), etc.). For tasks like defect detection, a phase called feature extraction is used in this method. Features are discrete areas of "interesting," descriptive, or informative patches in images. Feature extraction step could utilize a number of computer vision methods, including threshold segmentation, corner detection, and edge detection. Images are processed to extract as many features as possible, and these features are then used to define each object class. During the deployment phase, these definitions are looked up in other images. The three primary categories of traditional machine vision methods for surface defect identification are statistical-based approaches, filter-based approaches, and model-based approaches.

Statistical-based approaches: Statistical feature extraction [42] is a method used in computer vision to extract features from images that can be used for image recognition and classification. It involves using statistical techniques, such as calculating the mean, median, mode, and standard deviation of the pixel values in an image, to extract meaningful features from the data. These features can then be used as input to a ML algorithm, which can learn to recognize patterns in the data and make predictions about the contents of new images. For example, a simple statistical feature extraction technique for an image with a defect might involve calculating the mean and standard deviation of the pixel values for the entire image, as well as for smaller regions of the image, such as the background, defect region, etc. These features can then be used as input to a ML algorithm, which can learn to identify faces in new images based on these statistical features.

For an automated defect identification system, Win et al. [43] suggested two thresholding methods: the contrast-adjusted Otsu's approach and the contrast-adjusted median-based Otsu's method. Contrast-adjusted Otsu's method is a variant of Otsu's method, which is a popular method for image thresholding, or the process of converting a grayscale image into a binary image (i.e., an image with only two colors). In traditional Otsu's method, the threshold value is determined by maximizing the interclass variance between the black and white pixels in the image. However, this method may not work well in images with low contrast, where the difference in intensity between the black and white pixels is not very pronounced. Contrast-adjusted Otsu's method addresses this issue by first applying a contrast-stretching transformation to the image, which increases the contrast between the black and white pixels. This makes it easier for Otsu's method to accurately determine the threshold value. After the threshold value is determined, the image is thresholded using this value, resulting in a binary image. Contrast-adjusted median-based Otsu's method is another variant of Otsu's method, which uses the median of the image histogram data is used instead of the mean.

Gray level co-occurrence matrix (GLCM) and HU invariant moments were utilized by Zhang et al. [44] to extract features, and an adaptive evolutionary algorithm was used to choose the best features. GLCM is a powerful tool for analyzing the spatial relationship between pixels in an image, and it can be used to extract useful texture features for a variety of image processing tasks. The Hu moments are derived from the central moments by taking linear combinations of the moments, which makes them even more invariant to image transformations. This makes the Hu moments a powerful tool for shape analysis, as they can accurately describe the shape of an object in an image, regardless of its position, orientation, or size. By applying weight on the local neighborhood, Chu et al. [45] proposed a smoothed local binary pattern.

For the purpose of describing outline features in steel surface defects, Hu et al. [46] used Fourier shape descriptors. Ricci et al. [47] used canny operator to detect the defect edges. A two-level labeling system based on super pixels was proposed by zhao et al.[48]. Using this technique, super pixels were first grouped into subregions and then into individual pixels.

Filter-based approaches: Traditional filter-based feature extraction for computer vision involves using hand-crafted filters, such as Gabor filters, Laplacian of Gaussian (LoG) filters, and Difference of Gaussian (DoG) filters, to extract features from images. These filters are designed to highlight specific patterns and structures in the image, such as edges, corners, and textures. The output of the convolution is a set of feature maps, which encode the detected features in a compact representation. These feature maps can be fed into a classifier or other downstream model to perform tasks such as object recognition and detection. Traditional filter-based approaches are effective at capturing local and spatial information from images, but may not be as robust and generalizable as deep learning-based approaches.

A Gabor filter combination is used in [49, 50] to find the minute holes in steel slabs. To find seam cracks in steel plates, Choi et al. [51] used two Gabor filters, which have a good detection capability and effectively cut down on noise. In order to locate the defects, Wu et al. [52] employed the modular maximum of the interscale correlation of the wavelet coefficient; for the classification of the defects, they then used prior knowledge of the properties of the surface defect faults. The kernel locality preserving projection and a non-subsampled shearlet transform were both used by Liu et al. [53] to detect surface defects. Akdemir et al. [54] used wavelet transforms to glass surface flaws identification.

Model-based approaches:

Statistical model based defect detection [55] is a method of identifying defects in a product or process using statistical analysis. This approach involves building a statistical model of the normal behavior of the product or process, and then using this model to identify deviations from the norm that may indicate the presence of a defect. The specific steps involved in this process can vary depending on the specific application, but generally involve the following:

1. Collecting data on the normal behavior of the product or process
2. Developing a statistical model of this normal behavior
3. Using this model to identify deviations from the norm
4. Investigating these deviations to determine if they are the result of a defect

One advantage of using statistical modeling for defect detection is that it can help identify defects that are not easily detectable using other methods, such as visual inspection. This can be particularly useful for identifying defects in complex systems or products, where it may be difficult to identify defects by simply looking at them. Additionally, this approach can help identify patterns in the data that may indicate the presence of a defect, which can be useful for identifying defects that occur with low frequency or that are difficult to predict.

1.1.3.2 Deep learning-based approaches

The challenge with the traditional method is that one must decide which elements in a particular image are crucial. Feature extraction becomes more difficult as there are more classes to categorize. The selection of which features best define various types of defects must be determined by the developer through judgment and extensive trial and error process.

A deep learning-based machine vision approach [56], on the other hand, involves using deep learning algorithms and models to enable machines to perceive and understand their environment. This approach typically involves training deep learning models on large datasets of images and other sensory data, and using the trained models to extract features and make predictions about the environment. Deep learning-based approaches are often more accurate and robust than traditional machine vision approaches, and can enable machines to perform complex tasks such as object detection and recognition, scene understanding, and motion planning.

According to Stephen Marsland [57], There are several learning methods in deep learning that can be broadly classified into four categories: supervised, unsupervised, reinforcement, and evolutionary.

- **Supervised Learning [58]:** Supervised learning is the most commonly used learning method in deep learning. It is called supervised learning because the algorithm is provided with labeled training data, where the correct output for each input is known. The goal of supervised learning is to train the deep learning algorithm to predict the correct output for a new, unseen input. In the context of deep learning, this usually means training a deep neural network to perform a specific task, such as image classification.
- **Unsupervised Learning [59]:** Unsupervised learning is a learning method where the algorithm is not provided with labeled training data. Instead, the goal of unsupervised learning is to learn the underlying structure of the data, without any guidance from external labels. This can be achieved through methods such as clustering, where the algorithm groups similar data points

together, or dimensionality reduction, where the algorithm learns to represent the data in a lower-dimensional space.

- **Reinforcement Learning [60]:** Reinforcement learning is a learning method where the algorithm learns by receiving feedback in the form of rewards or penalties. The algorithm takes actions in an environment and receives a reward or penalty based on the outcome of the action. The algorithm then adjusts its behavior based on the received feedback, in an attempt to maximize the cumulative reward over time. Reinforcement learning is often used in applications where the goal is to control a system, such as playing a video game or controlling a robot.
- **Evolutionary Learning [61]:** Evolutionary learning is a learning method that is inspired by the process of natural selection. The algorithm generates a population of potential solutions and evaluates their performance based on a given fitness function. The solutions that perform well are then combined to generate a new population, while the solutions that perform poorly are discarded. This process is repeated over multiple generations, with the goal of finding the best solution for the task at hand. Evolutionary learning is often used in optimization problems where the solution space is too large to be searched exhaustively.

Several other learning techniques have been proposed to address specific challenges in the deep learning process, such as reducing the amount of labeled data needed for training, improving the ability to learn from limited data, adapting to changes in the data distribution over time, and resisting adversarial attacks. By incorporating these techniques into deep learning models, researchers aim to improve the generalization and robustness of the models and make them more applicable to real-world scenarios. These techniques includes Active learning [62], Contrastive learning [63], Curriculum learning [64], Multi-task learning [65], Meta learning [66], Multiple instance learning [67], Few-shot learning [68], Transfer learning [69], Incremental learning [70], Adversarial learning [71], and many more.

- **Active Learning [62]:** Active learning is a learning technique that allows models to improve their performance by actively selecting the most informative examples from a large pool of unlabeled data. The goal of active learning is to reduce the amount of human annotation required to train a model and increase its accuracy by only training on the most valuable examples. It is particularly useful in situations where collecting labeled data is difficult, time-consuming or expensive. For example, in some manufacturing scenarios, annotating images can be a labor-intensive task that requires specialized knowledge and skill. In such cases, active learning can help to reduce the amount of data that needs to be annotated and improve the quality of the model.

- **Contrastive learning [63]:** Contrastive learning is an innovative and effective approach for improving the performance of ML models. The basic idea behind contrastive learning is to leverage the data itself to provide supervision and learn the representations that are useful for the task at hand. This is achieved by defining a contrastive loss function that optimizes the model to discriminate between similar and dissimilar pairs of data. The goal of this process is to learn representations that are discriminative and capture the important information in the data. One of the key advantages of contrastive learning is that it requires fewer labeled examples than traditional supervised learning. This makes it a valuable tool for domains where labeled data is scarce or expensive to obtain. In such scenarios, contrastive learning can be used to pre-train the model and fine-tune it with a smaller number of labeled examples. This has been shown to significantly improve the performance of the models and reduce the amount of labeled data required.
- **Curriculum learning [64]:** Curriculum learning is a technique that aims to improve the performance of models by carefully designing the learning process. Unlike traditional ML approaches, which treat all examples equally, curriculum learning takes into account the difficulty of the examples and learns in a sequential manner, starting with the easiest examples and gradually moving on to the more difficult ones. This approach is motivated by the observation that humans often learn new concepts in a similar way, starting with the basics and gradually moving on to the more complex topics. The basic idea behind curriculum learning is to design a learning curriculum, or a sequence of examples, that starts with easy examples and gradually moves on to the more difficult ones. The curriculum is designed in such a way that it provides a smooth and incremental learning experience for the model. The model is first trained on the easy examples, and as it improves its performance, it is gradually exposed to the more difficult examples. This way, the model is able to learn incrementally, and its performance improves with each step.
- **Multi-task learning [65]:** Multi-task learning is a method that involves training a single model to perform multiple tasks simultaneously. It is motivated by the idea that learning multiple related tasks can lead to better performance on each task compared to learning them independently. This is because learning multiple tasks can provide additional information and constraints that can improve the learning process, leading to a more general and robust model. Multi-task learning can be applied to a wide range of tasks, such as image classification, and reinforcement learning. For example, in image classification,

a multi-task model might be trained to perform both object recognition and semantic segmentation.

- **Meta learning [66]:** Meta learning, also known as learning to learn, is a subfield of ML that aims to develop models that can quickly and efficiently learn new tasks from a limited amount of data. It is a learning process that trains models to learn how to learn. The idea behind Meta learning is to develop models that can generalize well across a wide range of tasks and can quickly adapt to new tasks using only a few examples. It has its roots in cognitive science, where it has been studied as a way to understand how humans learn and generalize their knowledge to new situations. In recent years, Meta learning has gained increasing attention in the ML community, as it has the potential to revolutionize the way models are trained and deployed. Meta learning models are trained on a set of tasks, each of which is characterized by its own data distribution, loss function, and task-specific parameters. The goal of Meta learning is to learn a meta-representation that can be used to quickly adapt to new tasks. This meta-representation can be thought of as a set of initialization parameters or a prior that can be fine-tuned on the new task. There are several approaches to Meta learning, including 1. model-agnostic Meta learning (MAML) [72], 2. gradient-based meta learning [73], and 3. metric-based meta learning [74].
  1. MAML [72] is one of the most widely used approaches, where the model is trained to quickly adapt to new tasks using gradient descent. In this approach, the model starts with a set of initialization parameters, and then fine-tunes these parameters on each task in the training set. The goal is to find a set of initialization parameters that work well across a wide range of tasks.
  2. Metric-based Meta learning [74] is another approach that is based on learning a similarity metric between tasks. In this approach, the model learns a distance metric that can be used to determine how similar one task is to another. This similarity metric can be used to find the most similar task in the training set to the new task, and then fine-tune the model on that task.
  3. Gradient-based Meta learning [73] is a third approach that is based on using the gradients of the loss function with respect to the model parameters. In this approach, the model is trained to quickly adapt to new tasks by using the gradients of the loss function to update the model parameters. This approach has been shown to be very effective, as it allows the model to learn how to quickly adjust its parameters to new tasks.

- **Multiple instance learning [67]:** Multiple Instance learning (MIL) is a type of supervised learning where the data is represented as a set of instances rather than individual instances. The main difference between MIL and traditional supervised learning is that in traditional supervised learning, the label of an instance is provided, whereas in multiple instance learning, the label is provided for the set of instances. This makes the problem of MIL more challenging, as the model has to make predictions based on the aggregate information from the set of instances. MIL can be seen as an extension of traditional supervised learning, where the goal is to predict the label of a set of instances based on the collective information from these instances. It has become a popular research area due to its ability to handle the uncertainty and vagueness of the data. The uncertainty arises from the fact that the data is often represented as a set of instances, and it is not clear which instances are most relevant to the prediction. The vagueness arises from the fact that the label is provided for the set of instances, and it is not clear how to assign the label to individual instances. MIL can be divided into two categories: positive instance learning [75] and negative instance learning [75]. In positive instance learning, the goal is to predict the label of a set of instances that contain at least one positive instance, whereas in negative instance learning, the goal is to predict the label of a set of instances that contain no positive instances. Positive instance learning is used in applications such as image classification, where the goal is to classify an image as containing a particular object or not. Negative instance learning is used in applications such as defect detection, where the goal is to predict whether a set of images contain a defect or not.
- **Few-shot learning [68]:** Few-shot learning refers to a technique in which a model is trained to recognize and classify new objects based on only a few examples. The goal of few-shot learning is to enable models to learn and generalize from small amounts of data, thereby reducing the need for large labeled datasets and speeding up the development process. Few-shot learning algorithms can be broadly categorized into two groups: metric-based approaches [76] and model-based approaches [77]. Metric-based approaches [76], such as k-NN and Siamese networks, rely on computing similarity between the new examples and previously seen examples. Model-based approaches [77], such as Prototypical Networks and Matching Networks, learn a model that maps examples to a compact representation and classifies new examples based on this representation. One of the key challenges in few-shot learning is to effectively balance between memorizing previously seen examples and generalizing to unseen examples. Despite the progress made in few-shot learning, there are still several open challenges that need to be

addressed. One challenge is to extend the few-shot learning framework to real-world problems with large number of classes and diverse data distributions. Another challenge is to make the few-shot learning models more interpretable and explainable, so that their predictions can be trusted and acted upon.

- **Transfer learning [69]:** Transfer learning is a technique that utilizes knowledge learned from one task to solve another related task. It is a powerful tool that enables the reuse of knowledge gained from previously learned models, enabling faster and more effective learning in new tasks. The key characteristic of transfer learning is that it leverages information learned from a previous task to solve a new task more effectively. This information can be in the form of knowledge about the relationship between features and labels, the structure of the data, or the features themselves. This information can be used to reduce the amount of data and computation required to learn a new task, and improve the performance of the learning algorithm. There are many benefits to transfer learning. One of the primary benefits is that it enables the rapid development of models for new tasks. By leveraging knowledge learned from previous tasks, the time required to train a new model is reduced, and the performance of the model is improved. This makes transfer learning particularly useful for tasks that require a large amount of data to be processed, or for tasks that are difficult to learn from scratch. Another benefit of transfer learning is that it can help overcome the limitations of traditional ML methods. For example, traditional ML algorithms often require large amounts of data to achieve good performance. This can be a major challenge for many applications, especially for tasks where data is scarce or expensive to acquire. Transfer learning can help overcome this challenge by leveraging information from related tasks, where data is more abundant, to learn the new task.
- **Incremental learning [70]:** Incremental learning refers to the process of incrementally acquiring new information and continuously updating model's parameters over time. It differs from traditional approaches to learning where all the information is absorbed in one go, and is often used in the context of ML, where it is desirable to efficiently train models on very large datasets. In incremental learning, the learning process is divided into multiple iterations, and new information is incrementally added to the existing knowledge base in each iteration. This is done in an online manner, which means that the model receives one data sample at a time and updates its parameters immediately after processing each sample. This allows the model to adapt to changes in the data distribution and avoid overfitting. One of the key benefits of incremental learning is that it is computationally more efficient than batch learning. Batch learning requires the entire dataset to be processed in one go,

which can be slow and computationally intensive, especially for large datasets. In contrast, incremental learning processes data in small chunks, reducing the computational overhead and allowing the model to be trained on much larger datasets. Another advantage of incremental learning is that it is more memory-efficient than batch learning. Batch learning requires the entire dataset to be stored in memory, which can be challenging for large datasets that do not fit into memory. In incremental learning, only a small portion of the data is stored in memory at any given time, reducing the memory requirements and allowing the model to be trained on larger datasets.

- **Adversarial learning [71]:** Adversarial learning focuses on the development of algorithms capable of learning from data that contains adversarial examples. Adversarial examples are inputs that have been specifically crafted to cause a ML model to produce incorrect outputs. Adversarial learning is motivated by the fact that ML models, particularly deep neural networks, can be easily fooled by adversarial examples. This is because these models typically rely on statistical patterns in the input data to make predictions, and adversarial examples can manipulate these patterns to produce incorrect outputs. Adversarial learning can be divided into two main categories: adversarial training and adversarial defense. Adversarial training involves modifying the training process of a ML model to include adversarial examples, with the goal of making the model more robust to these examples. Adversarial defense involves developing methods to detect and defend against adversarial examples during the testing phase. Adversarial training is often performed using the fast gradient sign method (FGSM), which involves adding a small, targeted perturbation to each input example during training to cause the model to misclassify it. The perturbation is chosen such that it is as small as possible while still causing the model to make an error. This process can be repeated multiple times to train the model to be more robust to adversarial examples. Adversarial defense is typically achieved by detecting and correcting the adversarial perturbations in the input data. This can be done using methods such as denoising autoencoders, adversarial training, and robust optimization.

CNNs are a popular deep learning architecture used in computer vision and image processing tasks. The architecture of a CNN is composed of multiple layers, each with a specific function, that work together to extract relevant features from the input data and classify it accurately. They are primarily composed of three types of layers: Convolutional Layers, Activation Layers, and Pooling Layers.

1. **Convolutional layers [78]:** Convolutional layers are the building blocks of CNNs, and they are responsible for learning the features of an image. They are

inspired by the idea that visual features can be learned by sliding a small filter (also known as a kernel or a weight matrix) over the input image, element-wise multiplying the entries and summing them up to produce a new feature map. This process is known as convolution, and it allows the CNN to learn local patterns in the input image.The convolution operation involves the application of a set of filters on the input data to identify specific patterns or features. The filters are used to detect edges, shapes, and other relevant features in the image. The number of filters used in a CNN determines the number of feature maps generated, and each filter is responsible for detecting a specific type of feature in the image. The filters are initialized randomly, and they are updated during training through backpropagation, so that they can learn to detect the most important features in the training data. The convolution operation outputs a feature map, which is then processed by the activation layer.

2. **Activation Layers [79]:** Activation Layers are used to introduce non-linearity into CNNs, and they are responsible for allowing the network to learn complex relationships between features. The most common activation function used in CNNs is the ReLU (Rectified Linear Unit) function, which sets all negative values to zero, but there are other activation functions available such as the Sigmoid and Tanh functions.
3. **Pooling Layers [79]:** Pooling Layers are used to reduce the spatial dimensions of the feature maps generated by Convolutional Layers. They work by sliding a small window (also known as a pooling kernel) over the feature map, computing a summary statistic (such as maximum or average) for the values in the window, and then downsampling the feature map by replacing each window with its summary statistic. Pooling Layers serve two main purposes: first, they reduce the computational cost of the ConvNet by reducing the number of parameters that need to be learned, and second, they provide some degree of translational invariance, meaning that the CNN can recognize the same feature in different parts of the image even if it is not in exactly the same position. There are two common types of pooling layers: Max Pooling [80] and Average Pooling [81]. Max Pooling [80] selects the maximum value in each window, while Average Pooling [81] computes the average of the values in each window. Max Pooling is typically used in CNNs because it has been found to work better in practice.

To sum up, all these layers work together to allow the network to learn the features of an image and make accurate predictions. Convolutional Layers learn local patterns in the input image, Activation Layers introduce non-linearity into the network, and Pooling Layers reduce the computational cost of the network and provide translational invariance.

There are many different tasks that can be performed by CNNs, but some of the most important include classification, object detection, and segmentation.

- **Classification:** Classification is the task of assigning a label to an input data point, based on its characteristics. In the context of deep learning, this usually means training a deep neural network to classify images, such as assigning a label of "defect" or "non-defect" to an image of an manufactured parts. Several popular image classification architectures such as LeNet [82], AlexNet [83], VGG-16 [84], GoogLeNet [85], ResNet [86], Inception-V3 [87], and DenseNet [88] have been used to achieve state-of-the-art results in various image classification tasks.
- **Object Detection:** Object detection is the task of locating and identifying objects within an image. This is typically achieved by training a deep neural network to detect objects by using bounding boxes around the objects. The deep learning algorithm is trained on a large dataset of images, where the objects of interest are labeled with bounding boxes.This bounding box can be understood as a set of coordinates that define the box. Currently, there are two main categories of object detection algorithms: two-stage detectors and single stage detectors. Two-stage detectors, such as Region-based CNNs (R-CNN) [89], Fast R-CNN [90], Faster R-CNN [91], Mask R-CNN [92], etc., first use a Region Proposal Network (RPN) to identify objects, and then classify the objects and perform bounding-box regression in a second stage. In contrast, single stage detectors, such as Single Shot Detection (SSD) [93] and You Only Look Once (YOLO) [94], detect objects directly on a grid to save time on generating region proposals.
- **Segmentation:** Segmentation is the task of dividing an image into multiple segments, where each segment corresponds to a different object or region of the image. This is typically achieved by training a deep neural network to predict a segmentation mask for an image, where each pixel of the mask is assigned a label corresponding to the object or region it belongs to. A few well-known image segmentation algorithms include the Fully Connected Network [95], SegNet [96], U-Net [97], ResUNet [98], among others. There are two main types of image segmentation: 1. instance segmentation [99] and 2. semantic segmentation [100].
  1. Instance segmentation [99] is a task that involves identifying and segmenting individual objects within an image, and assigning a unique label to each instance. This type of segmentation is particularly useful

for tasks such as object detection, where the goal is to locate and classify each object in the image.

2. Semantic segmentation [100], on the other hand, is a task that involves assigning a semantic label to each pixel in the image, such that all pixels with the same label belong to the same object or region of interest. This type of segmentation is useful for tasks such as scene understanding, where the goal is to understand the overall structure and context of the scene.

CNNs has been used by several researchers to find solutions to the industrial defect detection problem. In [101], the strip steel surface defect was classified using a semi-supervised method based on CNN. Natarajan et al. [102] employed transfer learning to extract the multi-level features from the industrial defect images and then input these features into SVM classifiers to avoid the over fitting brought on by small samples. A Multi-scale pyramidal pooling network was suggested by Masci et al. [103] for the classification of generic steel defects. He et al. [104] suggested a multi-group CNN (MG-CNN) to examine the defects of the steel surface. An end-to-end defect detection framework with multi-level characteristics was proposed in [105] to fully detect the strip steel surface defect. The network's output identified the defect locations using some dense bounding boxes and assigned them a category name. For the purpose of detecting surface defects on strip steel, Kou et al. [106] developed an end-to-end defect detection model based on YOLO-V3. To achieve image classification and defect segmentation in [107], a pretrained deep learning network is employed to extract multi-scale features from raw image patches. For the purpose of detecting texture surface defects, a multi-scale feature-clustering-based fully convolutional algorithm was presented in [108]. A multibranch U-Net was proposed by Neven et al. [109] for segmenting steel surface defect type and severity. For the salient object recognition of strip steel surface defects, Zhou et al. [110] developed an edgeaware multi-level interactive network. Encoder-decoder residual networks were used by Song et al. [111] to identify conspicuous objects in strip steel surface defects. To automate the surface defect segmentation, Dong et al. [112] presented a global context attention network and pyramid feature fusion. Although these approaches achieved exceptional performance in the flaws identification, they still need to be improved from different aspects.

1.2 Thesis motivation

Fasteners come in a wide range of sizes and shapes depending on the usage. As a result, MPI is the testing technique that is most frequently utilized to identify fastener defects out of all the non-destructive procedures. Other advantages of using MPI are its speed and portability. There is no need for a regular pre-cleaning schedule, and post-cleaning is usually unnecessary. MPI is often affordable, and doesn't require a thorough pre-cleaning of a fastener. MPI has a great sensitivity and is capable of identifying small, superficial surface defects. It is simple to operate and doesn't require extensive training. Due to MPI's extreme flexibility, it may examine components with atypical forms (external splines, crankshafts, connecting rods, etc.) as shown in Figure 1.8 .

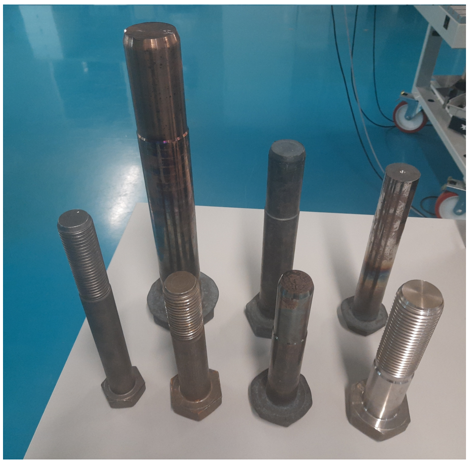


Fig. 1.8: An illustration of the different sizes and shapes of fasteners that are suited to various purposes

Therefore, at the Smart and Autonomous System Unit of the Tekniker (Spain), Universal WE Magnetic test bench (Figure 1.9) designed by MagnaFlux company for MPI was assembled. Then it was transferred to Erreka Fastening company (Spain) for use in mass production. It was assembled to completely magnetise the surface of fasteners up to 889 mm long in a single shot and designed to detect surface cracks at high speeds. Potentially cutting inspection time in half because the part can inspected in both directions at once. It was designed for high volume inspection environments to inspect for surface indication such as Fasteners manufacturing industries. The large surface shower automatically baths the entire part, further speeding up the inspection process.

At the Erreka Fastening company, traditionally, the MPI has been done manually by skilled inspectors, who visually inspect the products for defects (Figure 1.10). However, manual inspection is time-consuming and subject to human error, and it becomes infeasible for high-volume production lines. Therefore, there is a need for automated defect detection systems that can quickly and accurately identify defects in real-time.

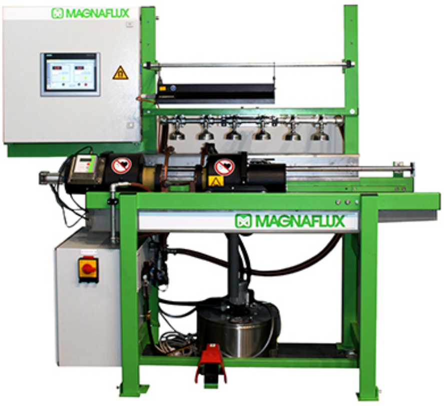


Fig. 1.9: The MPI machine

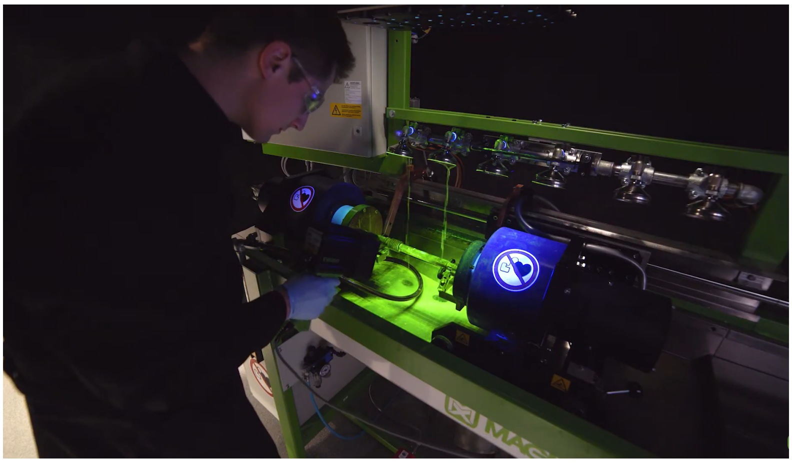


Fig. 1.10: A figure of the MPI performed by the skilled inspector

Conventional Computer vision method, the ability of computers to interpret and understand visual data, has the potential to revolutionize defect detection in manufacturing. By using ML algorithms trained on images of defective and non-defective products, a computer vision system can learn to recognize defects automatically. However, Conventional approaches to defect detection using computer vision often require significant manual effort in preprocessing and feature engineering, and they do not fully exploit the potential of modern deep learning techniques.

To address these challenges, this thesis aims to develop an end-to-end computer vision application for manufacturing defect detection that is easy to use and requires minimal manual effort. The proposed system, based on CNNs, will be able to process raw images of fasteners and output the presence and locations of defects, without the need for manual feature engineering or preprocessing. By leveraging the power of deep learning, the system will be able to learn to recognize defects automatically and achieve high accuracy in a variety of manufacturing scenarios.

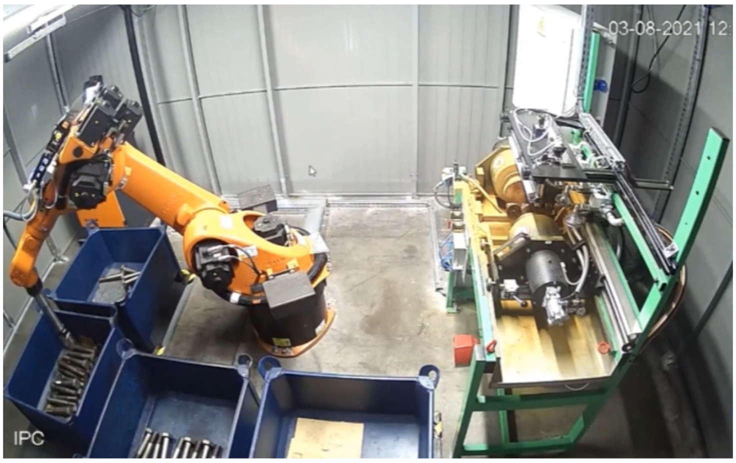


Fig. 1.11: A figure of the Pick-place robot picking fastener from the bin for inspection

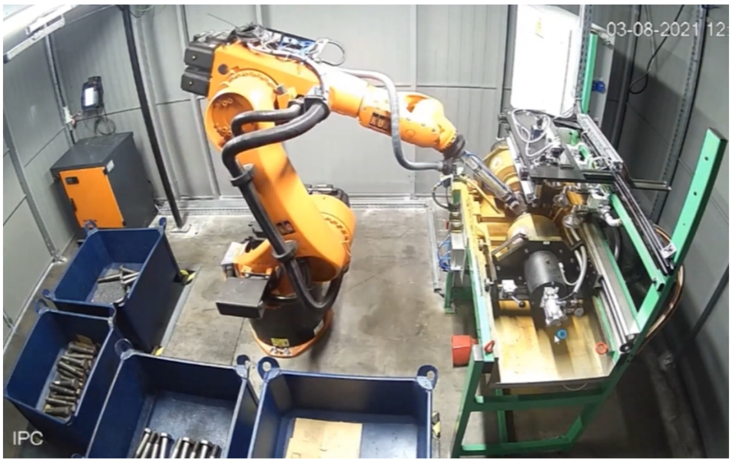


Fig. 1.12: A figure of the Pick-place robot performing MPI

CNNs have become a powerful tool for many tasks in computer vision and ML, but they are not without their challenges. One common problem is overfitting, where the CNN becomes too closely fit to the training data and performs poorly on unseen data. Another challenge is the need for regularization to prevent overfitting and improve the generalization performance of the model. In addition, many real-world applications require the use of CNNs with limited data, imbalanced classes, or inconsistent labels, which can make training and evaluation difficult.

Furthermore, there is a growing demand for CNNs to be explainable, especially in applications such as manufacturing defect detection, where it is important for humans to understand the basis for the model's predictions. Additionally, there are often constraints on the size and complexity of CNN models, particularly when they need to be deployed on devices with limited resources or integrated into robotic systems.

To address these challenges, this thesis aims to investigate and develop novel approaches to improving the performance, explainability, and efficiency of CNN models. Specifically, the research will focus on:

1. **Need for a novel CNN defect detection architecture:** One challenge that has consistently plagued CNN models is their tendency to overfit to the training data, leading to poor generalization performance on unseen data. This problem is especially prevalent in the field of defect detection, where the number of images with defects may be limited and the appearance of defects can vary significantly. Therefore, it is important to develop a novel CNN architecture for defect detection that is capable of preventing overfitting and improving the generalization performance of CNN models.
2. **Limited data and class imbalance problem:** Training CNNs can be challenging when faced with limited data, imbalanced classes, or inconsistent labels. These problems can lead to poor performance and generalization ability of the CNN model. Consequently, It is essential to develop certain methods that are more effective at addressing limited data, imbalanced classes, or inconsistent labels. Additionally, it is important to explore the trade-offs between these methods and evaluate their impact on the performance and generalization ability of the CNN model.
3. **Explaining the predictions of the CNN models:** CNN models are often considered "black boxes," making it difficult to understand how they make their predictions and which features they use to do so. This lack of transparency can be a major limitation, especially in situations where the CNN model is being used to make important decisions, such as in fasteners defect detection. As a result, it is also important to compare various techniques for explaining the predictions of CNN models and visualizing the features they use to make their decisions.
4. **CNN model compression:** State-of-the-art CNN models can be computationally intensive and require significant amounts of computational resources to run inference and storage, making them difficult to deploy on devices with limited resources, such as low power and IoT devices. In fasteners defect detection application, it is important to have real-time or near-real-time processing of image data. As a result, it is essential to develop an approach that can compress

CNN models to make them more efficient and suitable for deployment on these types of devices.

Overall, the goal of this research is to advance the state of the art in CNN models and enable their wider and more effective use in the case of automated defect detection applications. Finally, this work integrates pick-place robots (Figure 1.11 and Figure 1.12) and computer vision algorithms for automating MPI tasks that require precise positioning and handling of fasteners. In the context of defect detection, computer vision algorithms are used to identify defects in the fasteners being handled by the pick-place robot, and the robot can be programmed to handle these fasteners appropriately based on the presence or absence of defects.

2. PRESENTATION OF THE PUBLISHED WORK

2 Presentation of the published work

This doctoral thesis is presented as a compendium of publications, including five research articles, four of which have been published in international journals and the fifth in an international conference. The first part of this section provides a justification for the themes of the publications, and the second part summarizes each of the published works.

2.1 Justification of the thematic unity

The work presented in this thesis focuses on the the development of a CNN based defect detection model. The defect detection model requires a balanced approach that considers various factors such as explainability [113], light-weight [114], and both data-centric [115] and model-centric approaches [116]. By carefully considering these factors, it is possible to create a powerful and effective model that can accurately identify defects in a variety of contexts.

One of the main important considerations in the training of CNN-based detection models is the use of both data-centric approach [115] and model-centric approaches [116] iteratively. A data-centric approach for developing a CNN based defect detection model involves focusing on the quality and quantity of the data used to train the model. For example, it is important to have a diverse and representative set of images with defects to ensure that the model can accurately identify defects in a variety of contexts. It is also important to consider the annotation of the data, as the labels used to train the model should be accurate and consistent.

A model-centric approach [116], on the other hand, involves focusing on the design and architecture of the model itself. This can include experimenting with different layers, filters, and activation functions to see what works best for the specific task of defect detection. It is also important to consider the trade-offs between model accuracy and efficiency when designing the model. Both approaches are necessary for achieving optimal performance in CNN-based detection models. Using a well-designed model without sufficient high-quality training data can result in poor performance, while having large amounts of training data without a well-designed model can also lead to suboptimal results. By iteratively incorporating both approaches, the performance of the model can be continually improved until it reaches its full potential.

Another consideration in the development of a CNN based defect detection model is explainability. Explainability [113] refers to the ability of the model to provide reasoning and justification for its predictions. This is important in industrial applications where it is crucial to understand why a defect was identified or missed. One way to improve the explainability of a CNN based defect detection model is to use techniques such as saliency maps [117], which highlight the regions of the input that are most important for the model's prediction.

Another crucial final factor to take into account is the model's weight, or the amount of computation required to make predictions. A lightweight model is faster and more efficient, making it more practical for real-time applications. There are several ways to make a CNN based defect detection model lightweight, such as reducing the number of parameters in the model or using techniques such as pruning or quantization.

The research contributions of this thesis can be divided into three blocks: the data-centric approach, the model-centric approach, and the explainable and lightweight CNN model.

1. In the data-centric approach, we developed a Generative Adversarial Neural Network (GAN) model, named Magna-Defect-GAN, to create synthetic data as a solution to data scarcity. We also developed a novel augmentation approach of combining conventional image augmentation techniques with GAN-generated synthetic images.
2. In the model-centric approach, we developed a new deep learning architecture called Defect-aux-net, which is based on the concept of multi-task learning. Multi-task learning [65] involves training a single model to perform multiple related tasks simultaneously, in order to improve performance on all tasks.
3. In the third area of focus, we worked on creating explainable and lightweight CNN models. we used a technique called Gradient-weighted Class Activation Mapping (GRAD CAM) [117] to make the models more explainable, and used a combination of knowledge distillation (KD) and pruning techniques to make the models lighter and more efficient.

The following list includes the published articles that best demonstrate the research outcomes in these three areas:

- **Block 1:** The data-centric approach.
  - **Journal of Big Data 2021:** A survey on generative adversarial networks for imbalance problems in computer vision tasks
  - **Sensors 2023:** Intra-class image augmentation for defect detection using generative adversarial neural networks
- **Block 2:** The model-centric approach.
  - **Applied Sciences 2023:** Machine learning in manufacturing towards Industry 4.0: From ’For Now’ to ’Four-Know
  - **IEEE Transactions on Industrial Informatics 2023:** Attention Guided Multi-Task Learning for Surface defect identification
- **Block 3:** The explainable and light-weight CNN model.
  - **International Conference on Electrical, Computer and Energy Technologies (ICECET) 2022:** Vision Transformer based knowledge distillation for fasteners defect detection

Following the procurement and assembly of a magnaflux machine for magnetic particle inspection, we started our project by designing a custom image acquisition system. The goal of this system was to collect images of fastener with defect and then use CNN model to perform a defect detection task, identifying any abnormalities in the fasteners that we inspect. However, as we moved on to the preprocessing and model building steps, we encountered a significant challenge: our image data was highly imbalanced. Out of the 1000 parts that we inspected, only one would typically have a defect. This made it difficult for our CNN model to accurately detect defects [118], as it was not being presented with enough examples of positive cases to learn from. To address this issue, we turned to the most recent developments in GANs for addressing imbalance problems in image data. GANs [119] are a type of deep learning model that can generate synthetic images that are similar to real ones. By using GANs to create additional examples of defective parts, we tried to improve the balance of our image data and give our CNN model a better chance of success. In the first article, published in **Journal of Big Data 2021**, a survey on generative adversarial networks for addressing imbalance issues in computer vision tasks is thoroughly presented as the starting point for the research. In this article, we explored the use of GANs for addressing imbalanced datasets in computer vision tasks. We covered key concepts such as deep generative image models and GANs, and proposed a taxonomy for categorizing GAN-based techniques for addressing imbalance problems in computer vision tasks. These categories include image level imbalances in classification, object level imbalances in object detection, and pixel level imbalances in segmentation tasks. We also discuss the challenges and real-world implementations of using GANs for generating synthetic images to restore balance in imbalanced datasets and improve the performance of computer vision algorithms. There are several advantages of using GAN generated synthetic images [120] to solve the problem of data scarcity in deep learning.The synthetic images generated by GANs can be used to augment small or imbalanced datasets in order to improve the performance of deep learning models. For example, if a dataset for a computer vision task is small and imbalanced, the model trained on this dataset may not perform well due to the lack of sufficient data and the presence of imbalanced classes. By generating synthetic images using a GAN, it is possible to increase the size of the dataset and restore balance to the classes, which can improve the performance of the model.

In order to collect image data for building a CNN model to identify surface defects, we first prepared the surface of the fasteners according to the MPI standards and a suitable imaging device was selected to capture images of the surface. The surface was then examined using appropriate lighting conditions and imaging settings, and the resulting images were collected and labeled according to the presence or absence of defects and location of the defects. This labeling process involve manual annotation by quality control experts. The collected and labeled images can then be used to train a CNN model for surface defect identification. However, in order to build a defect detection model, data scarcity [121], diversity [122], and intra-class variations [123] can all pose challenges. Data scarcity [121] can make it difficult for the model to learn and generalize well, while low diversity [122] and large intra-class variations [123] can make it difficult for the model to accurately identify defects. To address these issues, in **Sensors 2023**, we presented a novel method by generating synthetic images using a GAN. The proposed Magna-Defect-GAN was trained on a collected defect dataset and was able to generate new synthetic images with large intra-class variations, which was then used to artificially increase the size of the training dataset and improve the performance of a defect identification model. We demonstrated that the proposed Magna-Defect-GAN model can generate realistic and high-resolution surface defect images and showed that this augmentation method can boost accuracy and be adapted to other surface defect identification models.

In **IEEE Transactions on Industrial Informatics 2023**, we presented a novel method for improving the performance of CNN based surface defect identification by leveraging auxiliary information beyond the primary labels. We proposed a deep learning model architecture called Defect-Aux-Net, which is based on multi-task learning [65] with attention mechanisms [124] and aims to exploit the rich additional information from related tasks in order to simultaneously improve the robustness and accuracy of the surface defect identification. Through experiments, we demonstrated that the proposed method can significantly improve the performance of state-of-the-art models, achieving an overall accuracy of 97.1%, Dice score of 0.926, and mean average Precision (mAP) of 0.762 on defect classification, segmentation, and detection tasks.

An application roadmap for ML in the manufacturing industry, with the goal of providing guidance and standards for developing ML solutions from ideation to deployment was presented in the **Applied Sciences 2023**. The roadmap is based on published research on the topic and includes two dimensions for formulating ML tasks (know-what, know-why, know-when, and know-how, and product, process, machine, and production) and an implementation pipeline starting from the early stages of ML solution development. Furthermore, we summarized available ML methods, including supervised [58], semi-supervised [125], unsupervised [59], and reinforcement methods [60], and discusses current challenges and future directions for ML applications in manufacturing. Finally we discussed the current challenges of using ML in manufacturing and suggest directions for future developments.

In real-time defect detection applications, both lightweight [114] and explainable CNN models [113] are highly important for a variety of reasons. Lightweight models [114] are often preferred for defect detection tasks because they can be more efficient in terms of computational resources and easier to deploy on low-power and resource-limited devices. This is especially important in industrial environments, where defect detection systems may need to operate in real-time with limited computational resources. Lightweight models [114] can also be more suitable for online learning and adaptation, as they can be more agile and easier to update with new data. Explainable CNNs, on the other hand, are designed to be transparent and interpretable, so that their decision-making processes can be understood and explained to humans. This is especially important in defect detection applications where the decisions made by the model have significant consequences. Explainable CNNs [117] can help to build trust and confidence in the model, by providing a way for humans to understand how the model is making decisions. They can also be useful for debugging and troubleshooting, and for improving the performance of the model. To achieve these two main goals, in **ICECET 2022**, we proposed a methodology that combines KD and pruning techniques to create a lightweight CNN model that is able to perform well on a defect detection task, while being small and efficient enough to be deployed on low-power and resource-limited devices. In addition, we employed GRADCAM [117] technique to create an explainable CNN model. It works by identifying the regions in the input image that are most important or relevant for the model's prediction, and visualizing these regions using a heatmap.

2.2 Summary of the Publications

The following is a brief summary of the research works that comprise this PhD study.

2.2.1 Journal of Big Data 2021: A survey on generative adversarial networks for imbalance problems in computer vision tasks

This article examines the recent developments GANs [119] based techniques for addressing imbalanced image data in computer vision tasks. Imbalanced data can lead to poor performance of defect detection algorithms and GANs [119] have gained attention as a potential solution due to their ability to model complex real-world image data. The paper covers the challenges and implementations of using GANs for synthetic image generation and proposes a taxonomy to categorize GAN-based techniques for addressing imbalances in computer vision tasks into three categories: image level imbalances in classification [82], object level imbalances in object detection [89], and pixel level imbalances in segmentation tasks [95]. The paper explains how GAN-based techniques can handle imbalanced data and improve the performance of computer vision algorithms.

Imbalanced data [118] can cause a model to be biased towards the majority class, leading to poor performance on the minority class and increased risk of overfitting. Data augmentation and GAN-based oversampling are two techniques that are commonly used to address the problem of imbalanced data in computer vision models.

In this PhD work, we used the combinations of both Data augmentation and GAN-based oversampling to mitigate the risk of overfitting and class imbalance problems. Data augmentation is a technique that artificially increases the size of a training dataset by applying various transformations to the original training examples. This can help to expose the model to a wider variety of inputs and reduce overfitting. One of the main advantages of data augmentation is that it can be applied to any type of data, including images [126], text [127], and time series data [128]. In image data, for example, data augmentation can be used to rotate [126], flip [129], zoom [126], or add noise [130] to images. This can help to expose the model to a wider variety of inputs, making it less likely to overfit and more robust to small perturbations in the input data. Another advantage of data augmentation is that it can be applied to data that is difficult or expensive to acquire. In cases where it is difficult or expensive to collect more data, data augmentation can be used to artificially increase the size of the training dataset, which can help to improve the generalization performance of the model.

In our case, acquiring high-quality and representative datasets for surface defect detection was a challenging task that was both difficult and expensive. The occurrence of surface defects was often very low, making it difficult to acquire a sufficient amount of data to train a computer vision model. Additionally, many surface defects were difficult to detect or reproduce, making it difficult to acquire data that was representative of real-world conditions. One of the main challenges in acquiring surface defect datasets was the low occurrence of defects. Surface defects were rare, making it difficult to collect a sufficient amount of data for training a computer model. For example, in fasteners manufacturing, the occurrence of surface defects can be as low as 0.1% or less. This made it difficult to collect a representative sample of the data, leading to a high risk of overfitting.

Another challenge is the collecting data for surface defects detection was expensive as it requires specialized equipment and trained personnel. For example, in the case of quality inspection of fasteners using MPI, collecting data requires the use of high-resolution imaging equipment and trained personnel to inspect the fasteners. This adds significant costs to the data collection process.

To overcome these challenges, we employed various data augmentation techniques to artificially increase the size of the dataset. By applying Feature Space Augmentation [131] and Data Space Augmentation [132], we were able to create a more diverse and informative dataset, which helped to improve the performance of our defect detection model. Despite the limited amount of data we had, we were able to achieve good results by using the data augmentation techniques.

Feature Space Augmentation [131] involves generating new features from the existing image data. This can be done by applying various mathematical operations to the existing image data, such as taking the Fourier transform or performing principal component analysis. By generating new features in this way, we created a more diverse and informative dataset, which helped to improve the performance of defect detection models.

Data Space Augmentation [132], on the other hand, involves applying various transformations to the data in order to generate new, artificially created data points. These transformations include techniques like Center crop, Horizontal flip, Rotation, Shear, Vertical flip, Translation, Noise injection, Color space transformations, Mixing images, Random erasing, Sharpness, Brightness, contrast, and Gaussian blur to image data.

Center crop [126]: Center crop is a technique used to artificially introduce variations in camera position to the training data. It involves cropping the image from the center, reducing the size of the image and removing the outer edges. This simulates the effect of the camera being positioned closer to the object being inspected, as well as removing any background information that may not be relevant to the detection of defects. Examples of Centre crop Augmentation are shown in Figure 2.1.

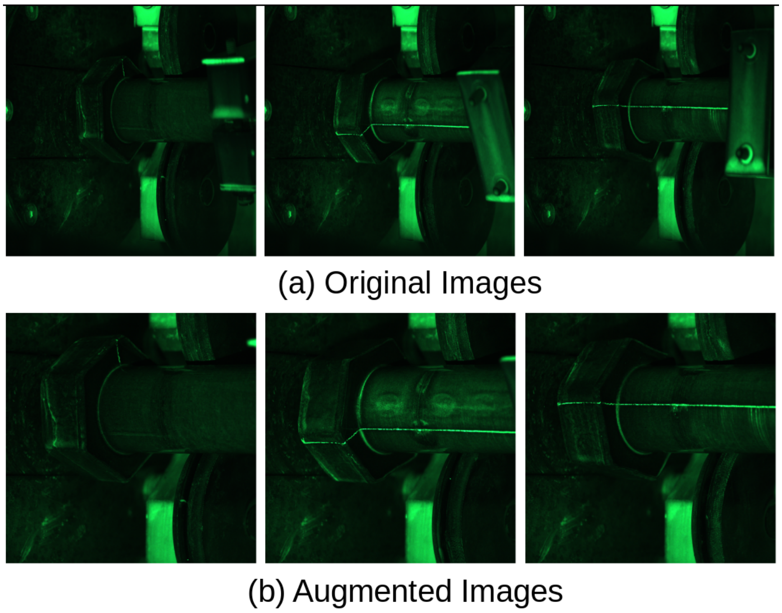
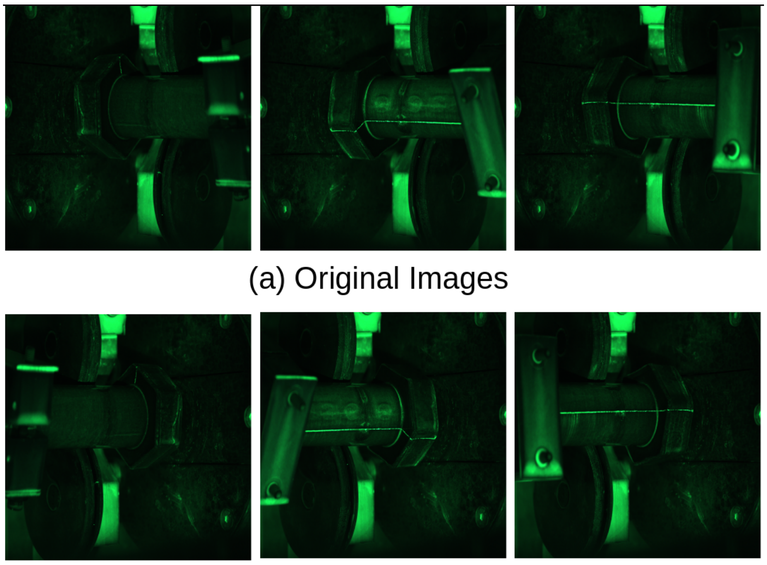


Fig. 2.1: Examples of Centre crop Augmentation

Horizontal flip [129]: Horizontal flip is a technique used to artificially introduce variations in camera position to the training data. It involves flipping the image horizontally, simulating the effect of the camera being positioned at a different angle from the object being inspected. This allows the model to learn to detect defects regardless of the angle at which the image was captured. Examples of Horizontal Flip Augmentation are shown in Figure 2.2.

Rotation [126]: Rotation is a technique used to artificially introduce variations in camera position to the training data. It involves rotating the image by a certain degree, simulating the effect of the camera being positioned at a different angle from the object being inspected. This allows the model to learn to detect defects regardless of the angle at which the image was captured. Examples of Rotation Augmentation are shown in Figure 2.3.

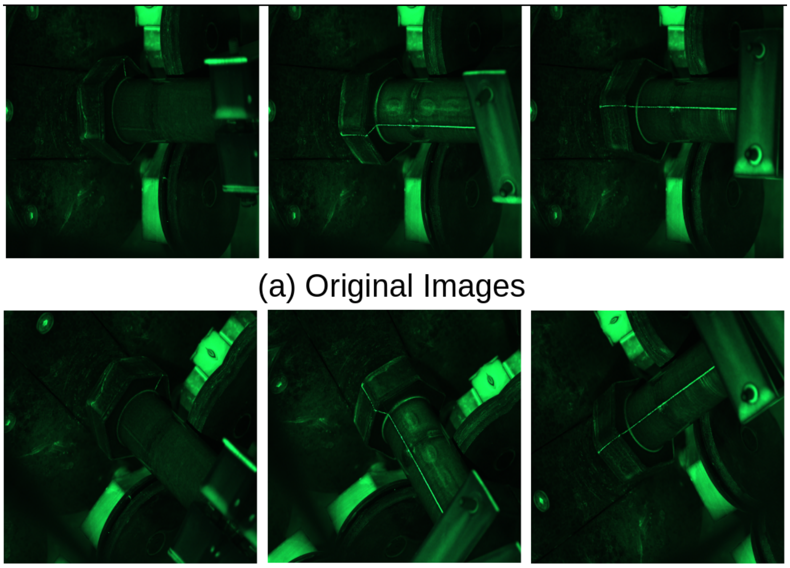
Shear [126]: Shear is a technique used to artificially introduce variations in camera position to the training data. It involves applying a shear transformation to the image, which distorts the image in a specific direction. This simulates the effect of



(a) Original Images

(b) Augmented Images

Fig. 2.2: Examples of Horizontal Flip Augmentation

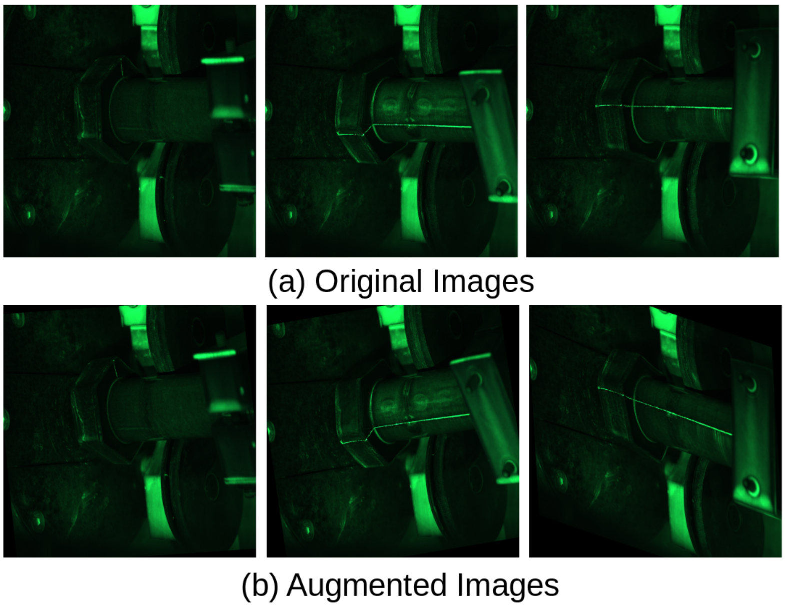


(a) Original Images

(b) Augmented Images

Fig. 2.3: Examples of Rotation Augmentation

the camera being positioned at a different angle from the object being inspected and allows the model to learn to detect defects regardless of the angle at which the image was captured. Examples of Shear Augmentation are shown in Figure 2.4.



(a) Original Images

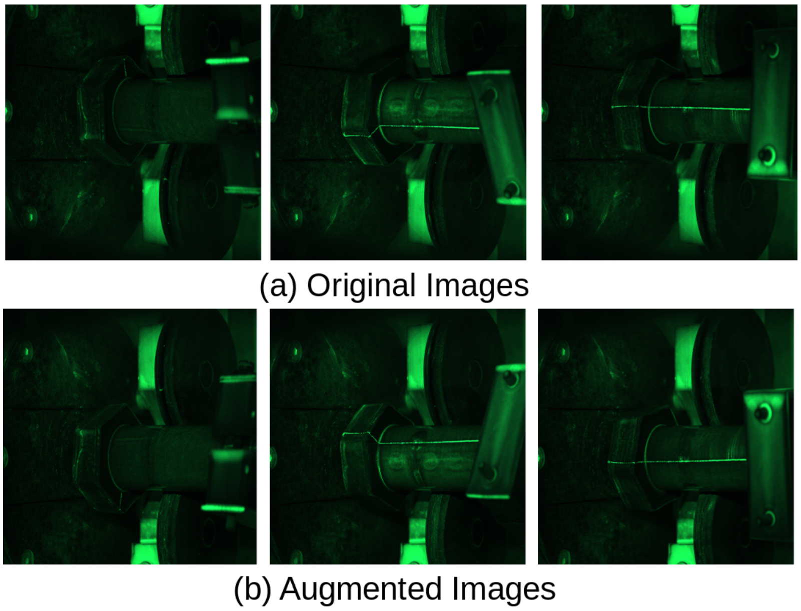
(b) Augmented Images

Fig. 2.4: Examples of Shear Augmentation

Vertical flip [126]: Vertical flip is a technique used to artificially introduce variations in camera position to the training data. It involves flipping the image vertically, simulating the effect of the camera being positioned at a different angle from the object being inspected. This allows the model to learn to detect defects regardless of the angle at which the image was captured. Examples of Vertical Flip Augmentation are shown in Figure 2.5.

Translation [126]: Translation is a technique used to artificially introduce variations in camera position to the training data. It involves moving the image in a specific direction by a certain amount, simulating the effect of the camera being positioned at a different location from the object being inspected. This allows the model to learn to detect defects regardless of the location of the camera when the image was captured.

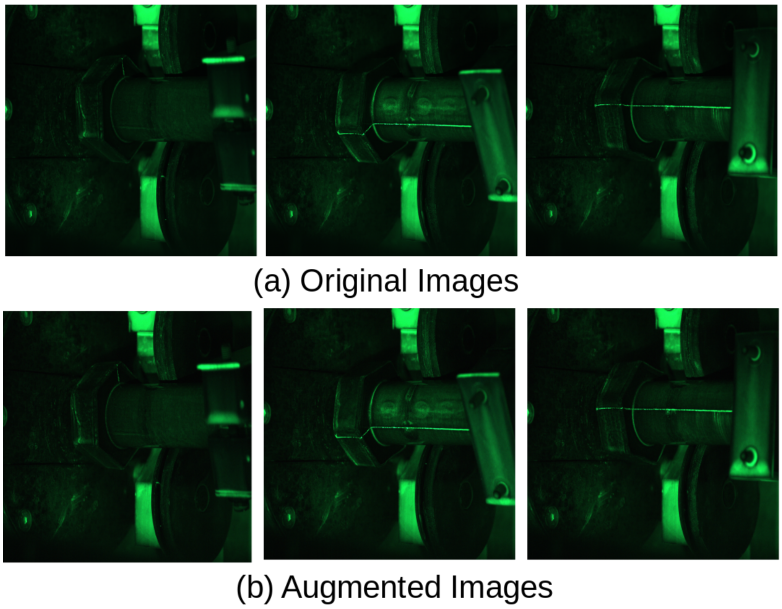
Noise injection [130]: Noise injection is a technique used to artificially introduce variations in lighting to the training data. It involves adding random noise to the image, simulating the effect of different lighting conditions. This allows the model to learn to detect defects regardless of the lighting conditions when the image was captured. Examples of Noise injection Augmentation are shown in Figure 2.6.



(a) Original Images

(b) Augmented Images

Fig. 2.5: Examples of Vertical Flip Augmentation

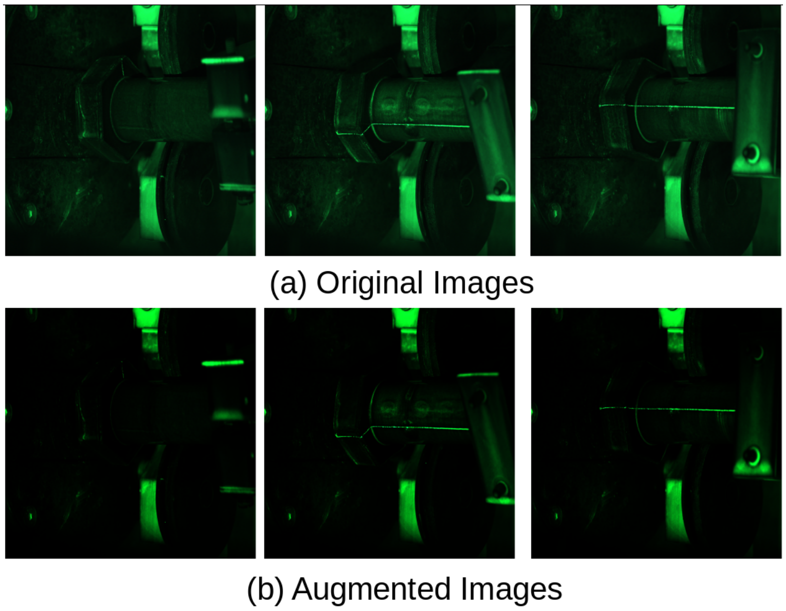


(a) Original Images

(b) Augmented Images

Fig. 2.6: Examples of Noise injection Augmentation

Color space transformations [133]: Color space transformations is a technique used to artificially introduce variations in lighting to the training data. It involves converting the image from one color space to another, such as from RGB to HSV. This simulates the effect of different lighting conditions and allows the model to learn to detect defects regardless of the lighting conditions when the image was captured. Examples of color space Augmentation are shown in Figure 2.7.


(a) Original Images

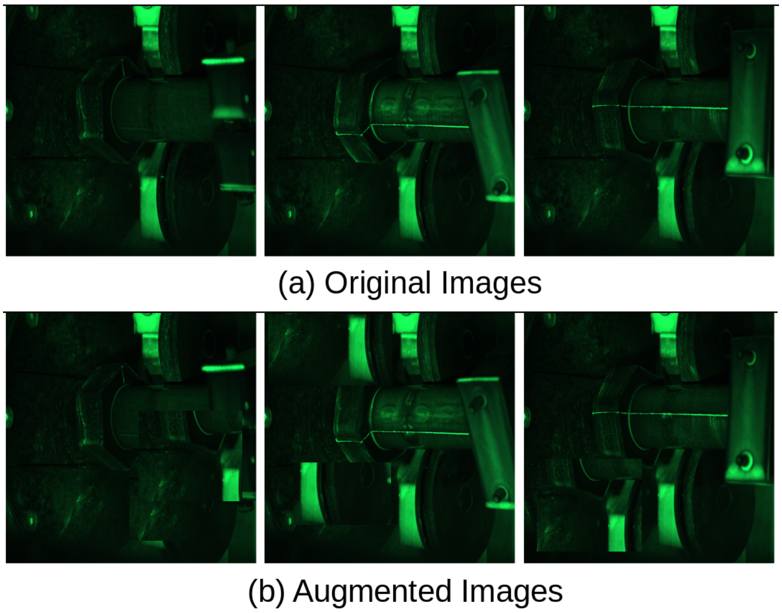
(b) Augmented Images

Fig. 2.7: Examples of color space Augmentation

Mixing images [134]: Mixing images is a technique used to artificially introduce variations in lighting to the training data. It involves combining two or more images, simulating the effect of different lighting conditions. This allows the model to learn to detect defects regardless of the lighting conditions when the image was captured. Examples of mixing images Augmentation are shown in Figure 2.8.

Random erasing [135]: Random erasing is a technique used to artificially introduce variations in lighting to the training data. It involves randomly erasing regions of the image, simulating the effect of different lighting conditions and allowing the model to learn to detect defects regardless of the lighting conditions when the image was captured. Examples of Random erasing Augmentation are shown in Figure 2.9.

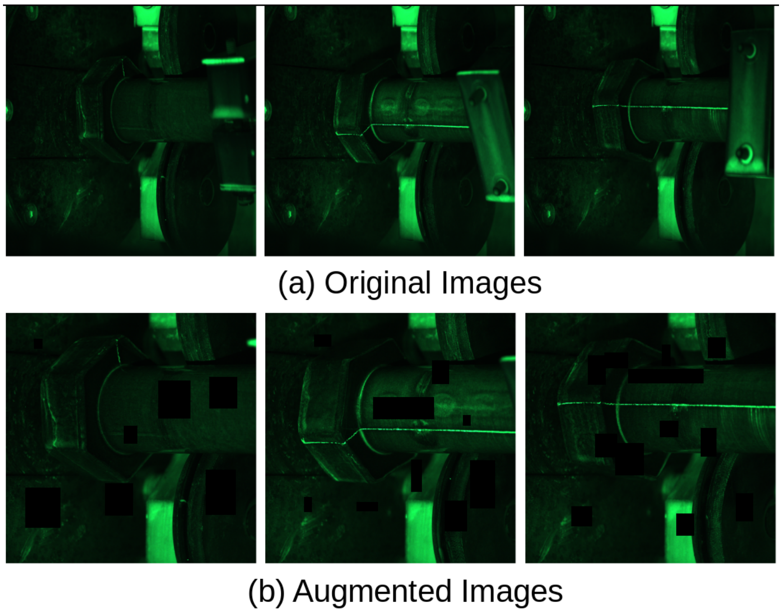
Sharpness [136]: Sharpness is a technique used to artificially introduce variations in lighting to the training data. It involves adjusting the sharpness of the image and simulating the effect of different lighting conditions. Examples of Sharpness Augmentation are shown in Figure 2.10.



(a) Original Images

(b) Augmented Images

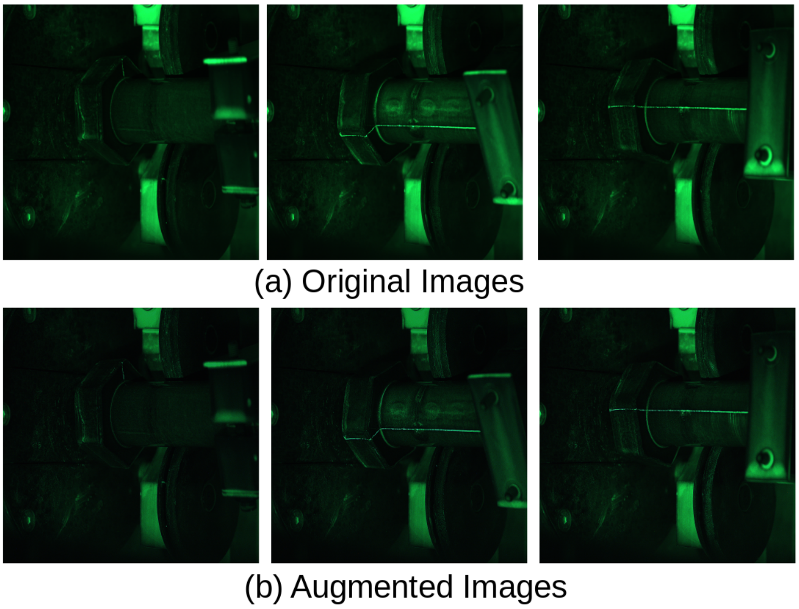
Fig. 2.8: Examples of mixing images Augmentation



(a) Original Images

(b) Augmented Images

Fig. 2.9: Examples of Random erasing Augmentation



(a) Original Images

(b) Augmented Images

Fig. 2.10: Examples of Sharpness Augmentation

By applying these transformations to the existing data, we generated new data points that are different from the original data, but still representative of the same underlying information. This helped to increase the size of the dataset and reduce overfitting.

Finally, Class imbalance, the focus of this article, is a common problem in defect detection tasks, where the number of examples of the minority class (defects) is much smaller than the number of examples of the majority class (non-defects). This can lead to several problems, such as poor performance of the defect detection model, overfitting, and a lack of robustness in real-world scenarios. In this article, we discussed the various issues that arise due to class imbalance in computer vision tasks and the various GAN based models that can be used to address these issues.

Class imbalance is a common problem in Classification, segmentation and object detection tasks, particularly in the context of defect detection. In this article, we proposed a taxonomy to summarize GANs based techniques for addressing imbalance problems in all three tasks as shown in Figure 2.11.

Defect classification: Surface defect classification is the task of identifying and categorizing different types of surface defects, such as scratches, dents, and stains, in images. This task is often performed using deep learning algorithms, such as CNNs, that are trained on labeled datasets of surface defects. class imbalance problems that can occur in defect classification task, including binary class imbalance, multi-class imbalance.

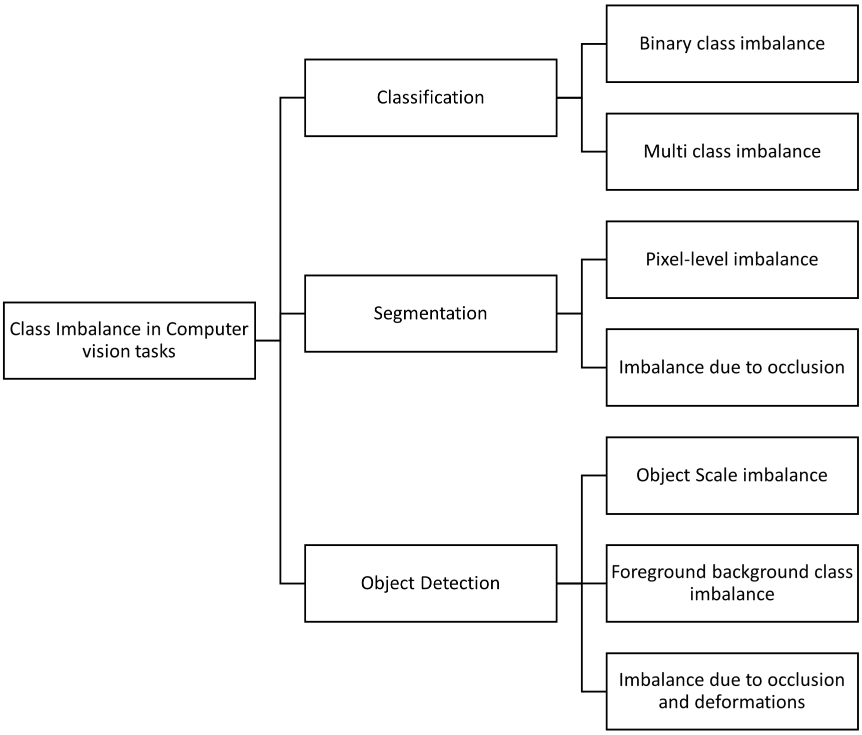

Class Imbalance in Computer vision tasks
Classification
Segmentation
Object Detection
Binary class imbalance
Multi class imbalance
Pixel-level imbalance
Imbalance due to occlusion
Object Scale imbalance
Foreground background class imbalance
Imbalance due to occlusion and deformations

Fig. 2.11: Proposed taxonomy of imbalanced problem in computer vision tasks

1. *Binary class imbalance:* Binary class imbalance refers to a situation where there is a significant difference in the number of instances of one class compared to the other class. In the context of defect classification, this could mean that there are many more non-defective items than defective items. This can be a major problem because it means that the classifier may be biased towards classifying everything as non-defective, since that is the majority class.
2. *Multi-class imbalance:* Multi-class imbalance refers to a situation where there is an imbalance in the number of instances of multiple classes. In the context of defect detection, this mean that there are a small number of instances of certain types of defects, while there are a large number of instances of other types of defects. This can be a major problem because it means that the classifier may be biased towards classifying everything as the majority class, since that is the most common class. There are several different scenarios that can occur with multi-class imbalance problems such as few minority-many majority classes, many minority-few majority classes, and many minority-many majority classes.
   - *Few minority-Many majority classes:* In this scenario, there are a small number of instances of one or more minority classes, while there are a large number of instances of one or more majority classes.

- *Many minority-Few majority classes:* In this scenario, there are a large number of instances of one or more minority classes, while there are a small number of instances of one or more majority classes.
- *Many minority-Many majority classes:* In this scenario, there are a large number of instances of one or more minority classes, while there are a large number of instances of one or more majority classes.

Defect segmentation: Surface defect segmentation is the task of identifying and isolating regions of an image that contain surface defects. This task is performed using image processing techniques, such as thresholding and edge detection, to extract the regions of interest from the image. Two specific types of imbalance that are particularly relevant to defect segmentation tasks: 1. imbalance due to occlusions and 2. pixel-wise imbalance.

1. *Imbalance due to occlusions:* Imbalance due to occlusions occurs when defects are partially or completely obscured by other objects or backgrounds. This can make it difficult for a CNN model to detect and segment these defects, as they are not fully visible in the images. As a result, the model may be trained on a dataset that is heavily skewed towards non-defect examples, leading to poor performance on defect examples.
2. *Pixel-wise imbalance:* Pixel-wise imbalance is another issue that can arise in defect segmentation tasks. In this case, the imbalance occurs at the pixel level, with non-defect pixels being much more prevalent than defect pixels. This can make it difficult for a CNN to accurately segment defects, as it is trained on a dataset that is heavily skewed towards non-defect pixels.

Defect detection: Surface defect detection is the task of identifying the presence of surface defects in images of surfaces. This task is often performed using deep learning algorithms, such as CNNs, that are trained on labeled datasets of surface defects. Detection can be performed at different levels of granularity, such as by detecting the presence of a defect or by detecting the specific location of a defect in an image. There are three main types of imbalance in the context of surface defect detection tasks: 1. foreground and background imbalance, 2. Defect scale imbalance, and 3. imbalance due to occlusion and deformation.

1. *Foreground and background imbalance:* Defects in the real-world datasets only occupy a small portion of the image, while the rest of the image is background. The imbalance between foreground (defect) and background can also hinder performance of the object detection algorithm. This can lead to a number of issues, including poor performance, biased predictions, and difficulty in training models.

2. *Defect scale imbalance:* Object scale imbalance refers to the imbalance in the size of defects in the dataset. This can occur when the dataset contains a wide range of defect sizes, with some defects being much larger or smaller than others. This can lead to poor performance, as models may struggle to detect smaller defects or may not be able to accurately locate larger defects. Additionally, this can lead to bias in the model, as it may be more likely to detect larger defects than smaller ones.
3. *Imbalance due to occlusion and deformation:* Imbalance due to occlusion and deformation refers to the imbalance in the appearance of defects in the dataset. This can occur when some defects are occluded or deformed in some way, making them harder to detect or locate. This can lead to poor performance, as models may struggle to detect occluded or deformed defects, or may not be able to accurately locate them. Additionally, this can lead to bias in the model, as it may be more likely to detect defects that are not occluded or deformed.

GANs [119] are a class of deep learning models that are used to generate new data that is similar to a given dataset. GANs consist of two main components: a generator and a discriminator. The generator is responsible for generating new data, while the discriminator is responsible for determining if the data generated by the generator is real or fake. GANs are trained using a two-player minimax game where the generator tries to generate data that can fool the discriminator, while the discriminator tries to correctly classify the generated data as fake. The generator and discriminator are trained simultaneously, and as the training progresses, the generator becomes better at fooling the discriminator, and the discriminator becomes better at distinguishing real from fake data. The main objective of GANs is to learn the underlying probability distribution of the given dataset, which can then be used to generate new data that is similar to the original dataset. GANs have been used in a wide range of applications, such as image synthesis, image super-resolution, and text-to-image synthesis.

The working principle of GANs can be broken down into two main steps: training the generator and training the discriminator. During the training of the generator, the objective is to minimize the following equation:

$$J(G) = -E[log(D(G(z)))] \tag{2.1}$$

Where G is the generator, D is the discriminator, and z is a random noise vector. The generator is trained to generate data that can fool the discriminator, which is achieved by minimizing the above equation.

During the training of the discriminator, the objective is to maximize the following equation:

$$J(D) = E[log(D(x))] + E[log(1 - D(G(z)))] \tag{2.2}$$

Where x is the real data and G(z) is the generated data. The discriminator is trained to correctly classify the generated data as fake, which is achieved by maximizing the above equation.

An approach to address class imbalance using GANs is to generate synthetic examples of the minority class. The GAN is trained on real examples of the minority class, and it generates new synthetic examples that are similar to the real ones. These synthetic examples can then be added to the dataset to balance the distribution of classes. By adding synthetic examples, the class imbalance problem can be alleviated, and the model can be trained on a more balanced dataset. This can lead to improved performance, especially in cases where the minority class is important and has a significant impact on the final predictions. Moreover, by generating synthetic examples, the size of the dataset can be increased, which can be beneficial for deep learning models, as they tend to perform better with larger datasets. In this article, a comprehensive examination of GAN models was conducted to address various class imbalance issues in computer vision tasks.

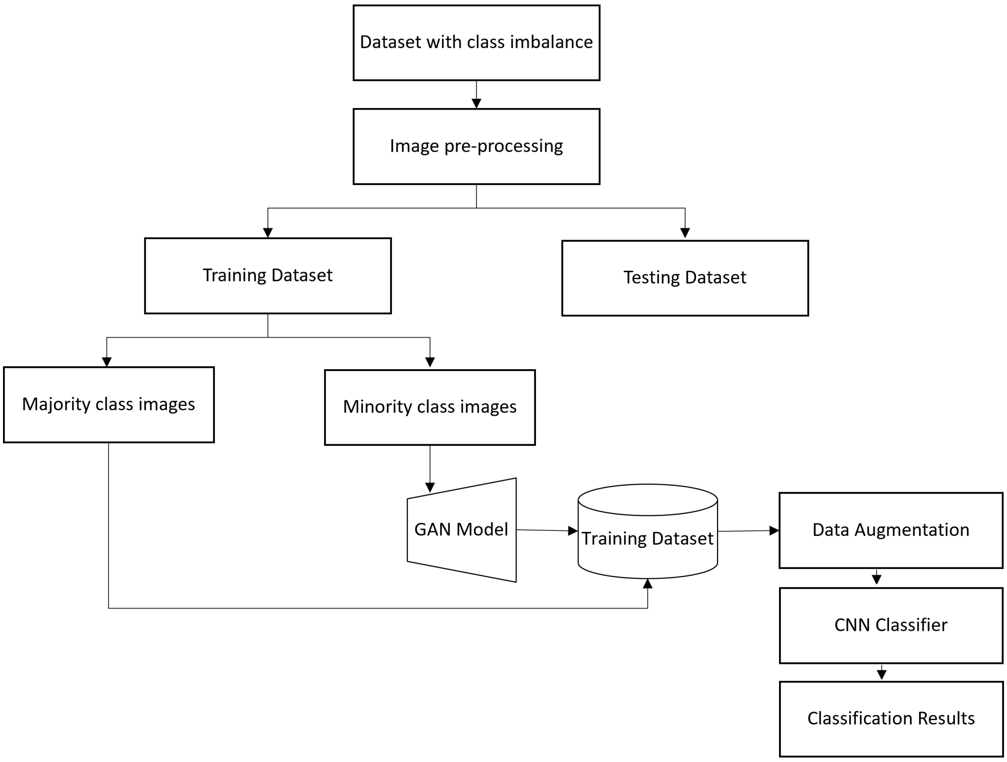


Fig. 2.12: Data Pipeline for combining GAN generated synthetic and augmented images.

Data pipeline for using the combination of GAN generated synthetic image and conventional image augmentation is a critical component in training CNN model. The goal of a data pipeline is to ensure that data is processed efficiently, effectively, and consistently from data collection to modeling. We also discussed a data pipeline that includes the following seven steps: data preprocessing, training and test split, GAN-based minority class oversampling, combining GAN-generated images and original training images, data augmentation, training classifier, and classification results.

- Data Preprocessing:The first step in the data pipeline is to preprocess the data. This involves cleaning, transforming, and normalizing the data to prepare it for modeling. Data preprocessing is crucial as it helps to improve the quality and accuracy of the data, remove noise and outliers, and ensure that the data is in a format that is suitable for modeling.
- Training and Test Split: The next step in the pipeline is to split the preprocessed data into two sets: training and testing. The training set is used to train the model, and the test set is used to evaluate the performance of the model. This step is important to prevent overfitting, which occurs when the model is trained too well on the training data, and performs poorly on the test data.
- GAN-based Minority Class Oversampling: In many real-world datasets, one class may be underrepresented, leading to class imbalance. This can negatively impact the performance of the model. To mitigate this issue, GAN-based minority class oversampling can be used. GANs are generative models that can generate new samples of the minority class based on the existing samples. The generated samples can be added to the training set, improving the balance of the classes.
- Combine GAN-generated Images and Original Training Images: The GAN-generated images and the original training images are combined to form a new training set. This new training set provides a better representation of the minority class, improving the overall performance of the model.
- Data Augmentation: Data augmentation is a technique used to increase the size of the training set by generating new, synthetic samples. These samples are generated by applying various transformations to the original training samples. This helps to increase the diversity of the training set, which can improve the robustness of the model and prevent overfitting.
- Train Classifier: The final step in the pipeline is to train a classifier using the combined and augmented training set. This can be done using any suitable ML algorithm, such as decision trees, random forests, or neural networks. The

goal of this step is to fit a model that accurately classifies the samples based on the features in the data.

- Classification Results: The final step is to evaluate the performance of the classifier by applying it to the test set. This can be done by calculating metrics such as accuracy, precision, recall, and F1 score. The results of the classification can then be used to fine-tune the model or to make informed decisions about the data.

The seven steps discussed in this article provide a framework for processing and preparing data for modeling Figure 2.12. By following these steps, It is feasible to ensure that the CNN models are robust, accurate, and effective.

Conclusion:

In conclusion, the challenge of imbalanced fastener image data posed a significant problem for our CNN model in accurately detecting defects. To address this issue, we delved into the most recent advancements in GANs for addressing imbalanced image data in computer vision tasks. Through categorizing imbalances into image level imbalances in classification, object level imbalances in object detection, and pixel level imbalances in segmentation tasks, we gained a deeper understanding of the problem. Additionally, we explored the combination of conventional data augmentation and synthetic images as a data-centric approach to improve model performance. Our study of state-of-the-art GAN models led us to the conclusion that a custom model tailored to magnetic particle inspection is needed for the fastener defect detection application.

2.2.2 Sensors 2023: Intra-class image augmentation for defect detection using Generative Adversarial Neural Networks

This paper describes a method for improving the generalization ability of CNN models used for surface defect identification using Data centric approach. The method proposed in this paper is a Pixel level image augmentation technique that uses image-to-image translation with GAN conditioned on fine-grained labels. A dataset of surface defects based on MPI is acquired, and the GAN model, called Magna-Defect-GAN, is trained on this dataset. The synthetic images generated by the GAN are then used to artificially inflate the size of the training dataset, resulting in improved accuracy for the defect identification model. The method is shown to be effective and adaptable to other surface defect identification models.

In the task of surface defect detection, GANs can be used to generate synthetic images of surface defects. The synthetic images can be used to artificially inflate the size of the training dataset, which can help to improve the performance of defect detection models. This is particularly useful in cases where there is a limited amount of real data available. By using GANs to generate synthetic images of surface defects, it is possible to increase the diversity of the training dataset, which can help to improve the generalization ability of the defect detection model. There are several different types of GANs that have been proposed in the literature. The most basic type of GAN is the vanilla GAN, which consists of a generator and a discriminator. Other types of GANs include Conditional GANs, which can be used to generate samples that are conditioned on a specific class label, and Wasserstein GANs, which use a different loss function than the standard GAN.

One of the most promising GAN architectures for image-to-image translation is the Pix2Pix GAN [137]. Pix2Pix GANs [137] are based on the vanilla GAN architecture, but they are trained using a specific type of data called paired data. Paired data consists of two images: one image that represents the input and one image that represents the output. The generator in a Pix2Pix GAN is trained to transform the input image into the output image. This can be used for various image-to-image translation tasks such as converting a black and white image to a color image, or converting a sketch to a photograph.

Deep Convolutional Generative Adversarial Networks [138] (DCGAN) is a type of GAN architecture that uses deep convolutional neural networks as the generator and discriminator. It was first proposed in 2015 and it introduced the use of strided convolutions in the generator and fractionally-strided convolutions in the discriminator, which allowed for the generation of higher-resolution images.

Progressive Growing of GANs [139] (Pro-GAN) is a GAN architecture that progressively increases the resolution of the generated images during training. The generator and discriminator networks start with a low resolution and are gradually increased during the training process. This allows the model to generate high-resolution images while avoiding the instability issues that can arise when training GANs with high-resolution images.

Laplacian Pyramid GANs [140] (LAPGAN) is a GAN architecture that uses a laplacian pyramid to generate images at different scales, which allows for the generation of high-resolution images. The architecture utilizes a generator network that generates images at multiple scales and a discriminator network that assesses the images at each scale.

Generative Recurrent Adversarial Networks [141] (GRAN) is a GAN architecture that uses a recurrent neural network (RNN) as the generator. The RNN generator generates the images one step at a time, which allows for the generation of more complex images than traditional feedforward GANs.

Diverging Discriminator GAN [142] (D2GAN) is a type of GAN that uses a diverging discriminator network, which allows the generator to learn more quickly. The discriminator network is designed to output a probability that is farther away from 0.5, which makes it easier for the generator to learn.

Single Image Non-Parametric GAN [143] (SinGAN) is a non-parametric GAN architecture that is trained on a single image, instead of a dataset of images. The generator network learns to generate images that are similar to the input image while the discriminator network learns to distinguish between the generated images and the input image.

Multi-scale Attention-based Discriminator GAN [144] (MADGAN) is a GAN architecture that uses multi-scale attention in the discriminator network, which allows for the generation of more detailed images. The generator network generates images at multiple scales and the discriminator network uses attention mechanisms to assess the images at each scale.

Conditional GAN [145] (cGAN) is a type of GAN that can generate samples conditioned on a specific class label. The generator network generates images based on the input class label while the discriminator network assesses the images based on both the class label and the image.

Auxiliary Classifier GAN [146] (ACGAN) is a cGAN that also includes an auxiliary classifier network in the discriminator. The classifier network is trained to predict the class label of the input image, which allows for the generation of more diverse samples.

Variational Autoencoder GAN [147] (VACGAN) is a GAN architecture that combines the strengths of VAEs (Variational Autoencoders) and GANs. The generator network is trained to generate images that are similar to the input image while the discriminator network is trained to assess the similarity between the generated image and the input image.

Information Maximizing GAN [148] (info-GAN) is a GAN architecture that maximizes the mutual information between the generator's input noise and the generated images. This allows the generator to learn a compact and interpretable representation of the data.

Self-Conditioned GAN[149] (SCGAN) is a GAN architecture that uses self-conditioning, which allows the generator network.

In the context of fastener surface inspection, two main problems arise with the use of above metioned GAN methods.

- Firstly, the image quality of the generated data needs improvement. The samples generated by GANs based on the vanilla GAN approach are often blurry, lacking in detail. The generated defects are also unrealistic and unstable, sometimes due to the phenomenon of mode collapse. Using these generated samples to fine-tune a defect inspection model would result in subpar performance.
- Secondly, the feature diversity of the generated defects is limited. The defects generated by vanilla GANs and latent-code-based generative methods are uncontrollable and lack diversity in features. This lack of diversity can have a negative impact on the generalization ability of the defect inspection model. It is crucial for the generated data to have a wide range of diverse features to ensure that the model can effectively generalize to real-world scenarios.

In this article, we propose a solution to the two problems of conventional GAN methods in data augmentation for fastener surface inspection. Our proposed method, called Magna-Defect-GAN, is a magnetic particle inspection-based defect GAN that utilizes a prior knowledge-based data augmentation method to overcome these limitations.

The first improvement of our method is the ability to generate defects with varied features in a controllable manner. This is achieved by integrating industrial prior knowledge into the generation process. The idea is inspired by the difference between human experts and machine inspection models. While machine inspection models often fail to inspect defects that are not in its training set, human experts can imagine unseen defects based on their prior knowledge.

In our method, this prior knowledge is encoded in the form of binary masks and also guidance vectors during the generation process. The binary masks serve as a way to incorporate industrial knowledge into the generation process, allowing us to create additional defects with different shapes, severities, scales, rotation angles, spatial locations, and part numbers. The model also includes strategies to enhance the quality of the images and stabilize the training process. The GAN architecture maps the given mask input to the sample space more efficiently by coupling a mask embedding vector, conditional label vector and latent noise vector. The generated samples are more diverse than traditional image-to-image translation GANs.By doing this, we ensure that the generated data has a wide range of diverse features, which is crucial for improving the generalization ability of the defect inspection model.

The second improvement of our method is better image quality. Our proposed method uses magnetic particle inspection to generate high-quality images of defects. The magnetic particle inspection method provides clear images with good details, making it well-suited for use in fastener surface inspection. This, in combination with the prior knowledge-based data augmentation, results in generated data that is not only diverse in features but also of high quality.

Magna-Defect-GAN Architecture:

The Magna-Defect-GAN model is a combination of three main components (as shown in Figure 2.13): a U-net style generator network, a discriminator network, and a pre-trained VGG feature extractor for style loss function.

Generator Architecture:

The training of an image-to-image translation GAN without a latent noise vector is a challenge due to the deterministic outputs that the model produces. In order to address this issue, Wang et al.[150] used a latent noise vector as an input to the generator model, in addition to the mask label. However, this approach still has some limitations in terms of overall feature projection efficiency. To improve upon this, our Magna-Defect-GAN model incorporates a novel approach. Our model first performs a mask embedding in the generator before the latent projection layer. This allows for a more efficient overall feature projection and helps to generate more diverse and detailed synthetic images.

The generator of our proposed GAN is based on a U-Net style design, which can be decomposed into two branches: the mask projection branch and the latent projection branch. The mask projection branch encodes the input mask into a 32-dimensional mask embedding vector through 7 convolution layers, each followed by a leaky rectified linear unit (Leaky ReLU).After the mask embedding, we concatenate the latent noise vector (132-dimensional vector) and the guide label vector with the mask embedding to improve sample space mapping and provide diverse texture detail in the synthetic images. Finally, the latent projection branch, whose inputs are the latent noise vector and the concatenated embedding and guide label, generates the output image.

In our proposed method, the image mask input provides the intended defect shape, position, and quantity, while the guide label provides the necessary defect background and thickness to generate a defect image. By combining the mask embedding, latent noise vector, and guide label, our Magna-Defect-GAN model is able to generate high-quality synthetic defect images that are diverse in features and of good quality.

Discriminator Architecture:

The discriminator architecture used in our proposed method, Magna-Defect-GAN, is a modified version of the Patch-GAN [151] architecture. Unlike traditional discriminator architectures, which classify the output and target image as being real or fake, the Patch-GAN discriminator uses a convolutional network that divides the input images into NxN patches.

The advantage of this approach is that it gives feedback on each region or patch of the image, enabling high frequency and encouraging detailed outputs by the generator. Typically, 70 x 70 patches are used to avoid tiling artifacts, but we found that using smaller patches in combination with style transfer losses results in sharper images while eliminating tiling artifacts. As a result, we use a patch size of 16 x 16.

The use of a Patch-GAN discriminator architecture [151] allows our method to focus on the details of the generated images, encouraging the generator to produce high-quality and realistic images. This, in turn, leads to a more effective data augmentation solution for fastener surface inspection, which is the goal of our proposed method.

Loss Function:

In our proposed method, Magna-Defect-GAN, we use a combination of three different losses in the GAN model: adversarial loss, style loss, and reconstruction loss.

- Adversarial loss is used to make sure that the generated images are realistic and not easily distinguishable from the original images. This is achieved by training the generator to trick the discriminator, which is trained to differentiate between real and fake images. The adversarial loss helps the generator to produce images that are indistinguishable from real images, which is essential for effective data augmentation in fastener surface inspection.The adversarial loss can be represented mathematically as follows:

  $$L_{adv} = log(D(x)) + log(1 - D(G(z))) \tag{2.3}$$

  where D(x) represents the discriminator's prediction of the real image x, G(z) represents the generator's output, and z is a random noise vector used as input to the generator. The goal of the generator is to minimize the adversarial loss by producing images that the discriminator cannot distinguish from real images.

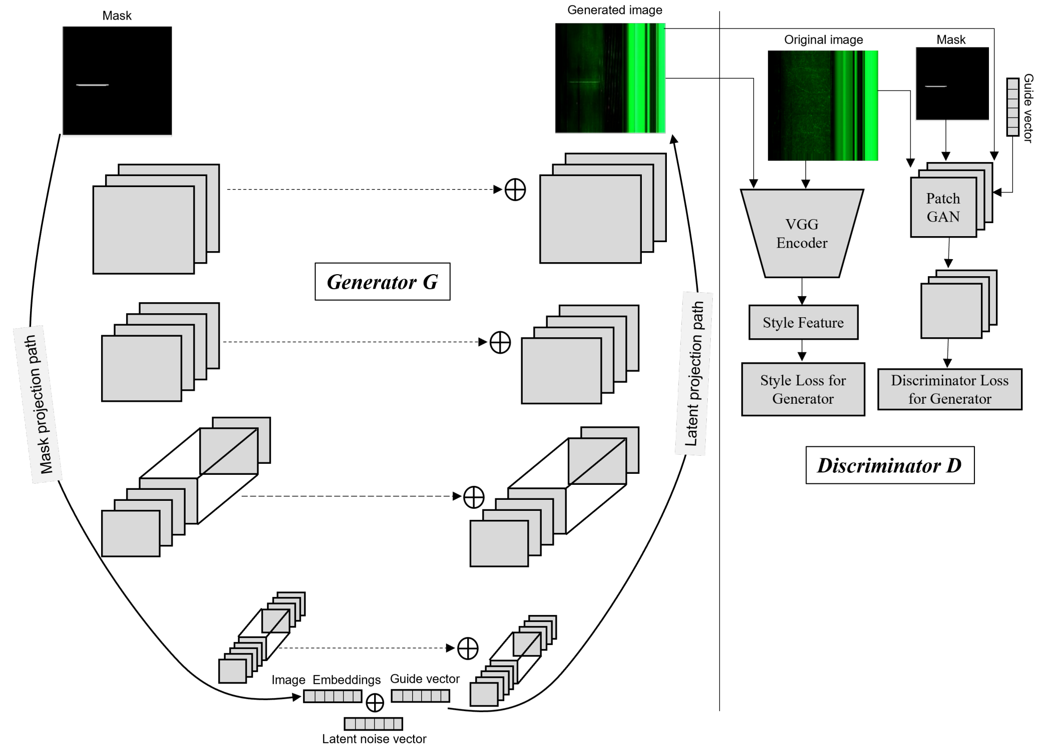


Fig. 2.13: The Magna-Defect-GAN model consists of a U-net styled generator network, a discriminator network, and a pre-trained VGG feature extractor. The generator network's mask projection path transforms input masks into image representations through encoder blocks. The latent projection path is utilized to produce the output image by combining the image representations, guide vectors, and latent noise vectors. The discriminator network D is trained to differentiate between real and generated images. The pre-trained VGG network is employed to extract features for computing the style loss.

- Style loss is used to ensure that the generated images have the same style as the original images. This is done by comparing the feature maps of the generated images to the feature maps of the original images. The style loss ensures that the generated images maintain the overall look and feel of the original images, which is important for maintaining the authenticity and realism of the generated images. We utilized the style loss at various levels in the original and generated image using a pretrained VGG model, similar to previous research [152]. The style information is evaluated as the level of correlation between the feature maps in a specific layer. The style loss is determined by comparing the mean and standard deviation between the feature maps generated by the generated image and the original image. To preserve the similarity between the style image and the generated image based on spatial information, the pair-wise correlation is calculated between all the feature vectors in the filters for each style layer. These feature correlations are given by Gram matrix

$G^l \in R^{N_l \times N_l}$, the inner product between the vectorized feature maps in layer l:

$$G_{ij}^l = \sum_{k=1}^{M_l} F_{ik}^l F_{jk}^l \tag{2.4}$$

Assume that there are A_l filters in total, each with a feature map of size B_l, and that we have G_{ij}^l, H_{ij}^l gram matrices for the style image and the generated image. Thus, we can calculate the overall style loss as follows:

$$L_{style} = \sum_l W_l \frac{1}{4A_l^2 B_l^2} \sum_{i,j} (G_{ij}^l - H_{ij}^l)^2 \tag{2.5}$$

where the weight given to layer l is w . Each W_l , in this case, contains the value $\frac{1}{4}$. Finally, reconstruction loss is used to ensure that the generated images are similar to the original images. This is done by comparing the generated images to the original images. The reconstruction loss helps the generator to produce images that are as close as possible to the original images, which is important for improving the accuracy and effectiveness of the data augmentation process. The reconstruction loss can be represented mathematically as follows:

$$L_{rec} = ||G(x) - x||_2^2 \tag{2.6}$$

where G(x) represents the generated image, and x represents the original image. The reconstruction loss measures the difference between the generated and original images, with a lower value indicating a better match in similarity.

The combination of these three losses in our GAN model provides a comprehensive solution for improving the quality and realism of the generated images, making our proposed method, Magna-Defect-GAN, a highly effective data augmentation solution for fastener surface inspection. The overall loss function used in our model can be represented as:

$$L = \lambda_1 * L_{adv} + \lambda_2 * L_{style} + \lambda_3 * L_{rec} \tag{2.7}$$

where λ_1, λ_2 and λ_3 are hyperparameters that control the relative importance of the style and reconstruction losses, respectively.

Evaluation Metrics:

Inception Score (IS) and Fréchet Inception Distance (FID) are two popular evaluation metrics for assessing the quality of GAN models.

IS is a measure of the quality and diversity of generated images. It combines two aspects of the generated images: their quality (based on how well the generated images match the real data distribution) and their diversity (based on how different the generated images are from each other). The IS is calculated as the average KL-divergence between the conditional class distribution and the marginal class distribution. The formula for IS is given as:

$$IS = \exp(E[KL(p(y|x)||p(y))]) \tag{2.8}$$

where x is a generated image, y is its corresponding class, p(y|x) is the predicted class distribution for the generated image, and p(y) is the marginal class distribution.

FID: The FID is a measure of the similarity between the distributions of real and generated images. The FID measures the Fréchet distance between the feature representation of the real data and the generated data. The Fréchet distance is the Earth Mover's Distance between two Gaussian distributions. The formula for FID is given as:

$$FID = ||\mu_{real} - \mu_{gen}||^2 + Tr(Cov_{real} + Cov_{gen} - 2 * \sqrt{Cov_{real} * Cov_{gen}}) \tag{2.9}$$

where μ_{real} and μ_{gen} are the mean vectors of the feature representations for the real and generated images, Cov_{real} and Cov_{gen} are the covariance matrices of the feature representations for the real and generated images, and Tr denotes the trace of the matrix.

Both the IS and FID scores are commonly used to evaluate the performance of generative models. Lower values of FID indicate that the distributions of real and generated images are more similar, while higher values of IS indicate that the generated images are of higher quality and more diverse.

Results and Discussion:

To evaluate the improvement in performance achieved by using synthetic images generated by GANs, we compare our approach with two widely used CNN models, ResNet [153] and EfficientNet [154], which are commonly employed in various defect classification tasks. The comparison of the defect classification metrics, including F1 score, precision, recall, and accuracy, is performed for the following training strategies:

(a) Model trained only with the original dataset: This refers to a CNN model that has been trained only on the original data, without any additional data or modifications. The model is only exposed to the raw data and has not seen any additional information.

(b) Model trained with the augmented dataset: This model is trained on an augmented version of the original dataset. The aim of this approach is to increase the diversity of the training data and reduce overfitting, as the model sees different variations of the same images.

(c) Model pretrained with synthetic dataset and fine-tuned with the original dataset: In this approach, the model is first trained on a synthetic dataset, which is a dataset generated using Magna-Defect-GAN, and then fine-tuned on the original dataset. The synthetic dataset serves as a pretraining step for the model, allowing it to learn general patterns and features that are common to both datasets. Fine-tuning on the original dataset helps the model to adapt to the specific characteristics of the task.

(d) Model pretrained with the ImageNet dataset and fine-tuned with the augmented dataset: ImageNet [155] is a large-scale dataset of images that is widely used for computer vision tasks. In this approach, the model is first trained on the ImageNet dataset [155], which provides the model with a good understanding of the common features and patterns in images. The model is then fine-tuned on the augmented version of the original dataset, allowing it to adapt to the specific characteristics of the task.

(e) Model pretrained with the ImageNet dataset and fine-tuned with the synthetic dataset: In this approach, the model is first trained on the ImageNet dataset [155] and then fine-tuned on the synthetic dataset. The aim is to leverage the general knowledge learned from the ImageNet dataset [155] and fine-tune it to the specific characteristics of the synthetic dataset.

(f) Model pretrained with the ImageNet dataset and fine-tuned with a mix of augmented and synthesized datasets: This approach combines the benefits of both augmentation and synthetic datasets. The model is first trained on the ImageNet dataset [155] and then fine-tuned on a combination of both augmented and synthetic datasets. This allows the model to learn from a diverse set of data and adapt to the specific characteristics of the task.

As shown in Table 2.1, the original data achieved accuracy and F1 scores of 80.8% and 0.801 for the ResNet model and 89.8% and 0.893 for the EfficientNet-B7 model. Data augmentation, such as flipping, rotating, and zooming, improved the accuracy and F1 scores to 83.5% and 0.868 for ResNet and 91.8% and 0.918 for EfficientNet-B7. Using a pretraining method with synthetic images and fine-tuning with original

Training Scheme	Model	Recall	Precision	F1-Score	Accuracy
a	ResNet [153]	0.684	0.969	0.801	0.808
	EfficientNet-B7 [154]	0.847	0.945	0.893	0.898
b	ResNet [153]	0.873	0.865	0.868	0.835
	EfficientNet-B7 [154]	0.895	0.943	0.918	0.918
c	ResNet [153]	0.863	0.907	0.88	0.875
	EfficientNet-B7 [154]	0.909	0.942	0.925	0.925
d	ResNet [153]	0.942	0.898	0.919	0.898
	EfficientNet-B7 [154]	0.958	0.935	0.946	0.947
e	ResNet [153]	0.945	0.934	0.939	0.927
	EfficientNet-B7 [154]	0.961	0.972	0.966	0.964
f	ResNet [153]	0.955	0.94	0.947	0.935
	EfficientNet-B7 [154]	0.969	0.977	0.973	0.972

Tab. 2.1: Data Centric Experiments on Surface Defect Dataset.

data resulted in higher accuracy and F1 scores of 87.5% and 0.88 for ResNet and 92.5% and 0.925 for EfficientNet-B7. Pretraining with ImageNet and fine-tuning with augmented data improved the accuracy and F1 scores to 89.8% and 0.919 for ResNet and 94.7% and 0.946 for EfficientNet-B7. Fine-tuning with a mix of synthetic and augmented data resulted in the best performance with accuracy and F1 scores of 93.5% and 0.947 for ResNet and 97.2% and 0.973 for EfficientNet-B7. The results show that both traditional data augmentation and GAN-generated synthetic images help prevent overfitting in a limited dataset.

Model	Inception Score ↑	FID ↓
Cycle [156]	2.88 ± 0.25	91.56
Pix2Pix [137]	3.08 ± 0.31	65.09
Magna-Defect-GAN	3.88 ± 0.36	**50.03**

Tab. 2.2: Table comparing the performance of the Magna-Defect-GAN Model.

According to the results shown in Table 2.2, the method we propose outperforms other methods in terms of IS and FID scores. There are a few factors that contribute to this superior performance. One reason is that combining the mask embedding vector, the conditional label vector, and the latent noise vector results in a more effective mapping of the sample space, resulting in more diverse and intricate textures in the generated images. Additionally, the use of a style loss in the Magna-Defect-GAN enhances the accuracy of the generated image with respect to background characteristics such as texture and color.

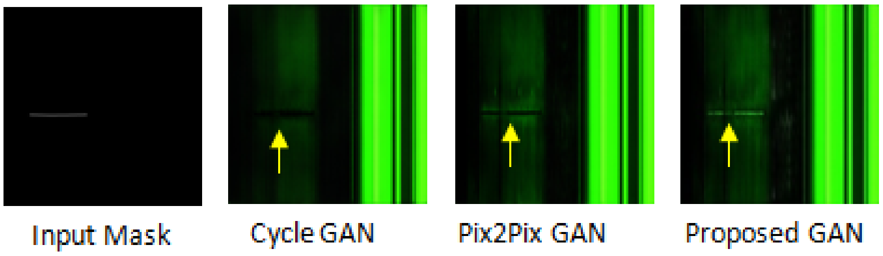


Fig. 2.14: Comparison between the synthetic image generated by the Magna-Defect-GAN and various image translation GANs.

It should be noted that other GAN models may generate many blank pixels in the defect region (as shown in Figure 2.14) in an effort to attain greater diversity, which can be detrimental for training a defect detection model as it would struggle to learn from the noisy images. However, the Magna-Defect-GAN model retains both structural consistency and fine background details, enhancing the ability of the defect detection model to learn from different appearances of defects under various levels of ambient light.

Conclusion:

In summary, the proposed method has several key findings that are worth highlighting.

- Firstly, by incorporating the mask embedding vector, the latent noise vector, and the discrete fine-grained guide labels, we were able to develop an improved conditional mask-to-image translation GAN. This GAN is capable of producing synthetic images that possess a high degree of intra-class diversity, including variations in the size, shape, position, thickness, brightness, and background of defects.
- Secondly, the proposed model was found to be more stable during the training process and the generated data samples were of higher quality compared to existing GAN models. This highlights the importance of carefully designing the GAN architecture and training procedure in order to obtain high-quality results.
- Thirdly, we concluded that GAN-based augmentation can be an effective tool for filling gaps in the discrete training data distribution and enhancing sources

of intra-class variation. However, it is important to keep in mind that GANs cannot expand the distribution beyond the extremes of the training dataset.

- Finally, we found that when training a defect detection model with a small dataset, a combination of conventional augmentations and GAN-generated synthetic images can be extremely helpful in avoiding overfitting. Conventional data augmentation can extrapolate the training data distribution, while GAN-based synthetic images add diversity by interpolating between the discrete data points in the manifold. These findings suggest that incorporating GAN-based augmentation into a training pipeline can be a useful tool for improving the robustness and performance of ML models in a variety of applications.

2.2.3 Applied Sciences 2023: Machine learning in manufacturing towards Industry 4.0: From ’For Now’ to ’Four-Know’

In this article, we explored the model-centric approach in detail after conducting a study on state-of-the-art GAN models in **Journal of Big Data 2021** and developing our custom GAN model specifically for the magnetic particle inspection in **Sensors 2023**. To gain a better understanding of current state-of-the-art model-centric approaches, we examined ML models within the context of Industry 4.0. The field of manufacturing is undergoing a significant transformation with the advent of Industry 4.0. Industry 4.0, also known as the fourth industrial revolution, is characterized by the integration of advanced technologies such as the IoT, big data analytics, and ML into the manufacturing process. This integration has the potential to revolutionize the way manufacturing is done, leading to improvements in efficiency, quality, and flexibility. Among these technologies, ML has been recognized as a key enabler of Industry 4.0, and its potential to improve manufacturing has been widely studied.

However, despite the increasing attention given to ML in the context of Industry 4.0, there is a lack of guidance on how to implement ML in the manufacturing industry. This paper aims to address this gap by investigating the ways in which ML can be implemented to improve manufacturing in its transition towards Industry 4.0. To achieve this, we formulated the following research questions:

- RQ1: How does ML benefit manufacturing and what are typical ML application cases?
- RQ2: How to develop an ML-based solution for problems in manufacturing engineering?
- RQ3: What are the challenges and opportunities in applying ML in manufacturing?

To answer these research questions, we systematically reviewed more than a thousand articles retrieved from two well-known research databases. The articles were classified within a two-dimensional framework, which takes value-based development stages on one axis and manufacturing levels on the other axis into account. The development stage dimension concerns visibility, transparency, predictive capacity, and adaptability, whereas the four considered manufacturing levels are: product, process, machine, and system. The systematic review and classification of the articles will provide a comprehensive understanding of the current state of research on the topic and will serve as a guide for practitioners looking to implement ML in the manufacturing industry.

ML has been applied in several fields of production engineering to solve a variety of tasks with different levels of complexity and performance. However, in spite of the enormous number of ML use cases, there is no guidance or standard for developing ML solutions from ideation to deployment. Therefore, this paper aims to address this problem by proposing an ML application roadmap for the manufacturing industry based on the state-of-the-art published research on the topic.

Firstly, the paper presents two dimensions for formulating ML tasks, ’Four-Know’ (Know-what, Know-why, Know-when, Know-how) and ’Four-Level’ (Product, Process, Machine, System). These are used for analyzing ML development trends in manufacturing. The ’Four-Know’ dimension includes the following aspects: Know-what refers to the knowledge of what the ML system should do, Know-why refers to the understanding of the underlying principles of the ML system, Know-when refers to the timing of when the ML system should be used, and Know-how refers to the knowledge of how the ML system should be implemented. The ’Four-Level’ dimension includes the following aspects: Product level refers to the use of ML to optimize product design and performance, Process level refers to the use of ML to optimize manufacturing processes, Machine level refers to the use of ML to optimize machine performance, and System level refers to the use of ML to optimize the overall manufacturing system.

Then, the paper provides an implementation pipeline starting from very early stages of a ML solution development. The pipeline includes the following steps: Problem definition, Data collection, Data preprocessing, Model selection, Model training, Model evaluation, and Model deployment. Furthermore, the paper summarizes the available ML methods, including supervised learning methods, semi-supervised methods, unsupervised methods and reinforcement methods, as well as their typical applications. Supervised learning methods are used for tasks such as classification and regression, semi-supervised methods are used for tasks such as anomaly detection, unsupervised methods are used for tasks such as clustering, and reinforcement methods are used for tasks such as control.

Finally, the paper discusses the current challenges during ML applications and outlines possible directions for future developments. Some of the challenges include data availability, data quality, interpretability, and scalability. Possible directions for future developments include using more advanced ML methods, developing more robust models, and implementing more advanced data management and preprocessing methods.

Applying ML in manufacturing normally involves the following six steps:

1. Data Collection: This is the first step in applying ML in manufacturing. It involves gathering data from various sources such as sensors, machines, and other equipment. The data collected can include information on production processes, machine performance, and product quality. It is essential to collect relevant and accurate data to train the ML models effectively.
2. Data Cleaning: After the data is collected, the next step is to clean the data. This step involves removing any irrelevant or inconsistent data, as well as handling missing or duplicate data. Data cleaning is an important step as it ensures that the data used for training the ML models is accurate and consistent.
3. Data Transformation: After the data is cleaned, it is transformed into a format that can be used for ML. This step involves converting the data into numerical values, normalizing the data, and transforming it into a format that can be used for ML.
4. Model Training: After the data is transformed, the next step is to train the ML models. This step involves using the cleaned and transformed data to train the models. Different types of ML algorithms can be used, such as supervised learning, unsupervised learning, and reinforcement learning. The goal is to find the best model that can accurately predict the results based on the input data.
5. Model Analysis: After the models are trained, the next step is to analyze the models. This step involves evaluating the performance of the models using various metrics such as accuracy, precision, and recall. The goal is to identify the best model that can accurately predict the results based on the input data.
6. Model Deployment: The final step is to deploy the model in the manufacturing process. This step involves integrating the trained models into the manufacturing process and using them to optimize production processes, improve product quality, and predict future trends. The deployed models

can continuously monitor the process and make adjustments in real-time to improve efficiency and reduce downtime.

Model development is the core of ML-based solutions, as the selection of an ML model plays a critical roles in the outcome. Therefore, We provided a comprehensive overview of ML methods and their potential possibilities in manufacturing applications, including supervised learning methods, semi-supervised learning methods, unsupervised learning methods, and reinforcement learning methods.

1. Supervised learning methods: Supervised learning [157] is a method of ML where the model is trained on labeled data, meaning the data used to train the model includes both input and output variables. The goal is to learn a mapping between inputs and outputs, such that the model can accurately predict the output given a new input. In the context of Industry 4.0 manufacturing engineering, supervised learning methods can be used for tasks such as product quality prediction, maintenance prediction, and process optimization.
    1.1 Tree-based methods: Tree-based methods [158] are a type of supervised learning method that uses a tree-like structure to represent the decisions and their possible consequences. These methods are commonly used for classification and regression tasks. Decision trees and Random Forest are examples of tree-based methods. In industry 4.0 manufacturing engineering, tree-based methods can be used for tasks such as product classification, process optimization, and maintenance prediction.
    1.2 Probabilistic-based methods: Probabilistic-based methods [159] are a type of supervised learning method that models the relationship between input and output variables using probability distributions. These methods are commonly used for classification and regression tasks. Examples of probabilistic-based methods include Naive Bayes, Logistic Regression, and Gaussian Mixture Model. In Industry 4.0 manufacturing engineering, probabilistic-based methods can be used for tasks such as product quality prediction, process optimization, and maintenance prediction.
    1.3 Neural-Network-based methods: Neural-Network-based methods [160] are a type of supervised learning method that models the relationship between input and output variables using a network of artificial neurons. These methods are commonly used for classification and regression tasks. Examples of neural-network-based methods include feedforward neural networks, convolutional neural networks, and recurrent neural networks. In Industry 4.0 manufacturing engineering, neural-network-based methods can be used for tasks such as product quality prediction, process optimization, and maintenance prediction.

2. Unsupervised learning methods: Unsupervised learning [161] is a method of ML where the model is trained on unlabeled data, meaning the data used to train the model only includes input variables, and the output variables are not provided. The goal is to find patterns or structure in the data without any prior knowledge of the output. In the context of Industry 4.0 manufacturing engineering, unsupervised learning methods can be used for tasks such as anomaly detection, process optimization, and product quality prediction.

   2.1 Dimensionality reduction: Dimensionality reduction [161] is a technique used to reduce the number of features in a dataset while preserving important information. This is important because datasets with high-dimensional features can be difficult to work with and may lead to poor performance of ML models. Two common techniques used for dimensionality reduction are Principal Component Analysis (PCA) and Autoencoder.

      2.1.1 PCA: PCA [161] is a technique that projects the data onto a new set of orthogonal (uncorrelated) axes, called principal components, which are ordered by the amount of variance they explain. The goal is to find the most important features of the data that explain most of the variance. In Industry 4.0 manufacturing engineering, PCA can be used for tasks such as anomaly detection, process optimization, and feature selection.

      2.1.2 Autoencoder: Autoencoder [162] is a type of neural network that is trained to reconstruct the input data. It is composed of two parts: an encoder that reduces the dimensionality of the data, and a decoder that reconstructs the original data. Autoencoder can be used for tasks such as anomaly detection, feature extraction, and compression.

   2.2 Clustering: Clustering [163] is a technique used to group similar data points together, called clusters. Clustering is a form of unsupervised learning and can be used for tasks such as anomaly detection, process optimization, and product quality prediction. Common clustering algorithms include k-means, hierarchical clustering, and density-based clustering.

   2.3 Association rule-based learning: Association rule-based learning is a technique used to find relationships between variables in a dataset. It is commonly used for market basket analysis and can be used to identify items that are frequently purchased together. In Industry 4.0 manufacturing engineering, association rule-based learning can be used for tasks such as process optimization, product quality prediction, and maintenance prediction.

3. Reinforcement learning: Reinforcement learning [164] is a type of ML that focuses on training agents to make decisions in an environment by taking actions and receiving rewards. In the context of manufacturing, reinforcement learning can be used to optimize various production processes and improve efficiency. In manufacturing environments, the reinforcement learning agent can be trained to make decisions based on real-time data inputs, such as machine utilization and inventory levels. The agent can learn how to maximize rewards by adjusting production parameters, such as the number of machines used, the production schedule, and the allocation of resources.

Conclusion: In conclusion, the proposed ML application roadmap for the manufacturing industry provides a valuable framework for understanding and developing ML solutions in this field. By using the Four-Know and Four-Level dimensions, developers can analyze the trends in ML development in manufacturing and use the implementation pipeline to guide the development process. Additionally, the paper provides an overview of the available ML methods and their typical applications, as well as outlining the current challenges and possible directions for future developments. By utilizing this ML application roadmap, we gained a deeper insight into the current model-centric solution approaches, which served as a starting point for developing our novel CNN model.

2.2.4 IEEE Transactions on Industrial Informatics 2023: Attention Guided Multi-Task Learning for Surface defect identification

This article presents a novel CNN model called Defect-Aux-Net, which is designed to detect defects in surfaces. The Defect-Aux-Net uses a technique called multi-task learning with attention mechanisms to make use of additional information from related tasks in order to improve the accuracy and robustness of the CNN when identifying surface defects.

Industrial visual inspection systems have come a long way in recent years with the advent of deep learning techniques, but despite the seemingly limitless potential of these complex CNN architectures, they are still barely utilized to their full potential [165]. This is attributed to several reasons in our case, one of the main being the inherent challenges associated with identifying small-scale surface defects.

The appearance of target surface defects varies greatly [116], depending on the type of material, imaging conditions, and camera position. This can cause issues for CNN models, making it difficult for them to identify small-sized defects accurately. Moreover, the distinction between tiny defects and non-defect components within an image can be challenging, leading to the appearance of false positives in images that are actually defect-free. It has created significant problems for our fastener industrial application where high accuracy is critical.

Another significant challenge with complex CNN models is their limited real-time applications. The long inference time [166] required to analyze images often results in higher computational resource and power consumption. This makes real-time applications of these models extremely limited and is a major roadblock to the widespread adoption of industrial visual inspection systems.

In this PhD research on surface defect detection using CNNs, there are not only the challenges discussed above, but also additional difficulties that arise during the training and testing phase. One of these difficulties is the problem of overfitting, which is a common issue when training CNN models. Overfitting [167] occurs when the model is too complex and is able to fit the noise in the training data, rather than the underlying patterns and features. This can happen when the model has too many layers or the number of neurons in each layer is too large. As a result, the model is able to achieve high accuracy on the training data, but performs poorly on the test data, as it is not able to generalize to new examples. Another problem that arises during the training of CNN models is underfitting. Underfitting [167] occurs when the model is not complex enough to capture the underlying patterns and features in the data. This can happen when the model has too few layers or the number of neurons in each layer is too small. As a result, the model is not able to accurately identify the patterns and features associated with surface defects, leading to poor performance on the test data. Balancing between underfitting and overfitting is critical [167] for ensuring that the model generalizes well and is able to perform accurately on new, unseen data. One method of detecting underfitting and overfitting is to plot the training and validation accuracy at each epoch during the training process. Figure 2.15 provides an example of what overfitting may look like when these accuracy values are plotted against the number of training epochs.

Additionally, Limited data problem [168] that can arise when working with real time surface defect detection using CNNs. In our case, the data available for training and testing the model is limited. This can be due to the difficulty in obtaining large amounts of labeled data, and the high cost of acquiring and labeling images. As a result, the model was not having enough data to learn from, leading to poor performance on the test data. To overcome these problems, in this article we used model centric approaches. We proposed a a Defect-Aux-Net model architecture that exploits auxiliary information beyond the primary labels to improve the

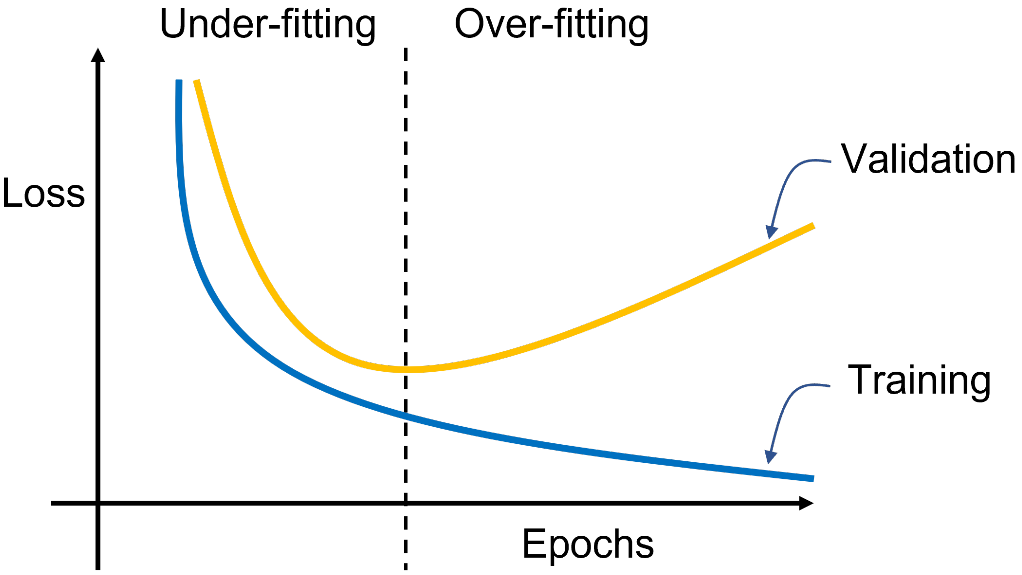


Fig. 2.15: The plot shows a turning point where the validation error begins to rise as the training rate continues to decrease. This is because the excessive training has caused the model to become too closely tailored to the training data, leading to poor performance on the testing set compared to the training set

generalization ability of surface defect identification tasks, and this method can help to solve the aforementioned issues.

- Defect-aux-Net is designed to overcome underfitting by incorporating additional information from related tasks. By using pixel-level segmentation masks and object-level bounding boxes, the model can learn more complex features and thus improve its performance in the surface defect identification task.
- To tackle overfitting, Defect-aux-Net utilizes attention mechanisms that exploit the rich additional information from related tasks. By focusing on the most relevant information, the model can learn more robust features and thus improve its performance on test data. This results in a model that is more resistant to overfitting and produces better results on unseen data.
- Defect-aux-Net also addresses the limited data problem by incorporating a multi-task learning scheme. This architecture allows the model to perform classification, segmentation, and detection of surface defects in a single network. The smaller number of parameters in the architecture makes the model more efficient in learning from a smaller number of labeled examples. This is especially important in real-world applications where labeled data may be limited.

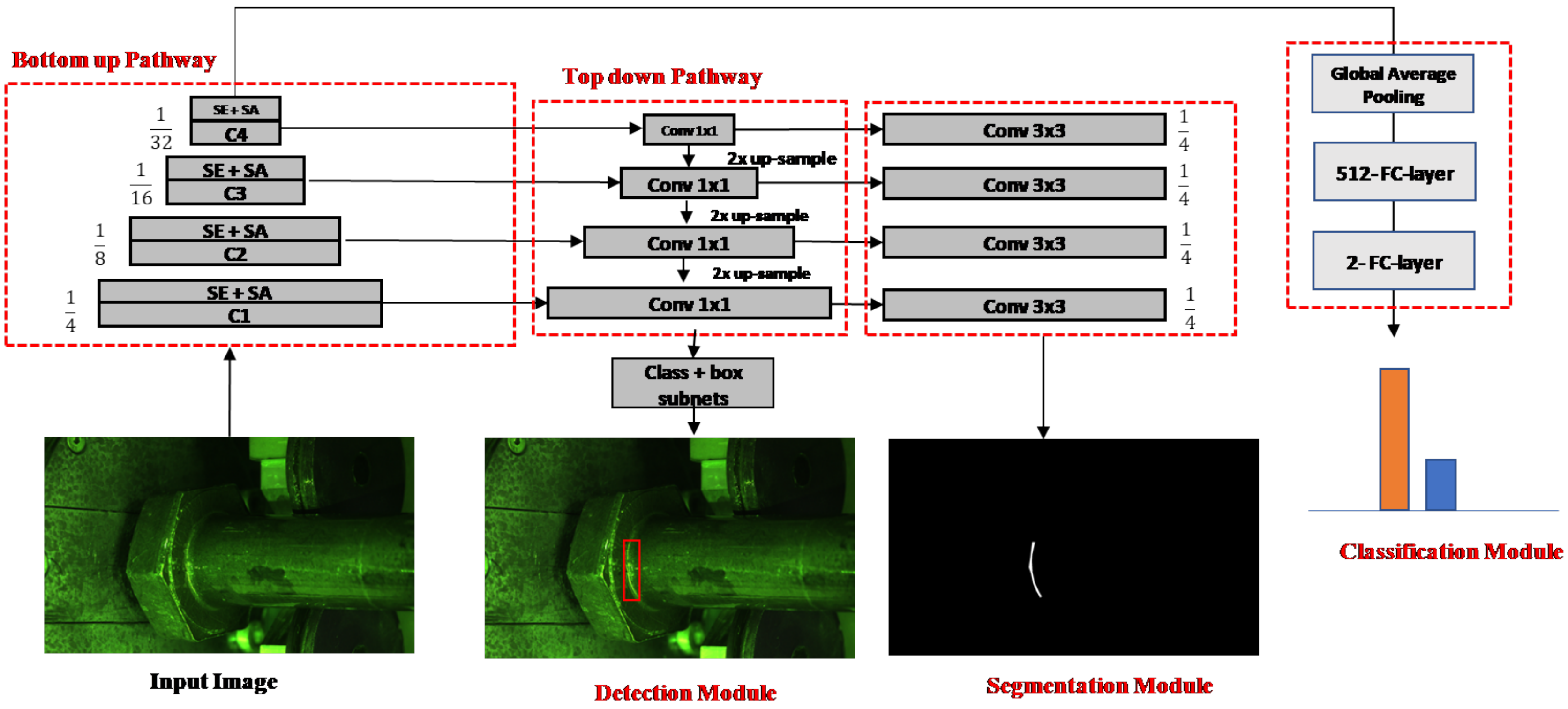


Fig. 2.16: A Pictorial representation of the proposed Defect-Aux-Net architecture, consisting of a classification, segmentation, and detection module that utilizes a multi-task loss function.

The proposed Defect-Aux-Net architecture is based on the Feature Pyramid Network (FPN)-semantic-segmentation model [169] and has been enhanced by adding the tasks of defect classification and detection to improve its generalization ability. This is achieved by utilizing image level information as a guiding principle.

To achieve this, a novel multi-task learning network was developed based on the FPN model. There are several different types of multi-task learning, each with their own strengths and weaknesses. Some of the most common types of multi-task learning include:

- Hard parameter sharing: This type of multi-task learning involves sharing the same set of parameters across all tasks. The model is trained to perform all tasks at once, with the same set of parameters. This approach is simple to implement and more efficient.
- Soft parameter sharing: This type of multi-task learning involves sharing some of the parameters across tasks, but also allowing for task-specific parameters. This approach allows for more flexibility in the model, as it can learn distinct features for each task. However, it can be more complex to implement and may require more data to train the model effectively.
- Multi-head architectures: This type of multi-task learning involves training separate heads for each task, with the heads sharing some or all of the parameters. This approach allows for more flexibility and can be more efficient than hard parameter sharing, but it can also be more complex to implement and may require more data to train the model effectively.
- Multi-task learning with attention mechanisms: This type of multi-task learning involves using attention mechanisms to focus on the most relevant information for each task. This approach can be more efficient as it allows the model to focus on the most important information for each task and can also help to overcome overfitting.

Defect-Aux-Net employs Multi-task learning with attention mechanisms based on the FPN model. The classification task is carried out in the bottom-up pathway of the network, while the segmentation is performed in the top-down pathway of the network. To create a bounding box, two sub-networks are employed in the top-down pathway. The first sub-network determines the class associated with the bounding box, while the second sub-network performs regression to adjust the position of the bounding box. This combination of tasks and sub-networks is designed to improve the overall accuracy and generalization ability of the model in identifying surface defects.

Defect-Aux-Net Architecture:

Defect-Aux-Net is inspired by two well-established deep learning architectures: FPN [169] and ResNet-50 [86]. Recognizing surface defects that vary greatly in scale is a major challenge in industrial machine vision systems. To address this issue, we use the FPN, which employs a hierarchical arrangement of convolutional filters to extract feature pyramids at different scales.

The FPN [169] consists of two pathways: the bottom-up and the top-down. The bottom-up pathway, also known as the encoder, is a standard convolutional neural network that is used for feature extraction. As we move up the network, the encoder gradually decreases the spatial resolution while building high-level feature maps. The top-down pathway is connected to the bottom-up pathway through lateral connections for efficient multi-scale feature fusion. It is designed to enhance the feature maps from the bottom-up pathway and build semantically rich feature maps at multiple scales by double upscaling. As a result, the feature pyramid has rich semantics at all levels, as the lower semantic features are interconnected to the higher semantics. A Pictorial representation of the proposed Defect-Aux-Net architecture is depicted in Figure 2.16.

Bottom-up pathway:

When selecting the core model for our proposed network, we tested several standard image classification architectures and ultimately chose ResNet-50 [86]. ResNet-50 [86] has a proven track record in performing well for surface defect classification, segmentation, and detection tasks. Additionally, ResNet-50 [86] has the advantage of using a stride of two for each scale reduction, making it easier to incorporate into FPNs for the top-down pathway. It is also a relatively small network based on modern standards, making it suitable for our limited labeled data problem.

However, there are two problems with existing ResNet-50 feature pyramids in the way they apply convolution operations to the input features. Firstly, the receptive field of the encoder only has information about the local region, so the global information is lost. Secondly, all feature maps are given equal magnitude of importance, but some feature maps are more important for the next layers than others. For instance, a feature map that contains edge information of the defects might be more important than another feature map that has background texture information (as shown in Figure 2.18 and Figure 2.19). To address this issue, we adopt the Squeeze-and-Excitation (SE) module in the encoder (Figure 2.17). The SE module consists of three components: Squeeze, Excite, and Scale. This allows for channel attention to be incorporated into the network, ensuring that the most important feature maps receive greater emphasis.

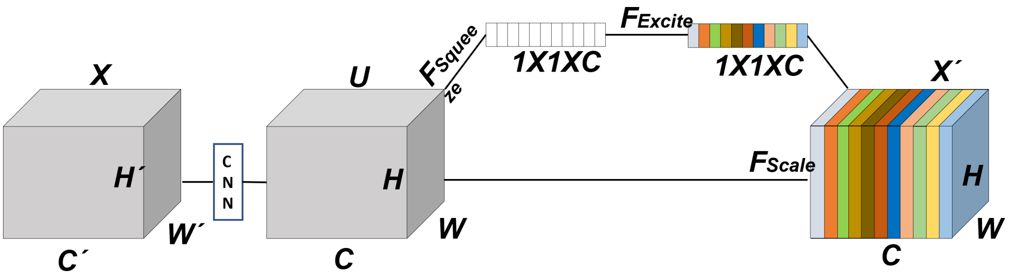


Fig. 2.17: The design of the Squeeze and Excite module.

SE attention mechanism is a method for improving the performance of CNNs by selectively focusing on relevant features in the input data. This mechanism is based on the idea of recalibrating the channel-wise feature responses by using global information to enhance the network's ability to learn more discriminative representations.

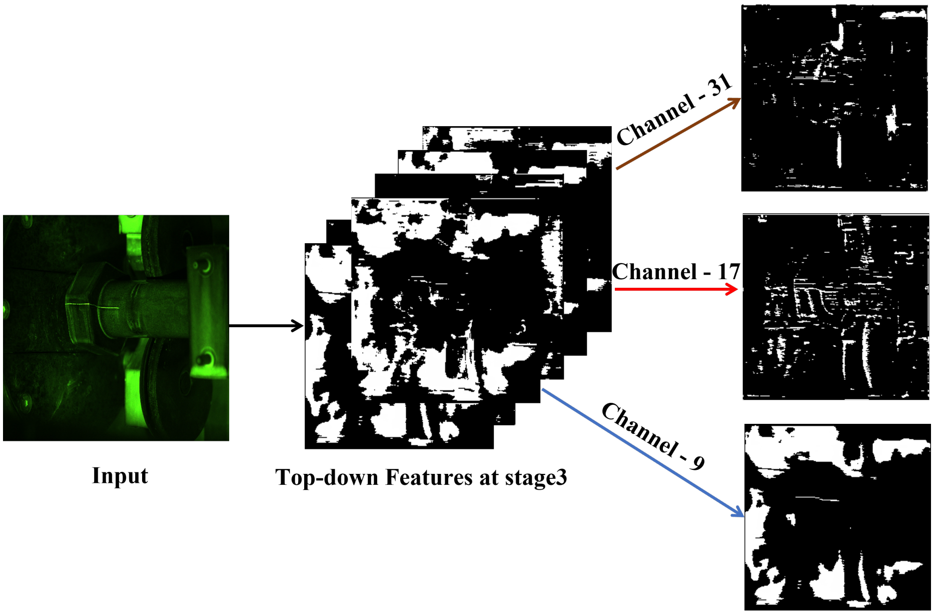


Fig. 2.18: Features of the samples in various channels of the top-down pathway at the third stage.

The SE mechanism consists of two main steps: the squeeze step and the excitation step. In the squeeze step, global information is obtained from the feature map by reducing the spatial dimensions through a global average pooling (GAP) operation. The result of this operation is a vector with a length equal to the number of channels, which represents the channel-wise feature responses. In the excitation step, this vector is passed through a fully connected (fc) layer followed by a non-linear activation function to learn a set of weights that represent the relative importance of each channel in the feature map. These weights are then used to enhance the channel-wise feature responses through an element-wise multiplication operation.

The mathematical representation of the SE mechanism can be described as follows:

$$Z_c = F_{squeeze} U_c = \frac{1}{HW} \sum_{m=1}^{H} \sum_{n=1}^{W} U_c(m,n) \tag{2.10}$$

The purpose of the squeeze component is to gather global information from each channel in a feature block U by performing a GAP operation across the spatial dimensions (H×W) for each channel U_c to obtain global statistics (1×1×C).

$$s = F_{excite} z, W = \sigma g z, W = \sigma(W_2 \rho W_1, z) \tag{2.11}$$

Once the squeeze component has gathered information globally, the excite component creates a set of weights for each channel using a Multi-Layer Perceptron (MLP) bottleneck structure. This structure features two layers that are fully connected and the output layer uses a sigmoid activation function. The MLP bottleneck is used to adjust the weights dynamically.

The conventional Resnet-50 architecture gives equal importance to each region in an image, making it challenging to distinguish the areas of interest from the background. In contrast, the spatial attention mechanism reduces background interferences by assigning a weight to each pixel in the feature map, allowing the system to focus on the most relevant parts of the image. This mechanism works by generating an efficient feature map summary through average and max-pooling operations along the channel axis, and then performing a convolutional layer followed by a sigmoid operation on the feature. The result is a spatial attention map that can be used to identify the most relevant parts of the image for further analysis.

The use of the spatial attention mechanism has a number of advantages over traditional methods for surface defect detection. Firstly, it reduces the number of false positive detections by focusing on the most relevant parts of the image. Secondly, it increases the accuracy of the defect detection process, as the system is able to distinguish between surface defects and background interferences. Thirdly, it is computationally efficient, as the feature map summary is generated through simple pooling operations and the convolutional layer is performed on the reduced feature map.

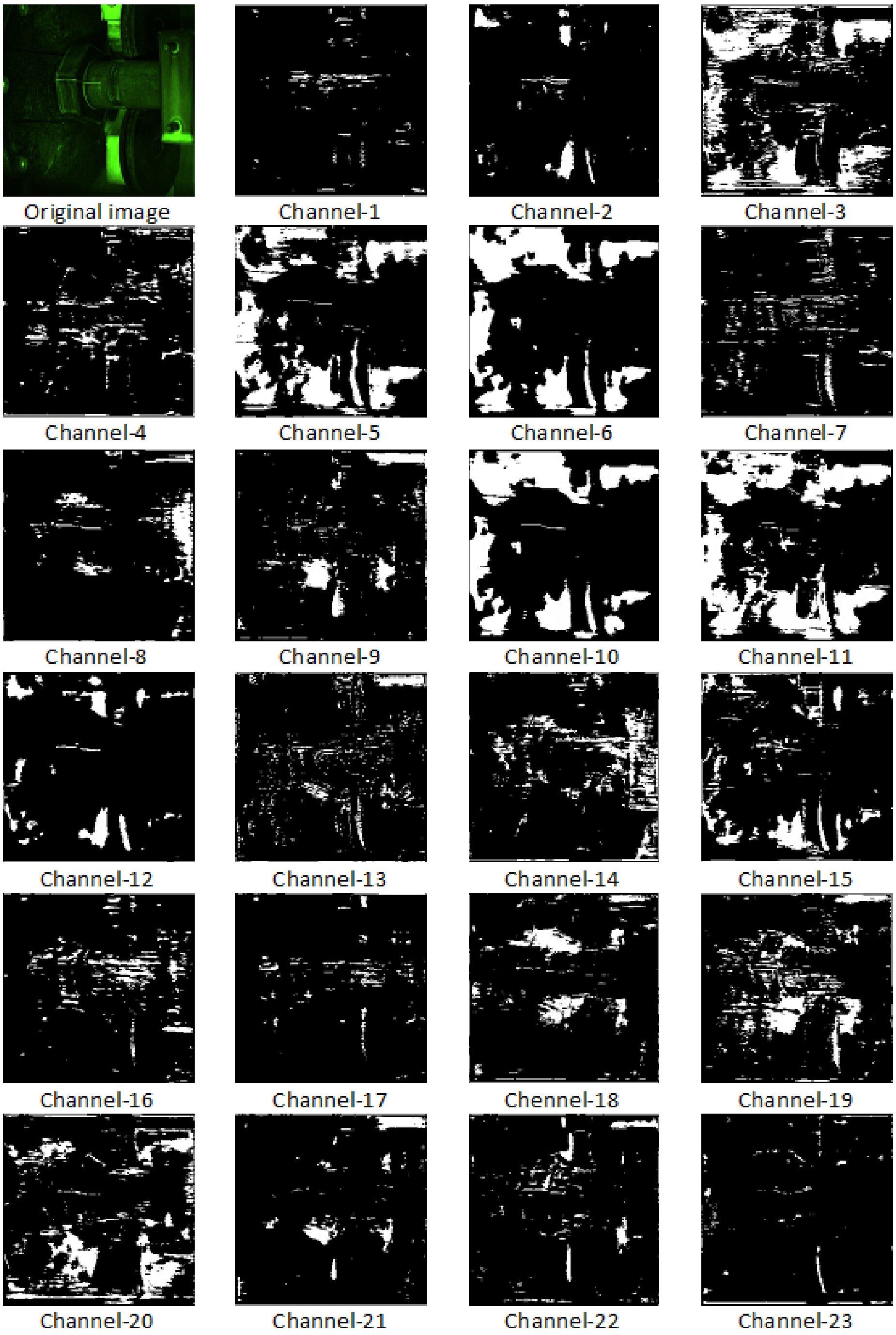


Fig. 2.19: Acquired CNN Features of the images in the first 23 channels of the third stage's top-down pathway

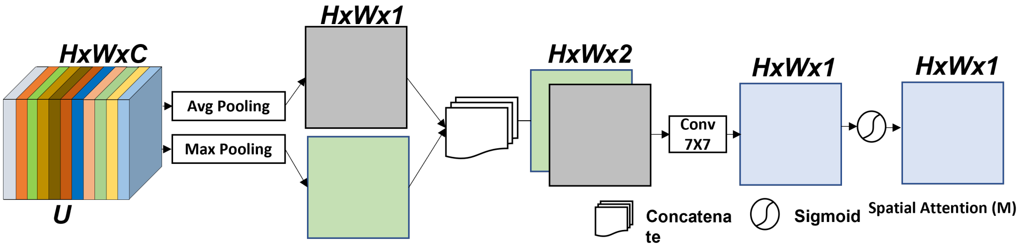


Fig. 2.20: The design of the Spatial Attention component.

The Resnet architecture uses a combination of residual blocks and identity blocks to process images and identify surface defects. A residual block consists of two blocks: an identity block and a convolution block. The identity block is used when the input and output dimensions are the same, while the convolution block is used when the dimensions are different. The purpose of the residual block is to allow the network to learn the residual mapping from the input to the output, rather than trying to fit a direct mapping.

In addition to the residual blocks, we have integrated the SA and SE modules into the residual block as shown in Figure 2.21. The SE module is designed to enhance the representation of important features within the feature map by using a channel-wise attention mechanism. The SA module, on the other hand, focuses on the most relevant parts of the image by assigning a weight to each pixel in the feature map (Figure 2.20).

The integration of the SA and SE modules into the residual block enhances the ability of the Resnet architecture to identify surface defects. The SE module helps to highlight important features within the feature map, while the SA module focuses on the most relevant parts of the image. This results in a more accurate and efficient defect detection process.

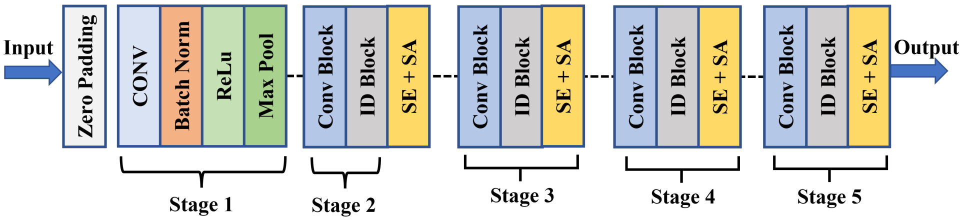


Fig. 2.21: FPN Bottom-Up structure with attention module.

Top-down pathway:

The deep features from the bottom-up pathway in the Resnet architecture are upsampled through a combination of convolutions and bilinear up-sampling operations until all the feature maps reach ¼ scale. The purpose of this up-sampling process is to obtain a higher resolution feature map, which is more representative of the surface defects present in the image.

After the up-sampling process, the attention module outputs from the bottom-up pathway are fused with the top-down pathway through lateral connections. This multi-scale feature fusion allows the network to extract features at different levels of abstraction, which is essential for detecting surface defects of different sizes and shapes.

The fusion process begins by applying a 1 x 1 convolutional filter to the feature maps C_2,C_3,C_4,and C_5. This reduces the number of channels in the feature maps to a fixed number, making them more manageable for the next step in the fusion process. The reduced feature maps are then merged with the corresponding top-down feature map through element-wise addition. The outputs from this step are then summed and transformed into a pixel-wise output.

Segmentation Branch:

The top-down pathway in the Resnet architecture is focused on classifying pixels into a pre-defined set of classes through the segmentation branch. However, the real-world datasets often contain far more pixels corresponding to the background compared to surface defects, which can result in the model becoming biased towards the background.

To overcome this pixel-wise class imbalance, we utilize the Dice loss, which calculates the overlap of the pixels in the predicted mask with the ground truth label through the use of the Dice coefficient. The Dice loss function is mathematically defined as:

$$L_{seg} = 1 - \frac{2y\hat{y} + 1}{y + \hat{y} + 1} \tag{2.12}$$

The ground truth label is represented by y_i and the predicted label is represented by $\hat{y}_i$. The Dice coefficient has a range of values from 0 to 1, with 1 indicating a complete and exact match between the pixels.

Classification Branch:

The bottom-up pathway provides rich and abstract feature representations of the input image. To utilize these representations, we perform global average pooling on the feature maps from the bottom-up pathway to obtain a feature vector. This feature vector is then fed into a sigmoid or softmax layer, depending on the type of classification being performed. In our approach, we use binary cross-entropy (BCE) as the classification loss function. Mathematically, the loss can be defined as:

$$L_{class} = \frac{1}{k}\sum_{1}^{k} CE(y - \hat{y}) \tag{2.13}$$

Where, y_i is the ground truth label, $\hat{y}_i$ is the predicted label of i^{th} sample, k is the total number of samples. CE is the binary cross entropy function.

Object detection Branch:

In our approach, bounding boxes and their associated classes are extracted through the use of box regression and classification subnets at each level of the top-down pathway. The classification subnet predicts the probability of defect presence at each spatial location in the input image. To facilitate the regression of the bounding boxes, a box regression subnet is attached to the top-down pathway in parallel to the classification subnet.

To tackle the problem of class imbalance, we employ focal loss [170], which is an improved version of cross-entropy. The focal loss [170] focuses the learning process on hard negative examples, thus addressing the imbalance issue. The focal loss is mathematically defined as:

$$L_{detection} = \alpha_t * (1 - p_t)^{\gamma} * \log(p_t) \tag{2.14}$$

Where, α_t is the weight parameter per class and γ is the hyper parameter focuses on hard negative samples. We choose α_t =0.25 and γ = 4 as suggested in [170].

Loss Function:

Our proposed method integrates three loss functions from the classification, segmentation, and detection tasks, which provide mutual sources of inductive bias for each task. The segmentation and detection loss functions signal back to the entire model, including both the bottom-up and top-down pathways, while the classification loss only signals back to the bottom-up pathway.

To leverage the heterogeneous annotations and jointly optimize multiple tasks, we combine and weight these three losses into a multi-task loss, L_M. The combination is performed as follows:

$$L_M = \beta L_{class} + \beta_1 L_{Seg} + \beta_2 L_{detection} \tag{2.15}$$

Where, β, β_1, and β_2 are weight parameters. We tested with different combinations of weight parameters and found that $\beta=\beta_1=\beta_2=1$ yields the best result for all the tasks.

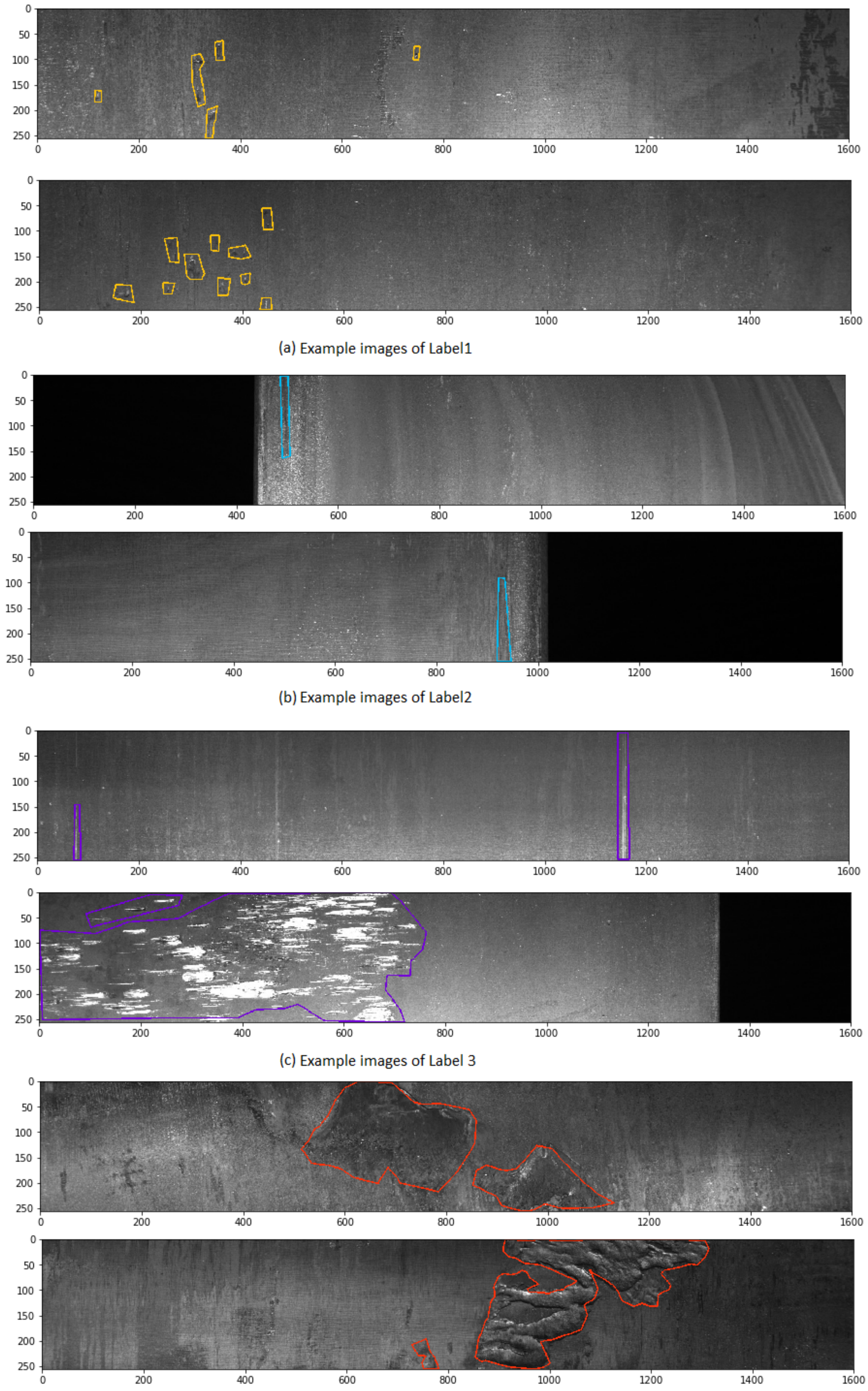
Dataset:

We evaluate the performance of Defect-Aux-Net on two challenging datasets, the Severstal steel sheet dataset [171] and our own fastener defect dataset, named as TekErreka dataset. The recent publication of the largest industrial steel sheet surface defect dataset by Severstal, the largest steel and steel-related mining company, has opened up new opportunities for research in the field of surface defect identification. The dataset, which is annotated by the company's technical experts, is highly imbalanced and contains pixel-wise masks of grayscale images of size 1600×256. The complexity of the dataset lies in the inter-class similarities between defective and defect-free examples, which makes the task of surface defect identification challenging. The Severstal dataset is highly imbalanced as shown in Figure 2.23, with a large number of defect-free images and a small number of images with defects. This imbalance makes the task of surface defect identification challenging because it is difficult to achieve high accuracy in identifying defects when the majority of the images are defect-free. This problem is compounded by the inter-class similarities between defective and defect-free examples, which make it difficult for CNN algorithms to differentiate between the two. Examples of Severstal steel with four categories of defects is depicted below in Figure 2.22.

In order to evaluate the performance of surface defect identification algorithms, a validation and test set must be selected from the Severstal dataset. In this study, 10% and 20% of the original 12,568 images were randomly selected as the validation and test data, respectively. This random selection helps to ensure that the results of the evaluation are representative of the entire dataset and that the performance of the algorithms is not affected by any biases in the selection process.

Data Preprocessing:

We used image resizing to 600x600 pixels and min-max standardization for normalization in our Defect-Aux-Net model. The resizing provided a balance between having enough detail and features in the images and being computationally feasible. The normalization using min-max standardization rescaled the raw pixel values to a range of 0 to 1, which helped the optimizer in the model avoid getting stuck in a local minimum. To further improve the diversity of the training set and reduce overfitting, we utilized augmentation techniques and GAN generated synthetic images as discussed in the subsection 2.2.1.



(a) Example images of Label1

(b) Example images of Label2

(c) Example images of Label 3

(d) Example images of Label 4

Fig. 2.22: Examples of Severstal steel with four categories of defects.

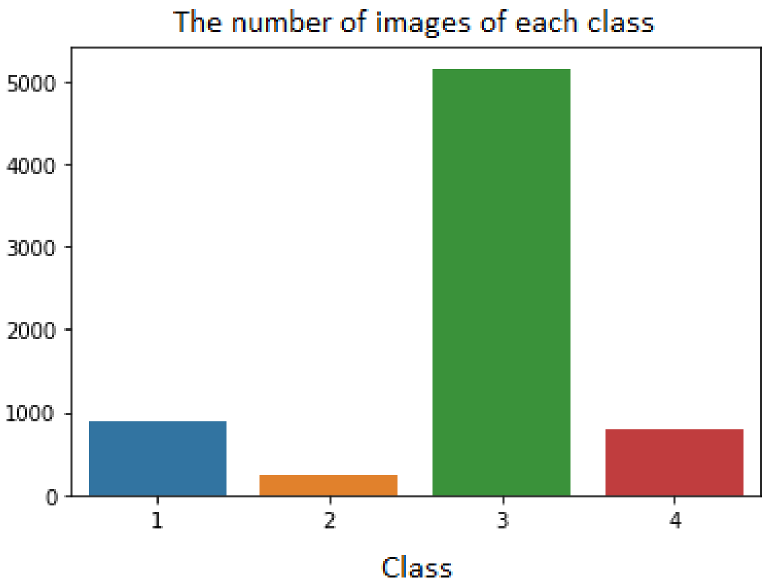


Fig. 2.23: Class distribution of Severstal dataset

Evaluation Metrics:

The classification results are assessed through various metrics such as precision, recall, F1-score, and binary accuracy.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.16}$$

$$Precision = \frac{TP}{TP + FP} \tag{2.17}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.18}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{2.19}$$

The true positive (TP) refers to correctly identified surface defects, true negative (TN) refers to correctly identified non-defect images, false positive (FP) refers to incorrectly classified images as surface defect, and false negative (FN) refers to incorrectly classified images as non-defect. Precision measures the accuracy of surface defect classification, while recall calculates the ratio of correctly classified surface defect images to the total number of such images. F1-score is a combination of precision and recall, providing a single score to measure performance. Binary accuracy represents the overall performance of the classification task. To evaluate the segmentation results, the Dice score and Intersection-over-Union (IoU) are used to calculate the overlap between the predicted and target binary masks. Lastly, the mean average precision (mAP) is employed to assess the defect detection results by comparing the detected bounding boxes to the ground truth bounding boxes.

Results and Discussion:

In the evaluation of the proposed approach, the performance of the classification task was compared to state-of-the-art deep learning architectures. The segmentation and detection modules were removed from the network specifically for the evaluation of the classification task. The results of the experiments are summarized in Table 2.3 and it was found that the majority of errors were false positives. The visual similarity between defects and surface noise was the cause of these false positive errors. It was observed that the proposed Defect-Aux-Net approach achieved an overall accuracy of 92.9% to 99.4% across all defect types on the Severstal dataset.

Model	Dataset	Class	Recall	Precision	F1-Score	Accuracy
Resnet-50 [2]	Severstal	Class1	0.454	0.403	0.427	0.831
		Class2	0.591	0.533	0.561	0.958
		Class3	0.918	0.847	0.881	0.811
		Class4	0.857	0.852	0.854	0.963
	TekErreka	Class1	0.759	0.979	0.855	0.949
SEResnet-50 [24]	Severstal	Class1	0.508	0.556	0.531	0.875
		Class2	0.617	0.58	0.598	0.97
		Class3	0.98	0.816	0.891	0.817
		Class4	0.559	0.94	0.701	0.94
	TekErreka	Class1	0.803	0.968	0.878	0.955
Effecientnet-B0 [5]	Severstal	Class1	0.891	0.859	0.875	0.964
		Class2	0.872	0.732	0.796	0.984
		Class3	0.943	0.963	0.953	0.929
		Class4	0.946	0.924	0.935	0.983
	TekErreka	Class1	0.858	0.928	0.892	0.958
Defect-Aux-Net	Severstal	Class1	0.891	0.926	0.908	0.975
		Class2	0.957	0.9	0.928	0.994
		Class3	0.982	0.929	0.955	0.929
		Class4	0.946	0.94	0.943	0.985
	TekErreka	Class1	0.887	0.939	0.912	0.971

Tab. 2.3: Performance comparison of Defect-aux-Net with the competing classification models.

The proposed multi-task learning approach showed to be superior in performance compared to other models, and it was also found that incorporating the segmentation task improved the performance of the classification task and vice versa. To test the approach's effectiveness in handling limited data, experiments were conducted by removing a portion of the training data at 90%, 75%, and 50%. The results showed that the accuracy decreased as the training data size decreased, as shown in Figure 2.24 and Table 2.4. Despite reducing the training data size, the proposed Defect-Aux-Net showed consistent performance, even when only 50% of the original training data was used.

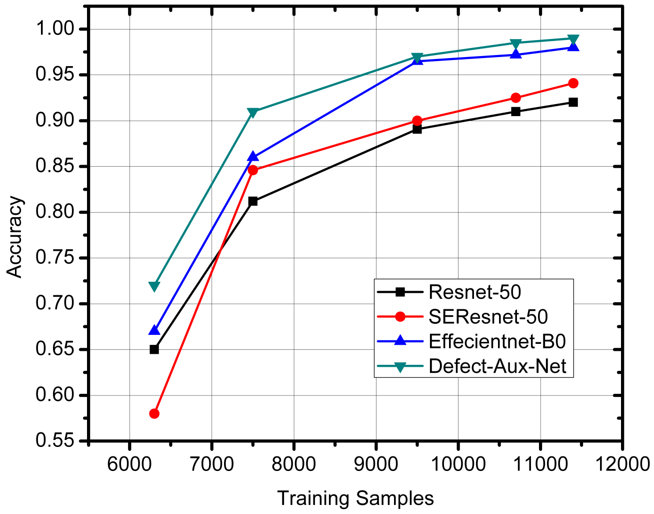


Fig. 2.24: Plot showing the relationship between the size of the training data and the classification accuracy on the Severstal dataset.

	50%	**60%**	**75%**	**85%**	**90%**
Resnet-50	0.65	0.812	0.89075	0.91	0.92
SEResnet-50	0.58	0.846	0.9	0.925	0.941
Effecientnet-B0	0.67	0.86	0.965	0.972	0.98
Defect-Aux-Net	0.72	0.91	0.97	0.985	0.99

Tab. 2.4: Ablation study on the effect of Training data size.

To assess the significance of the attention mechanisms in Defect-Aux-Net, we conducted experiments to compare the accuracy of the network with and without the spatial and channel attention mechanisms (squeeze and excite) on the TekErreka dataset, as shown in Table 2.5. Additionally, we also tested the combination of both spatial and channel attention mechanisms to determine their impact on the network's performance. These experiments aimed to verify the importance of the attention mechanisms in Defect-Aux-Net and their contribution to its overall accuracy and effectiveness in solving the classification task.

Model	Accuracy	Parameters (M)
Defect-Aux-Net (without attentions)	0.962	33.2
Defect-Aux-Net (with SE attention)	0.968	35.7
Defect-Aux-Net (Spatial attention)	0.963	33.5
Defect-Aux-Net (with SE + Spatial attention)	**0.971**	**36.2**

Tab. 2.5: Ablation study on the effect of using attention mechanisms on the Tekerreka dataset.

In our Study, the proposed approach was compared with other object detection algorithms on the TekErreka dataset. The comparative models included SSD [93], RetinaNet [170], and Cascade R-CNN [172]. The results of these comparisons are presented in Figure 2.25, which shows the mAP scores of the various detection models for the TekErreka dataset. The results demonstrate that Defect-Aux-Net was able to achieve a higher mAP score compared to the alternative networks. The mAP of the proposed algorithm was 17.95%, 43.77%, and 26.03% higher than that of RetinaNet, SSD, and Cascade R-CNN, respectively. These results indicate the superior performance of the proposed Defect-Aux-Net in comparison to the other object detection algorithms on the TekErreka dataset.

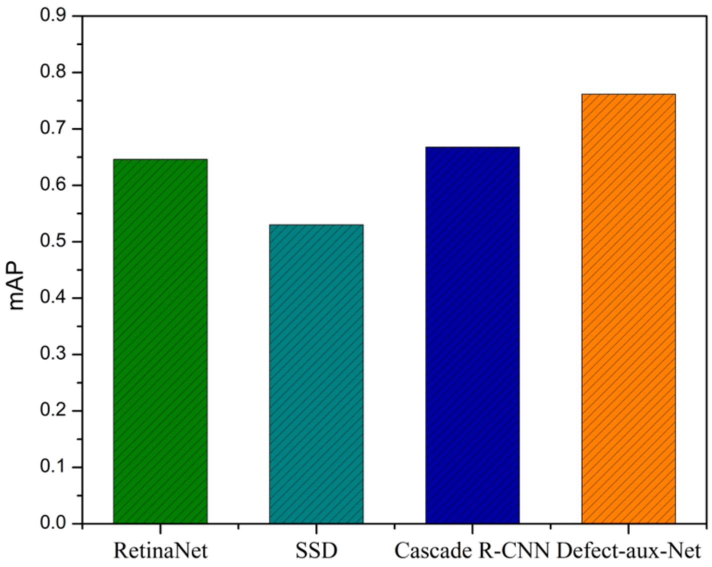


Fig. 2.25: Comparison of the mAP between the state-of-the-art detection models and the Defect-aux-Net.

To evaluate the performance of various loss functions, we conducted a series of experiments on the TekErreka dataset. We trained the Defect-Aux-Net using two different methods: first using the BCE, Jaccard, or Dice loss function as the sole segmentation loss and then using a combination of BCE, Jaccard and Dice loss. The results of these experiments are displayed in Table 2.6. Our goal was to determine the effectiveness of the different loss functions and to see which method produces the best results.

Loss Function	IoU	Dice
BCE	0.892	0.911
Dice	**0.903**	**0.926**
Jaccard	0.9	0.913
Dice + BCE	0.901	0.92
Jaccard + BCE	0.899	0.912

Tab. 2.6: Performance of the Defect-Aux-Net on different loss functions for the defect segmentation task.

Our experimental results also demonstrated that the proposed multi-task learning strategy outperforms the state-of-the-art segmentation models in terms of segmentation performance. The Dice and Intersection over Union (IoU) scores of the various segmentation models on the Severstal dataset are shown in Figure 2.26 and Figure 2.27, respectively. The results reveal that Defect-Aux-Net achieved higher scores for all classes compared to the other segmentation models. Table 2.7 summarizes the performance of the various networks on the TekErreka dataset. The experimental results from Table 2.7 indicate that the proposed multi-task learning approach can improve the performance of its corresponding single task model. By incorporating the classification-guidance module, Defect-Aux-Net avoids over-segmentation of defects in complex backgrounds, allowing for improved segmentation results.

Model	Iou	Dice
FPN [11]	0.881	0.902
LinkNet [28]	0.876	0.895
Unet [10]	0.832	0.856
PSPNet [29]	0.885	0.917
Defect-Aux-Net	**0.903**	**0.926**

Tab. 2.7: Performance of the competing segmentation models on the Tek-Erreka Dataset

In addition to evaluating the performance of the model, we also evaluated the impact of using a multi-task learning framework on inference time. To do this, we compared the inference time of our proposed approach, which uses a multi-task

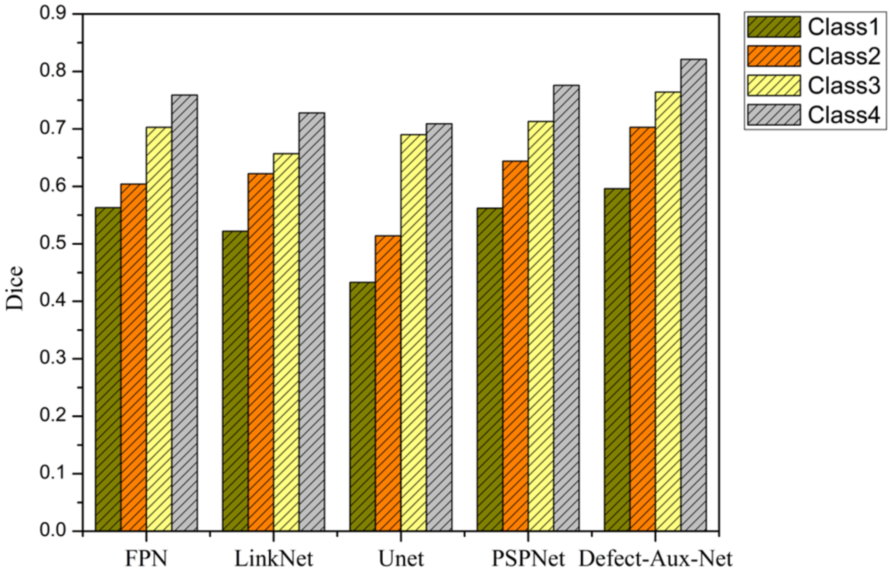


Fig. 2.26: Comparison of Dice scores between the state-of-the-art segmentation methods and the proposed approach for each type of defect.

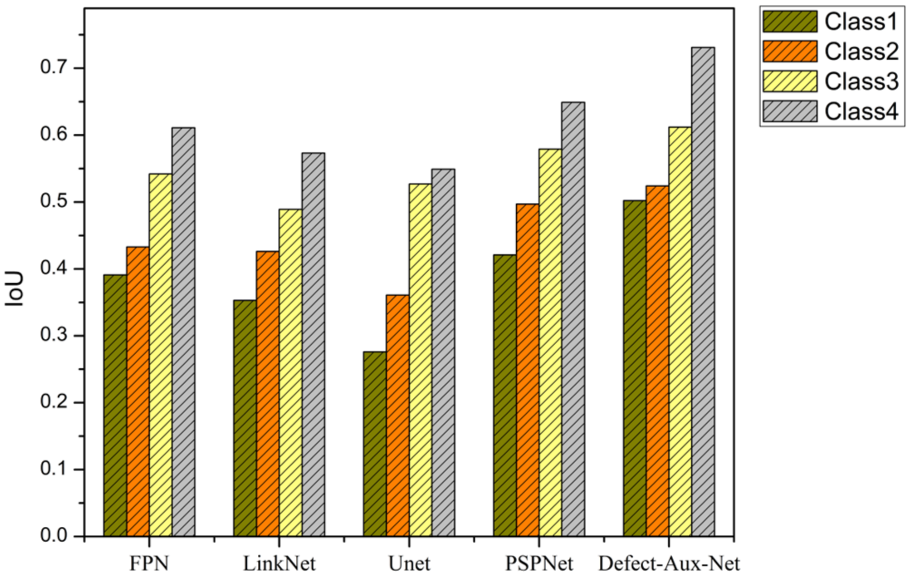


Fig. 2.27: Comparison of IOU between the state-of-the-art segmentation methods and the proposed approach for each type of defect.

learning framework, to a conventional single task network, where each task requires a separate pass through the network during inference. The inference time was measured on a computer with an Intel Core processor and the CPU specification is summarized in Table 2.8.

CPU Specification	
CPU Processor type	Intel(R) Xeon(R)
Processor Base Frequency	2.20 GHz
Total Cores	1

Tab. 2.8: CPU Specification.

Model	**Task**	**Task Name**	**Inference time (s)**	**Parameters (M)**
Single Task Networks	Task 1	Classification (ResNet-50)	0.0654	23.5
	Task 2	Segmentation (ResNet-50 FPN)	0.1106	26.9
	Task 3	Detection (ResNet-50 RetinaNet)	0.1780	34.0
	Total	Classification + Segmentation + Detection	0.3540	84.4
Multitask Network	Multitask	Classification + Segmentation + Detection (Defect- Aux-Net)	**0.1927**	**36.2**

Tab. 2.9: Comparison of the inference time between the Defect-Aux-Net model and the baseline model.

From the results presented in Table 2.9, we can see that our proposed framework leads to a significant reduction in the model size, with a 57.1% decrease compared to solving each task independently. Additionally, compared to the single task network, the inference time of our proposed network was reduced by 45.5%. This demonstrates the effectiveness of the multi-task learning framework in reducing the computational cost of the model without sacrificing its performance.

Conclusion: In this paper, we introduced a novel CNN architecture, named Defect-Aux-Net, which leverages an attention-guided multi-task learning scheme to perform automated surface defect detection. This novel architecture integrates three critical tasks, namely classification, segmentation, and defect detection, into a unified framework, resulting in improved performance and efficiency. The Defect-Aux-Net architecture extends the FPN with a Resnet-50 encoder, forming a hybrid architecture that balances the strengths of both networks. Additionally, we proposed a hybrid loss function that combines multiple loss terms to enhance the model's performance. Evaluation of the Defect-Aux-Net on the TekErreka dataset showed outstanding results, with an overall accuracy of 97.1%, a Dice score of 0.926, and a mAP of 0.762 on the classification, segmentation, and detection tasks. These results demonstrate the effectiveness of the Defect-Aux-Net in detecting surface defects and its potential for widespread use in a variety of industrial applications.

2.2.5 International Conference on Electrical, Computer and Energy Technologies (ICECET) 2022: Vision Transformer based knowledge distillation for fasteners defect detection

With the completion of building CNN models for defect detection through both iterative data-centric approaches in **Journal of Big Data 2021** and **Sensors 2023**, and model-centric approaches in **IEEE Transactions on Industrial Informatics 2023** and **Applied Sciences 2023**, we moved on to model compression techniques in this article. Our objective is to optimize and improve the efficiency of our models through compression without sacrificing their accuracy.

Model compression in CNN refers to the process of reducing the size and computational complexity of a CNN without sacrificing its performance. This is a crucial aspect of deep learning application as the models are becoming increasingly large and complex, making them impractical for deployment in real-world applications.

One of the main reasons for the need for model compression in CNN is the memory constraints of hardware devices. CNN models often require a large amount of memory to store the model parameters and intermediate activations. This can be a major bottleneck in deploying CNN models on devices with limited memory, such as smartphones, edge devices, and embedded systems. In such cases, model compression can be used to reduce the memory footprint of the model and make it possible to deploy it on these devices.

Another reason for the need for model compression is computational efficiency. CNN models require a large amount of computational resources to perform forward and backward passes during training and inference. This can make them impractical for deployment in real-world applications, where the available computational resources are limited. Model compression can be used to reduce the computational requirements of the model, making it possible to run it on devices with limited computational resources, such as embedded systems and edge devices.

In addition to the practical benefits of model compression, there are also theoretical reasons for its importance. For example, it has been shown that CNN models often have a large number of redundant parameters that do not contribute significantly to the model's performance. Model compression techniques can be used to remove these redundant parameters, resulting in a smaller and more compact model that is easier to deploy and maintain.

There are several techniques for model compression in CNN, including pruning [173], quantization [174], low-rank factorization [175] and KD [176].

Pruning:

Pruning [173] is a popular model compression technique that involves removing neurons or connections that are deemed to be redundant or not useful for the task at hand. In the context of CNNs, pruning is used to reduce the size and computational complexity of the model without sacrificing its performance. This is especially important in real-world deployment scenarios, where limited computational resources and memory are often available.

Weight pruning:

There are several types of pruning techniques that have been proposed for CNNs, each with its own strengths and limitations. One of the most widely used pruning techniques is weight pruning [177], which involves removing the smallest weight values in the model. This technique is simple to implement and can achieve significant reductions in the size of the model. However, it can also lead to a loss of accuracy, as removing weights can affect the overall function of the model.

Neuron pruning:

Another popular pruning technique is neuron pruning [178], which involves removing entire neurons from the model. This technique can achieve even greater reductions in the size of the model compared to weight pruning, but it is also more computationally intensive and can lead to a greater loss of accuracy. To mitigate this, neuron pruning techniques often use more sophisticated methods for selecting which neurons to remove, such as using importance scores or criteria based on the structure of the model.

Structured pruning:

A third type of pruning technique is structured pruning [179], which involves removing entire channels or filters from the model. This technique is especially effective in reducing the size of the model, as entire channels can be removed with minimal impact on the performance of the model. Structured pruning techniques can be used to prune the model during the training process, or they can be used to fine-tune a pre-trained model. Structured pruning procedures aims to keep the network's accuracy while enhancing its efficiency.

Hybrid pruning:

Finally, there are hybrid pruning [180] techniques, which combine multiple pruning techniques to achieve the best results. For example, a hybrid pruning technique might first use weight pruning to reduce the size of the model, and then use neuron pruning to further reduce the size while minimizing the loss of accuracy.

Quantization:

Quantization is a widely used model compression technique in deep learning, particularly in CNNs. It involves reducing the precision of the model parameters, which can significantly reduce the memory footprint of the model and increase its computational efficiency.

There are several types of quantization techniques used in CNNs, each with its own strengths and limitations. These techniques can be broadly classified into two categories: weight quantization [174] and activation quantization.

Weight quantization:

Weight quantization [174] involves reducing the precision of the model weights from floating-point values to integers or fixed-point values. This can result in a substantial reduction in memory footprint, as integer and fixed-point values take up much less memory compared to floating-point values. However, weight quantization can also result in a reduction in accuracy, as the reduced precision of the weights may lead to a loss of information. To mitigate this issue, various weight quantization techniques have been proposed, including uniform quantization, non-uniform quantization, and k-means quantization.

Activation quantization:

Activation quantization [181] involves reducing the precision of the activations generated during forward propagation. This can further reduce the memory footprint of the model and increase its computational efficiency. Similar to weight quantization, activation quantization can also result in a reduction in accuracy, as the reduced precision of the activations may lead to a loss of information. To mitigate this issue, various activation quantization techniques have been proposed, including uniform quantization, non-uniform quantization, and k-means quantization.

Mixed-precision quantization:

Another type of quantization technique that is commonly used in CNNs is mixed-precision quantization [182]. This involves using different precisions for different parts of the model, such as using higher precision for the weights and lower precision for the activations. This can provide a balance between memory efficiency and accuracy, as the high precision weights ensure that the information is not lost, while the low precision activations reduce the memory footprint and computational complexity of the model.

Low-rank factorization:

The main idea behind low-rank factorization is to decompose the model weights into a lower-dimensional representation, which can significantly reduce the memory footprint and computational complexity of the model. This is achieved by factorizing the weight tensors into the product of two or more low-rank matrices, which capture the essential information contained in the original weights.

There are several types of low-rank factorization techniques used in CNNs, each with its own strengths and limitations. These techniques can be broadly classified into two categories: matrix factorization and tensor factorization.

Matrix factorization: Matrix factorization techniques [183] involve factorizing the weight matrices into the product of two or more low-rank matrices. This can significantly reduce the memory footprint and computational complexity of the model, as the size of the weight matrices is reduced. The most common matrix factorization technique used in CNNs is singular value decomposition (SVD), which decomposes the weight matrices into the product of three matrices: a diagonal matrix containing the singular values, a unitary matrix containing the left singular vectors, and a unitary matrix containing the right singular vectors.

Tensor factorization: Tensor factorization techniques [184] involve factorizing the weight tensors into the product of two or more low-rank matrices. This can provide an even more compact representation of the model weights, as the size of the weight tensors is reduced. The most common tensor factorization technique used in CNNs is tensor train (TT) decomposition, which decomposes the weight tensors into the product of a sequence of matrices with a fixed rank.

low-rank approximation: Another type of low-rank factorization technique that is widely used in CNNs is low-rank approximation [175]. This involves approximating the weight tensors with a lower-rank tensor, which captures the essential information contained in the original weights. Low-rank approximation can be achieved using various methods, such as SVD, TT decomposition, and matrix factorization.

knowledge distillation:

The basic idea behind KD [176] is to use the outputs of a larger, more complex model, called the teacher model, to train a smaller, simpler model, called the student model. By using the teacher model's outputs as supervision, the student model can learn the features more efficiently as shown in Figure 2.28. In the context of fastener defect detection, using KD to train a lightweight CNN model can significantly reduce the inference time while maintaining a high level of performance.

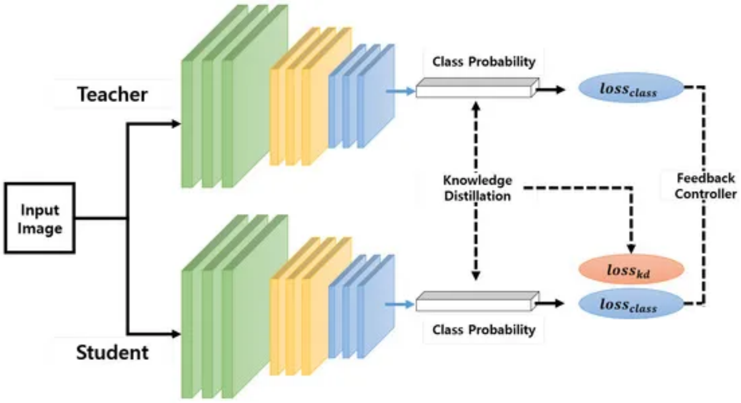


Fig. 2.28: Overview of the knowledge distillation architecture [185].

In this study, we focused on utilizing the advantages of KD and quantization technique to build fast and lightweight CNN models that can be deployed in low-power and resource-limited devices for real-time fastener defect detection system. We employ a compact smaller CNN model (Resnet18) as a student network and a larger pretrained Defect-Aux-Net model as a teacher network in the KD framework. By using a smaller model as the student network, we can reduce the computational complexity and inference time. Additionally, by using a pretrained model as the teacher network, we can leverage the knowledge learned from a large dataset and achieve a high level of performance.

Furthermore, we applied quantization techniques to the student network to reduce the precision of the parameters and activations, which leads to a reduction in the memory and computation requirements. This allows the model to be deployed in low-power and resource-limited devices with minimal computational and storage resources.

In this work, we used Grad-CAM [117] to visualize the predictions of the trained student model for fastener defect identification. To do this, we applied Grad-CAM [117] to a set of training examples and analyzed the resulting activation maps. As shown in Figure 2.29, our results indicated that the student network was basing its predictions not on background or noise components within the fasteners, but on the actual defects. This observation is important because it shows that the student network has learned to effectively identify and focus on the relevant features in the input images. Furthermore, the use of Grad-CAM helped us to gain a better understanding of how the student network was making its predictions. By visualizing the class activations, we were able to see which regions of the fasteners were most important for the model to correctly classify the images as containing defects or not.

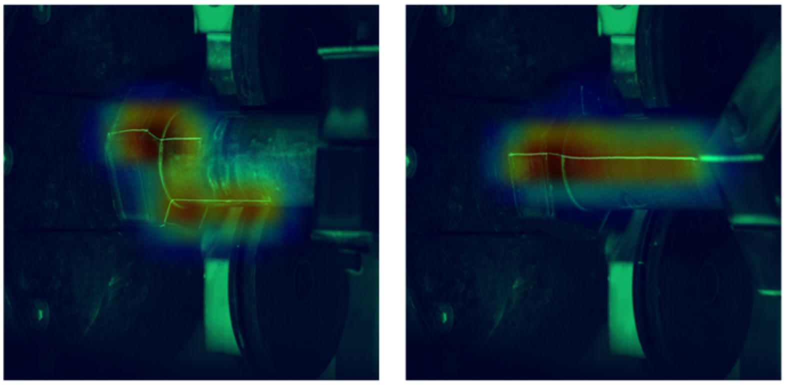


Fig. 2.29: Example results of GRAD-CAM visualization

Conclusion:

In conclusion, our study focused on utilizing the advantages of KD and quantization techniques to build fast and lightweight CNN models that can be deployed in low-power and resource-limited devices for real-time fastener defect detection system. By using a light-weight ResNet 18 as the student network in the KD framework, we were able to significantly reduce the model size and speed up the real-time inference on edge devices. Moreover, to further optimize the student network, we used quantization as a post-model optimization step. This quantization method allowed us to achieve 4x reduction in the model size and 4x reduction in memory bandwidth requirements. This approach can be applied to various manufacturing industries for efficient and accurate fastener defect detection, leading to increased production efficiency and improved product quality.

3. RESEARCH ARTICLES

Research Articles

3

Sampath *et al. J Big Data (2021) 8:27*
https://doi.org/10.1186/s40537-021-00414-0

Journal of Big Data

SURVEY PAPER **Open Access**

A survey on generative adversarial networks for imbalance problems in computer vision tasks

Vignesh Sampath[1,2*], Iñaki Maurtua[1], Juan José Aguilar Martín[2] and Aitor Gutierrez[1]

*Correspondence: vignesh.sampath@tekniker.es
[1] Autonomous and Intelligent Systems Unit, Tekniker, Member of Basque Research and Technology Alliance, Eibar, Spain
Full list of author information is available at the end of the article

Abstract

Any computer vision application development starts off by acquiring images and data, then preprocessing and pattern recognition steps to perform a task. When the acquired images are highly imbalanced and not adequate, the desired task may not be achievable. Unfortunately, the occurrence of imbalance problems in acquired image datasets in certain complex real-world problems such as anomaly detection, emotion recognition, medical image analysis, fraud detection, metallic surface defect detection, disaster prediction, etc., are inevitable. The performance of computer vision algorithms can significantly deteriorate when the training dataset is imbalanced. In recent years, Generative Adversarial Neural Networks (GANs) have gained immense attention by researchers across a variety of application domains due to their capability to model complex real-world image data. It is particularly important that GANs can not only be used to generate synthetic images, but also its fascinating adversarial learning idea showed good potential in restoring balance in imbalanced datasets.

In this paper, we examine the most recent developments of GANs based techniques for addressing imbalance problems in image data. The real-world challenges and implementations of synthetic image generation based on GANs are extensively covered in this survey. Our survey first introduces various imbalance problems in computer vision tasks and its existing solutions, and then examines key concepts such as deep generative image models and GANs. After that, we propose a taxonomy to summarize GANs based techniques for addressing imbalance problems in computer vision tasks into three major categories: 1. Image level imbalances in classification, 2. object level imbalances in object detection and 3. pixel level imbalances in segmentation tasks. We elaborate the imbalance problems of each group, and provide GANs based solutions in each group. Readers will understand how GANs based techniques can handle the problem of imbalances and boost performance of the computer vision algorithms.

Keywords: Generative adversarial neural networks, Imbalanced data, Object detection, Segmentation, Classification, Deep learning, Deep generative model

Introduction

Recent developments in Convolutional Neural Networks (ConvNets) have led to substantial progress in the performance of computer vision tasks applied across various domains such as self-driving cars [1], medical imaging [2], agriculture [3, 4], manufacturing [5], etc. The availability of big data [6], together with increased computing capabilities is the predominant reason for the recent success. Image acquisition is the first step in the development of computer vision algorithms. When the acquired image is not adequate, the desired task may not be possible to achieve. Image classification [7], object detection [8] and segmentation [9] are the fundamental building blocks of the computer vision tasks. All these methods use deep ConvNets with enormous layers and have a very high number of parameters that need to be tuned. Therefore, they demand a huge amount of representative data to improve their performance and generalization ability. While the amount of visual data is increasing exponentially, many of the real-world datasets suffer from several forms of imbalance. Handling imbalances in the image dataset is one of the pervasive challenges in the field of computer vision.

Image classification is the task of classifying an input image according to a set of possible classes. Classification algorithms learn to isolate important distinguishing information about an object in an image like shape or color and ignore irrelevant parts of an image such as plane background or noise. Several popular image classification architectures such as LeNet [7], AlexNet [10], VGG-16 [11], GoogLeNet [12], ResNet [13], Inception-V3 [14], DenseNet [15] take an input image and then pass it through several convolutional and pooling layers. Convolutional layer helps to extract features from the input image, while a pooling layer reduces the dimension. Several successive convolutional and pooling layers may follow, depending on the layout and intent of the architecture. The result is a set of feature maps reduced in size from the original image that through a training process have learned to distill information about the content in the original image. All extracted feature maps are then transformed into a single vector that can be fed into a series of fully connected neural network to obtain a probability distribution of class scores. The predicted class for the input image can be extracted from this probability distribution.

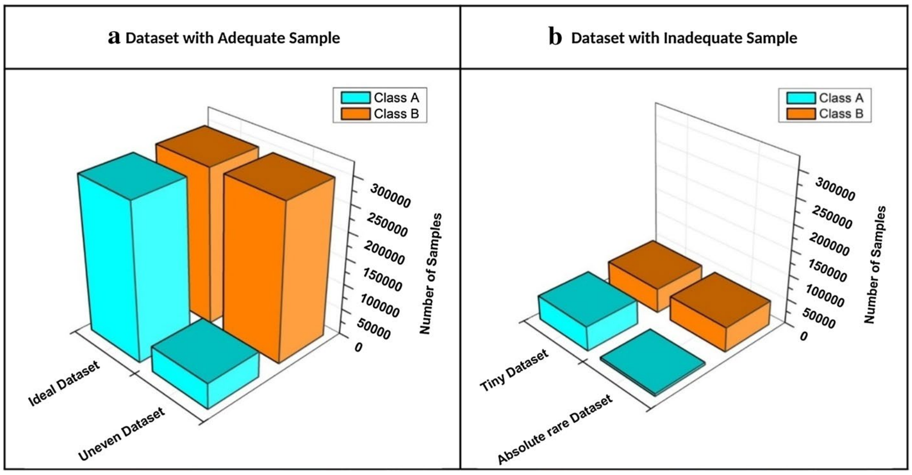
These architectures are typically designed to work well with balanced datasets, but a common issue with real-world datasets is the imbalance of observed classes. The most commonly known imbalance problem in a task of image classification is the class imbalance. Class imbalance in the real-world image datasets is ubiquitous and can have an adverse effect on the performance of ConvNets [16]. These datasets usually fall into four categories in terms of its size and imbalance [17]:

1. The ideal datasets are the one that contain an adequate and equal or almost equal number of samples within each class. An equal probability is assigned to all classes during training to update parameters of the network and approach the minimum value of the error function. A wide range of standard machine learning algorithms can be applied for the ideal datasets.
2. The datasets with an adequate number of samples where some instances of classes are rarer than other instances of classes are said to be uneven datasets. Even though these datasets have adequate number of samples, it is costly and may not be possible for experts to manually inspect huge unlabeled datasets to annotate.
3. Tiny datasets are not easily available, and they can be difficult to collect. Such datasets have an equal number of samples within each class, but they are almost impossible to collect due to privacy restriction and other reasons.

4. Absolute rare datasets have a limited number of samples and substantial class imbalance. Reasons for class imbalance in these datasets can vary but commonly the problem arises because of: (a) Very limited number of experts available for data collection; for an example, generation of medical imaging datasets requires specialized equipment and well trained medical practitioners for data acquisition (b) Enormous manual effort required to label datasets; and (c) Scarcity of samples of specific class leading to class imbalance. Consequently, the size of the dataset and class imbalance problem becomes a bottleneck that prevents us from tapping the true potential of ConvNets. Figure 1 illustrates different types of datasets in terms of its size and imbalance.

Class imbalance in a dataset can stem from either between classes (inter class imbalance) or within class (intra class imbalance). Inter class imbalance occurs when a minority class contains a smaller number of instances when compared to instances belonging to the majority class. Classifiers built using inter class imbalanced datasets are most likely to predict minority class as rare occurrences, even sometimes assumed as outlier or noise which results in misclassification of minority classes [18]. Minority classes are often of greater interest and significance, that needs to be cautiously handled. For example, in a rare disease medical diagnosis where there is a vital need to distinguish such a rare medical condition among the normal populations. Any kind of diagnosis errors will cause stress to the patient and further complications. It is therefore very important that deep learning models [19] built using such datasets should be able to achieve a higher detection rate on minority classes.

Intra class imbalance in a dataset can also deteriorate the performance of the classifier. An Intra-class imbalance can be viewed as the attribute bias within a class, in other words inter-class imbalance in fine-grained visual categorization. For example, a class of dog samples can be further categorized by dog color, pose variations and dog breeds. Imbalances in such categories (intra class imbalance) is an unavoidable problem in


a Dataset with Adequate Sample

b Dataset with Inadequate Sample

Fig. 1 Distribution of different type of datasets (**a**) Dataset with adequate sample (**b**) Dataset with inadequate sample

datasets of many classification tasks such as modality based medical image classification [19], fine grained attribute classification [20], person re-identification [21], age [22] and pose invariant face recognition [23].

Several attempts have been made to overcome the problem of class imbalance by using different approaches and techniques. These techniques can be grouped into data-level approaches, algorithm level methods and hybrid techniques. While data level approaches modify the distribution of training set to restore balance by adding or removing instances from the training dataset, algorithm level methods change the objective function of the classifier to increase the importance of the minority class. Hybrid techniques combine algorithm level methods with data level approaches. Next few paragraphs will inform readers about some of the traditional techniques available to counter the class imbalance problem.

- *Resampling* To counteract the class imbalance problem, two types of re-sampling can be applied: One is under sampling by deleting samples from the majority class and another is oversampling by duplicating samples from the minority class [24]. Re-sampling method balances the dataset but fails to provide any additional information to the training set. The other limitations of this method include: oversampling results in over fitting problem while under sampling leads to substantial loss of information [25]. The quantity of under-sampling and oversampling is generally determined using experimental methods and empirically established [26]. In order to yield additional information to the training set, synthetic oversampling methods create new samples instead of duplicates to add equilibrium to skewed distribution. The Synthetic Minority Oversampling Technique (SMOTE) [27] is a popular synthetic oversampling method that aims to generate synthetic samples based on randomly selected K-nearest neighbors. SMOTE does not take account of the distribution of data between the classes. Adaptive synthetic sampling (ADASYN) approach [28] uses a weighted distribution for different minority classes according to their learning difficulties to adaptively generate synthetic data samples. Cluster based oversampling [29] technique divides the input space into various clusters and then incorporates sampling to alter the sample size. Many traditional synthetic oversampling techniques such as SMOTE or ADASYN are only suitable for low dimensional tabular data which restricts their application in a high dimensional image data. In addition, all the aforementioned techniques generate data by either deleting or averaging existing data, and hence may fail to improve classification performance.
- *Augmentative oversampling* Data augmentation is another commonly used technique to inflate the size of the training dataset [30]. Augmentation such as translation, cropping, padding, rotation and horizontal flipping introduces small modifications in the image data, but not all these modifications will improve the performance of a classifier. There is no standard method that can decide whether any particular augmentation strategy can improve results until the training process is complete. As training ConvNets is a time-consuming process [31], only a restricted amount of augmentation strategy is likely to be tested before model deployment. Also, the diversity that can be obtained from small modifications of the images is relatively small. In addition to balancing classes by oversampling, augmentation techniques

also serve as a kind of regularization in deep neural network architecture and hence reduce the chance of over fitting. There is no consensus about the best strategy for combining different augmentation strategies together. Therefore, more advanced augmentation techniques such as mixing images depend on expert knowledge for validation and labelling [32]. A complete survey of Image data augmentation for deep learning has been compiled by Shorten et al. [32].

- *Semi-supervised learning (SSL)* SSL [33] is one of the most attractive ways to improve classification performance where we have access to small number of labeled samples x x along with large amount of unlabeled samples (Uneven dataset). SSL uses the combination of supervised and unsupervised learning techniques. It makes use of small labeled samples as the training set to train the model in a supervised manner, and then use the trained model to predict on the remaining unlabeled portion of the dataset. The process of labeling each sample of unlabeled data with the individual outputs predicted for them using the trained model is known as pseudo labeling. After labeling the unlabeled data through the pseudo labeling process, classification model is trained on both the actual and pseudo labeled data. Pseudo labeling is an interesting paradigm to annotate large-scale unlabeled data that potentially takes many tedious hours of human labor to manually label them. However, SSL relies on assumptions about the underlying marginal distribution of input data $p(x)$, both the labeled and unlabeled samples are assumed to have the same marginal distribution. This marginal distribution $p(x)$ should contain information about the posterior distribution $p(y|x)$. A complete list of semi supervised learning is detailed in [34].
- *Cost sensitive learning* Majority of the classification algorithms assume that misclassification costs of both minority and majority classes are the same. Cost-sensitive learning [35] pays more attention to misclassification costs of the minority class through a cost matrix.

The most straightforward and commonly used approach in ConvNets is the data driven strategy, because deep ConvNets with enormous layers have a very high number of parameters to be tuned, it is prone to overfitting when trained on a small sized dataset. Data level approaches inflate the training data size that serves as regularization and hence reduce the chance of overfitting in deep neural network architecture. Traditional data-level techniques suffer the following drawbacks, particularly when used for the class imbalance problem in high-dimensional image data.

a. Synthetic instances created using traditional data level approaches may not be the true representative of the training set.
b. Synthetic data generation is achieved either by duplication or linear interpolation which does not generate new examples that are atypical and puzzle the classifier decision boundaries, and hence fail to improve overall performance.
c. In Medical images, augmentation techniques are restricted to minor alteration on an image, as they abide by strict standards. Additionally, the types of augmentation one can use vary from problem to problem. For instance, heavy augmentations such as geometric transformations, random erasing, and mixing images might damage semantic content of the medical image.

d. Applying data augmentation in an absolute rare dataset may not provide the variations required to produce a distinct sample to add equilibrium to skewed distribution.
e. Dealing with the class imbalance in fine-grained visual categorization is challenging because it involves large intra-class variability and small inter-class variability.
f. Most of the techniques are designed only for binary classification problems. Multi class imbalance problems are generally considered much harder than their binary equivalents for many reasons. For Instance, there can be several combinations of minority-majority classes, i.e., they may include: 1. Few minority-Many majority classes, 2. Many minority-Few majority classes, and 3. Many minority-Many majority classes.

Class imbalance in image classification tasks has been widely explored and studied. In addition to class imbalance, there are many different forms of imbalances that can impede performance of other computer vision tasks such as object detection and image segmentation. Object detection, which deals with localization and classification of multiple objects in a given image, is another challenging and significant task in computer vision. The typical way of localizing an object in an image is by drawing a bounding box around the object. This bounding box can be interpreted as a collection of coordinates that define the box. Nowadays, object detection algorithms fall into two broad categories: two-stage detectors and single stage detectors. On one hand, two stage detector such as Region-based Convolutional Neural Networks (R-CNN) [8], Fast R-CNN [36], Faster R-CNN [37], Mask R-CNN [38], etc. employ a Region Proposal Network (RPN) to search objects in the first stage, and then process these region of interests for object classification and bounding-box regression in the second stage. On the other hand, single stage detectors such as Single Shot Detection (SSD) [39], You Only Look Once (YOLO) [40], etc. perform detection on a grid that avoids spending too much time on generating region proposals. Instead of locating objects perfectly, they prioritize speed and recognition. Therefore, one stage object detectors are fast and simple, whereas two stage detectors are more accurate.

Despite the recent advances, applying object detection algorithms to the real-world datasets such as in-car video [41], transportation surveillance images [42] that contain objects with large variance of scales (Objects scale imbalance) remains challenging. Physical size of a same object at different distances from the camera would appear as different size. Singh et al. [43] showed that object level scale variation greatly affects the overall performance of object detectors. Many solutions have been proposed to address the object scale imbalance. Scale aware fast R-CNN [44] uses an ensemble of two object detectors, one for detecting the large and medium scale objects and other for the small scale objects, and then combines them to produce final predictions. Multi-scale Image Pyramids such as SNIP [43] and SNIPER [45] use an image pyramid to build multi scale feature representation. Feature Pyramid Networks (FPN) [46] combine feature hierarchies at different scales to predict objects at different scales.

Objects in the real-world datasets only occupy a small portion of the image, while the rest of the image is background. Both single and two stage algorithms approximately evaluate about 10^4 to 10^5 locations per image [47], yet just a few locations have objects.

The imbalance between foreground (object) and background can also hinder performance of the object detection algorithm. Furthermore, object detection algorithms should be invariant to deformation and occluded objects. In Pedestrian detection Dataset [48], for instance, more than 70% of pedestrians are occluded in at least one frame of a video clip and about 19% of pedestrians are occluded in all frames, where the occlusions are ranked as heavy in almost half of such cases. Dollar et al. [48] highlight that the performance of pedestrian detection using standard detectors declines substantially even under partial occlusion, and drastically under severe occlusion. Data augmentation based on random erasing [49] is a frequently used technique that forces detectors to pay attention to the entire object in an image, rather than just a portion of it. Yet, this technique is not guaranteed to be advantageous in all the conditions. Because skewed distributions arise even within deformed and occluded objects as some of the occlusions and deformations are uncommon that they hardly occur in practical scenarios [50].

Image segmentation that classifies every pixel in an image suffers from pixel level imbalances, as are other computer vision tasks.Some of the well-known image segmentation algorithms include Fully connected network [9], SegNet [51], U-Net [52], ResU-Net [53] etc. Image segmentation is essential for a variety of tasks, including: Urban scene segmentation for autonomous driving [54], industrial inspection [55] and cancer cell segmentation [56]. Datasets of all these tasks suffer from pixel level imbalance. For example, In Urban street scene dataset [57], Pixels corresponding to sky, building and road are far numerous than pixels of pedestrian and bicyclist. This is due to the fact that the area covered by sky, buildings and roads are more than pedestrians and bicyclists in the image. Similarly, In brain tumour image segmentation dataset [58], MRI images have more healthy brain tissue pixels than cancerous tissue pixels. The most frequently used loss function for image segmentation task is a pixel wise cross entropy loss [59]. This loss assigns equal weights to all the pixels, evaluates the prediction for each pixel individually and then averages over all pixels. In order to mitigate this problem, many works have been done which modify the pixel wise cross entropy loss function. The standard cross entropy loss is modified in Weighted cross entropy [52], Focal loss [47], Dice Loss [60], Generalised Dice Loss [61], Tversky loss [62], Lovász-Softmax [63] and Median frequency balancing [51], so as to assign higher importance to rare pixels. Although modified loss functions are efficient for some imbalances, such functions undergo severe difficulties when it comes to highly imbalanced datasets, as seen with medical image segmentations.

In contrast to all the traditional approaches described above, Generative adversarial Neural Networks (GANs) aim to learn underlying true data distributions from the limited available images (both minority and majority class), and then use the learned distributions to generate synthetic images. This raises an interesting question on whether GANs can be used to generate synthetic images for the minority class of various imbalanced datasets. Indeed, recent developments of GANs suggest that being capable to represent complex and high dimensional data can be used as a method of intelligent oversampling. GANs utilize the ability of neural networks to learn a function that can approximate model distribution as close as possible to true distribution. Particularly, they do not rely on prior assumptions about the data distribution and can generate synthetic images with high visual fidelity. This significant property allows GANs to be applied to any kind of imbalance problem in computer vision tasks. GANs can not only be able to generate a fake image, but also offer a way to change something about the original image. In other words, they can learn to produce any desired number of classes (such as, objects, identities, people, etc.), and across many variations (such as, viewpoints, light conditions, scale, backgrounds, and more). There are a wide variety of GANs reported in the literature, each with their own strengths to alleviate imbalance problem in computer vision tasks. For instance, AttGAN [64], IcGAN [65], ResAttrGAN [66], etc. are a specific variant of GANs that are commonly used for facial attribute editing tasks. They learn to synthesize not only a new face image with desired attributes but also preserves attribute independent details. Recently, GANs have been combined with a wide range of existing object detection and image segmentation algorithms to overcome the problem of imbalance and improve their performance.

The original GANs architecture [67] contains two differentiable functions represented by two networks, a generator G and a discriminator D. The learning procedure of GANs is to simultaneously train a discriminator D and a generator G. It follows an adversarial two-player, zero-sum game. An intuitive way of understanding GAN is with the police and the counterfeiter anecdote. The generator network is like a group of counterfeiters trying to produce fake money and make it look genuine. The police attempt to discover counterfeiters using fake money, yet at the same time need to let every other person spend their real money. Over time, the police show signs of improvement at identifying fake cash, and the forgers improve at faking it. In the end, the counterfeiters are compelled to make ideal copies of real money. High resolution and realistic minority class images generated using learned model distribution can be used to balance the class distribution and mitigating effect of over fitting by inflating the training dataset size. GANs solve the problem of generating data when there is not enough data to begin with and they require no human supervision. GANs can provide an efficient way to fill in holes in the discrete distribution of training data. In other words, they can transform the discrete distribution of training data to continuous, providing an additional data by non-linear interpolation between the discrete points. Bowles et al. [68] argues that GANs offer an access to unlock additional information from a dataset. In fact, Yann LeCun, the facebook vice president and chief AI scientist, referred to GANs as "*the most interesting thing that has happened to the field of machine learning in the last 10 years*".

In this survey, as opposed to other related surveys on class imbalance, that present class imbalance in tabular data, we focus on wide range of imbalance in high dimensional image data by following a systematic approach with a view to help researchers establish a detailed understanding of GAN based synthetic image generation for the imbalance problems in computer vision tasks. Furthermore, our survey covers imbalances in a wide range of computer vision tasks in contrast to other surveys that are limited to image classification tasks.

The key contributions of this survey are presented as follows:

- In this survey paper, we review current research work on GAN based synthetic image generation for the imbalance problems in visual recognition tasks spanning from 2014 to 2020. We group these imbalance problems in a taxonomic tree with three main groups: Classification, Object detection and Segmentation (Fig. 2).

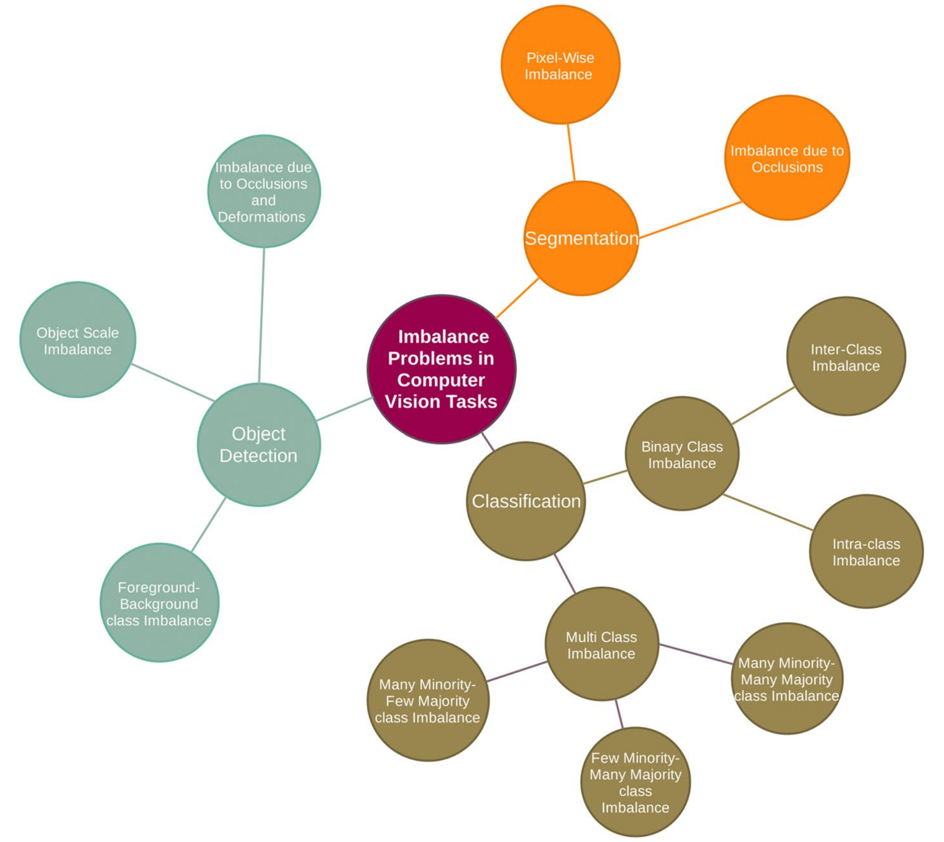

Pixel-Wise Imbalance
Imbalance due to Occlusions
Segmentation
Imbalance due to Occlusions and Deformations
Object Scale Imbalance
Imbalance Problems in Computer Vision Tasks
Inter-Class Imbalance
Object Detection
Binary Class Imbalance
Classification
Intra-class Imbalance
Foreground-Background class Imbalance
Multi Class Imbalance
Many Minority-Many Majority class Imbalance
Many Minority-Few Majority class Imbalance
Few Minority-Many Majority class Imbalance

Fig. 2 Proposed taxonomy for the review of imbalanced problem in computer vision tasks

- Also, we provide necessary material to inform research communities about the latest development and essential technical components in the field of GAN based synthetic image generation.
- Apart from analyzing different GAN architectures, our survey focuses heavily on real world applications where GAN based synthetic images are used to alleviate imbalances and fills a research gap in the use of synthetic images for the imbalance problems in visual recognition tasks.

The remainder of this paper is organized as follows: "Deep Generative image models" section gives readers necessary background information on generative models. "Generative adversarial Neural Network" section discusses selected GAN variants from the architecture, algorithm, and training tricks perspective in detail. In "Taxonomy of class imbalance in visual recognition tasks" section, we provide a brief explanation on various types of imbalances encountered in visual recognition tasks and how the GAN based synthetic image is used to rebalance, followed by GAN variants from the application perspective. "Discussion and Future work" section identifies and enumerates our perspective and possible future research direction. Finally, we conclude the paper in "Conclusion" section.

Deep generative image models

Deep Generative model is an important family of unsupervised learning methods that are dedicated to describe the underlying distribution of unlabeled training data and learn to generate brand new data from that distribution. Color image data [32] is pixel values encoded into a three-dimensional stacked array, made up of height, width, and three-color channels. Modeling the distribution of image data is extremely challenging as natural images are high dimensional and highly structured [69]. This challenge has led to a rich variety of neural network based generative image models, each having their own advantages. Research into neural network based generative models for image generation has a long history. Restricted Boltzmann Machines [70–72] and their deep variants [73–75] are a popular class of probabilistic models for image generation. Now the generative image models can be grouped into three broad categories: 1. Autoregressive models, 2. Latent variable models and 3. Adversarial learning-based models.

Autoregressive models (ARs) aim to estimate a distribution over images (density estimation) using a joint distribution of the pixels in the image by casting it as a product of conditional distributions [76]. ARs transform the problem of joint modeling into a sequence problem, where, given all the pixels previously generated, one learns to predict the next pixel. But a highly powerful sequence model is needed to model the highly non-linear and long span auto correlations between the pixels. Based on this idea, many research articles have been published that use different sequence models from deep learning to model the complex conditional distribution. Fully visible belief network (FVBN) [77, 78] is one of the tractable explicit density models that use chain rule to factorize likelihood of an image x into product of one dimension distributions, where n × n. pixels in the greyscale image is taken row by row as a one dimensional sequence $\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3 \ldots, \mathrm{x}_{\mathrm{n}^2}$. The joint likelihood p(x) is explicitly computed as the product of the conditional probabilities over the pixels. The conditional distribution of each pixel in an image is calculated as shown in Eq. (1).

$$p(x) = \prod_{j=1}^{n^2} p(x_j|x_1, x_2, ...x_{j-1}) \tag{1}$$

ven all the preceding pixels $x_1, x_2 \ldots x_{j-1}$, the value $p(x_j|x_1, x_2, ...x_{j-1})$ is the probability of the j-th pixel x_j. Each pixel is dependent on previous pixels that have been already generated. The pixel generation starts from the corner, continues pixel by pixel and row by row. In the case of an RGB image, each pixel value in an individual RGB color is jointly computed by three values, one for each of the RGB color channels. The conditional distribution $p(x_j|X < j)$ can be rewritten as the following product (Eq. (2)) where green channel is conditioned on channel red and blue channel is conditioned on channels red and green.

$$p(x_j, R|X < j)p(x_j, G|X < j, x_j, R)p(x_j, B|X < j, x_j, R, x_j, G) \tag{2}$$

Generating an image pixel by pixel using this approach is sequential, computationally intense, and a very slow process as each of the colour channels is conditioned on the other channels as well as on all the pixels generated previously (Fig. 3).

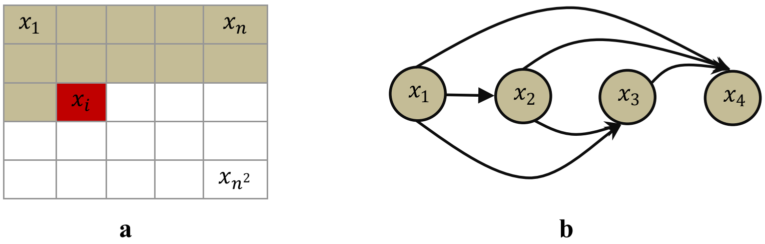


Fig. 3 Autoregressive models train a network that models conditional distribution of each pixel given all previous pixels. The image is processed pixel-by-pixel in (**a**) Raster scan order and (**b**) Sequentially predicts pixels

Neural Autoregressive Density Estimator (NADE) [79] aims to learn a joint distribution using a neural network to parametrize the factors of p(x). The output layer of the NADE is designed to predict n conditional probability distributions, each node in the output layer corresponds to one of the factors in the joint distribution. Hidden representation for each output node is computed using only relevant inputs, i.e. only previous i − 1 input variables are connected to the ith output. By implementing a neural network, NADE allows weights sharing that reduces the number of parameters to learn a joint distribution using stochastic gradient descent.

Recurrent neural networks (RNN) have been proved to excel at various sequential tasks, such as speech recognition [80], speech synthesis [81], handwriting recognition [82], and image to text [83]. Particularly, Long Short-Term Memory (LSTM) layers [84], transformers and self-attention mechanism [85] are the robust architecture for modeling long range sequence data with auto correlations like time series data, natural languages etc. In order to have a long-term memory, LSTM layer adds gates to the RNN. It has an input to state component and a recurrent state to state component that together determine the gates of the layer. Theis et al. [86] used spatial LSTM (sLSTM), a multidimensional LSTM which is suitable for image modeling because of its spatial structure. However, an immense amount of time is needed to train the LSTM layers considering the number of pixels in the larger datasets such as CIFAR-10 [87] and ImageNet [88].

Van den Oord et al. [69] designed two variants of recurrent image models: PixelRNN and PixelCNN. The pixel distributions of the natural images are modeled with two-dimensional LSTM (spatial LSTMs) and convolutional networks in PixelRNN and PixelCNN respectively. Convolution operation enables PixelCNNs to generate pixels faster than PixelRNNs, given the large number pixels in natural images. But typically, PixelRNNs achieve higher performance when compared to PixelCNNs. Gated PixelCNN [89] is another interesting paradigm to generate diverse natural images with a density model conditioned on prior information along with previously generated pixels. The prior information h in Eq. (4) can be any vector, including class labels or tags.

$$p(x|h) = \prod_{j=1}^{n^2} p(x_j|x_1, x_2, ...x_{j-1}, h) \tag{3}$$

A lot of work on improving performance of PixelCNN has been reported in literature by introducing new architectures, loss functions and different training tricks.

PixelCNN+ +[90] enhances the performance of PixelCNN by proposing numerous modifications while retaining its computational performance. Major modifications include: 1. Intensity of a pixel is viewed as 8-bit discrete random variables and modeled using 256-softmax output in pixelCNN. In contrast, PixelCNN+ +uses discretized logistic mixture likelihood to model each pixel as real valued output. 2. It simplifies the model structure by conditioning on entire pixels, instead of RGB sub space. 3. PixelCNN+ +employs down-sampling by using convolution of stride 2 in order to capture structure at multiple resolutions 4.Short cut connections are added to compensate the loss of information due to down-sampling. 5. PixelCNN+ +also introduces model regularization using dropouts. Pixel Snail [91] incorporates a self-attention mechanism in PixelCNN to have access to long term temporal information.

Latent variable models on the other hand, aim to represent high dimensional image data (observable variables) into lower dimensional latent space (latent variables). Latent variables as opposed to observable variables are variables that are not directly observed but inferred through a model from other variables that are observed directly. One advantage of using latent variable is that it reduces dimensionality of data. High dimensional observable variables can be aggregated in a model to represent an underlying concept making it easier to understand the data.

Autoencoders are one of the latent variable models that take unlabeled high dimensional image data x, after encoding them into lower dimensional feature representation z, try to reconstruct them as accurately as possible. The lower dimensional feature z is a compressed representation of an input image, as a result, the autoencoder must decide which of the features in an image are the most important, essentially acting as a feature extraction engine or dimensionality reduction. They are typically very shallow neural networks, and usually consist of an input layer, an output layer, and a hidden layer. Autoencoders with nonlinear encoder and decoder functions learn to project image data onto a nonlinear manifold, which are capable of performing powerful nonlinear generalization compared to principle component analysis (PCA). They are trained with backpropagation, using a metric called Reconstruction loss. Reconstruction loss measures the amount of information that was lost when an autoencoder tried to reconstruct the input, using pixel wise L1 or L2 distance. In other words, pixel wise distance between original images x and reconstructed images $\hat{x}$. Autoencoders with a small loss value can produce reconstructed images that look very similar to the original images.

Traditionally, autoencoders are used for data denoising, data compression and dimensionality reduction. There are many variants of autoencoder proposed in the literature [92–97]. Deep autoencoders [93] use a stack of layers as encoder and decoder instead of limiting to a single layer. Sparse autoencoders [94] have a larger number of hidden neurons than the input or output neurons, but only a fraction of hidden neurons are permitted to be active at once. ConvNets are used as encoder and decoder in convolutional autoencoders [98]. In order to learn a function that is robust to minor variations in its training dataset, contractive autoencoders [96] add a penalty term to its objective function. Denoising autoencoders [92] are stochastic forms of the basic autoencoder that add white noise to the training data to reduce a situation of learning the identity function.

An autoencoder is tweaked to predict the n-conditional distributions rather than just reconstructing the inputs in Masked Autoencoder Density Estimator (MADE)

[99]. In the standard fully connected autoencoder ith output unit depends on all the input units, but in order to predict the conditional distributions, ith output unit should depend only on previous $i-1$ input variables. MADE modifies the autoencoder using a binary mask matrix to ensure each output unit is connected only to relevant input units (Fig. 4). As opposed to autoencoders that are used for an image abstraction, MADE is designed for image generation using learnt distribution (Fig. 4).

Variational Autoencoders (VAEs) [97] are the most popular class of autoencoders. In VAEs, the encoder instead of outputting a latent vector directly, outputs mean μ and variance σ vectors which constitutes latent probability distributions $q_{\emptyset}(z|x)$ from which a latent vector is sampled. This means that given the same input image, no two latent vectors sampled are the same, which forces the decoder to learn the mapping from a region of a latent space to a reconstruction rather than just from a single point resulting in a much smoother reconstructed image. Unlike traditional autoencoders, which are only able to reconstruct images similar to training set, VAEs can generate new images close to training set. VAEs are trained by maximizing the variational lower bound (Eq. (4)) also known as evidence lower bound [100].

$$\mathcal{L}_{VAE}(\theta,\emptyset;x,z)=\underbrace{D_{KL}(q_{\emptyset}(z|x)||p(z))}_{Latent\ loss}-\underbrace{E_{z\sim q_{\emptyset}(z|x)}\left(logP_{\theta}(x|z)\right)}_{Reconstruction\ loss} \tag{4}$$

The first term in Eq. (4) is the Latent loss which regularizes the distribution of q to be Gaussian normal distribution $\mathcal{N}(0,1)$ by minimizing Kullback–Leibler divergence (KL divergence). KL divergence measures similarity between the latent probability distribution and the prior distribution using relative entropy. KL divergence from probability distribution q to p is defined to be

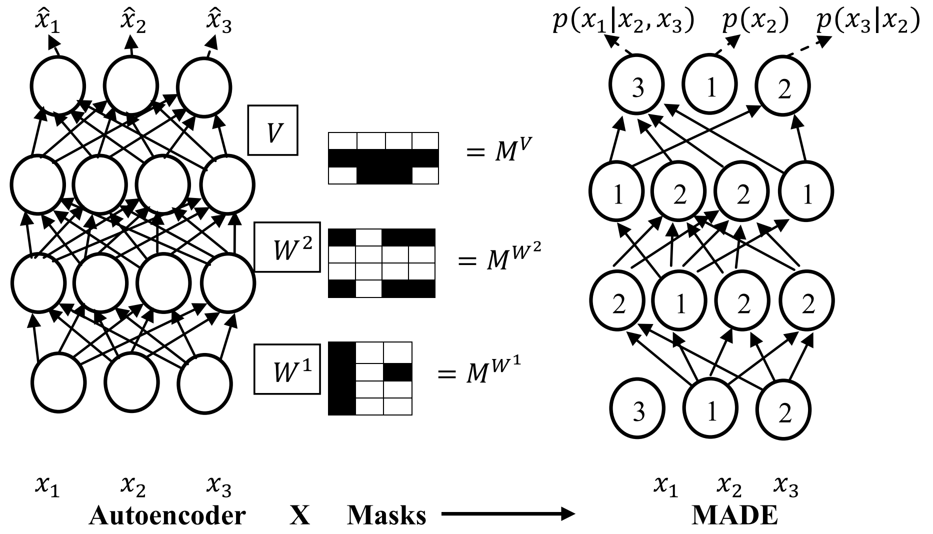


Fig. 4 An illustration of Masked Autoencoder Density Estimator (MADE) [99]. A set of connections in an autoencoder is removed using multiplicative binary masks, such that each output unit is connected only to relevant input units

$$D_{KL}(q||p) = \sum_x q(x)log\frac{q(x)}{p(x)} \tag{5}$$

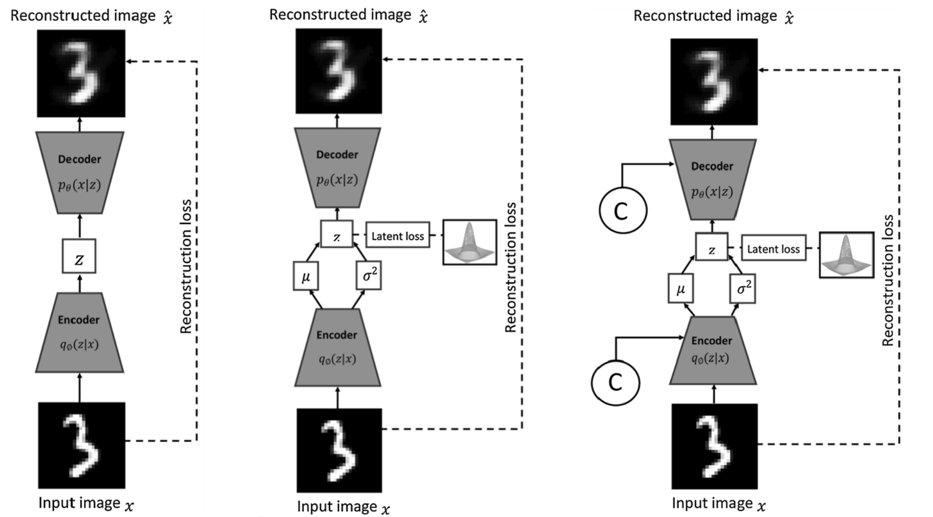
The latent loss is high when the latent probability distribution does not resemble a standard multivariate Gaussian and it is low when the resemblance between those two distributions is close. Given input data x, a probabilistic encoder encodes them to latent representation z with distribution $q_{\emptyset}(z|x)$ and a probabilistic decoder decodes $p_{\theta}(x|z)$. Latent loss enforces the posterior distribution of latent representation z to match with an arbitrary prior distribution $p(z)$. In other words, it imposes a restriction in z, such that input data x are distributed in a latent space following a specified arbitrary prior distribution. The second term, reconstruction loss is pixel wise Binary cross entropy between original image x and reconstructed image $\hat{x}$.

The numerous modifications have been made over basic VAEs that was initially introduced in [97]. The Conditional VAE (CVAE) [101] is a conditioned version of standard VAEs (Fig. 5c) to generate diverse reconstructed images conditioned on additional information such as class labels, facial attributes etc. Variational lower bound of CVAE is written as

$$\mathcal{L}_{CVAE}(\theta, \emptyset; x, z, c) = D_{KL}(q_{\emptyset}(z|x, c)||p(z, c)) - E_{z\sim q_{\emptyset}(z|x)}\left(logP_{\theta}(x|z, c)\right) \tag{6}$$

Beta VAE (β-VAE) [102] is another modified form of original VAE intended to learn disentangled latent representations that capture the independent features of a given image. It introduces additional hyper parameter β that balances the latent and reconstruction loss. Variational lower bound of β-VAE is defined as

$$\mathcal{L}_{\beta-VAE}(\theta, \emptyset, \beta; x, z) = \beta[D_{KL}(q_{\emptyset}(z|x)||p(z))] - E_{z\sim q_{\emptyset}(z|x)}\left(logP_{\theta}(x|z)\right) \tag{7}$$



a ***Autoencoders*** **b** ***Variational Auto Encoders*** **c** ***Conditional Variational Auto Encoders***

Fig. 5 The architecture of (**a**) Autoencoders; **b** Variational Auto Encoders; **c** Conditional Variational Auto Encoders

When $\beta = 1$ in Eq. (7), it corresponds to the standard VAE framework. β-VAE with $\beta > 1$ pushes the model to learn disentangled representation. Deep Convolutional Inverse Graphics Network (DC-IGN) [103] replaced feed forward neural networks in the encoder and decoder of VAEs with convolution and deconvolution operators respectively. Importance weighted VAE (IWVAE) [104] learns richer and more complex latent space representation than VAEs from importance weighting. Convolutional VAE is combined with the PixelCNN in PixelVAE [105] and Variational lossy autoencoder [106]. Deep Recurrent Attentive Writer (DRAW) [107] networks combine spatial attention mechanism with a sequential variational autoencoder. In order to avoid problems of posterior collapse, Vector Quantized VAE (VQ-VAEs) [108] learns discrete latent representation instead of continuous normal distribution. VQ-VAEs combine VAEs with ideas from vector quantization to get a sequence of discrete latent variables. VQ-VAE 2 [109] is a Hierarchical multi-scale VQ-VAE combined with a self-attention mechanism for generating high resolution images.

Adversarial models try to model the distribution of the real data through an adversarial process. Generative adversarial neural networks based on game theory, introduced by Goodfellow et al. [67] in 2014, is arguably one of the best innovations in recent years. The word adversarial in generative adversarial neural networks means that the two neural networks, the generator and the discriminator are in a competition with each other. The learning procedure of GAN is to simultaneously train a discriminator D and a generator$G$. The generator network takes a noise vector z in a latent space as an input, then runs that noise vector through a differentiable function to transform the noise vector z to create a fake but plausible image x:$G(z) \rightarrow x$. At the same time, the discriminator network, which is essentially a binary classifier, tries to distinguish between the real images (label 1) and artificially generated images by generator network (label 0):$D(x) \rightarrow [0,1]$. Therefore, the objective function of GANs can be defined as

$$\min_{G}\max_{D} V(D,G) = E_{x\sim p_r(x)}[logD(x)] + E_{z\sim p_z(z)}[\log(1-D(G(z)))] \tag{8}$$

Given random noise vector z and real image x, the generator attempts to minimize $\log(1-D(G(z))$ and the discriminator attempts to maximize $logD(x)$ in Eq. (8). For fixed$G$, the optimal D is given by

$$D^*(x) = \frac{p_r(x)}{p_g(x)+p_r(x)} \tag{9}$$

Theoretically, when G is trained to its optimal, the generated data distribution $p_g(x)$ gets closer to the real data distribution $p_r(x)$. If $p_g(x) = p_r(x), D^*(x)$ in Eq. (9) becomes ½. This means that the discriminator is maximally puzzled and cannot distinguish fake images from real ones. When the discriminator D is optimal, the loss function for the generator G can be visualized by substituting in D* (x) Eq. (8).

$$\begin{aligned} G^* &= \max_{D} V\left(G,D^*\right) = E_{x\sim p_r(x)}\left[logD^*(x)\right] + E_{x\sim p_g(x)}\left[\log\left(1-D^*(x)\right)\right] \\ &= E_{x\sim p_r(x)}\left[log\frac{p_r(x)}{\frac{1}{2}[p_g(x)+p_r(x)]}\right] + E_{x\sim p_g(x)}\left[log\frac{p_g(x)}{\frac{1}{2}[p_g(x)+p_r(x)]}\right] - 2log2 \end{aligned} \tag{10}$$

The definition of Jensen-Shannon divergence (D_{JS}) between two probability distributions $p_g(x)$ and $p_r(x)$ is defined as

$$D_{JS}(p_r||p_g) = \frac{1}{2}D_{KL}(p_r||\frac{p_r + p_g}{2}) + \frac{1}{2}D_{KL}(p_g||\frac{p_r + p_g}{2}) \tag{11}$$

Therefore, Eq. (10) is equal to

$$G^* = 2D_{JS}(p_r(x)||p_g(x)) - 2log2 \tag{12}$$

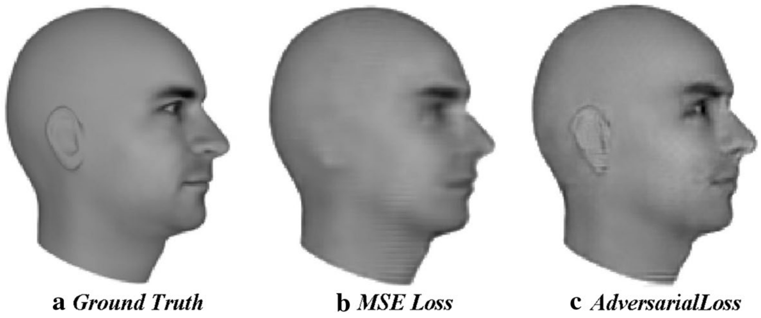
Essentially, the loss for the generator *G* minimizes the Jensen-Shannon divergence between the generated data distribution $\mathrm{p_g(x)}$ and the real data distribution $\mathrm{p_r(x)}$ when discriminator *D* is optimal. Jensen-Shannon divergence is a smooth, symmetric version of the KL divergence. Huszar [110] believes that the main reason behind the great success of GANs is replacing asymmetric KL divergence loss function in the classical approach to symmetric JS divergence.

Mean squared error used in latent variable models such as autoencoder, averages all the possible features in an image and generate blurry images. In contrast, adversarial loss preserves the features using discriminator networks that detect an absence of any features as an unrealistic image. An example of this is the study carried out by Lotter et al. [111], in which models trained using mean square loss and adversarial loss to predict the next image frame in a video sequence are compared. A model trained using mean square loss generates blurry images as shown in Fig. 6, where ear and eyes are not sharply defined as they could be. Using an additional adversarial loss, features like the eyes and ear remain preserved very well, because an ear is the recognizable pattern, and the discriminator network would not accept any sample that is missing an ear.

This section has attempted to provide readers a brief introduction to the current state of deep generative image models. A quick summary of this section is depicted below in Fig. 7.

Despite remarkable achievements in generating sharp and realistic images, GANs suffer from certain drawbacks.

- *Non convergence* Both generator and discriminator networks in GANs are trained simultaneously using gradient descent in a zero-sum game. As a result, improve-



a *Ground Truth* **b** *MSE Loss* **c** *AdversarialLoss*

Fig. 6 An illustration of the importance of an adversarial loss [111]

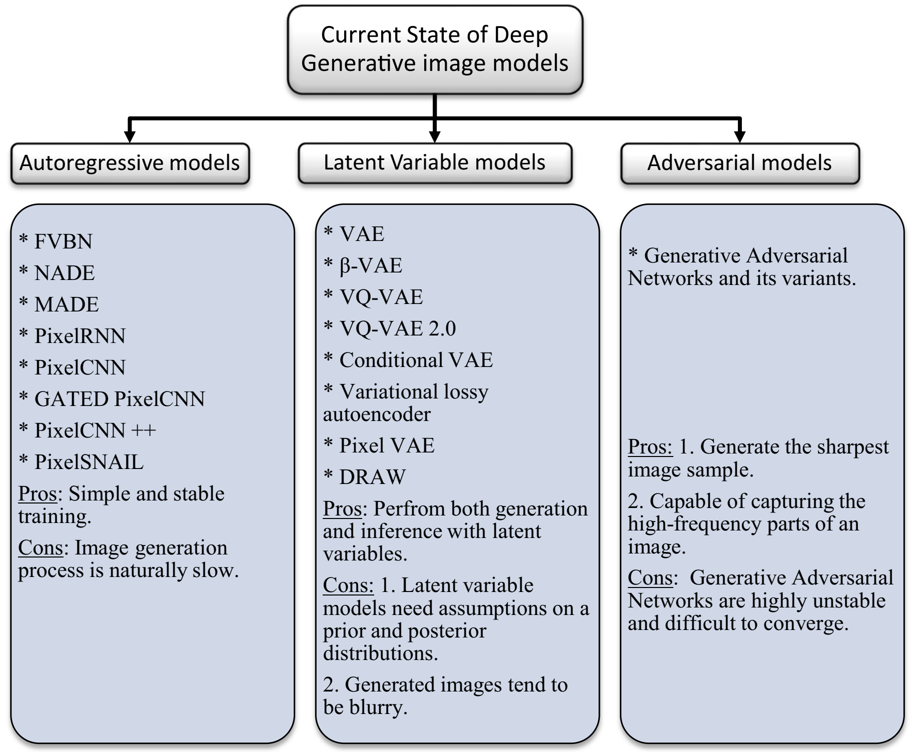

Current State of Deep Generative image models
Autoregressive models
Latent Variable models
Adversarial models
* FVBN
* NADE
* MADE
* PixelRNN
* PixelCNN
* GATED PixelCNN
* PixelCNN ++
* PixelSNAIL
Pros: Simple and stable training.
Cons: Image generation process is naturally slow.
* VAE
* β-VAE
* VQ-VAE
* VQ-VAE 2.0
* Conditional VAE
* Variational lossy autoencoder
* Pixel VAE
* DRAW
Pros: Perfrom both generation and inference with latent variables.
Cons: 1. Latent variable models need assumptions on a prior and posterior distributions.
2. Generated images tend to be blurry.
* Generative Adversarial Networks and its variants.
Pros: 1. Generate the sharpest image sample.
2. Capable of capturing the high-frequency parts of an image.
Cons: Generative Adversarial Networks are highly unstable and difficult to converge.

Fig. 7 Comparative summary of Deep generative models discussed in "Deep Generative image models" section

ment of the generator network comes at the expense of discriminator and vice versa. Hence there is no guarantee of GANs convergence.

- *Mode collapse* Generator network achieves a state where it continues to generate samples with little variety, although trained on diverse datasets. This form of failure is referred to as mode collapse.
- *Vanishing gradient problems* If the discriminator is perfectly trained early in the training process, then there would be no gradients left to train the generator due to vanishing gradients.

Therefore, many GAN-variants have been proposed to overcome these drawbacks. These GAN-variants can be grouped into three categories:

1. *Architecture variants* In terms of architecture of generator and discriminator networks, the first proposed GANs use the Multi- layer perceptron (MLP). Owing to the fact that ConvNets work well with high resolution image data taking into account of the spatial structure of data, a Deep Convolutional GAN (DCGAN) [112] replaced the MLP with the deconvolutional and convolutional layers in generator and discriminator networks respectively.

Autoencoder based GANs such as AAE [113], BiGAN [114], VAE-GAN [115], DEGAN [116], VEEGAN [117] etc., have been proposed to combine their construction power of autoencoders with the sampling power of GANs.

Conditional based GANs like Conditional GAN (CGAN) [118], Auxiliary Classifier GAN (ACGAN) [119], VACGAN [120], infoGAN [121], and SCGAN [122] focused on controlling mode of data being generated by conditioning model on conditional variable.

2. *Training tricks* GANs are difficult to train. Improved trainings tricks such as feature matching, minibatch discrimination, historical averaging, one-sided label smoothing, and Two Time-Scale Update Rule have been suggested to ensure that GANs converge to achieve Nash equilibrium.
3. *Objective variants* In order to improve the stability and overcome vanishing gradient problems, different objective functions have been explored in [123–130].

The following section of this review moves on to describe in greater detail the selected GAN variants.

Generative adversarial neural networks

Architecture variants

The performance and training stability of GANs are highly influenced by the architecture of the generator and the discriminator networks. Various architecture variants of GANs have been proposed that adopt several techniques to improve performance and stability.

i. *Conditional based GAN Variants*

The standard GAN [67] architecture does not have any control on the modes of data being generated. Van den Oord et al. [89] argue that the class conditioned image generation can significantly enhance the quality of generated images. Several conditional based GANs have been proposed that learn to sample from a conditional distribution $p(x|y)$ instead of marginal $p(x)$. Conditional based GANs variants (Fig. 8) can be classified into two groups: 1. Supervised and 2. Unsupervised conditional GANs.

Supervised conditional GANs variants require a pair of images and corresponding prior information such as class label. The prior information could be class labels, textual descriptions, or data from other modalities.

cGAN Mirza and Osindero [118] proposed conditional Generative Adversarial Network (cGAN), to have a control on kind of data being generated by conditioning the model on prior information y. Both discriminator and generator in cGAN are conditioned by feeding y as additional input. Using this prior information, cGAN is guided to generate output images with desired properties during the generation process.

ACGAN Auxiliary classifier Generative Adversarial Network (ACGAN) [119] is an extension of the cGAN architecture. The discriminator in the ACGAN receives only the image, unlike the cGAN that gets both the image and the class label as input. It is modified to distinguish real and fake data as well as reconstruct class labels. Therefore, in addition to real fake discrimination, the discriminator also predicts class label of the image using an auxiliary decoder network.

VACGAN The major problem with ACGAN is that it will affect the training convergence because of mixing the loss of classifier and discriminator into a single loss. Versatile Auxiliary Generative Adversarial Network (VACGAN) [120] separates out classifier loss by introducing a classifier network in parallel to the discriminator.

No prior information is used in unsupervised conditional GAN variants to control on modes of the image being generated. Instead, feature information such as hair color, age, gender etc. is learned during the training process. Therefore, they need an additional algorithm to decompose the latent space into disentangled latent vector c, which contains the meaning features, and standard input noise vector z. The content and representation of an image is then controlled by noise vector z and disentangled latent vector c respectively.

Info-GAN Information maximizing Generative Adversarial Network (Info-GAN) [121] splits an input latent space into the standard noise vector z and additional latent vector c. The latent vector c is then made meaningful disentangled representation by maximizing the mutual information between latent vector c and generated images $G(z, c)$ using additional Q network.

SC-GAN Similarity constraint Generative Adversarial Network (SC-GAN) [122] attempts to learn disentangled latent representation by adding the similarity constraint between latent vector c and generated images $G(z, c)$. Info-GAN uses an extra network to learn disentangle representation, while SC-GAN only adds an additional constraint to a standard GAN. Therefore, SCGAN simplifies the architecture of Info-GAN.

ii. *Convolutional based GAN*

DCGAN Deep Convolutional Generative Adversarial Network (DCGAN) [112] is the first work that deploys convolutional and transpose-convolutional layers in the discriminator and generator, respectively. The salient features of the DCGAN architecture are enumerated as follows:

- First, the generator in DCGAN consists of fractional convolutional layers, batch normalization layers and ReLU activation functions.
- Second, the discriminator is composed of strided convolutional layers, batch normalization layers and Leaky ReLU activation functions.
- Third, uses Adaptive Moment Estimation (ADAM) optimizer instead of stochastic gradient descent with momentum.

iii. *Multiple GANs*

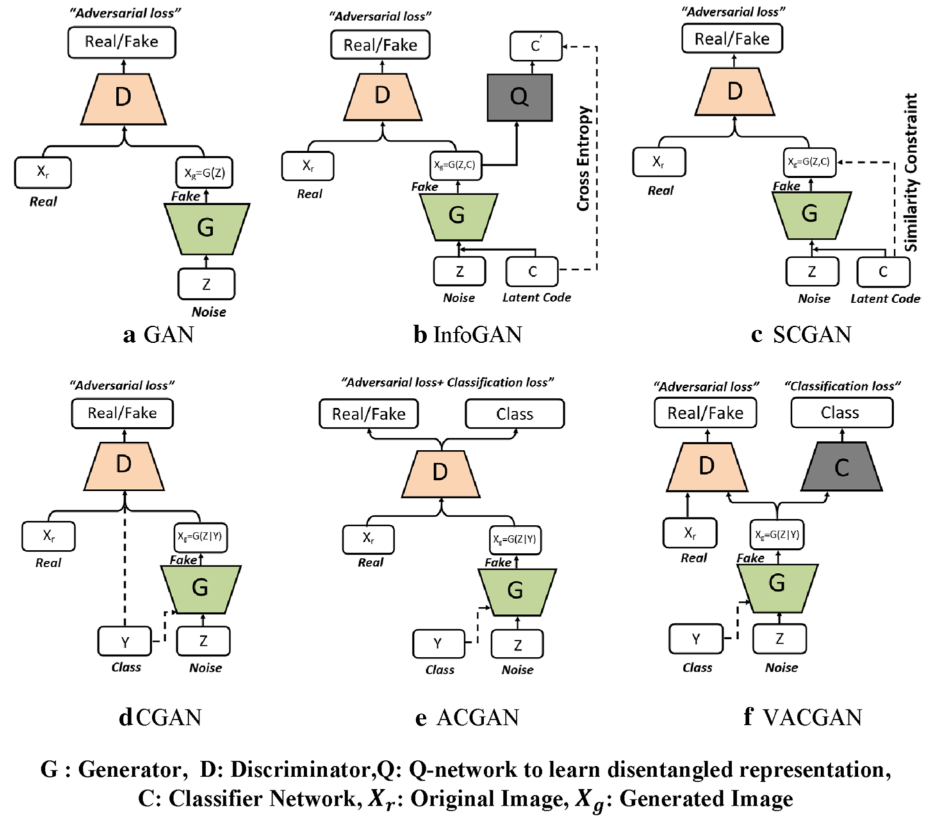


Fig. 8 A schematic view of (**a**) the vanilla GAN and (**b–f**) variants of Conditional GANs

In order to accomplish more than one goal, several frameworks extend the standard GAN to either multiple discriminators, generators, or both (Fig. 9).

ProGAN In an attempt to synthesize higher resolution images Progressive Growing of Generative Adversarial Network (ProGAN) [131] stacks each layer of the generator and discriminator in a progressive manner as training progresses.

LAPGAN Laplacian Generative Adversarial Network (LAPGAN) [132] is proposed for the generation of high quality images. This architecture uses a cascade of ConvNets within a Laplacian pyramid framework. LAPGAN utilizes several Generator-Discriminator networks at multiple levels of a Laplacian Pyramid for an image detail enhancement. Motivated by the success of sequential generation, Im et al. [133] introduced Generative Recurrent Adversarial Networks (GRAN) based on recurrent network that generate high quality images in a sequential process, rather than in one shot.

D2GAN Dual discriminator Generative Adversarial Network (D2GAN) [134] employs two discriminators and one generator to address the problem of mode collapse. Unlike GANs, D2GAN formulates a three-player game that utilizes two discriminators to minimize the KL and reverse KL divergences between true data and the generated data distribution.

MADGAN Multi-agent diverse Generative Adversarial Network (MADGAN) [135] incorporates multiple generators that discover diverse modes of the data while

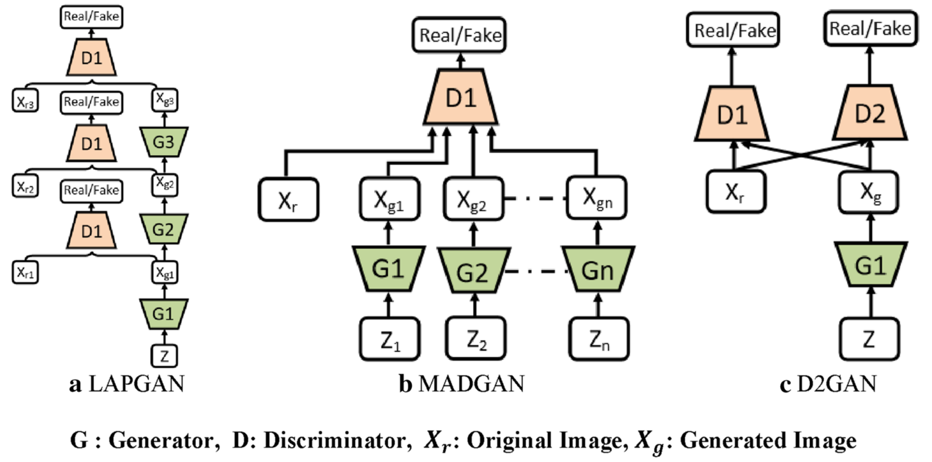


Fig. 9 A schematic view of Variants of GANs with multiple discriminators and generators: **a** LAPGAN, **b** MADGAN and **c** D2GAN

maintaining high quality of generated images. To ensure that different generators learn to generate images from different modes of the data, the objective of discriminator is modified to detect the generator which generated the given fake image along with discriminating the real and fake images.

CoGAN Coupled GAN(CoGAN) [136] is used for generating pair of like images in two different domains. CoGAN is composed of a set of GANs–GAN1 and GAN2, each accountable for synthesizing images in one domain. It leans a joint distribution from two-domain images which are drawn individually from the marginal distributions.

CycleGAN and DiscoGAN [137] use two generators and two discriminators to accomplish unpaired image to image translation tasks. CycleGAN [138] adopts the concept of cycle consistency from machine translation, where a sentence translated from English to Spanish and translate it back from Spanish to English should be identical.

iv. *Autoencoder based GAN Variants*

The standard GANs architecture is unidirectional and can only map from latent space z to data space x, while autoencoders are bidirectional. The latent space learned by encoders is the distribution that contains compressed representation of the real images. Several variants of GANs that combine GAN and encoder architecture are proposed to make use of the distribution learned by encoders (Fig. 10). Attributes editing of an image directly on data space x is complex as image distributions are highly structured and high dimensional. Interpolation on latent space can facilitate to render complicated adjustments in the data space x.

DEGAN In standard GANs architecture, the input to the generator network is the noise vector that is randomly sampled from a Gaussian distribution $N(0, 1)$, which may create a deviation from the true distribution of real images. Decoder Encoder Generative adversarial Network (DEGAN) [116] adopt decoder and encoder structure from VAE, pretrained on the real images. The pretrained decoder and encoder structure transform

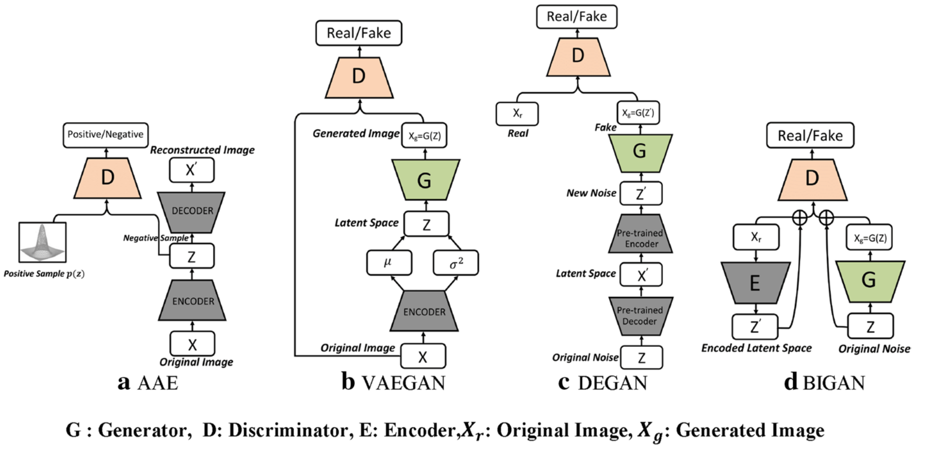


Fig. 10 A schematic view of Variants of GANs based on Encoder and decoder architecture: **a** AAE, **b** VAEGAN, **c** DEGAN and **d** BIGAN

random Gaussian noise to distribution that contains intrinsic information of the images which is used as input of the generator network.

VAEGAN Variational autoencoder Generative Adversarial Network (VAEGAN) [115] jointly trains VAE and GAN by replacing the decoder of VAE with GAN framework. VAEGAN employs feature wise adversarial loss of GAN in lieu of element wise reconstruction loss of VAE to improve quality of image generated by VAE. In addition to latent loss and adversarial loss, VAEGAN uses content loss, also known as perceptual loss, which compares two images based on high level feature representation from pre-trained VGG Network [11].

AAE Unlike VAEGAN that discriminates in data space, adversarial autoencoders (AAE) [113] imposes a discriminator on the latent space as learning the latent code distribution is simpler than data distribution. The discriminator network discriminates between a sample drawn from latent space and from the distribution $p(z)$ that we are trying to model.

ALI and BiGAN In addition to generator network, Adversarially Learned Inference (ALI) [114] model and Bidirectional Generative Adversarial Network (BiGAN) contain an encoder component E that simultaneously learn inverse mapping of the input data x to the latent code z. Unlike other variants of GAN where the discriminator network receives only real or artificially generated images, in the BiGAN and ALI model, the discriminator network receives both image and latent code pair.

VEEGAN [117]: addresses the problem of mode collapse through addition of a reconstruction network that reverses the action of the generator network. Reconstruction network takes in synthetic images then transforms them to noise, while generator network takes noise as an input and reconstructs them into synthetic image. In addition to adversarial loss, difference between the reconstructed noise and initial noise is used to train the network. Both generator and reconstruction networks are jointly trained, which encourages generator network to learn true distribution, hence solving the mode collapse problem.

Several other GANs have been proposed for image super resolution. The goal of super resolution is to upsample low resolution images to a high resolution one. Ledig et al. proposed Super-Resolution GAN (SRGAN) [139] for image super resolution,which takes poor quality image as input, and generates high quality image with 4 × resolution. The generator of the SRGAN uses very deep convolutional layers with residual blocks. In addition to an adversarial loss, SRGAN includes a content loss. The content loss is computed as the euclidean distance between the feature maps of the generated high quality image and the ground truth image, where feature maps are obtained from a pretrained VGG19 [140] network. Zhang et al. [141] combined a self attention mechanism with GANs (SAGAN) to handle long range dependencies that make the generated image look more globally coherent. Image-to-image translation GANs such as Pix2Pix GAN [142], Pix2pix HD GAN [143], and CycleGAN [137] learn to map an input image from a source domain to an output image from a target domain. A summary of architectural variants of GANs are summarized in Table 1.

Objective variants

The main objective of GAN is to approximate the real data distribution. Hence, minimizing distance between the real data distribution (p_r) and the GAN generated data distribution (p_g) is a vital part of training GAN. As stated in "Deep Generative image models" section, standard GAN [67] uses Jensen Shannon divergence to measure similarity between real and generated data distributions $D_{JS}(p_r||p_g)$. However, JS divergence fails to measure distance between two distributions with negligible or no overlap. To improve performance and achieve stable training of GAN, several distances or divergence measures have been proposed instead of JS divergence.

WGAN Wasserstein Generative Adversarial Network (WGAN) [123] replaces JSD from the standard GAN with the Earth mover Distance (EMD). EMD also known as Wasserstein Distance (WD) can be interpreted informally as minimum amount of work to move earth (quantity of mass) from the shape of one distribution p(x) to that of another distribution q(x) so as to match shape of both the distributions. WD is smooth and can provide meaningful distance measure between distributions with negligible or no overlap. WGAN imposes an additional Lipchitz constraint to use WD as the loss in the discriminator, where it deploys weight clipping to enforce weights of the discriminator to satisfy Lipchitz constraint after each training batch.

WGAN-GP Weight clipping in the discriminator of a WGAN greatly diminishes its capacity to learn and often fails to converge. WGAN-GP [124] is an extension of WGAN that replaces weight clipping with gradient penalty to enforce discriminator to satisfy Lipchitz constraint. Furthermore, Petzka et al. [125] proposed a new regularization method, also known as WGAN-LP, that enforces the Lipschitz constraint.

LSGAN Least squares Generative Adversarial Network (LSGAN) [126] deploys least square loss instead of the cross entropy loss in discriminator of the standard GAN to overcome the problem of Vanishing gradient as well as improving quality of generated image.

EBGAN Energy Based GAN (EBGAN) [127] uses auto-encoder architecture to construct the discriminator as an energy function instead of a classifier. The Energy of EBGAN is the mean squared reconstruction error of an autoencoder, providing lower

Table 1 An overview of GANs variants discussed in "Architecture variants" section

Categories	GAN Type	Main Architectural Contributions to GAN
Basic GAN	GAN [67]	Use Multilayer perceptron in the generator and discriminator
Convolutional Based GAN	DCGAN [112]	Employ Convolutional and transpose-convolutional layers in the discriminator and generator respectively
	PROGAN [131]	Progressively grow layers of GAN as training progresses
Condition based GANs	cGAN [118]	Control kind of image being generated using prior information
	ACGAN [119]	Add a classifier loss in addition to adversarial loss to reconstruct class labels
	VACGAN [120]	Separate out classifier loss of ACGAN by introducing separate classifier network parallel to the discriminator
	infoGAN [121]	Learn disentangled latent representation by maximizing mutual information between latent vector and generated images
	SCGAN [122]	Learn disentangled latent representation by adding the similarity constraint on the generator
Latent representation based GANs	DEGAN [116]	Utilize the pretrained decoder and encoder structure from VAE to transform random Gaussian noise to distribution that contains intrinsic information of the real images
	VAEGAN [115]	Combine VAE and GAN
	AAE [113]	Impose discriminator on the latent space of the autoencoder architecture
	VEEGAN [117]	Add reconstruction network that reverse the action of generator network to address the problem of mode collapse
	BiGAN [114]	Attach encoder component to learn inverse mapping of data space to latent space
Stack of GANs	LAPGAN [132]	Introduce Laplacian pyramid framework for an image detail enhancement
	MADGAN [135]	Use multiple generators to discover diverse modes of the data distribution
	D2GAN [134]	Employ two discriminators to address the problem of mode collapse
	CycleGAN [137]	Use two generators and two discriminators to accomplish unpaired image to image translation task
	CoGAN [136]	Use two GANs to learn a joint distribution from two-domain images
Other variants	SAGAN [141]	Incorporate self-attention mechanism to model long range dependencies
	GRAN [133]	Recurrent generative model trained using adversarial process
	SRGAN [139]	Use very deep convolutional layers with residual blocks for image super resolution

energy to the real images and high energy to generated images. EBGAN exhibits faster and more stable behavior than standard GAN during training.

Same as EBGAN, Boundary Equilibrium GAN (BEGAN) [128], Margin adaptation GAN [129] and dual agent GAN [130] also deploy an auto-encoder architecture as the discriminator. The discriminator loss of BEGAN uses Wasserstein distance to match the distributions of the reconstruction losses of real images with the generated images.

There are also several other objective functions based on Cramer distance [144], Mean/covariance Minimization [145], Maximum mean discrepancy [146], Chi-square [147] have been proposed to improve performance and achieve stable training of GAN.

Training tricks

While research on various GANs architectures and objective functions continue to improve the stability of training, there are several training tricks proposed in the literature intended to achieve excellent training performance. Radford et al. [112] showed using leaky rectified activation functions in both generator and discriminator layers gave higher performance over using other activation functions. Salimans et al. [148] proposed several heuristic approaches which can improve the performance, and training stability of GANs. First, feature matching, changes the objective of the generator to minimize the statistical difference between features of the generated and real images. In this way, the discriminator is trained to learn important features of the real data. Second, minibatch discrimination, where the discriminator process batch of samples, rather than in isolation that helps prevent mode collapse, as the discriminator can identify if the generator continues to generate sample with little variety. Third, historical averaging, that takes the running average of parameters in the past and penalizes if there is a large difference between parameters, which can help the model to converge to an equilibrium. Finally, one-sided label smoothing provides smoothed labels to the discriminator instead of 0 or 1, which can smooth the classification boundary of the discriminator.

Sønderby et al. [149] proposed the idea of crippling the discriminator by introducing noise to the samples rather than labels, which prevents the discriminator from overfitting. Heusel et al. [150] used a separate learning rate for generator and discriminator, and trained GANs by a Two Time-Scale Update Rule (TTUR) to ensure that model converge to a stationary local Nash equilibrium. To stabilize the training of the discriminator, Miyato et al. [151] proposed normalization technique called spectral normalization.

Taxonomy of class imbalance in visual recognition tasks

This section describes different GANs applied to imbalance problems in various visual recognition tasks. We group the imbalance problems in a taxonomy with three main types: 1. Image level imbalances in classification 2. object level imbalances in object detection and 3. pixel level imbalances in segmentation tasks. Understanding this taxonomy of imbalances will provide a valuable framework for further research into synthetic image generation using GAN.

Class imbalances in classification

Image classification is the task of classifying an input image according to a set of possible classes. Classification can be broken down into two separate problems: binary classification and multi-class classification. Binary classification involves assigning an input image into one of two classes, whereas in multi-class classification two or several classes are involved. A classic example of a binary image classification problem is the identification of cats or dogs in each input image. Image dataset with high imbalance [152], which includes inter-class imbalance and intra-classes imbalance, results in poor classification performance.

Inter class imbalance

Inter-class imbalance refers to a binary image classification problem where a minority class contains a smaller number of instances when compared to instances belonging to the majority class. Inter class imbalance in a dataset is described in terms of the imbalance ratio. The ratio between the numbers of instances of the majority class and those of the minority class is called the imbalance ratio (IR). For example, binary class imbalance with imbalance ratio of 1:1000 means that for every one-instance in a minority class, there are 1000 instances in the majority class. Datasets with a high imbalance ratio are harmful because they bias the classifier towards majority class predictions.

Synthetic images generated using GAN can be used as an intelligent oversampling technique to solve class imbalance problems. The general flowchart of GAN-based oversampling technique is depicted in Fig. 11. This GAN-based oversampling technique not only increases the representation of the minority class, but it may also help to prevent over fitting.

Shoohi et al. [153] have used DCGAN to restore balance in the distributions of imbalanced malaria dataset. Generated synthetic images from DCGAN are used to achieve 100% balance ratio by oversampling minority class and thus reduce the false positive rate of classification. Their original dataset contains 18,258 cell images, (13,779 parasitized

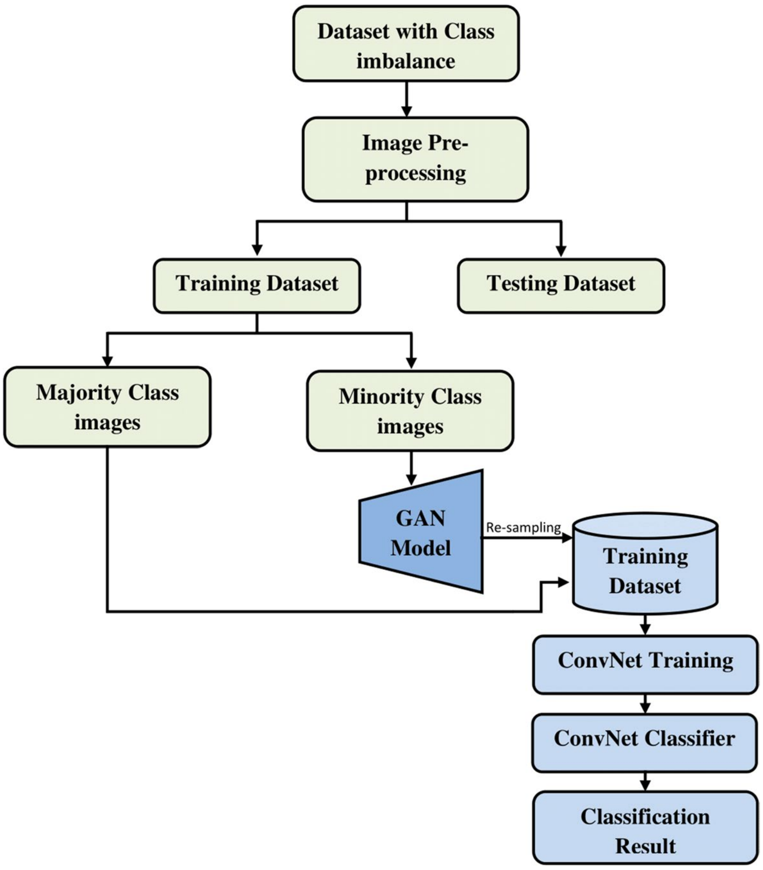


Fig. 11 flowchart of GAN-based oversampling technique

cells, 4,479 uninfected cells). After using an imbalanced dataset to achieve 50% accuracy, they observed an increase to 94.5% accuracy once they added the DCGAN-generated samples.

Niu et al. [154] introduced surface defect-generation adversarial network (SDGAN), using D2 adversarial loss and cycle consistency loss for industrial defect image generation. SDGAN is trained to generate defective images from defect-free images. D2 adversarial loss enables the SDGAN to generate defective images of high image quality and diversity, while cycle consistency loss helps to translate defective images from defect-free images. Surface defect classifier trained on the images synthesized by the SDGAN achieved 0.74% error rate and, also proved to be robust to uneven and poor lighting conditions.

Mariani et al. [155] argued that the few examples in minority class may not be sufficient to train GANs, so they introduced a new architecture called Balancing GAN (BAGAN). BAGAN utilizes all available images of minority and majority classes, and then tries to achieve class balance by implementing class conditioning in the latent space. Learning useful features from majority classes can help the generative model to generate images for minority classes. An autoencoder is employed to learn an exact class-conditioning in the latent space.

Most of the work done in utilizing GANs based synthetic images for class imbalance and comparing the resulting classification performance have been performed in medical image datasets [152, 156–158], and [159]. In the study of Wu et al. [156], class conditional GAN with mask infilling (ciGAN) is trained to generate examples of mammogram lesions for addressing class imbalance in mammogram classification. Instead of generating malignant images from scratch, ciGAN simulates lesions on non-malignant images. For every non-malignant image, ciGAN generates a malignant lesion onto it using a mask from another malignant lesion. On the DDSM (Digital Database for Screening Mammography) Dataset [152], synthetic images generated using ciGAN improves classification performance by 0.014 AUC over baseline model and 0.009 AUC compared to standard augmentation techniques alone.

The vast majority of studies in bio-medical domain used cycle-GAN [138] to generate synthetic medical images. Muramatsu et al. [157] tested the use of a cycle-GAN to synthesis mammogram lesion images from different organs in mammogram classification. They translated CT images with lung nodules to mammogram lesion images using cycle-GAN and found classification accuracy improved from 65.7% to 67.1% with generated images.

For breast cancer detection, Guan and Loew [158] compared the usefulness of DCGAN-generated mammograms and traditional image augmentation method in a mammogram classification task. On the DDSM Dataset [152], the GAN based oversampling method performed about 3.6% better accuracy than traditional image augmentation techniques.

Most recently, Waheed et al. [159] proposed a variant of ACGAN, called Covid-GAN for the generation of synthetic Chest X-Ray (CXR) images to restore balance in the imbalanced dataset. Their dataset contains 721 images of Normal CXR and 403 images of Covid-CXR collected from three publicly accessible databases: (1) COVID-19 Chest X-ray Dataset Initiative [160], (2) IEEE Covid Chest X-ray dataset [161] and

(3) COVID-19 Radiography Database [162]. The generator network in the Covid-GAN is stacked on top of the discriminator. At the beginning of the training process, the layers of the discriminator are freezed and thus, only the generator network gets trained via the discriminator. However, the author offers no explanation for the significance of stacking. They observed improved classification accuracy from 85 to 95% when the classifier is trained on combination of original and synthetic images.

The effectiveness of using synthetic images to balance the class distribution is fairly a recent idea that has not been widely tested and understood. At low resolution image datasets, adding synthetic images with original images have shown to improve performance of the classifiers, but at the higher resolution image datasets these synthetic images become obvious to distinguish from the real one. This is due to the fact that the higher resolution images allow for finer textures and details, and hence will need more cautious modifications by GAN so as not to distort the natural patterns occurring in the high-resolution image dataset. Improving the resolution of GAN samples and testing their effectiveness is an interesting area of future work.

Intra class imbalance

Another type of imbalance that deteriorates performance of the classification problem is the intra-class imbalances. The techniques used for inter-class imbalance can be extended to intra-class imbalance if the datasets have detailed labels. However, in real world datasets, data acquisition with a detailed label is rare because acquiring detailed dataset is costly, and sometimes even not feasible [163]. In many cases, collecting images is tiresome, like 1. capturing images of the same person with glasses and without them, 2. Images of the same person face with varying poses, facial attributes, etc. In some cases, such as the gender swapping, it is not feasible to collect images of the same person as both male and female. Therefore, those techniques for inter-class imbalance are hard to solve intra-class imbalance.

Hase et al. [163] presented an interesting idea to combine clustering technique with GANs designed for solving intra class imbalance. The proposed architecture consists of the generator G, the discriminator D, and the pre-trained feature extractor F (Fig. 12). The key idea is to generate clusters of images in each class in the feature space, and synthesize images conditioned on class and cluster while estimating the clusters of

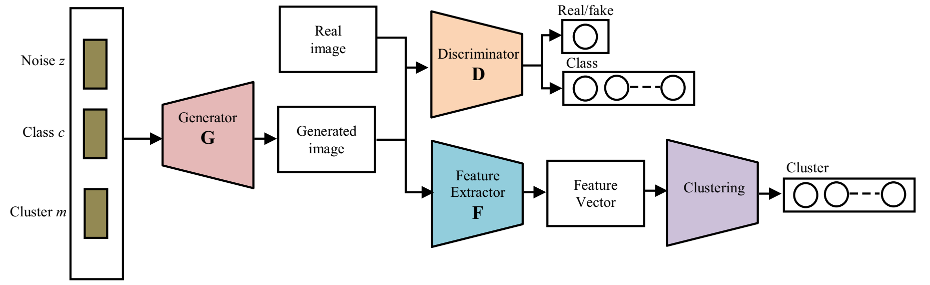


Fig. 12 Architecture diagram of clustering based GAN for solving intra-class imbalance presented by Hase et al. [163]

generated images. The generator *G* is trained to generate an equal number of images for each class and cluster, so that the distribution of both inter and intra class become uniform.

Utilizing clustering techniques in the feature space to divide the images into groups for an automatic pattern recognition in the dataset is a promising area for future work. Additionally, it will be interesting to see how the performance of GAN changes with different types of clustering methods such as Hierarchical clustering, Fuzzy clustering, Density-based clustering, etc.

A semantically decomposed GAN (SD-GAN) proposed by Donahueet al. [164] adopts Siamese networks that learn to generate images across both inter and intra class variations. Both GANs and Siamese networks have two networks. But unlike GANs, where the two networks compete with each other, the two networks in Siamese networks are similar and working one beside the other. They learn to compare output of the two networks on two different inputs and measure their similarity. For example, Siamese networks can measure the probability that two signatures are made by the same person. A combination of GAN and Siamese networks in SD-GAN can learn to synthesize photo-realistic variations (such as, viewpoints, light conditions, scale, backgrounds, and more) of an original input image.

Many studies have reported the problem of an intra-class imbalance owing to age, gender, race and pose attribute variations in face recognition tasks [165–168]. Several variants of GAN have been proposed to address this issue, some focusing on modifying one or more facial attributes, others on generating high quality face images with distinctive pose variations.

Facial attribute editing Human face attributes are highly imbalanced in nature. Attributes can be combined to generate descriptions at multiple levels. For instance, one can describe “white-female” at the category level, or “white-female blond-hair black-eyes wearing necklace” at the attribute level. Attribute level imbalances are inevitable in facial recognition datasets (Fig. 13). As an example, Bald persons with a mustache wearing neckties are 14 to 45 times less likely to occur in the CelebA dataset [169].

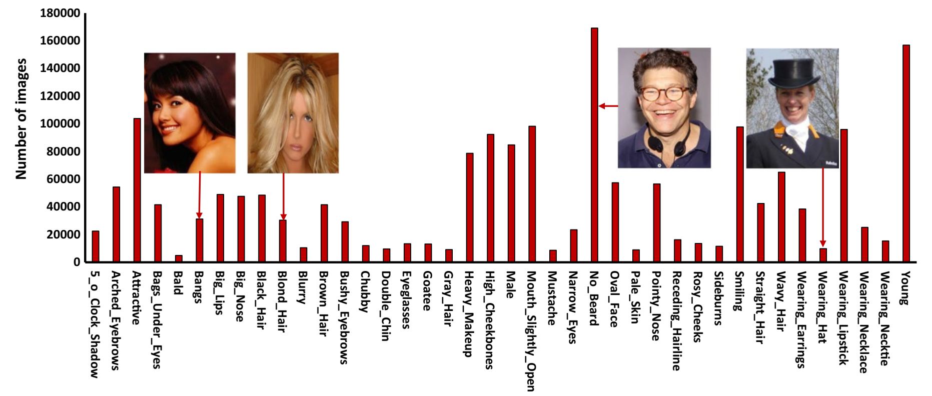


Fig. 13 Imbalanced distribution of 40 binary face attributes (positive and negative) on CelebA dataset [169]

Face attribute editing aims to edit the face image by modifying single or multiple attributes while preserving other details. It is challenging because some of the face attributes are locally distributed, such as 'bangs', 'wavy hair', and 'mustache', but some are globally attributed such as 'chubby', 'smiling' and 'attractive'. Several GANs based methods have been proposed to achieve face attribute editing tasks.

Anders et al. [115] proposed a model that combines VAE and GAN together and learns to map the facial images into latent representation. The derived latent representations are then used to find the attribute manipulating direction. For a given facial attribute (e.g., blond hair), the training dataset can be separated into two groups that images with or without blond hair, then the manipulation direction can be computed as the difference between the mean latent representation of two groups. However, such latent representation contains highly correlated attributes, that results in unexpected changes of other attributes, e.g., adding mustache always makes a female become a male as mustache objects are always associated with male in the training set.

He et al. [64] showed how single or multiple facial attributes of a face image can be manipulated by using encoder-decoder architecture. i.e., to generate and modify a face image with the required attributes, while preserving realism of the image (Fig. 14). They have introduced encoder-decoder architecture in GAN to handle this task. Encoder in the encoder-decoder architecture maps a facial image onto a latent representation and facial attribute editing is accomplished by decoding the latent representation conditioned on the expected attributes. The authors applied an attribute classification constraint to guarantee that the attributes are correctly edited. Meanwhile, reconstruction learning is employed to ensure the attributes excluding details are well preserved.

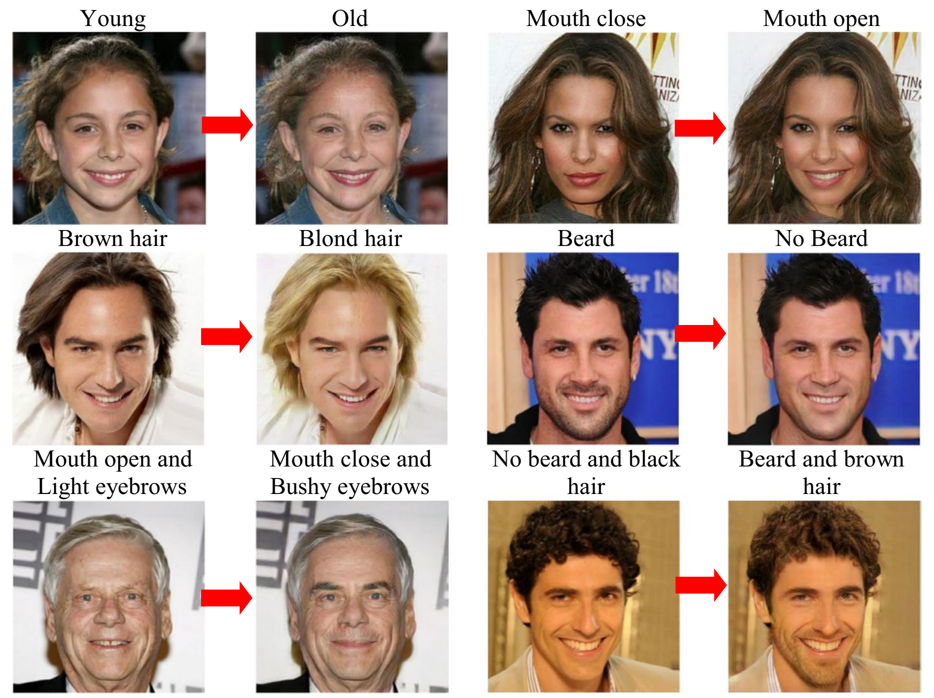


Fig. 14 Face attribute editing examples created by AttGAN [64]

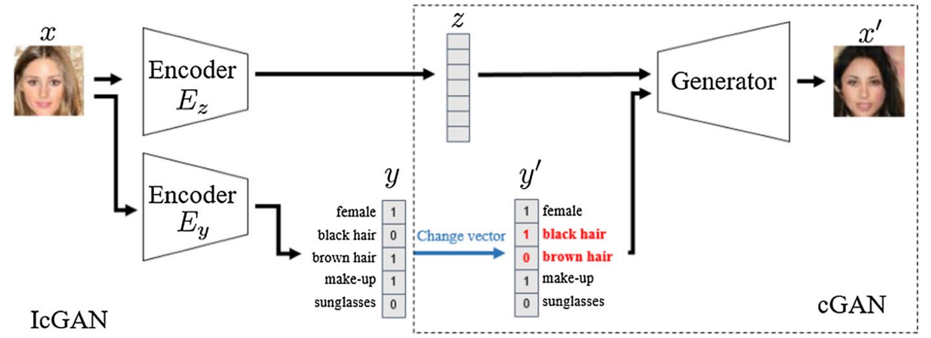


Fig. 15 illustration of invertible conditional GAN presented by Perarnau et al. [65]

Perarnau et al. [65] proposed an invertible conditional GAN (IcGAN) that is equipped with two encoders to inversely map from input facial images into conditional vector y and latent vector z, which, as a result can be manipulated to generate a new face image with desired attributes. IcGAN is a multi-stage training algorithm that first trains a cGAN [118] to map from conditional vector y and latent vector z to real images, and in a second step learns its inverse mapping from generated images to conditional vector y and latent vector z in a supervised manner (Fig. 15). In this way, by changing the conditional vector y, IcGAN allows to control attribute relevant features (e.g. hair color) while latent vector z allows to modify attribute irrelevant features (e.g. pose, background).

Tao et al. [66] argued that the facial attribute editing is an image-to-image translation problem, which aims to transfer facial images from the source domain to the target domain. Their proposed model contains three major parts: an encoder, a decoder, and a residual attributes extractor. The encoder and decoder together constitute a generator, whose main aim is to generate a facial image with desired attributes. The encoder maps the facial images into latent representation and the decoder reconstructs (generates) the image from this representation along with attribute vectors. The main purpose of residual attributes extractor is to learn the gap between the original input and the desired output in the feature space and back propagate error signal to supervise the generation process.

Zhangi et al. [170] have used the design principle of Adversarially Regularized U-net (ARU-net), instead of conventional encoder and decoder architecture to learn facial attribute editing and generation tasks together during training. The symmetric skip connection technique is used to pass on the details from encoder to decoder, which preserves the attribute irrelevant features. In this architecture, the ARU-net is integrated with GANs that results in ARU-GAN to perform facial attribute editing. The ARU-GAN consists of four major components: the ARU-net for preserving attribute irrelevant features, the adversarial network to constrain the latent representation, the discriminator to distinguish between real and fake image, and the attribute classifier to ensure the desired attributes are edited.

Zhang et al. [171] introduced a spatial attention mechanism into GANs for only modifying attribute relevant parts and keeping attribute irrelevant parts unchanged. SaGAN [141] is used to locate and manipulate attribute-relevant part more precisely. The generator of the proposed architecture consists of an attribute manipulation network (AMN) and a spatial attention network (SAN). Given a face image, SAN learns to localize the attribute-specific region and then AMN edit the face image with the desired attributes in the specific region located by SAN.

The major downside with the current approaches is that the input to GAN should be frontal face images. It will be interesting to explore a new architecture that can be trained to modify the attributes of side-view or any arbitrary views.

Person re-identification Person re-identification [172] is another challenging task worth mentioning, which are adversely affected due to significant intra class imbalance. Intra class variations caused by rotation (varying poses) are often larger than the inter-person dissimilarities used to differentiate the face images [173]. Recent face-recognition surveys [174, 175] identified pose variation as one of the prominent unresolved issues in face-recognition task. For instance, in order to maintain the highest standard of security, a smart video system needs to be able to detect a person invariant to pose (Fig. 16).

Qian et al. [176] introduced a pose-normalized GAN model (PN-GAN) for alleviating the effects of pose variation. Given any pedestrian image and a desirable pose as input, the model utilized a desirable pose to produce a synthetic image of the same identity with the original pose replaced with the desirable pose (Fig. 17). After this, the authors trained the re-identification model with the original images and generated pose-normalized images to extract two sets of features. Finally, they fused the two types of features as the final feature. As a result, the features extracted from the synthesized images improved the generalization ability of the re-identification model.

To address person re-identification challenges in complex scenarios, Wei et al. [177] proposed a model called Person Transfer Generative Adversarial Network (PTGAN) for implausible person image style transfer from source domain to target domain, across datasets with different styles, such as backgrounds, poses, seasons, lightings, etc. The domain transfer procedure in PTGAN is inspired by CycleGAN [138]. Different from

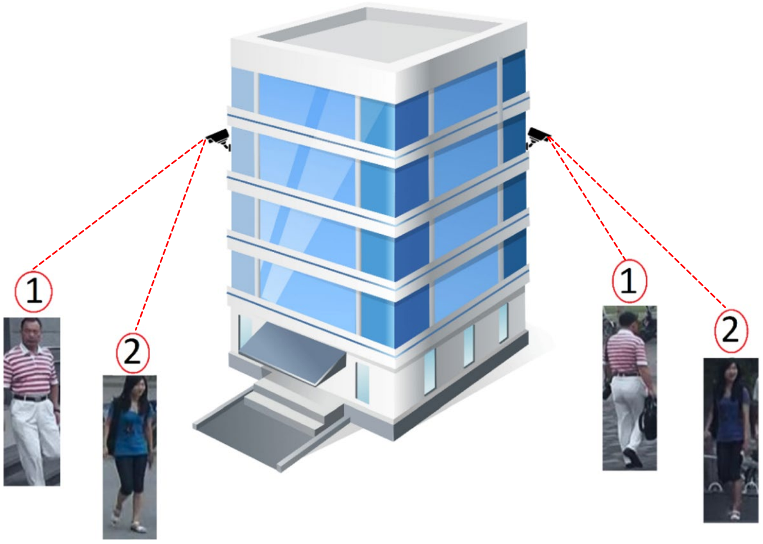


Fig. 16 Example of Person reidentification task. Person reidentification is a key element in video surveillance that deals with matching images of same person over many non-overlapping camera views

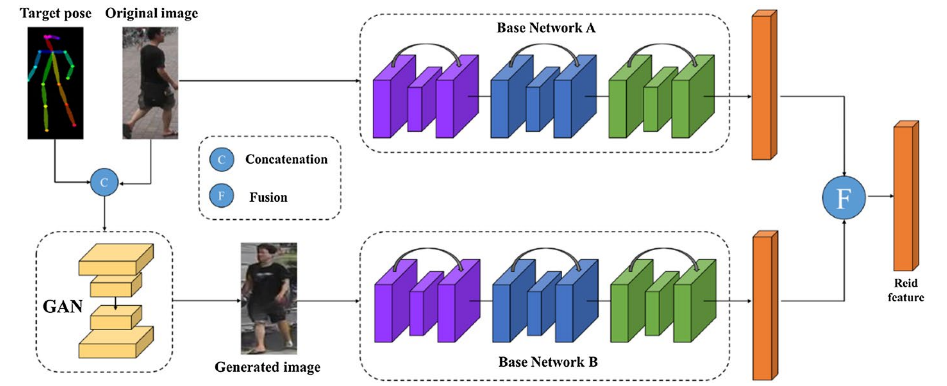


Fig. 17 Architecture diagram of pose-normalized GAN presented by Qian et al. [176]

Cycle-GAN [138], PTGAN incorporates additional constraints on the person foregrounds to make sure the stability of their identities during transfer. Compared with Cycle-GAN, PTGAN generates high resolution person images, where person identities are unchanged, and the styles are transformed.

Being a cross-camera tracking and human retrieval task, person re-identification often suffers from image style variations resulting from different cameras. Therefore, Zhong et al. [178] designed a camera style adaption model for adjusting ConvNet training. They have used CycleGAN [138] for transferring images from one camera to the style of another camera. Given that both original and style transferred images, identification discriminative embedding (IDE) is used to train the ConvNet model. Particularly, authors have used ResNet-50 pre-trained on ImageNet dataset as backbone and follow the fine-tuning strategy.

Pedestrian images suffer from information loss when transferring from one camera to the style of another camera. Deng et al. [179] presented a model, named similarity preserving cycle consistent generative adversarial network (SPGAN), which is composed of a CycleGAN and a Siamese network (SiaNet). CycleGAN learns to translate pedestrian images from one domain to another domain, and the contrastive loss induced by the SiaNet pulls close a translated image and its counterpart in the source domain, and moves away the translated image and any image in the target domain.

Ge et al. [180] presented a Feature Distilling Generative Adversarial Network (FD-GAN) that aims at learning identity related and pose-unrelated person representations. The proposed model adopts a Siamese structure with multiple novel discriminators on human poses (pose discriminator) and identities (identity discriminator). The idea behind FD-GAN is to learn pose-unrelated and identity-related features of pedestrian image, then it can be used to generate the same pedestrian image but with different target poses.

Although GAN-based methods described above have achieved excellent performance in image-based person re-identification, it still needs considerable effort to tackle the video-based identification datasets. Future work seeks to expand to use GAN for generating a sequence of images for the video-based identification datasets.

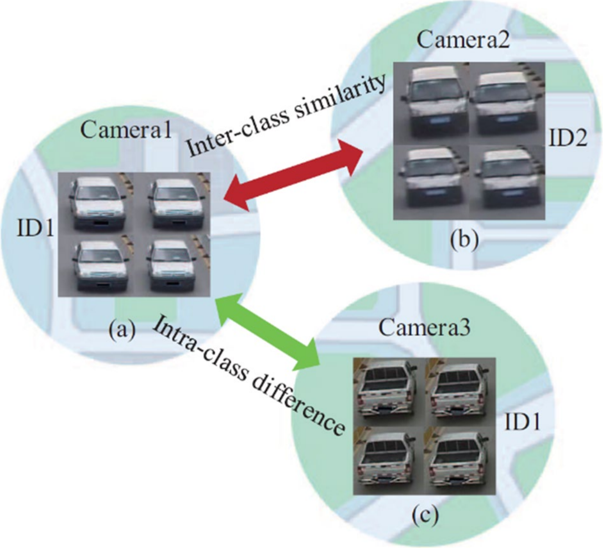


Fig. 18 illustration of challenges in vehicle Re-identification provided by Zheng et al. [181]

Vehicle re-identification Vehicle Re-identification task is even more challenging as it suffers from large intra-class differences caused by viewpoint and illuminations variations, and inter-class similarity primarily for different identities with the similar look (Fig. 18).

Zhou et al. [182] proposed a model called Cross view GAN to generate images in different viewpoints of the same vehicle. Cross view GAN composed of classification, generator, and discriminator network. First, classification network is trained to learn vehicle intrinsic features such as model, color, and type information. In addition to intrinsic features, it also learns viewpoint features. Then the generative network is conditioned on the average feature of the expected viewpoint and vehicle's intrinsic features to infer images of the same vehicle in other viewpoints. The discriminator network learns to distinguish real images from the generated images, while ensuring images are generated with correct attributes.

Wu et al. [183] improved the discriminative power of the ResNet-50 model for the Vehicle re-ID task by simultaneously training with initial labeled images and DCGAN generated unlabeled images. They further explore the effectiveness of using DCGAN generated images on a wide range of vehicle re-ID datasets and show improved performance of vehicle re-identification.

Fine-grained image classification The fine-grained image classification is also attributed to major variations in the intra-class and minor inter class variations [184]. It is a difficult task for two reasons. First, the training samples of each class are inadequate. Second, the differences between different classes of images are quite small [185]. As an example, it is very difficult to identify the images of Shetland Sheepdog from that of Collie dog. Similarly, the images of Sayornis and Gray Kingbird are quite difficult to distinguish (Fig. 19).

Fu et al. [184] developed a model called Fine grained conditional GAN (F-CGAN) to solve fine grained class dependent image synthesis problems. F-CGAN consists of

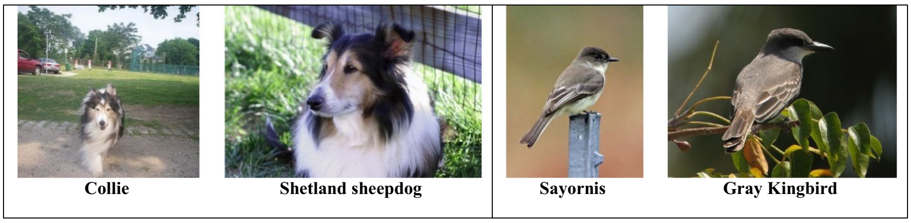


Fig. 19 Sample images from the Stanford Dogs dataset [186] and the Caltech-UCSD Birds dataset [187], which exhibits minor inter-class variations and major intra-class variations

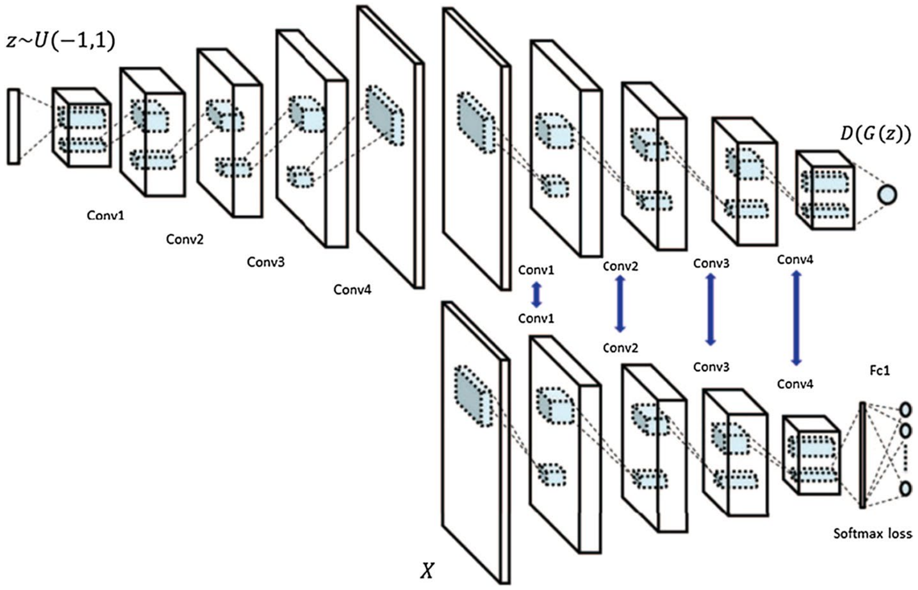


Fig. 20 Complete fine-grained Plankton classifier architecture used by Wang et al. [188]

three main components: 1. a 2-stage GAN, 2. a fine-grained feature preserver and 3. a multi-task classification model. The 2-stage GAN generates high resolution images, the fine-grained feature preserver targets to capture fine grained details and the multi-task classification model utilizes generated image data to improve fine grained classification accuracy.

Wang et al. [188] find that the discriminator in GANs learns a hierarchical identification features of the fine-grained classes and discriminate pattern of the fine-grained training samples. They use the architecture pictured below to implement the fine-grained Plankton classification task (Fig. 20). The main idea is to train a fine-grained classifier that shares weights with discriminator of the DCGAN, which forces discriminator to concentrate on features of small classes. On WHOI-Plankton dataset [189], F1 score of the classifier improved by over 7%.

Typically, medical image datasets contain both general labels, e.g., “male”, “female” and disease specific detailed labels [190]. It is mentioned that the complexity and nature of data is hard to learn by using a single GAN. Hence, T. Koga et al. [190] connected two GANs in series, one for learning general features and other for detailed features. The first GAN generates diverse images, which takes a noise vector and general labels as inputs.

The second GAN receives synthetic images generated by the first GAN, and disease specific detailed labels as inputs, and generates the final fine-grained medical images.

Multiclass imbalance

In many real world problems such as emotion classification [191], plant disease classification [192], medical image classification [193], industrial defect classification [194] etc., it is more likely that more than one class exists and needs to be recognized. Multiclass classification has been shown to suffer more learning difficulties than binary class classification, because multiclass classification increases the data complexity and intensifies the imbalanced distribution [195]. Three types of imbalance could occur to the multiclass datasets: few minority-many majority classes, many minority-few majority classes, and many minority-many majority classes. Shuo Wang et al. [196] studied the impact of all different types of multiclass imbalances and showed that they negatively affect minority class and overall performance.

An example of few minority-many majority class imbalance is an emotion classification, as some classes of emotions like disgust are relatively uncommon compared to common emotions like happy or sad. Zhu et al. [197] employed cycle-GAN which can synthesize uncommon emotion classes like disgusted from the frequent classes (Fig. 21). In addition to adversarial and cycle consistency loss, they use least square loss from LSGAN to avoid vanishing gradient problems. Employing cycle-GAN based data minority class data augmentation achieved 5–10% increase in the overall accuracy. They also found that enlarging minority classes also increases accuracy of other majority classes.

Weather Image classification is another example of few minority-many majority class imbalance, because some types of weather, like snow, is relatively rare compared to sunny, hazy and rainy days. Li et al. [198] used DCGAN to generate images of minority classes in training. They found that the GAN-based data augmentation technique led to margin clarity between classes and hence improvement in classification performance.

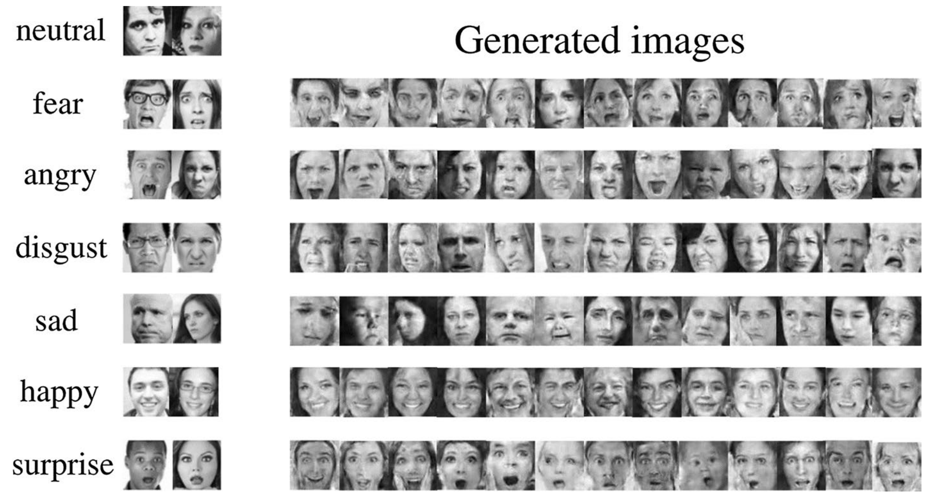


Fig. 21 On emotion classification task [197], the images on the left are original data and the rest are images generated by cycle-GAN

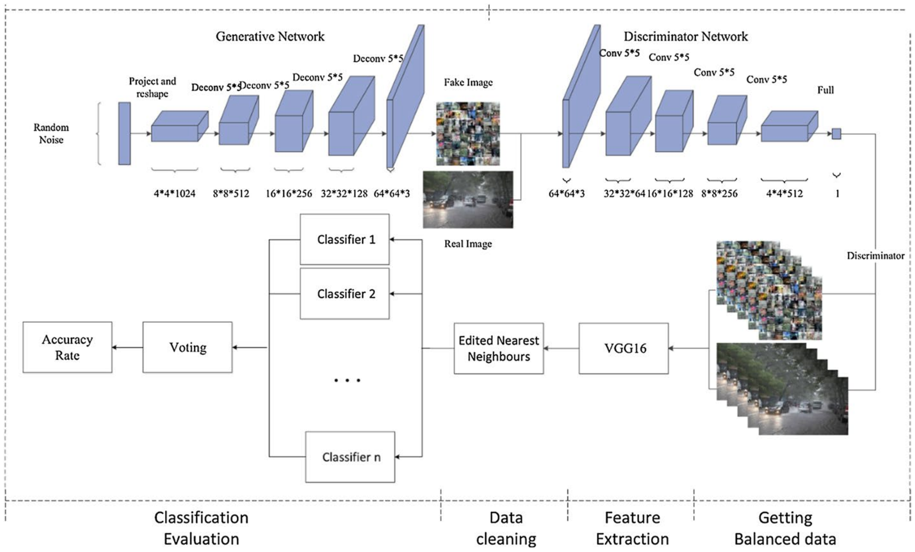


Fig. 22 Illustration from Huang et al. [199] showing how the Ensemble learning is integrated with GAN Framework

Huang et al. [199] presented an interesting idea to combine ensemble learning with GANs designed to address the class imbalance problem in weather classification. The proposed method comprised of three ingredients as depicted in (Fig. 22): 1. DCGAN to generate synthetic images and balance the training dataset 2. Nearest neighbor method to remove any possible outlier images generated by DCGAN 3. An ensemble learning method to combine the classification results of the multiple classifiers so as to achieve better results.

The use of DCGAN was tested by Salehinejad et al. [193] in the task of chest pathology classification. Using chest X-ray images, they build a deep ConvNet classifier to classify 5 different anemic classes. Their dataset is highly imbalanced, contains three majority and two minority classes (Fig. 23a). The synthetic images generated using DCGAN were used to balance and augment the original imbalanced dataset. They demonstrated that a combination of the original imbalanced dataset and generated images improves the accuracy of deep ConvNet classifier in comparison to the same classifier trained with original imbalanced dataset alone. On chest X-ray dataset [193], a mean classification accuracy improved from 70.87 to 92.10%.

Frid-Adar et al. [200] also showed that generating synthetic liver lesion images using DCGAN can improve classification results. They combined standard augmentation techniques and DCGAN generated synthetic images to train a classifier. Their liver lesion dataset contains 182 computed tomography images (65 hemangiomas, 64 metastases and 53 cysts). By adding the synthetic images to standard data augmentation, their classification performance increased from 78.6% sensitivity and 88.4% specificity using standard augmentations to 85.7% sensitivity and 92.4% specificity using DCGAN-based synthetic images.

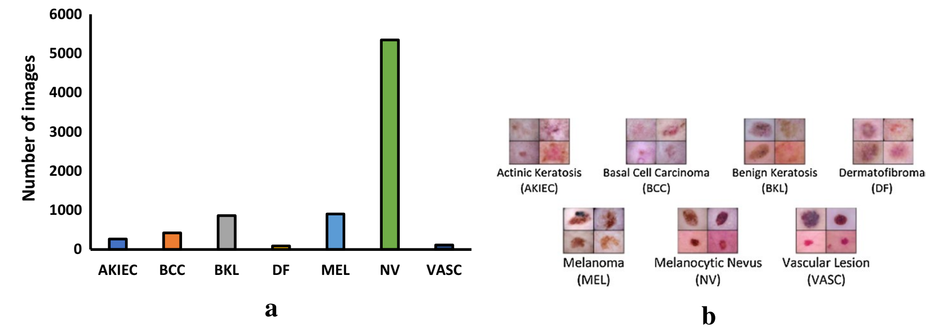


Fig. 24 **a** Distribution of the seven skin lesion class labels of the ISIC 2018 dataset [202]. **b** Sample images from each class

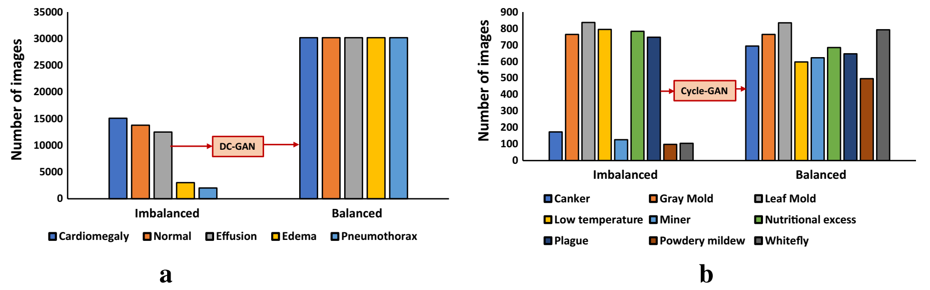


Fig. 23 The distributions of (**a**) Chest X-ray image dataset [193] and (**b**) tomato plant disease dataset [192], before (left) and after class balancing using GANs (right)

Rashid et al. [201] tested the effectiveness of using GANs to generate skin lesion images. Using ISIC 2018 dataset [202], they built a CNN classifier to classify 7 different skin lesions as depicted in Fig. 24. These classes are highly imbalanced, and the GAN is used as a method of intelligent oversampling.

Nazki et al. [192] used Cycle-GAN to alleviate multiclass imbalance problem in tomato plant disease classification. Their tomato plant disease dataset contains 2789 images, highly suffered from class imbalance in 9 disease categories (Fig. 23b). Using Cycle-GAN, they translated images from the healthy tomato leaves to underrepresented diseased tomato leaves. This study demonstrated that the synthetic image generated by Cycle-GAN can be used as an augmented training set to improve the performance of classifier.

Bhatia et al. [203] sought out to compare synthetic images generated using WGAN-GP against the standard data augmentation in the context of multiclass image classification. They artificially introduced class imbalance in two balanced datasets of CIFAR-10 [87] and FMNIST [204], and studied the effects of multiclass imbalance on classification performance. On the CIFAR-10 [87] dataset, classification performance improved from 80.84% accuracy and 0.806 F1-score using standard data augmentation to 81.89% accuracy and 0.812 F1-score using WGAN-GP. On FMNIST [204] dataset, performance improved from 91.9% accuracy and 0.921 F1-score using augmentation to 92.8% accuracy and 0.923 F1-score using WGAN-GP.

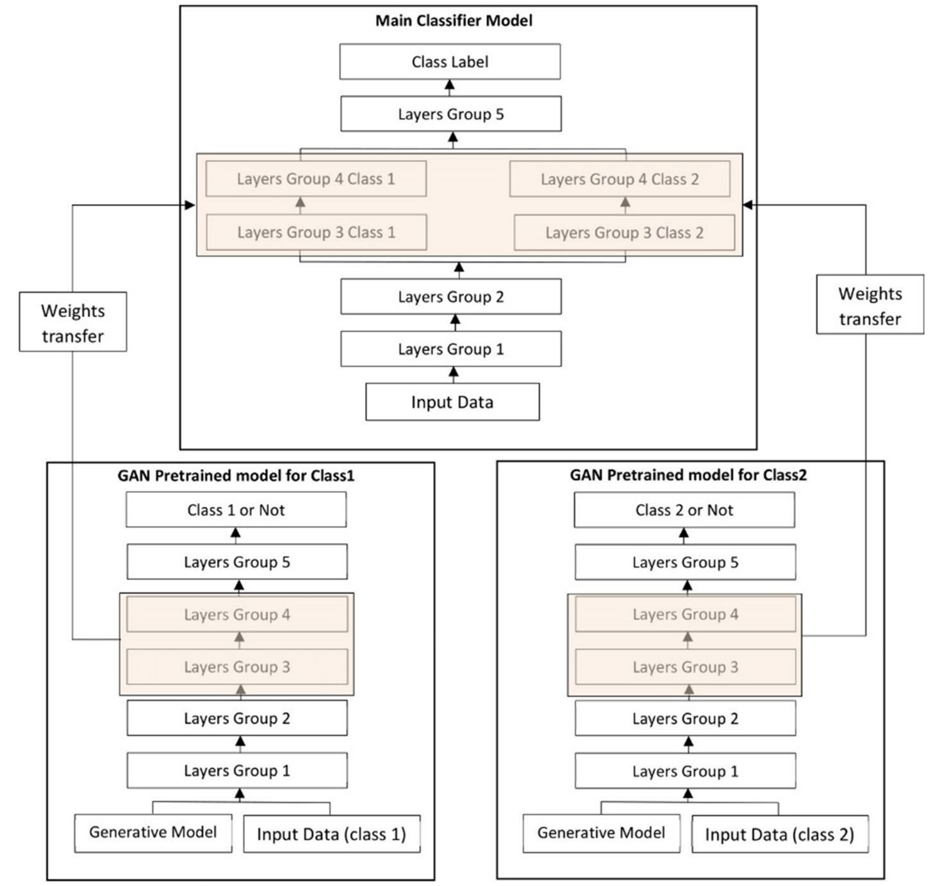


Fig. 25 Illustration of the class expert generative adversarial network architecture [205]

An idea of GANs based transfer learning technique for multiclass imbalance problem is proposed by Fanny et al. [205]. Their architecture named class expert generative adversarial network (CE-GAN) makes use of multiple GANs models, a separate GANs for each class. Feature maps in the main classifier are arranged in parallel, with each feature maps pre-trained to identify the characteristics of a single class in the training data (Fig. 25). The weights of the pretrained feature maps are transferred from discriminators of the GANs to main classifier model for further training in a supervised mode.

The GAN-based synthetic images served as an intelligent oversampling technique and can address the problem of multi-class imbalance to a greater extent. However, synthetic images must be used with caution because if the quality of the synthesized images is not high, this would lead to additional noise to the original datasets.

Object level imbalances in object detection

Object-scale imbalance

One pervasive challenge in the scale invariant object detection is large scale variance across object instances, and particularly, detecting small objects are more challenging than medium and large-scale objects. As per MS COCO definition [206], Objects

Table 2 The definitions and statistics of the small, medium, and large objects as MS COCO [206]

Object category	Spatial dimension		Object count %	Total object area %
	Minimum	Maximum		
Small	0×0	32×32	41.43	1.23
Medium	32×32	96×96	34.32	10.18
Large	96×96	$\infty \times \infty$	24.24	88.59

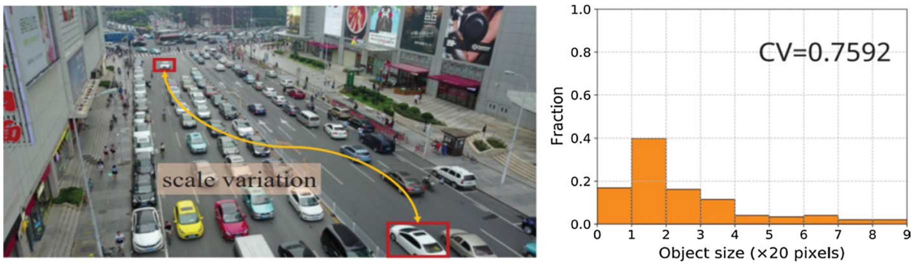


Fig. 26 Example of scale variation and the scale (object size) distribution of the VisDrone2019 dataset objects in pixels [207]

with size less than 32×32 pixels are small, size between 32×32 to 96×96 pixels are considered as medium and objects with size greater than 96×96 pixels are large objects (Table 2). On the one hand, small objects in MS COCO dataset accounts for only 1.23% of total object area, on the other hand, medium and large-scale objects are over 98% of object area. Object detection algorithms should be able to detect both small objects as well as medium and large objects. Detecting small objects are essential in many real-world applications. For instance, detecting distant or small objects in the high-resolution driving scene images captured from cars is essential for achieving autonomous driving. Many distant objects, such as traffic lights or cars, are imperceptible as shown in Fig. 26. Haoyue et al. [207] measure the extent of scale variation using the coefficient of variation (CV), determined as the ratio of the standard deviation to the mean of the object scale. The bigger the CV, the more complicated the problem of scale variation.

There can be three reasons why detecting small objects are more complicated than larger one: 1. Small objects occupy a much smaller area, and consequently there exists lack of diversity where small objects are located in the image, 2. There are comparatively less images in the dataset containing small objects which may bias any object detection algorithm to concentrate more on medium and large-scale objects, and 3. The activations of small objects become smaller and smaller with each pooling layer in a standard convNet architecture as it progressively reduces the spatial size of an image.

To overcome the problem of scale imbalance, two different strategies based on GAN have been proposed in the literature. Commonly adopted strategy is to convert low resolution small object features into high resolution features [208] using GAN. Diversity of the small object locations in the images are enhanced by copy-pasting small object instances several times in each image through adversarial processes [209].

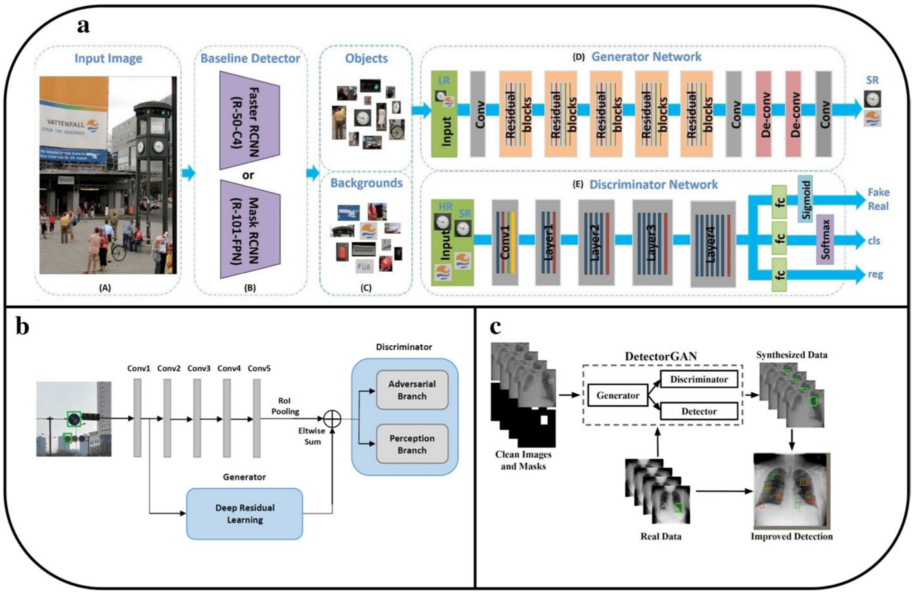


Fig. 27 Architecture diagram of (**a**) SOD-MTGAN [213] (**b**) Perceptual GAN [208] and (**c**) Detector GAN [209]

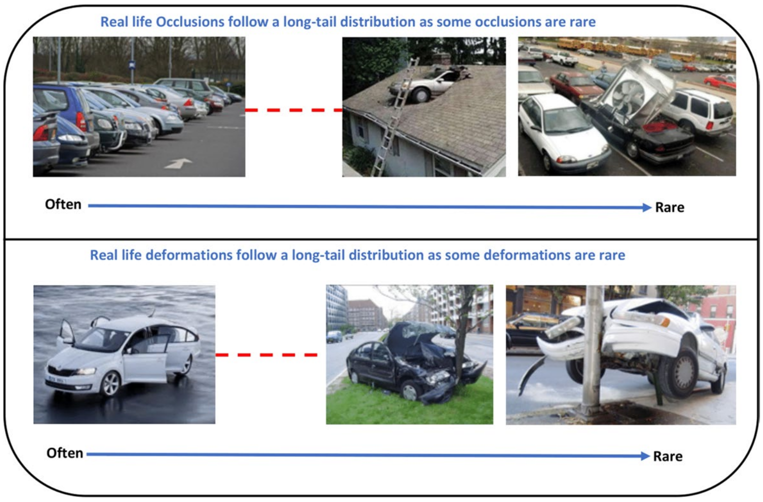

Real life Occlusions follow a long-tail distribution as some occlusions are rare
Often
Rare
Real life deformations follow a long-tail distribution as some deformations are rare
Often
Rare

Fig. 28 Imbalanced distribution of occluded, partially occluded and heavily occluded objects in VisDrone-DET2018 dataset [215]

Li et al. [208] utilized a GAN framework that transforms poor representation of small-scale objects to super-resolved large objects. The generator attempts to generate super resolution features for the small objects. The discriminator in this framework is decomposed into two branches, namely, a perceptual branch and an adversarial branch. An adversarial branch is trained to discriminate between real large-scale objects and generated super resolution objects while a perceptual branch helps to make sure that the generated super-resolved object is useful for the detection (Fig. 27b). They tested the effectiveness of this framework on Tsinghua-Tencent 100 k dataset [210], PASCALVOC dataset [211] and Caltech pedestrian benchmark [212].On the PASCAL VOC 2007 dataset [211], The Average precision (AP) of small objects such as plant, chair, bottle and boat increased by 10%, 15.1%, 21.9% and 10% respectively, compared with Faster-RCNN.

Bai et al. [213] used baseline detectors such as Faster RCNN [36], Mask RCNN [214] to crop an input image into smaller regions (generate ROIs) and then use generator network to reconstruct up-scaled version (super resolved) of cropped regions, while the discriminator perform multiple tasks that discriminates the real from the high resolution generated images, perform classification and regress the bounding box co-ordinates (object location) simultaneously (Fig. 27a).

Lanlan Liu et al. [209] proposed a Detector GAN that combines and optimizes both GANs and object detector together. The generator is trained with both adversarial and training loss, which generates multiple small objects in an image that are hard to detect by the detector and hence enhance the robustness of the detector (Fig. 27c).

Imbalance due to occlusions and deformations

Like the object scale imbalance, occluded and deformed objects in the images follow a skewed distribution. For instance, occlusion from other cars due to urban traffic or parking lots is more common than from an air conditioner as shown in Fig. 28. The performance object detection is often suffered from imbalance due to occluded and deformed objects. Zhu et al. [215] define occlusion ratio to measure the degree of occlusion, determined as the fraction of pixels being occluded. As per VisDrone-DET2018 dataset [215], objects with occlusion ratio greater than 50% are heavy occlusion, ratio between 1 to 50% are considered as partial occlusion and objects with 0% occlusion ratio are categorized as no occlusion. The bar chart below (Fig. 29) depicts the imbalanced distributions of occluded, partially occluded and heavily occluded objects in VisDrone-DET2018 dataset [215].

One way to build the robust object detector invariance to occlusion and deformation is to generate realistic images of these rare occurrences using GANs, and then train the object detector with the generated images. Adversarial object detection could be another interesting way to generate all possible occlusions or deformations on the feature maps that make recognition hard. The object detector is simultaneously trained to overcome the difficulties imposed by the adversarial task.

Wang et al. [216] utilized the adversarial spatial dropout to simulate all kinds of rare deformations and occlusions on the feature maps that are hard for the object detector to detect. Unlike traditional methods [49] that add occlusions on foreground objects in pixel space, they focused on feature space. Their architecture (Fig. 30) comprised of two networks: Adversarial Spatial Dropout Network (ASDN) and Adversarial Spatial Transformer Network (ASTN) to create occlusion and deformation respectively. On

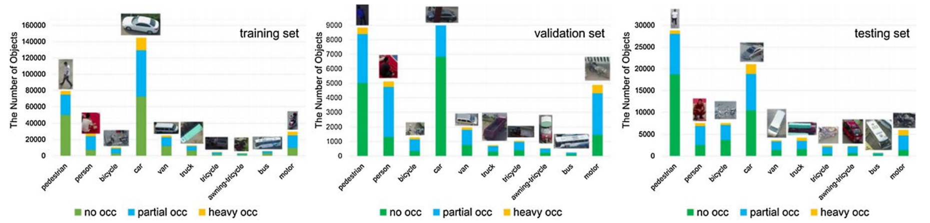


Fig. 29 Illustration of real world occlusions and deformations provided by Wang et al. [216]

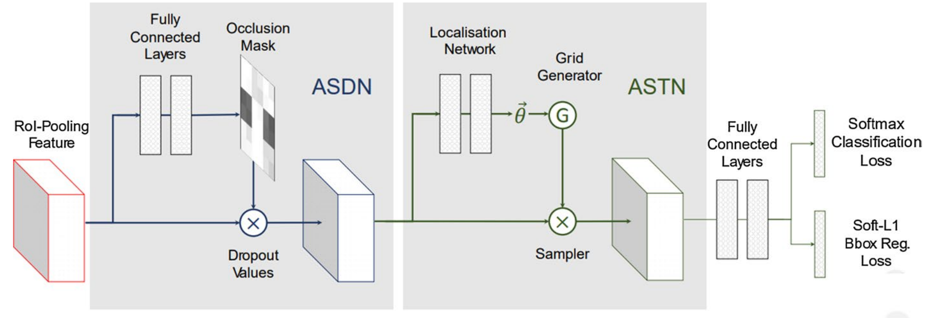


Fig. 30 Architecture diagram to generate hard examples for training [216]

VOC2007 and VOC2012 datasets, this architecture achieved an increase in mean Average Precision (mAP) of 2.3% and 2.6% respectively compared to the Fast-RCNN [36].

Inspired by this architecture, Chen et al. [217]. proposed Adversarial Occlusion Aware Face Detection (AOFD) to overcome the problem of limited occluded face image in training dataset. As opposed to cropping or erasing, Dwibedi et al. [218]. utilized GAN to insert new objects on the images by cut and paste. This method can be extended by inserting occluded and deformed objects on the training images.

Taking full advantage of GANs and combining them into different ConvNet architectures is a recent trend in object detection. These kinds of architectures are often called a three-player GAN. In an attempt to improve performance of detection and classification, three-player GAN only generates hard-to-classify samples. Particularly, the use of faster R-CNN with GANs has improved the state of-the-art benchmarks. Testing the performance of different combinations in comparison to current state of-the-art models is an interesting area for future work.

Foreground–background object class imbalance

Both single stage and two stage object detection algorithms evaluate multiple regions in an image during the training stage. But only a few regions contain foreground (positive), the rest are background (negative). Many of the background examples are easy to classify and offer an uninformative training signal. Just a few background examples provide rich information for training. The imbalance between foreground (objects) and easily classified background overwhelms cross entropy loss and gradients from converging. Some form of hard sampling is a commonly used method by the object detection algorithms to account for this imbalance. The most straightforward and simple hard sampling method is uniform random sampling that randomly selects a subset of negative and positive examples (uniformly distributed) for evaluation. Hard negative mining is another hard sampling method that selects hard samples as negative examples instead of random selection to improve the detection performance.

Unlike hard sampling methods, GAN addresses the problem of foreground background imbalance by directly injecting hard positive and negative synthetic examples into the training dataset. Task aware data synthesis proposed by Tripathi et al. [219]. uses GAN based approach to generate hard positive examples that improve the detectors classification accuracy. Their architecture utilizes three competing networks (Fig. 31):

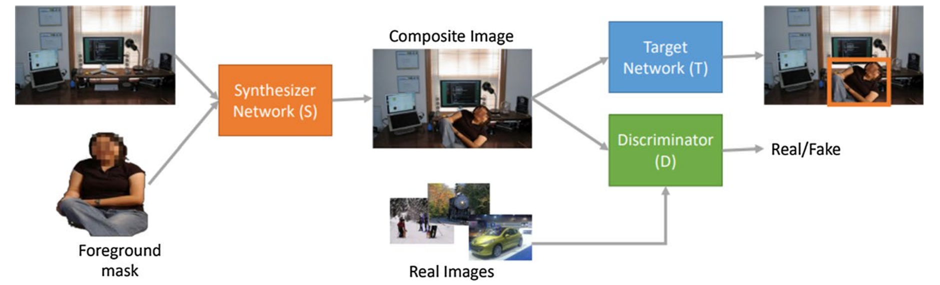


Fig. 31 Pipeline of task aware image synthesis used by Tripathi et al. [219]

a synthesizer (S), a discriminator (D) and the target network (T).Given a background image and a hard-positive foreground mask, synthesizer aims to optimally paste foreground mask onto the background image to produce a realistic image that can fool both the target and discriminator networks. The discriminator network provides necessary feedback to the synthesizer which ensures the realism of the generated composite image. The target network is a pre-trained object detector such as SSD and faster R-CNN. On the VOC person detection dataset, this architecture achieved a performance improvement of up to 2.7%.

Wang et al. [220] presented an interesting idea of object detection via progressive and selective instance-switching (PSIS). Given a pair of training images, PSIS synthesizes a new pair of images by swapping objects of the same class between an original pair of images by also considering scale and shape information of the objects. Generating more training images by swapping objects of low-performing classes improves overall detection accuracy.

Gene-GAN [221] proposed by Zhou et al. employ an encoder and a decoder architecture to replace an object in an image with a different object from a second image. Given an image, Encoder decomposes it into the background and object feature vectors, while decoder reconstructs a new image by transplanting an encoded object to it.

Pixel level imbalances in segmentation

Pixel-wise class imbalance

GANs are being employed to solve pixel level class imbalance problem in segmentation tasks that have a negative influence on segmentation accuracy. The use of image to image translation GANs for a pixel-level augmentation on segmentation tasks was tested by Liu et al. [222]. Particularly, they used Pix2pix HD GAN [143] to translate semantic label maps to realistic images. Semantic object labels from the original dataset such as street, car, pedestrian etc. are recombined to synthesize new label maps which can balance the semantic label distribution. Then the new balanced label maps are translated to realistic images by Pix2pix HD GAN. To further understand the effectiveness of this method, a study was conducted by balancing one to many label classes on original label maps. On the Cityscapes dataset [57] this resulted in an improved mean accuracy of a specific class up to 5.5% and the average overall segmentation accuracy up to 2%.

Shadow detection is a segmentation problem in which there are substantially lesser shadow pixels than non-shadow pixels in training images. Nguyen et al. [223] presented

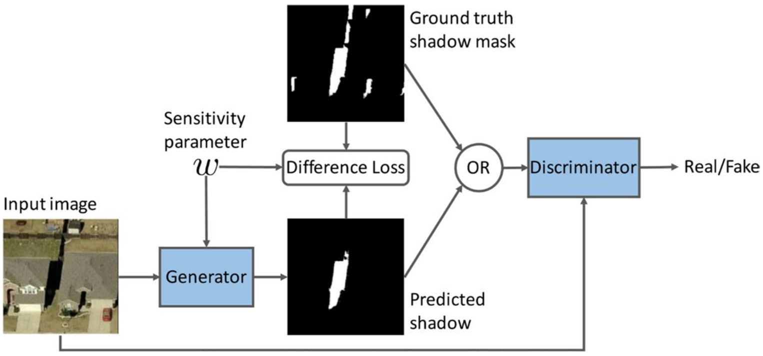


Fig. 32 Illustration of Sensitivity conditional GAN [223]

Sensitivity conditional GAN (ScGAN), an extension of cGAN [118], tailored to tackle the challenging problem of pixel-level imbalance. To balance shadow and non-shadow pixel imbalance during training process, Sensitivity parameter W is introduced in ScGAN that controls how much to penalize the false positive prediction. Notably, the Sensitivity parameter W is made tunable by allowing it to interact with the generator in addition to loss function (Fig. 32). ScGAN achieved up to 17% error reduction on UCF [224] and SBU [225] dataset with respect to the previous state-of-the-art model.

Voxel GAN architecture proposed by Rezaei et al. [226] is a 3D GAN model to address the pixel level imbalance problem in the brain tumor segmentation task as the majority of the pixels belongs to the healthy region and only few pixels belongs to tumor region. Voxel GAN is made of 3D segmentor network to learn generating segmentation labels from 3D MRIs, and a discriminative network to differentiate generated segmentation labels from real labels. The segmentor and discriminator are trained by mix of adversarial loss with weighted $\ell 1$ loss and weighted categorical cross-entropy loss to reduce the negative impact of pixel imbalance.

Similar to this work, Rezaei et al. [227] used similar loss function by mixing adversarial loss and weighted categorical accuracy loss to handle imbalanced training dataset of whole heart segmentation tasks. Balancing through ensemble learning by combining two discriminators to improve their generalization ability of the GAN was tested by Rezaei et al. [228] in medical image semantic segmentation task. One discriminator classifies whether the generated segmentation label is real or fake. Another discriminator is trained to predict false positives and false negatives. Final segmentation mask is generated through adding the false negatives and removing the false positives predicted by this discriminator.

Imbalance due to occlusions in segmentation

GANs are also very efficient in segmentation of natural settings with severe occlusion and large-scale changes [229]. Sa et al. [230] describe that occlusion is a key challenge in segmenting dense scenes. Objects in dense scenes often occlude each other, which lead to severe information loss. In many cases, segmentation algorithms cannot infer the appearance of the objects beyond their visible parts, which may prevent it from making accurate decisions if a person purposely covers the face. GANs offer a new way to

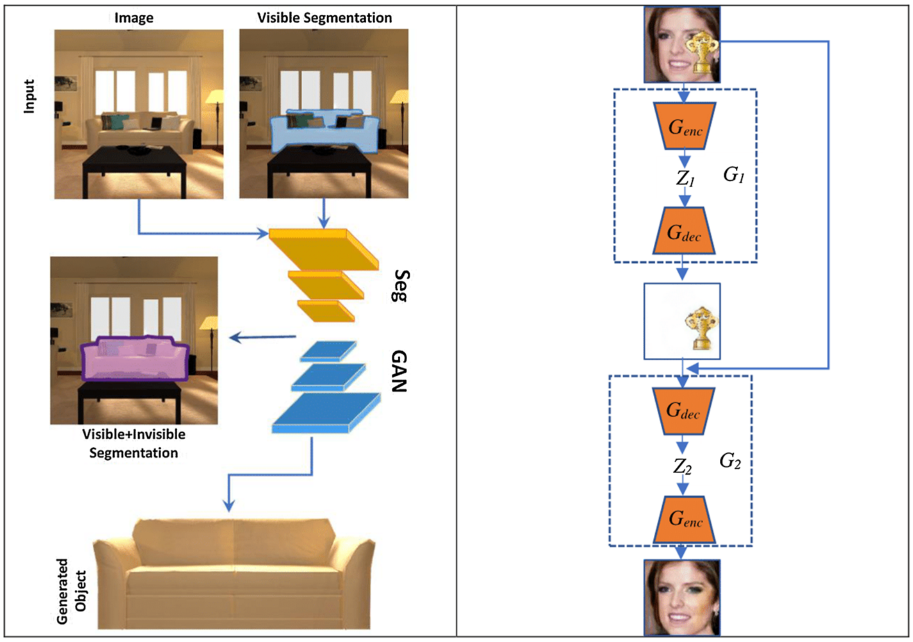


Fig. 33 Illustration of SeGAN [231] (left) and Occlusion-Aware GAN [232] (right)

generate the invisible parts of objects, i.e., learns to complete the appearance of occluded objects.

SeGAN [231], developed by Ehsani et al., is an interesting framework to segment the invisible part of the object and then generate the appearance by painting the invisible parts. The proposed framework uses a segmentor, a generator, and a discriminator to combine segmentation and generation tasks (Fig. 33). The segmentor takes an image and segmentation mask of the visible region of an object as an input, and then predicts an intermediate mask of the entire occluded object. The generator and discriminator are trained to generate an object image in which the invisible regions of the object are reconstructed.

Dong et al. [232] proposed a two stage model, named Occlusion-Aware GAN (OA-GAN), to remove arbitrary facial occlusions, e.g., faces with mask, microphone, cigarette, etc. OA-GAN is equipped with two GANs: The first GAN G_1 is designed to disentangle the occlusion, and the second GAN G_2 is trained to generate the occlusion free images given the generated occlusions.

Discussion

To provide a detailed overview and better comparison of various studies for imbalances in computer vision, the surveyed works have been summarized in Table 3.

GANs based methods that address the imbalance problem in classification tasks aim to increase the classification accuracy for the minority classes. Many of these methods use image-to-image translation to generate minority class images from one of the majority classes, while others generate minority class images from the random noise vector. GANs based intelligent oversampling [197] method outperforms both traditional sampling and data augmentation methods in classifying imbalanced image data. However, it

Table 3 Comparative summary of GANs for the problem of imbalances in computer vision

Category	Imbalance type	Study	Application
Binary classification	Inter class imbalance	DCGAN [153]	Malaria disease classification
	Inter class imbalance	SDGAN [154]	Industrial defect classification
	Inter class imbalance	BAGAN [155]	Image classification
	Inter class imbalance	CiGAN [156]	Mammogram classification
	Inter class imbalance	CycleGAN [157]	Mammogram classification
	Inter class imbalance	DCGAN [233]	Mammogram classification
	Inter class imbalance	CovidGAN [159]	Covid19 classification
	Intra class imbalance	Clustering + GAN [163]	Imbalanced intra class classification
	Intra class imbalance	Semantically decomposed GAN [234]	Imbalanced intra class classification
	Intra class imbalance	VAE + GAN [115]	Facial Attribute editing
	Intra class imbalance	AttGAN [64]	Facial Attribute editing
	Intra class imbalance	IcGAN [65]	Facial Attribute editing
	Intra class imbalance	ResAttr-GAN [66]	Facial Attribute editing
	Intra class imbalance	ARU-net [170]	Facial Attribute editing
	Intra class imbalance	SaGAN [171]	Facial Attribute editing
	Intra class imbalance	PN-GAN [176]	Person reidentification
	Intra class imbalance	PTGAN [177]	Person reidentification
	Intra class imbalance	CycleGAN [178]	Person reidentification
	Intra class imbalance	SPGAN [179]	Person reidentification
	Intra class imbalance	FDGAN [180]	Person reidentification
	Intra class imbalance	Cross view GAN [182]	Vehicle reidentification
	Intra class imbalance	DCGAN [183]	Vehicle reidentification
	Intra class imbalance	F-CGAN [184]	Fine grained classification
	Intra class imbalance	DCGAN + Fine grained Classifier [188]	Fine grained classification
	Intra class imbalance	General-to-Detailed GAN [190]	Fine grained classification
Multi class classification	Few minority-many majority class imbalance	Cycle GAN [197]	Emotion classification
	Few minority-many majority class imbalance	DCGAN [198]	Weather classification
	Few minority-many majority class imbalance	DCGAN + Ensemble learning [199]	Weather classification
	Few minority-many majority class imbalance	DCGAN [193]	Chest pathology classification
	Few minority-many majority class imbalance	DCGAN [200]	liver lesion classification
	Many majority- Few minority class imbalance	DCGAN [201]	Skin lesion classification
	Many majority- Many minority class imbalance	Cycle-GAN [192]	Plant disease classification
	Many majority- Many minority class imbalance	WGAN-GP [203]	Multi class classification
	Many majority- Many minority class imbalance	CE-GAN [205]	Multi class classification

Table 3 (continued)

Category	Imbalance type	Study	Application
Object detection	Object Scale imbalance	Perceptual GAN [208]	Traffic sign detection
	Object Scale imbalance	SOD-MTGAN [213]	Small object detection system
	Object Scale imbalance	Detector GAN [209]	Pedestrian and disease detection
	Imbalance due to occlusions and deformations	Adversarial-Fast-RCNN [216]	Occluded object detection
	Imbalance due to occlusions and deformations	Adversarial Occlusion-aware Face Detector [217]	Occluded face detection
	Imbalance due to occlusions and deformations	Cut-Paste GAN [218]	Occluded object detection
	Foreground Background object class imbalance	Task-aware synthetic data generation [219]	Object detection
	Foreground Background object class imbalance	Gene-GAN [221]	Object detection
	Foreground Background object class imbalance	PSIS [220]	Object detection
Segmentation	Pixel wise Imbalance	Sensitivity conditional GAN [118]	Shadow detection
	Pixel wise Imbalance	Pix2pix HD GAN [143]	Imbalanced pedestrian image segmentation
	Pixel wise Imbalance	Voxel GAN [226]	Brain tumor segmentation
	Pixel wise Imbalance	GAN + ensemble learning [228]	Medical image semantic segmentation
	Pixel wise Imbalance	GAN + Weighted categorical loss [227]	Heart image segmentation
	Imbalance due to occlusions	SeGAN[231]	Invisible part generation and Segmentation
	Imbalance due to occlusions	Occlusion-Aware GAN [232]	Occlusion free image generation

is not clear how much synthetic images must be blended with original images to achieve the maximum performance of the classifiers. Additionally, synthetic images would lead to additional noise to the original training dataset if the quality of the synthesized images is poor. Therefore, most of the surveyed methods in GANs based intelligent oversampling methods [197] focused mainly on balancing distribution as well as improving quality of the generated images.

Image-to-image translation [138] methods used for inter-class imbalance problem cannot be extended to solve intra-class imbalance as it is difficult to acquire image datasets with detailed labels. The interesting way to solve this problem is to employ clustering techniques in the feature space of the GANs to divide the images into different groups for automatic pattern recognition in the dataset. Improving the performance of the clustering techniques that clearly find the difference among clusters, is an area of future work.

GANs and encoder network hybrid models have a good potential to address intra class imbalance problem in face recognition and re-identification tasks. The key idea of these models is to work on latent code space rather than the pixel space. This is because for manipulating a fine grained image category, e.g., hair color, the latent code representation will operate only on that single latent code (hair color), whereas the pixel space will edit every single pixel in an image.

The fascinating approaches to use GANs for the problem of object level imbalances in object detection tasks fall into two general categories: 1. Generating more rare examples as intelligent oversampling used for class imbalance. These generated rare examples are introduced into the training dataset to address imbalance problems. 2. Learn an adversary in combination with original object detection algorithms. This adversary modifies the features to solve imbalance problems instead of generating examples in pixel space. i.e., to generate hard-to-detect samples by performing feature space manipulations.

The capability of super-resolution GANs are being used to up-sample small blurred objects into fine-scale ones and to recover detailed spatial information for accurate small object detection. This technique combines super-resolution GANs with object detection algorithms to solve the imbalances due to object size. The power of adversarial process is being used to increase the diversity of the small object locations in the images by copy-pasting small object instances several times at different locations.

Making the best use of GANs and combining them into U-Net architectures is an interesting way to solve pixel level imbalances in segmentation tasks. These architectures often use a weighted loss function to mitigate the pixel level imbalances. Combination of image in painting GANs with U-Net architectures has the great potential use in segmenting hidden objects. This technique is not only efficient in segmentation tasks, but also to infer the appearance of the objects beyond their visible parts. Overall, combining different deep learning models with adversarial process can provide a way to solve many other open problems in the computer vision field.

Future work

Even though GANs can be used as an effective way to unlock additional information from a dataset, the synthetic images generated by GANs cannot replace the real images completely. However, a blend of different proportions of real and GANs generated images are extremely useful to improve the diversity of the training samples and increase performance of the classifiers. Our future work intends to study the influences of blending different propositions of GANs generated images and real images on the classification performance. There are a very limited number of comparative studies that compare effectiveness of using GAN based synthetic images with other traditional methods for intra-class imbalances. We also intend to conduct the comparative study in order to validate the effectiveness of using synthetic images for intra class imbalances.

Inflating the size of the dataset brings another problem: One of the most significant limitations in computer vision experiments is computational resources. Sophisticated computer vision models trained on inflated dataset can perform complex tasks, the problem however is, how do we deploy such massive architecture on edge devices for instant usage. Handling this problem using knowledge distillation is non-trivial and an active field of research. Knowledge distillation is model compression technique in which a smaller network is trained with the help of the sophisticated pretrained model to achieve the similar accuracy. This training process is often referred to as "teacher-student", where the sophisticated pretrained model is the teacher and the smaller network is the student. Wang et al. [235] combine GANs and knowledge distillation to improve the efficiency of the student network in object detection. Similar to this work, we will attempt to further implement GANs and knowledge distillation combinations to other computer visions tasks.

As research on GANs are developing and maturing, assessment of performance has become essential. Evaluation metrics helps to quantitatively measure how well GANs models are performing, also to assess the relative performance of GANs. Very often the performance of GANs is measured by the manual inspection of the visual fidelity of generated images. However, the manual inspection is cumbersome, subjective, time-consuming, and sometimes misleading. Lack of universal evaluation metrics can impede the development of GANs. Introducing new performance measures to evaluate both diversity and fidelity of generated images is a very important area for future work.

Manually designing GANs architecture for a given task is time-consuming and sometimes has a tendency of errors. This drawback has led researchers to move on to the next stage of automating GANs architecture in the form of neural architecture search (NAS). Another interesting area of further research is to use meta-heuristic search algorithms that assist architectural search and find optimal GANs architecture which outperforms human created GANs models.

Achieving equilibrium between the generator and discriminator of the GANs can take a long time relative to other deep neural networks. Distributed training of GAN through parallelization and cluster computing is another important area of future work to cut down the training time.

Most of the applications of the GANs so far have been for creating synthetic images. GANs are not limited to the visual domain and can be also applied to non-visual applications. For example, Paganini et al. [236] used GANs to predict the outcome of high energy particle physics experiments. Instead of using explicit Monte Carlo simulation of the real physics of every step, the GANs learn by example what outcome is likely to occur in each situation. The GANs reduce the computational cost of high energy particle simulation, enough to save millions of dollars' worth of supercomputer time. We believe that the invention of new applications using this powerful tool will be continued in the future.

Conclusion

This paper surveys various GANs architectures that have been used for addressing the different imbalance problems in computer vision tasks. In this survey, we first provided detailed background information on deep generative models and GAN variants from the architecture, algorithm, and training tricks perspective. In order to present a clear roadmap of various imbalance problems in computer vision tasks, we introduced taxonomy of the imbalance problems. Following the proposed taxonomy, we discussed each type of problems separately in detail and presented the GANs based solutions with important features of each approach and their architectures. We focused mainly on the real-world applications where GAN based synthetic images are used to alleviate class imbalance. In addition to the thorough discussion on the imbalance problems and their solutions, we addressed many open issues that are crucial for computer vision applications.

Synthetic but realistic images generated using the methods discussed in this survey have the potential to mitigate the class imbalance problem while preserving the extrinsic distribution. Many of the methods surveyed in this paper tackled the highly complex imbalances by combining GANs architecture with different other deep learning frameworks. Specifically, the use of autoencoders with GANs has offered an effective way to perform feature space manipulations instead of complex pixel space operations.

Synthetic images generated by GANs cannot be used as the complete replacement for real datasets. However, the blend of real and GANs generated images have enormous potential to increase the performance of the deep learning model. Looking into the future, GAN-related research in image as well as non-image data domains to address the problem of imbalances and limited training dataset would continue to expand. We conclude that the future of GANs is promising and there are clearly a lot of opportunities for further research and applications in many fields.

Abbreviations

ConvNets: Convolutional neural networks; SMOTE: Synthetic minority oversampling technique; ADASYN: Adaptive synthetic sampling; IHM: Instance hardness measure; SSL: Semi-supervised learning; R-CNN: Region-based convolutional neural networks; RPN: Region proposal network; YOLO: You only look once; SSD: Singe shot detection; SNIP: Scale normalization for image pyramids; FPN: Feature pyramid networks; RNN: Recurrent neural networks; LSTM: Long short-term memory; PCA: Principle component analysis; MADE: Masked autoencoder density estimator; ARs: Autoregressive models; FVBNs: Fully visible belief networks; RGB: Red Green blue; NADE: Neural autoregressive density estimator; MADE: Masked autoencoder density estimator; VAEs: Variational auto encoders; CVAE: Conditional variational auto encoders; DC-IGN: Deep convolutional inverse graphics network; IWVAE: Importance weighted Variational Auto Encoders; VQ-VAEs: Vector quantized variational auto encoders; DRAW: Deep recurrent attentive writer; EMD: Earth mover Distance; TTUR : Two time-scale update rule; DDSM: Digital database for screening mammography; ARU-net: Adversarially regularized U-net; AMN: Attribute manipulation network; SiaNet: Siamese network; CV: Coefficient of variation; AP: Average precision; ASTN: Adversarial spatial transformer network; ASDN: Adversarial spatial dropout network; mAP: Mean average precision; AOFD: Adversarial occlusion aware face detection; PSIS: Progressive and selective instance-switching; ADAM: Adaptive moment estimation optimizer; ReLU: Rectified linear unit; GANs: Generative adversarial neural networks; cGAN: Conditional generative adversarial network; ACGAN: Auxiliary classifier generative adversarial network; VACGAN: Versatile Auxiliary classifier generative adversarial network; InfoGAN: Information maximizing generative adversarial network; SCGAN: Similarity constraint generative adversarial network; DCGAN: Deep convolutional generative adversarial network; ProGAN: Progressive growing of generative adversarial network; LAPGAN: Laplacian generative adversarial network; GRAN: Generative recurrent adversarial networks; D2GAN: Dual discriminator generative adversarial network; MADGAN: Multi-agent diverse generative adversarial network; CoGAN: Coupled generative adversarial network; DEGAN: Decoder encoder generative adversarial network; VAEGAN: Variational autoencoder generative adversarial network; AAE: Adversarial autoencoders; ALI: Adversarially learned inference; BiGAN: Bidirectional generative adversarial network; SRGAN: Super-resolution generative adversarial network; SAGAN: Self-attention generative adversarial network; WGAN: Wasserstein generative adversarial network; WGAN-GP: Wasserstein generative adversarial network with gradient penalty; LSGAN: Least squares generative adversarial network; EBGAN: Energy based generative adversarial network; BEGAN: Boundary equilibrium generative adversarial network; SD-GAN: Surface defect-generative adversarial network; BAGAN: Balancing generative adversarial network; ciGAN: Conditional infilling generative adversarial network; IcGAN: Invertible conditional generative adversarial network; PNGAN: Pose-normalized generative adversarial network; PTGAN: Person transfer generative adversarial network; SPGAN: Similarity preserving cycle consistent generative adversarial network; FD-GAN: Feature distilling generative adversarial network; F-CGAN: Fine grained conditional GAN; CE-GAN: Class expert generative adversarial network; ScGAN: Sensitivity conditional generative adversarial network; OAGAN: Occlusion-aware generative adversarial network.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions on the paper. Also, we acknowledge the members of the Autonomous and Intelligent Systems Unit, Tekniker, for valuable discussions and collaborations.

Authors' contributions

VS performed the primary literature review and analysis of this survey, and also drafted the manuscript. IM, JJAM and AG worked with VS to develop the article's framework and focus. IM and JJAM double checked the manuscript and provided several advanced ideas for this manuscript. All authors read and approved the final manuscript.

Funding

This research work was undertaken in the context of DIGIMAN4.0 project ("Digital Manufacturing Technologies for Zero defect", https://www.digiman4-0.mek.dtu.dk/). DIGIMAN4.0 is a European Training Network supported by Horizon 2020, the EU Framework Programme for Research and Innovation (Project ID: 814225). This research was also partly supported by the ELKARTEK project KK-2020/00049 3KIA of the Basque Government.

Availability of data and materials
Not applicable.

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Author details
[1] Autonomous and Intelligent Systems Unit, Tekniker, Member of Basque Research and Technology Alliance, Eibar, Spain. [2] Design and Manufacturing Engineering Department, Universidad de Zaragoza, 3 María de Luna Street, Torres Quevedo Bld, 50018 Zaragoza, Spain.

Received: 30 July 2020 Accepted: 16 January 2021
Published online: 29 January 2021

References

1. Nugraha BT, Su SF, Fahmizal. Towards self-driving car using convolutional neural network and road lane detector. Proceedings of the 2nd International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology, ICACOMIT 2017. 2017;2018-Janua:65–9.
2. Yadav SS, Jadhav SM. Deep convolutional neural network based medical image classification for disease diagnosis. J Big Data. 2019. https://doi.org/10.1186/s40537-019-0276-2.
3. Gutierrez A, Ansuategi A, Susperregi L, Tubío C, Rankić I, Lenža L. A Benchmarking of learning strategies for pest detection and identification on tomato plants for autonomous scouting robots using internal databases. J Sensors. 2019. https://doi.org/10.1155/2019/5219471.
4. Santos L, Santos FN, Oliveira PM, Shinde P. Deep learning applications in agriculture: a short review. Advances in intelligent systems and computing. Fourth Ibe. 2020. https://doi.org/10.1007/978-3-030-35990-4_12.
5. Wang T, Chen Y, Qiao M, Snoussi H. A fast and robust convolutional neural network-based defect detection model in product quality control. Int J Adv Manufactur Technol. 2018;94:3465–71.
6. Hashemi M. Enlarging smaller images before inputting into convolutional neural network: zero-padding vs interpolation. J Big Data. 2019. https://doi.org/10.1186/s40537-019-0263-7.
7. Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE . 1998;86:2278–324. http://ieeexplore.ieee.org/document/726791/
8. Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. 2014 IEEE Conference on Computer Vision and Pattern Recognition . IEEE; 2014. p. 580–7. http://ieeexplore.ieee.org/document/6909475/
9. Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) . IEEE; 2015. p. 3431–40. http://arxiv.org/abs/1605.06211
10. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Adv Neural Informat Process Syst. 2012;2:1097–105.
11. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 3rd International Conference on Learning Representations, ICLR 2015–Conference Track Proceedings. 2015;1–14.
12. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going Deeper with Convolutions. CoRR . 2014; abs/1409.4. https://arxiv.org/abs/1409.4842
13. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. Proceedings of the IEEE computer society conference on computer vision and pattern recognition. 2016. p. 770–8. http://arxiv.org/abs/1512.03385
14. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) . IEEE; 2016. p. 2818–26. http://arxiv.org/abs/1512.00567
15. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) . IEEE; 2017. p. 2261–9. http://arxiv.org/abs/1608.06993
16. Buda M, Maki A, Mazurowski MA. A systematic study of the class imbalance problem in convolutional neural networks. Neural Netw. 2018;106:249–59. https://linkinghub.elsevier.com/retrieve/pii/S0893608018302107
17. Al-Stouhi S, Reddy CK. Transfer learning for class imbalance problems with inadequate data. Knowl Informat Syst. 2016;48:201–28. https://doi.org/10.1007/s10115-015-0870-3
18. Ali A, Shamsuddin SM, Ralescu AL. Classification with class imbalance problem: a review. Int J Adv Soft Comput Applicat. 2015;7:176–204.
19. Zhang J, Xia Y, Wu Q, Xie Y. Classification of medical images and illustrations in the biomedical literature using synergic deep learning. 2017. http://arxiv.org/abs/1706.09092
20. Dong Q, Gong S, Zhu X. Imbalanced deep learning by minority class incremental rectification. IEEE Transactions on Pattern Analysis and Machine Intelligence . 2019;41:1367–81. https://ieeexplore.ieee.org/document/8353718
21. Zhang Y, Li B, Lu H, Irie A, Ruan X. Sample-Specific SVM learning for person re-identification. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) . IEEE; 2016. p. 1278–87. http://ieeexplore.ieee.org/document/7780512/
22. Sawant MM, Bhurchandi KM. Age invariant face recognition: a survey on facial aging databases, techniques and effect of aging. Artific Intell Rev. 2019;52:981–1008. https://doi.org/10.1007/s10462-018-9661-z.

23. Mostafa E, Ali A, Alajlan N, Farag A. Pose Invariant Approach for Face Recognition at Distance. Berlin : Springer; 2012. p. 15–28. https://doi.org/10.1007/978-3-642-33783-3_2.
24. Japkowicz N, Stephen S. The class imbalance problem: a systematic study. Intell Data Analy. 2002;6:429–49. https://doi.org/10.5555/1293951.1293954.
25. Chawla NV. Data mining for imbalanced datasets: an overview. data mining and knowledge discovery handbook. New York : Springer-Verlag; 2009. p. 853–67. https://doi.org/10.1007/0-387-25465-X_40.
26. Chawla NV, Japkowicz N, Kotcz A. Special Issue on Learning from Imbalanced Data Sets. ACM SIGKDD Explorations Newsletter. 2004; 6: 1–6. https://doi.org/10.1145/1007730.1007733
27. Chawla N V., Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: Synthetic minority over-sampling technique. J Artific Intell Res. 2011;16:321–57. https://doi.org/10.1613/jair.953. https://arxiv.org/abs/1106.1813
28. Haibo He, Yang Bai, Garcia EA, Shutao Li. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence) . IEEE; 2008. p. 1322–8. http://ieeexplore.ieee.org/document/4633969/
29. Puntumapon K, Rakthamamon T, Waiyamai K. Cluster-based minority over-sampling for imbalanced datasets. IEICE Transactions on Information and Systems . 2016;E99.D:3101–9. https://www.jstage.jst.go.jp/article/transinf/E99.D/12/E99.D_2016EDP7130/_article
30. Simard PY, Steinkraus D, Platt JC. Best practices for convolutional neural networks applied to visual document analysis. Seventh International Conference on Document Analysis and Recognition, 2003 Proceedings . IEEE Comput. Soc; p. 958–63. http://ieeexplore.ieee.org/document/1227801/
31. Lemley J, Bazrafkan S, Corcoran P. Deep Learning for Consumer Devices and Services: Pushing the limits for machine learning, artificial intelligence, and computer vision. IEEE Consumer Electronics Magazine . 2017;6:48–56. http://ieeexplore.ieee.org/document/7879402/
32. Shorten C, Khoshgoftaar TM. A survey on image data augmentation for deep learning. J Big Data. 2019;6:60. https://doi.org/10.1186/s40537-019-0197-0.
33. Wu H, Prasad S. Semi-Supervised Deep Learning Using Pseudo Labels for Hyperspectral Image Classification. IEEE Transactions on Image Processing . 2018;27:1259–70. http://ieeexplore.ieee.org/document/8105856/
34. van Engelen JE, Hoos HH. A survey on semi-supervised learning. Mach Learn. 2020;109:373–440. https://doi.org/10.1007/s10994-019-05855-6.
35. Thai-Nghe N, Gantner Z, Schmidt-Thieme L. Cost-sensitive learning methods for imbalanced data. The 2010 International Joint Conference on Neural Networks (IJCNN) . IEEE; 2010. p. 1–8. http://ieeexplore.ieee.org/document/5596486/
36. Girshick R. Fast R-CNN. 2015 IEEE International Conference on Computer Vision (ICCV) . IEEE; 2015. p. 1440–8. http://ieeexplore.ieee.org/document/7410526/
37. Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence . 2017;39:1137–49. http://ieeexplore.ieee.org/document/7485869/
38. He K, Gkioxari G, Dollar P, Girshick R. Mask R-CNN. IEEE Transactions on pattern analysis and machine intelligence. 2020;42:386–97. https://ieeexplore.ieee.org/document/8372616/
39. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, et al. SSD: Single Shot MultiBox Detector. In: Leibe B, Matas J, Sebe N, Welling M, editors. Cham: Springer International Publishing; 2016. p. 21–37. Doi: https://doi.org/10.1007/978-3-319-46448-0_2
40. Redmon JSDRGAF. (YOLO) You Only Look Once. Cvpr. 2016;
41. Yan X, Gong H, Jiang Y, Xia S-T, Zheng F, You X, et al. Video scene parsing: an overview of deep learning methods and datasets. Computer Vision and Image Understanding . 2020;201:103077. https://linkinghub.elsevier.com/retrieve/pii/S1077314220301120
42. Hsu Y-W, Wang T-Y, Perng J-W. Passenger flow counting in buses based on deep learning using surveillance video. Optik . 2020;202:163675. https://linkinghub.elsevier.com/retrieve/pii/S0030402619315736
43. Singh B, Davis LS. An analysis of scale invariance in object detection–SNIP. 2018 IEEE/CVF Conference on computer vision and pattern recognition. IEEE; 2018. p. 3578–87. https://ieeexplore.ieee.org/document/8578475/
44. Yang F, Choi W, Lin Y. Exploit All the Layers: Fast and Accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) . IEEE; 2016. p. 2129–37. http://ieeexplore.ieee.org/document/7780603/
45. Singh B, Najibi M, Davis LS. SNIPER: Efficient Multi-Scale Training. 32nd conference on neural information processing systems. Montréal; 2018. http://arxiv.org/abs/1805.09300
46. Lin T-Y, Dollar P, Girshick R, He K, Hariharan B, Belongie S. Feature Pyramid Networks for Object Detection. 2017 IEEE conference on computer vision and pattern recognition (CVPR). IEEE; 2017. p. 936–44. http://ieeexplore.ieee.org/document/8099589/
47. Lin T-Y, Goyal P, Girshick R, He K, Dollar P. Focal Loss for Dense Object Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2020;42:318–27. https://ieeexplore.ieee.org/document/8417976/
48. Dollar P, Wojek C, Schiele B, Perona P. Pedestrian detection: a benchmark. 2009 IEEE Conference on Computer Vision and Pattern Recognition . IEEE; 2009. p. 304–11. https://ieeexplore.ieee.org/document/5206631/
49. Zhong Z, Zheng L, Kang G, Li S, Yang Y. Random Erasing Data Augmentation. 2017. http://arxiv.org/abs/1708.04896
50. Wang X, Shrivastava A, Gupta A. A-Fast-RCNN: Hard positive generation via adversary for object detection. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE; 2017. p. 3039–48. http://arxiv.org/abs/1704.03414
51. Badrinarayanan V, Kendall A, Cipolla R. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017;39:2481–95. http://arxiv.org/abs/1511.00561
52. Ronneberger O, Fischer P, Brox T. U-Net: Convolutional networks for biomedical image segmentation. 2015. p. 234–41. http://arxiv.org/abs/1505.04597

53. Diakogiannis FI, Waldner F, Caccetta P, Wu C. ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data. ISPRS Journal of Photogrammetry and Remote Sensing . 2020;162:94–114. https://linkinghub.elsevier.com/retrieve/pii/S0924271620300149
54. Yurtsever E, Lambert J, Carballo A, Takeda K. A survey of autonomous driving: common practices and emerging technologies. 2019. http://arxiv.org/abs/1906.05113
55. Tabernik D, Šela S, Skvarč J, Skočaj D. Segmentation-based deep-learning approach for surface-defect detection. 2019. http://arxiv.org/abs/1903.08536
56. Rizwan I Haque I, Neubert J. Deep learning approaches to biomedical image segmentation. Informatics in Medicine Unlocked. 2020;18:100297. https://linkinghub.elsevier.com/retrieve/pii/S235291481930214X
57. Cordts M, Omran M, Ramos S, Rehfeld T, Enzweiler M, Benenson R, et al. The cityscapes dataset for semantic urban scene understanding. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2016;2016-Decem:3213–23.
58. Menze BH, Jakab A, Bauer S, Kalpathy-Cramer J, Farahani K, Kirby J, et al. The multimodal brain tumor image segmentation benchmark (BRATS). IEEE Transac Med Imag. 2015;34:1993–2024. http://ieeexplore.ieee.org/document/6975210/
59. Murphy KP. Machine learning: a probabilistic perspective (Adaptive Computation and Machine Learning series). Cambridge: The MIT Press; 2012.
60. Milletari F, Navab N, Ahmadi S-A. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. 2016 Fourth International Conference on 3D Vision (3DV) . IEEE; 2016. p. 565–71. http://ieeexplore.ieee.org/document/7785132/
61. Crum WR, Camara O, Hill DLG. Generalized Overlap Measures for Evaluation and Validation in Medical Image Analysis. IEEE Transact Med Imag. 2006;25:1451–61. http://ieeexplore.ieee.org/document/1717643/
62. Salehi SSM, Erdogmus D, Gholipour A. Tversky loss function for image segmentation using 3D fully convolutional deep networks. 2017. p. 379–87. http://arxiv.org/abs/1706.05721
63. Berman M, Triki AR, Blaschko MB. The Lovasz-Softmax Loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition . IEEE; 2018. p. 4413–21. https://ieeexplore.ieee.org/document/8578562/
64. He Z, Zuo W, Kan M, Shan S, Chen X. AttGAN: Facial attribute editing by only changing what you want. IEEE transactions on image processing . 2019;28:5464–78. https://ieeexplore.ieee.org/document/8718508/
65. Perarnau G, van de Weijer J, Raducanu B, Álvarez JM. Invertible Conditional GANs for image editing. Conference on Neural Information Processing Systems . 2016. http://arxiv.org/abs/1611.06355
66. Tao R, Li Z, Tao R, Li B. ResAttr-GAN: Unpaired deep residual attributes learning for multi-domain face image translation. IEEE Access . 2019;7:132594–608. https://ieeexplore.ieee.org/document/8836502/
67. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial nets. Adv Neural Inf Process Syst. 2014;3:2672–80.
68. Bowles C, Chen L, Guerrero R, Bentley P, Gunn R, Hammers A, et al. GAN Augmentation: augmenting training data using generative adversarial networks. 2018; http://arxiv.org/abs/1810.10863
69. Oord A van den, Kalchbrenner N, Kavukcuoglu K. Pixel recurrent neural networks. 2016; http://arxiv.org/abs/1601.06759
70. Sejnowski MIJTJ. Learning and relearning in boltzmann machines. Graphical models: foundations of neural computation, MITP. 2001;
71. McClelland DERJL. Information processing in dynamical systems: foundations of harmony theory. parallel distributed processing: explorations in the microstructure of Cognition: Foundations, MITP. 1987;194–281.
72. Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science. 2006;313:504–7.
73. Salakhutdinov R, Hinton G. Deep Boltzmann machines. J Machine Learn Res. 2009;5:448–55.
74. Lee H, Grosse R, Ranganath R, Y. Ng A. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. Computer Science Department, Stanford University . 2009;8. http://robotics.stanford.edu/~ang/papers/icml09-ConvolutionalDeepBeliefNetworks.pdf
75. Hinton GE, Osindero S, Teh Y-W. A fast learning algorithm for deep belief nets. Neural Comput. 2006;18:1527–54. https://doi.org/10.1162/neco.2006.18.7.1527.
76. Ramachandran P, Paine T Le, Khorrami P, Babaeizadeh M, Chang S, Zhang Y, et al. Fast generation for convolutional autoregressive models. 2017; http://arxiv.org/abs/1704.06001
77. Frey BJ. Graphical models for machine learning and digital communication. Cambridge: MIT Press; 1998.
78. Frey BJ, Hinton GE, Dayan P. Does the Wake-sleep algorithm produce good density estimators? Advances in neural information processing systems . 1996;13:661–70. http://www.cs.utoronto.ca/~hinton/absps/wsperf.pdf%5Cnpapers2://publication/uuid/BCC0547E-7C14-42EC-8693-D800C5819C79
79. Uria B, Côté M-A, Gregor K, Murray I, Larochelle H. Neural autoregressive distribution estimation. J Mach Learn Res. 2016;17:1–37. http://arxiv.org/abs/1605.02226
80. Schuller B, Wöllmer M, Moosmayr T, Rigoll G. Recognition of noisy speech: a comparative survey of robust model architecture and feature enhancement. EURASIP J Audio Speech Music Process. 2009;2009:942617. http://asmp.eurasipjournals.com/content/2009/1/942617
81. Yang S, Lu H, Kang S, Xue L, Xiao J, Su D, et al. On the localness modeling for the self-attention based end-to-end speech synthesis. Neural Netw. 2020;125:121–30. https://linkinghub.elsevier.com/retrieve/pii/S0893608020300447
82. Ghosh R, Vamshi C, Kumar P. RNN based online handwritten word recognition in Devanagari and Bengali scripts using horizontal zoning. Pattern Recognit. 2019;92:203–18. https://linkinghub.elsevier.com/retrieve/pii/S0031320319301384
83. Chen J, Zhuge H. Extractive summarization of documents with images based on multi-modal RNN. Future Generat Comput Syst. 2019;99:186–96. https://linkinghub.elsevier.com/retrieve/pii/S0167739X18326876
84. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9:1735–80. https://doi.org/10.1162/neco.1997.9.8.1735.

85. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. arXiv . 2017; http://arxiv.org/abs/1706.03762
86. Theis L, Bethge M. Generative Image Modeling Using Spatial LSTMs. Proceedings of the 28th International Conference on Neural Information Processing Systems–Volume 2. Cambridge: MIT Press; 2015. p. 1927–1935.
87. Krizhevsky A. Learning multiple layers of features from tiny images . 2009. http://www.cs.toronto.edu/~kriz/cifar.html
88. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, et al. ImageNet large scale visual recognition challenge. Int J Comput Vis. 2015;115:211–52. https://doi.org/10.1007/s11263-015-0816-y.
89. Oord A van den, Kalchbrenner N, Vinyals O, Espeholt L, Graves A, Kavukcuoglu K. Conditional image generation with PixelCNN Decoders. http://arxiv.org/abs/1606.05328
90. Salimans T, Karpathy A, Chen X, Kingma DP. PixelCNN++: Improving the PixelCNN with discretized logistic mixture likelihood and other modifications. 2017; http://arxiv.org/abs/1701.05517
91. Chen X, Mishra N, Rohaninejad M, Abbeel P. PixelSNAIL: an improved autoregressive generative model. 2017. http://arxiv.org/abs/1712.09763
92. Vincent P, Larochelle H, Bengio Y, Manzagol P-A. Extracting and composing robust features with denoising autoencoders. Proceedings of the 25th international conference on Machine learning - ICML '08 . New York: ACM Press; 2008. p. 1096–103. https://linkinghub.elsevier.com/retrieve/pii/S0925231218306155
93. Baldi P. Autoencoders, unsupervised learning, and deep architectures . PMLR; 2012. http://proceedings.mlr.press/v27/baldi12a.html
94. Y. Ng A. Sparse autoencoder .https://web.stanford.edu/class/cs294a/sparseAutoencoder.pdf
95. Masci J, Meier U, Cireşan D, Schmidhuber J. Stacked convolutional auto-encoders for hierarchical feature extraction. 2011. p. 52–9. https://doi.org/10.1007/978-3-642-21735-7_7
96. Rifai S, Vincent P, Muller X, Glorot X, Bengio Y. Contractive auto-encoders: explicit invariance during feature extraction. ICML. 2011.
97. Kingma DP, Welling M. Auto-encoding variational bayes. 2013; http://arxiv.org/abs/1312.6114
98. Tan S, Li B. Stacked convolutional auto-encoders for steganalysis of digital images. Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific. IEEE; 2014. p. 1–4.
99. Germain M, Gregor K, Murray I, Larochelle H. MADE: Masked autoencoder for distribution estimation. 2015. http://arxiv.org/abs/1502.03509
100. Schmidhuber J. Learning factorial codes by predictability minimization. Neural Comput. 1992;4:863–79. https://doi.org/10.1162/neco.1992.4.6.863.
101. Sohn K, Yan X, Lee H. Learning structured output representation using deep conditional generative models. Adv Neural Informat Process Syst. 2015;2015-Janua:3483–91.
102. Higgins I, Matthey L, Pal A, Burgess C, Glorot X, Botvinick M, et al. -VAE: Learning basic visual concepts with a constrained variational framework. 5th International Conference on Learning Representations, ICLR 2017–Conference Track Proceedings. 2019;1–13.
103. Kulkarni TD, Whitney W, Kohli P, Tenenbaum JB. Deep convolutional inverse graphics network. 2015. http://arxiv.org/abs/1503.03167
104. Huang C-W, Sankaran K, Dhekane E, Lacoste A, Courville A. Hierarchical Importance Weighted Autoencoders. In: Chaudhuri K, Salakhutdinov R, editors. Long Beach, California, USA: PMLR; 2019. p. 2869–78. http://proceedings.mlr.press/v97/huang19d.html
105. Gulrajani I, Kumar K, Ahmed F, Taiga AA, Visin F, Vazquez D, et al. PixelVAE: A latent variable model for natural images. 2016; Ahttp://arxiv.org/abs/1611.05013
106. Chen X, Kingma DP, Salimans T, Duan Y, Dhariwal P, Schulman J, et al. Variational Lossy Autoencoder. 2016. http://arxiv.org/abs/1611.02731
107. Gregor K, Danihelka I, Graves A, Rezende DJ, Wierstra D. DRAW: A recurrent neural network for image generation. 2015. http://arxiv.org/abs/1502.04623
108. Oord A van den, Vinyals O, Kavukcuoglu K. Neural Discrete Representation Learning. 31st Conference on Neural Information Processing Systems . Long Beach, California, USA; 2017. http://arxiv.org/abs/1711.00937
109. Razavi A, Oord A van den, Vinyals O. Generating diverse high-fidelity images with VQ-VAE-2. Advances in neural information processing systems 32. 2019. http://arxiv.org/abs/1906.00446
110. Huszár F. How (not) to Train your generative model: scheduled sampling, likelihood, adversary? 2015. http://arxiv.org/abs/1511.05101
111. Lotter W, Kreiman G, Cox D. Deep Predictive coding networks for video prediction and unsupervised learning. 2016. http://arxiv.org/abs/1605.08104
112. Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. 2015. http://arxiv.org/abs/1511.06434
113. Makhzani A, Shlens J, Jaitly N, Goodfellow I, Frey B. Adversarial Autoencoders. 2015; Available from: http://arxiv.org/abs/1511.05644
114. Dumoulin V, Belghazi I, Poole B, Mastropietro O, Lamb A, Arjovsky M, et al. Adversarially Learned Inference. 2016. http://arxiv.org/abs/1606.00704
115. Larsen ABL, Sønderby SK, Larochelle H, Winther O. Autoencoding beyond pixels using a learned similarity metric. 2015. http://arxiv.org/abs/1512.09300
116. Zhong G, Gao W, Liu Y, Yang Y. Generative Adversarial networks with decoder-encoder output noise. 2018. http://arxiv.org/abs/1807.03923
117. Srivastava A, Valkov L, Russell C, Gutmann MU, Sutton C. VEEGAN: Reducing Mode Collapse in GANs using implicit variational learning. 2017. http://arxiv.org/abs/1705.07761
118. Mirza M, Osindero S. Conditional generative adversarial nets. 2014. http://arxiv.org/abs/1411.1784
119. Odena A, Olah C, Shlens J. Conditional image synthesis with auxiliary classifier GANs. 2016. http://arxiv.org/abs/1610.09585

120. Bazrafkan S, Corcoran P. Versatile auxiliary classifier with generative adversarial network (VAC+GAN), Multi Class Scenarios. 2018. http://arxiv.org/abs/1806.07751
121. Chen X, Duan Y, Houthooft R, Schulman J, Sutskever I, Abbeel P. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. 2016. http://arxiv.org/abs/1606.03657
122. Li X, Chen L, Wang L, Wu P, Tong W. SCGAN: disentangled representation learning by adding similarity constraint on generative adversarial nets. IEEE Access . 2019;7:147928–38. https://ieeexplore.ieee.org/document/8476290/
123. Arjovsky M, Chintala S, Bottou L. Wasserstein GAN. 2017. http://arxiv.org/abs/1701.07875
124. Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville A. Improved training of Wasserstein GANs. 2017. http://arxiv.org/abs/1704.00028
125. Petzka H, Fischer A, Lukovnicov D. On the regularization of Wasserstein GANs. 2017. http://arxiv.org/abs/1709.08894
126. Mao X, Li Q, Xie H, Lau RYK, Wang Z, Smolley SP. Least squares generative adversarial networks. 2016. http://arxiv.org/abs/1611.04076
127. Zhao J, Mathieu M, LeCun Y. Energy-based Generative Adversarial Network. 2016. http://arxiv.org/abs/1609.03126
128. Berthelot D, Schumm T, Metz L. BEGAN: Boundary Equilibrium Generative Adversarial Networks. 2017. http://arxiv.org/abs/1703.10717
129. Wang R, Cully A, Chang HJ, Demiris Y. MAGAN: Margin adaptation for generative adversarial networks. 2017. http://arxiv.org/abs/1704.03817
130. Zhao J, Xiong L, Jayashree K, Li J, Zhao F, Wang Z, et al. Dual-agent GANs for photorealistic and identity preserving profile face synthesis. Advan Neural Informat Process Syst. 2017;2017:66–76.
131. Karras T, Aila T, Laine S, Lehtinen J. Progressive growing of GANs for improved quality, stability, and variation. 2017; http://arxiv.org/abs/1710.10196
132. Denton E, Chintala S, Szlam A, Fergus R. Deep generative image models using a laplacian pyramid of adversarial networks. Advances in Neural Information Processing Systems 28 . 2015. http://arxiv.org/abs/1506.05751
133. Im DJ, Kim CD, Jiang H, Memisevic R. Generating images with recurrent adversarial networks. 2016; http://arxiv.org/abs/1602.05110
134. Nguyen TD, Le T, Vu H, Phung D. Dual discriminator generative adversarial Nets. 2017; http://arxiv.org/abs/1709.03831
135. Ghosh A, Kulharia V, Namboodiri V, Torr PHS, Dokania PK. Multi-agent diverse generative adversarial networks. 2017. http://arxiv.org/abs/1704.02906
136. Liu M-Y, Tuzel O. Coupled generative adversarial networks. conference on neural information processing systems. 2016. http://arxiv.org/abs/1606.07536
137. Kim T, Cha M, Kim H, Lee JK, Kim J. Learning to discover cross-domain relations with generative adversarial networks. 2017. http://arxiv.org/abs/1703.05192
138. Zhu J-Y, Park T, Isola P, Efros AA. Unpaired Image-to-image translation using cycle-consistent adversarial networks. 2017 IEEE International Conference on Computer Vision (ICCV) . IEEE; 2017. p. 2242–51. http://arxiv.org/abs/1703.10593
139. Ledig C, Theis L, Huszar F, Caballero J, Cunningham A, Acosta A, et al. Photo-realistic single image super-resolution using a generative adversarial network. 2016; http://arxiv.org/abs/1609.04802
140. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 2014; http://arxiv.org/abs/1409.1556
141. Zhang H, Goodfellow I, Metaxas D, Odena A. Self-Attention Generative Adversarial Networks. 2018; http://arxiv.org/abs/1805.08318
142. Isola P, Zhu J-Y, Zhou T, Efros AA. Image-to-image translation with conditional adversarial networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE; 2017. p. 5967–76. http://ieeexplore.ieee.org/document/8100115/
143. Wang T-C, Liu M-Y, Zhu J-Y, Tao A, Kautz J, Catanzaro B. High-resolution image synthesis and semantic manipulation with conditional GANs. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition . IEEE; 2018. p. 8798–807. https://ieeexplore.ieee.org/document/8579015/
144. Bellemare MG, Danihelka I, Dabney W, Mohamed S, Lakshminarayanan B, Hoyer S, et al. The cramer distance as a solution to biased wasserstein gradients. 2017. http://arxiv.org/abs/1705.10743
145. Mroueh Y, Sercu T, Goel V. McGan: mean and covariance feature matching GAN. 2017. http://arxiv.org/abs/1702.08398
146. Li C-L, Chang W-C, Cheng Y, Yang Y, Póczos B. MMD GAN: towards deeper understanding of moment matching network. 2017. http://arxiv.org/abs/1705.08584
147. Mroueh Y, Sercu T. Fisher GAN. 2017. http://arxiv.org/abs/1705.09675
148. Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A, Chen X. Improved techniques for training GANs. 2016. http://arxiv.org/abs/1606.03498
149. Sønderby CK, Caballero J, Theis L, Shi W, Huszár F. Amortised MAP inference for image super-resolution. 2016. http://arxiv.org/abs/1610.04490
150. Heusel M, Ramsauer H, Unterthiner T, Nessler B, Hochreiter S. GANs trained by a two time-scale update rule converge to a local nash equilibrium. 2017. http://arxiv.org/abs/1706.08500
151. Miyato T, Kataoka T, Koyama M, Yoshida Y. Spectral normalization for generative adversarial networks. 2018. http://arxiv.org/abs/1802.05957
152. Heath M, Bowyer K, Kopans D, Moore R, Kegelmeyer WP. Digital database for screening mammography . https://www.mammoimage.org/databases/
153. Shoohi LM, Saud JH. Dcgan for handling imbalanced malaria dataset based on over-sampling technique and using cnn. Medico-Legal Update. 2020;20:1079–85.
154. Niu S, Li B, Wang X, Lin H. Defect image sample generation With GAN for Improving defect recognition. IEEE Transactions on Automation Science and Engineering . 2020;1–12. https://ieeexplore.ieee.org/document/9000806/

155. Mariani G, Scheidegger F, Istrate R, Bekas C, Malossi C. BAGAN: Data Augmentation with Balancing GAN. 2018; http://arxiv.org/abs/1803.09655
156. Wu E, Wu K, Cox D, Lotter W. Conditional infilling GANs for data augmentation in mammogram classification. 2018. p. 98–106. Doi: https://doi.org/10.1007/978-3-030-00946-5_11
157. Muramatsu C, Nishio M, Goto T, Oiwa M, Morita T, Yakami M, et al. Improving breast mass classification by shared data with domain transformation using a generative adversarial network. Comput Biol Med. 2020;119:103698. https://linkinghub.elsevier.com/retrieve/pii/S001048252030086X
158. Guan S. Breast cancer detection using synthetic mammograms from generative adversarial networks in convolutional neural networks. J Med Imag. 2019;6:1. https://doi.org/10.1117/1.JMI.6.3.031411.full.
159. Waheed A, Goyal M, Gupta D, Khanna A, Al-Turjman F, Pinheiro PR. CovidGAN: Data augmentation using auxiliary classifier GAN for improved Covid-19 detection. IEEE Access . 2020;8:91916–23. https://ieeexplore.ieee.org/document/9093842/
160. COVID-19 Chest X-Ray dataset initiative. https://github.com/agchung/Figure1-COVID-chestxray-dataset
161. Cohen JP, Morrison P, Dao L, Roth K, Duong TQ, Ghassemi M. COVID-19 Image data collection: prospective predictions are the future. 2020. http://arxiv.org/abs/2006.11988
162. Covid19 radiography database. https://www.kaggle.com/tawsifurrahman/covid19-radiography-database
163. Hase N, Ito S, Kanaeko N, Sumi K. Data augmentation for intra-class imbalance with generative adversarial network. In: Cudel C, Bazeille S, Verrier N, editors. Fourteenth International Conference on Quality Control by Artificial Vision . SPIE; 2019. p. 56. Available from: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/11172/2521692/Data-augmentation-for-intra-class-imbalance-with-generative-adversarial-network/https://doi.org/10.1117/12.2521692.full
164. Donahue C, Lipton ZC, Balsubramani A, McAuley J. Semantically Decomposing the Latent Spaces of Generative Adversarial Networks. 2017; http://arxiv.org/abs/1705.07904
165. Wang Y, Gong D, Zhou Z, Ji X, Wang H, Li Z, et al. Orthogonal deep features decomposition for age-invariant face recognition. 2018. p. 764–79. https://doi.org/10.1007/978-3-030-01267-0_45
166. Gong D, Li Z, Lin D, Liu J, Tang X. Hidden factor analysis for age invariant face recognition. 2013 IEEE International Conference on Computer Vision. IEEE; 2013. p. 2872–9. http://ieeexplore.ieee.org/document/6751468/
167. Yin X, Liu X. Multi-task convolutional neural network for pose-invariant face recognition. IEEE Transactions on Image Processing. 2018;27:964–75. http://ieeexplore.ieee.org/document/8080244/
168. Carcagnì P, Del CM, Cazzato D, Leo M, Distante C. A study on different experimental configurations for age, race, and gender estimation problems. EURASIP J Image Video Process. 2015;2015:37. https://doi.org/10.1186/s13640-015-0089-y.
169. Ziwei L, Ping L, Xiaogang W, Tang X. Large-scale CelebFaces attributes (CelebA) Dataset. 2018. http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html
170. Zhang J, Li A, Liu Y, Wang M. Adversarially Regularized U-Net-based GANs for facial attribute modification and generation. IEEE Access . 2019;7:86453–62. https://ieeexplore.ieee.org/document/8754728/
171. Zhang G, Kan M, Shan S, Chen X. Generative adversarial network with spatial attention for face attribute editing. 2018. p. 422–37. https://doi.org/10.1007/978-3-030-01231-1_26
172. Zheng Z, Yang X, Yu Z, Zheng L, Yang Y, Kautz J. joint discriminative and generative learning for person re-identification. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) . IEEE; 2019. p. 2133–42. https://ieeexplore.ieee.org/document/8954292/
173. Zhang X, Gao Y. Face recognition across pose: a review. pattern recognition . 2009;42:2876–96. https://linkinghub.elsevier.com/retrieve/pii/S0031320309001538
174. Tan X, Chen S, Zhou Z-H, Zhang F. Face recognition from a single image per person: a survey. pattern recognition. 2006;39:1725–45. https://linkinghub.elsevier.com/retrieve/pii/S0031320306001270
175. Zhao W, Chellappa R, Phillips PJ, Rosenfeld A. Face recognition. ACM computing surveys. 2003;35:399–458. http://portal.acm.org/citation.cfm?doid=954339.954342
176. Qian X, Fu Y, Xiang T, Wang W, Qiu J, Wu Y, et al. Pose-Normalized Image Generation for Person Re-identification. 2018. p. 661–78. https://doi.org/10.1007/978-3-030-01240-3_40
177. Wei L, Zhang S, Gao W, Tian Q. Person Transfer GAN to bridge domain gap for person re-identification. 2018 IEEE/CVF conference on computer vision and pattern recognition . IEEE; 2018. p. 79–88. https://ieeexplore.ieee.org/document/8578114/
178. Zhong Z, Zheng L, Zheng Z, Li S, Yang Y. Camera style adaptation for person re-identification. 2018 IEEE/CVF conference on computer vision and pattern recognition. IEEE; 2018. p. 5157–66. https://ieeexplore.ieee.org/document/8578639/
179. Deng W, Zheng L, Ye Q, Yang Y, Jiao J. Similarity-preserving image-image domain adaptation for person re-identification. 2018; http://arxiv.org/abs/1811.10551
180. Ge Y, Li Z, Zhao H, Yin G, Yi S, Wang X, et al. FD-GAN: Pose-guided Feature Distilling GAN for robust person re-identification. Adv Neural Informat Process Syst. 2018;2018:1222–33.
181. Zheng A, Lin X, Li C, He R, Tang J. Attributes guided feature learning for vehicle re-identification. 2019; http://arxiv.org/abs/1905.08997
182. Zhou Y, Shao L. Cross-View GAN Based Vehicle Generation for Re-identification. Procedings of the British Machine Vision Conference 2017 . British Machine Vision Association; 2017. http://www.bmva.org/bmvc/2017/papers/paper186/index.html
183. Wu F, Yan S, Smith JS, Zhang B. Vehicle re-identification in still images: application of semi-supervised learning and re-ranking. Signal Processing: Image Communication . 2019;76:261–71. https://linkinghub.elsevier.com/retrieve/pii/S0923596518305800
184. Fu Y, Li X, Ye Y. A multi-task learning model with adversarial data augmentation for classification of fine-grained images. Neurocomputing . 2020;377:122–9. https://linkinghub.elsevier.com/retrieve/pii/S0925231219313748

185. Ge Z, Bewley A, McCool C, Corke P, Upcroft B, Sanderson C. Fine-grained classification via mixture of deep convolutional neural networks. 2016 IEEE Winter Conference on Applications of Computer Vision (WACV) . IEEE; 2016. p. 1–6. http://ieeexplore.ieee.org/document/7477700/
186. Khosla A, Jayadevaprakash N, Yao B, Fei-Fei L. Novel dataset for fine-grained image categorization. Proc IEEE Conf Comput Vision and Pattern Recognition. 2011
187. Welinder P, Branson S, Mita T, Wah C, Schroff F. Caltech-ucsd Birds 200. Caltech-UCSD Technical Report . 2010;200:1–15. http://www.flickr.com/
188. Wang C, Yu Z, Zheng H, Wang N, Zheng B. CGAN-plankton: Towards large-scale imbalanced class generation and fine-grained classification. 2017 IEEE International Conference on Image Processing (ICIP) . IEEE; 2017. p. 855–9. http://ieeexplore.ieee.org/document/8296402/
189. Orenstein EC, Beijbom O, Peacock EE, Sosik HM. WHOI-Plankton-a large scale fine grained visual recognition benchmark dataset for plankton classification. 2015; http://arxiv.org/abs/1510.00745
190. Koga T, Nonaka N, Sakuma J, Seita J. General-to-Detailed GAN for infrequent class medical images. 2018; http://arxiv.org/abs/1812.01690
191. Zhu X, Liu Y, Qin Z, Li J. Data Augmentation in emotion classification using generative adversarial networks. 2017; http://arxiv.org/abs/1711.00648
192. Haseeb Nazki, Jaehwan Lee, Sook Yoon DSP. Image-to-image translation with GAN for Synthetic Data augmentation in plant disease datasets. Smart Media J. 2019;8:46–57. http://kism.or.kr/file/memoir/8_2_6.pdf
193. Salehinejad H, Valaee S, Dowdell T, Colak E, Barfett J. Generalization of deep neural networks for chest pathology classification in X-Rays using generative adversarial networks. ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing–Proceedings. 2018;2018-April:990–4.
194. Lu Y-W, Liu K-L, Hsu C-Y. Conditional Generative Adversarial Network for Defect Classification with Class Imbalance. 2019 IEEE International Conference on Smart Manufacturing, Industrial & Logistics Engineering (SMILE) . IEEE; 2019. p. 146–9. https://ieeexplore.ieee.org/document/8965320/
195. Shuo Wang, Xin Yao. Multiclass imbalance problems: analysis and potential solutions. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) . 2012;42:1119–30. http://ieeexplore.ieee.org/document/6170916/
196. Shuo W, Xin Y. Multiclass Imbalance Problems: Analysis and Potential Solutions. IEEE Transact Syst Man Cybernet Part B. 2012;42:1119–30.
197. Zhu X, Liu Y, Qin Z, Li J. Data augmentation in emotion classification using generative adversarial networks. 2017.
198. Li Z, Jin Y, Li Y, Lin Z, Wang S. imbalanced adversarial learning for weather image generation and classification. 2018 14th IEEE International Conference on Signal Processing (ICSP) . IEEE; 2018. p. 1093–7. https://ieeexplore.ieee.org/document/8652272/
199. Huang Y, Jin Y, Li Y, Lin Z. Towards imbalanced image classification: a generative adversarial network ensemble learning method. IEEE Access . 2020;8:88399–409. https://ieeexplore.ieee.org/document/9086504/
200. Frid-Adar M, Diamant I, Klang E, Amitai M, Goldberger J, Greenspan H. GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. Neurocomputing. 2018;321:321–31.
201. Rashid H, Tanveer MA, Aqeel Khan H. Skin lesion classification using GAN based data augmentation. 2019 41st annual international conference of the IEEE engineering in medicine and biology society (EMBC). IEEE; 2019. p. 916–9. https://ieeexplore.ieee.org/document/8857905/
202. Tschandl P, Rosendahl C, Kittler H. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Scientific Data . 2018;5:180161. http://www.nature.com/articles/sdata2018161
203. Bhatia S, Dahyot R. Using WGAN for improving imbalanced classification performance. AICS 2019. 2019.
204. Xiao H, Rasul K, Vollgraf R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. 2017;1–6. http://arxiv.org/abs/1708.07747
205. Fanny, Cenggoro TW. Deep learning for imbalance data classification using class expert generative adversarial network. Procedia Comput Sci. 2018;135:60–7.
206. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al. Microsoft COCO: Common objects in context. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2014;8693 LNCS:740–55.
207. Bai H, Wen S, Chan SHG. Crowd counting on images with scale variation and isolated clusters. Proceedings–2019 International Conference on Computer Vision Workshop, ICCVW 2019. 2019;18–27.
208. Li J, Liang X, Wei Y, Xu T, Feng J, Yan S. Perceptual generative adversarial networks for small object detection. 2017 IEEE conference on computer vision and pattern recognition (CVPR) . IEEE; 2017. p. 1951–9. http://ieeexplore.ieee.org/document/8099694/
209. Liu L, Muelly M, Deng J, Pfister T, Li LJ. Generative modeling for small-data object detection. Proceedings of the IEEE International Conference on Computer Vision. 2019; 2019-Octob: 6072–80.
210. Zhu Z, Liang D, Zhang S, Huang X, Li B, Hu S. Traffic-Sign Detection and Classification in the Wild. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) . IEEE; 2016. p. 2110–8. http://ieeexplore.ieee.org/document/7780601/
211. Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A. The pascal visual object classes (VOC) challenge. Int J Comput Vision. 2010;88:303–38. https://doi.org/10.1007/s11263-009-0275-4.
212. Dollar P, Wojek C, Schiele B, Perona P. Pedestrian detection: an evaluation of the state of the art. IEEE transactions on pattern analysis and machine intelligence . 2012;34:743–61. http://ieeexplore.ieee.org/document/5975165/
213. Bai Y, Zhang Y, Ding M, Ghanem B. SOD-MTGAN: Small object detection via multi-task generative adversarial network. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2018;11217 LNCS:210–26.
214. He K, Gkioxari G, Dollar P, Girshick R. Mask R-CNN. 2017 IEEE International Conference on Computer Vision (ICCV) . IEEE; 2017. p. 2980–8. http://ieeexplore.ieee.org/document/8237584/

215. B SC, Koznek N, Ismail A, Adam G, Narayan V, Schulze M. Computer Vision–ECCV 2018 Workshops . European Conference on Computer Vision 2018. 2019. https://doi.org/10.1007/978-3-030-11021-5
216. Wang X, Shrivastava A, Gupta A. A-Fast-RCNN: Hard positive generation via adversary for object detection. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) . IEEE; 2017. p. 3039–48. http://ieeexplore.ieee.org/document/8099807/
217. Chen Y, Song L, He R. Adversarial occlusion-aware face detection. 2017; http://arxiv.org/abs/1709.05188
218. Dwibedi D, Misra I, Hebert M. Cut, Paste and learn: surprisingly easy synthesis for instance detection. 2017 IEEE International conference on computer vision (ICCV) . IEEE; 2017. p. 1310–9. http://ieeexplore.ieee.org/document/8237408/
219. Tripathi S, Chandra S, Agrawal A, Tyagi A, Rehg JM, Chari V. Learning to generate synthetic data via compositing. 2019 IEEE/CVF Conference on computer vision and pattern recognition (CVPR) . IEEE; 2019. p. 461–70. https://ieeexplore.ieee.org/document/8953554/
220. Wang H, Wang Q, Yang F, Zhang W, Zuo W. Data augmentation for object detection via progressive and selective instance-switching. 2019; http://arxiv.org/abs/1906.00358
221. Zhou S, Xiao T, Yang Y, Feng D, He Q, He W. GeneGAN: Learning object transfiguration and object subspace from unpaired data. procedings of the british machine vision conference 2017. British Machine Vision Association; 2017. http://www.bmva.org/bmvc/2017/papers/paper111/index.html
222. Liu S, Zhang J, Chen Y, Liu Y, Qin Z, Wan T. Pixel Level Data Augmentation for Semantic Image segmentation using generative adversarial networks. ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) . IEEE; 2019. p. 1902–6. https://ieeexplore.ieee.org/document/8683590/
223. Nguyen V, Vicente TFY, Zhao M, Hoai M, Samaras D. Shadow detection with conditional generative adversarial networks. 2017 IEEE International Conference on Computer Vision (ICCV). IEEE; 2017. p. 4520–8. http://ieeexplore.ieee.org/document/8237745/
224. Zhu J, Samuel KGG, Masood SZ, Tappen MF. Learning to recognize shadows in monochromatic natural images. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition . IEEE; 2010. p. 223–30. http://ieeexplore.ieee.org/document/5540209/
225. Vicente TFY, Hou L, Yu C-P, Hoai M, Samaras D. Large-Scale Training of Shadow Detectors with Noisily-Annotated Shadow Examples. 2016. p. 816–32. https://doi.org/10.1007/978-3-319-46466-4_49
226. Rezaei M, Yang H, Meinel C. voxel-GAN: adversarial framework for learning imbalanced brain tumor segmentation. 2019. p. 321–33. https://doi.org/10.1007/978-3-030-11726-9_29
227. Rezaei M, Yang H, Meinel C. Recurrent generative adversarial network for learning imbalanced medical image semantic segmentation. Multimedia Tools Applications. 2020;79:15329–48. https://doi.org/10.1007/s11042-019-7305-1.
228. Rezaei M, Yang H, Meinel C. Conditional generative refinement adversarial networks for unbalanced medical image semantic segmentation. 2018; http://arxiv.org/abs/1810.03871
229. Gongal A, Amatya S, Karkee M, Zhang Q, Lewis K. Sensors and systems for fruit detection and localization: a review. Comput Electron Agric. 2015;116:8–19.
230. Sa I, Ge Z, Dayoub F, Upcroft B, Perez T, McCool C. DeepFruits: a fruit detection system using deep neural networks. Sensors . 2016;16:1222. http://www.mdpi.com/1424-8220/16/8/1222
231. Ehsani K, Mottaghi R, Farhadi A. SeGAN: Segmenting and Generating the Invisible. 2018 IEEE/CVF conference on computer vision and pattern recognition . IEEE; 2018. p. 6144–53. https://ieeexplore.ieee.org/document/8578741/
232. Dong J, Zhang L, Zhang H, Liu W. Occlusion-Aware GAN for Face De-Occlusion in the Wild. 2020 IEEE international conference on multimedia and expo (ICME) . IEEE; 2020. p. 1–6. https://ieeexplore.ieee.org/document/9102788/
233. Guan S. Breast cancer detection using synthetic mammograms from generative adversarial networks in convolutional neural networks. J Med Imag. 2019;6:1.
234. Donahue C, Lipton ZC, Balsubramani A, McAuley J. Semantically decomposing the latent spaces of generative adversarial networks. 2017;
235. Wang W, Hong W, Wang F, Yu J. GAN-Knowledge distillation for one-stage object detection. IEEE Access . 2020;8:60719–27. https://ieeexplore.ieee.org/document/9046859/
236. Paganini M, de Oliveira L, Nachman B. CaloGAN: Simulating 3D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks. Phys Rev D. 2018;97:014021. https://doi.org/10.1103/PhysRevD.97.014021.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

sensors

MDPI

Article

Intraclass Image Augmentation for Defect Detection Using Generative Adversarial Neural Networks

Vignesh Sampath [1,2,*], Iñaki Maurtua [1], Juan José Aguilar Martín [2], Ander Iriondo [1], Iker Lluvia [1] and Gotzone Aizpurua [1]

[1] Smart and Autonomous System Unit, Tekniker, Member of Basque Research & Technology Alliance, 20600 Eibar, Spain
[2] Department of Design and Manufacturing Engineering School of Engineering and Architecture, University of Zaragoza, 50009 Zaragoza, Spain
* Correspondence: vignesh.sampath@tekniker.es; Tel.: +34-943-206-744

Abstract: Surface defect identification based on computer vision algorithms often leads to inadequate generalization ability due to large intraclass variation. Diversity in lighting conditions, noise components, defect size, shape, and position make the problem challenging. To solve the problem, this paper develops a pixel-level image augmentation method that is based on image-to-image translation with generative adversarial neural networks (GANs) conditioned on fine-grained labels. The GAN model proposed in this work, referred to as Magna-Defect-GAN, is capable of taking control of the image generation process and producing image samples that are highly realistic in terms of variations. Firstly, the surface defect dataset based on the magnetic particle inspection (MPI) method is acquired in a controlled environment. Then, the Magna-Defect-GAN model is trained, and new synthetic image samples with large intraclass variations are generated. These synthetic image samples artificially inflate the training dataset size in terms of intraclass diversity. Finally, the enlarged dataset is used to train a defect identification model. Experimental results demonstrate that the Magna-Defect-GAN model can generate realistic and high-resolution surface defect images up to the resolution of 512 × 512 in a controlled manner. We also show that this augmentation method can boost accuracy and be easily adapted to any other surface defect identification models.

Keywords: class imbalance; convolutional neural network; defect detection; GAN; image augmentation; limited data; synthetic images; transfer learning

Citation: Sampath, V.; Maurtua, I.; Aguilar Martín, J.J.; Iriondo, A.; Lluvia, I.; Aizpurua, G. Intraclass Image Augmentation for Defect Detection Using Generative Adversarial Neural Networks. *Sensors* **2023**, *23*, 1861. https://doi.org/10.3390/s23041861

Academic Editor: Kuo-Liang Chung

Received: 27 December 2022
Revised: 19 January 2023
Accepted: 3 February 2023
Published: 7 February 2023

1. Introduction

Nondestructive testing (NDT) plays an essential role in industrial applications that can benefit directly from computer vision algorithms. They are widely employed in the manufacturing sector to detect defects, including scratches, flaws, pores, leaks, fractures, and cracks. In addition to impairing the aesthetic of the corresponding object, these defects on the object surface may also have a negative impact on quality control or even pose serious manufacturing safety risks [1]. The traditional procedures of performing NDT methods are more susceptible to the effects of human factors, which can result in different outcomes for the same test. Therefore, the incorporation of automation and computer vision techniques is desirable. Computer vision models excel at inspecting object details and defect detection tasks because of their speed, accuracy, and repeatability.

MPI is used to inspect a wide variety of manufactured products in different forms including castings, forgings, and weldments. The principle of magnetism is used in MPI to find defects in magnetic materials such as steel, iron, nickel, cobalt, etc. The first step in MPI is to magnetize the component parallel to its surface that is to be inspected. In the case of defects on or near the surface of the component, the defects create a leakage field.

Then, the iron particles in wet suspended form are applied onto the component. In the places of leakage fields, the particles are attracted and clustered. The defects can provide visible indications under ultraviolet light [2]. There are several factors that influence the effectiveness of MPI. The main factors include:

1. Part geometry: The shape and size of the part being inspected can affect the effectiveness of the inspection. For example, it may be more difficult to detect defects in thin or small parts compared to larger or thicker parts.
2. Material properties: The material properties of the part being inspected can also affect the effectiveness of the inspection. For example, nonferromagnetic materials may not be suitable for magnetic particle inspection.
3. Surface finish: a rough or uneven surface can make it more difficult to detect defects using magnetic particle inspection.
4. Magnetizing force: The strength of the magnetizing force applied during the inspection can affect the sensitivity of the inspection. A stronger magnetizing force may be more effective at detecting smaller defects.
5. Particle size and type: The size and type of magnetic particles used in the inspection can also affect the effectiveness of the inspection. Smaller particles may be more sensitive to defects but may be more difficult to see.
6. Light intensity: the intensity of the light used to illuminate the magnetic particles can affect the visibility of the particles and the ability to detect defects.

Collecting defective images with different combinations of these factors (intraclass variations) at a large scale is expensive due to the low possibility of defecting occurrence [3]. It leads to several difficulties in acquiring defect data with a high range of variability and hence poor generalization ability of a defect detection model. One of the most challenging tasks in developing a defect detection model is to improve its generalization ability.

To address the issue of the insufficient generalization ability of a defect detection algorithm caused by the limited data problem, in this paper, an improved conditional mask-to-image translation GAN-based data augmentation method is proposed. GAN-based intraclass augmentation is used to artificially increase the size and diversity of the dataset, which can improve the performance of the model. Intraclass image augmentation refers to the process of applying various types of data augmentation techniques to images within the same class in order to increase the variability of the training dataset. This can help to improve the generalization performance of a machine learning model by providing it with more examples of the same class with different variations. Unlike previous work, for our generator, we use a U-Net-based network, we couple the mask embedding vector with the latent noise vector and the discrete fine-grained guide labels (Figure 1), and for our discriminator, we use a PatchGAN classifier [4]. Coupling embedding vectors with fine-grained guide labels and latent noise vectors leads to conditioning the data generation process in a controlled manner. With the mask, our Magna-Defect-GAN model can generate diverse defect images, such as by changing defect size, shape, location, position, etc. We also allow more diversity, such as by changing the background, thickness, and brightness of the defects.

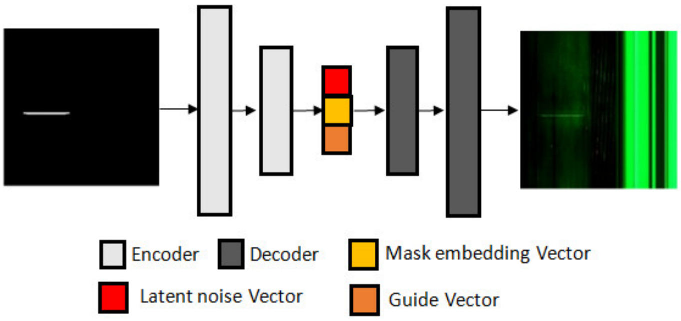


Figure 1. Structure of the proposed Magna-Defect-GAN.

To verify the effectiveness of the proposed network, we acquired a defect dataset using a line scan camera from an MPI apparatus, located at Erreka Fastening solutions, in a controlled manner. The Magna-Defect-GAN model is trained for augmenting data samples. The defect detection accuracies of a convolutional neural network (CNN) model before and after data augmentation are compared. The experimental results show that the Magna-Defect-GAN is more robust in generating controllable realistic and high-resolution defect images than other existing GAN models.

The key contributions of this paper are as follows: (1) we present a surface defect dataset acquired from a line scan camera that is essential for defect detection in cylindrical objects; (2) we propose combinations of the mask, latent vector, and guide vectors (background, thickness, and brightness vectors) as a means of controlling the conditions of the synthesized images; (3) we present a novel conditional mask-to-image GAN that utilizes the interpretable guide vectors, and the Magna-Defect-GAN is employed to augment training data at pixel level; and (4) we validate the effectiveness of the proposed pixel-level data augmentation by training the CNN model with various training schemes using synthetic and original data. The defect detection model trained by the combination of original and augmented data alleviates the problem of overfitting and overcomes all biases present in a limited dataset. Several forms of biases in the limited dataset such as background, lighting, defect position, shape, size, etc., are drastically lessened with the help of GAN-generated synthetic images.

The remainder of the paper is structured as follows: In the next section, existing work on the classical and GAN-based data augmentation methods are described in detail. In Section 3, an experimental platform for defect image acquisition is established in the laboratory. The GAN-based data augmentation models are built to generate synthetic images for enhancing intraclass diversity in a limited data regime, and some comparative experiments are performed in Section 4 to test the efficacy of the GAN-based data augmentation. The effectiveness of the Magna-Defect-GAN-based data augmentation is examined in Section 5, and the findings are reported. In Section VI, conclusions are drawn.

2. Related Work

There are thousands of parameters in even a lightweight CNN model that need to be trained. When employing deep CNN models with numerous layers or when working with a small number of training images, there is a risk of overfitting. The most widely used method to reduce overfitting is data augmentation, which artificially inflates the dataset size. By exposing the defect detection model to a wider range of variations in the data, data augmentation can help the model learn to generalize better and reduce overfitting. For example, if the model is trained on images of defects that all have the same orientation, it may not be able to recognize defects that have a different direction. However, if the model is also trained on images of defects that are rotated or flipped, it may be able to recognize defects in a wider range of orientations. This encompasses classic augmentation techniques such as affine and color transformations [5]. Even though classic augmentation techniques serve as an implicit regularization, they are limited in augmentation diversity. Several methods have been introduced to increase the effectiveness of data augmentation. Zhong et al. [6] proposed a random erasing augmentation technique to make sure that the CNN pays attention to the entire image rather than its subset. Random erasing works by discarding a random n × m rectangle patch in an image and masking it with random values. The disadvantage of using random erasing in defect identification applications is that it is not always a label-preserving augmentation. Moreno-Barea et al. [7] injected random noise into images that can help the model to learn more robust features. Combining different augmentation techniques can result in an enormously expanded dataset size. However, it is not ensured to be beneficial. Cubuk et al. [8] proposed an autoaugment policy based on the reinforcement learning algorithm to search for an optimal combination of augmentation techniques. Perez and Wang [9] developed an augmentation method based on the neural style transfer algorithm, which employs neural nets to transfer style and classify the image.

Recently, GAN-based data augmentation gained momentum in the field of computer vision [10]. GAN models can be used to create synthetic images such that they retain similar characteristics to the original training data. One way to use GAN-generated synthetic images in defect detection is to train a defect detection model on a combination of real and synthetic images. Using GANs to generate synthetic images can be useful for augmenting the training dataset for a defect detection model. By adding synthetic images to the training dataset, it is possible to increase the diversity of the dataset and improve the generalization ability of the model. The objective of GAN-based augmentation is to generate synthetic images to increase diversity and the amount of the original dataset. Several modifications of the original GAN [11] have been proposed to improve the performance and stability of GAN training including DCGAN [12], Pro-GAN [13], LAPGAN [14], GRAN [15], D2GAN [16], SinGAN [17], and MADGAN [18]. However, these GAN models have a limitation in that the generated synthetic images cannot be controlled. Conditional variants of GANs such as cGAN [4], ACGAN [19], VACGAN [20], info-GAN [21], and SCGAN [22] have been proposed to overcome the limitation. GAN models have proved to excel at several other computer vision tasks including image super-resolution [23], image denoising [24], and text to image synthesis [25]. In the area of manufacturing, image-to-image translation is the most pertinent use of GAN.

In 2016, P. Isola et al. [26] developed a conditional variant of GAN called Pix2Pix (pixel to pixel) GAN as a general solution to image-to-image translation tasks. In this case, the generator takes an image from one domain and is tasked to convert it into an image in another domain by minimizing reconstruction as well as the adversarial loss. Several variants of Pix2Pix GAN have been proposed to enhance the quality of the translated images. To reduce the blurriness of the translated images, Wang et al. [27] replaced the reconstruction loss with a feature-matching loss. Unsupervised variants of image-to-image translation GANs such as Disco-GANs [28] and Cycle-GANs [29] were proposed.

The ability to generate industrial images with defects in a controlled manner is highly desirable by the industry 4.0 machine learning community. In particular, given that the pixels corresponding to the background are far more numerous than the pixels of defects, a mask-guided stochastic generator for augmentation of industrial data could potentially yield improvements in detection and classification algorithms. To actually realize this gain, our model employs numerous strategies to produce controllable, realistic, and high-resolution synthetic industrial images as well as to enhance the quality of images and stabilize the training process. We propose a new GAN architecture that maps a given mask input to the sample space more efficiently by coupling the mask embedding vector, conditional label vector, and latent noise vector. Compared with the traditional image-to-image translation GANs described above, the samples generated by the GAN model are more diverse.

In the context of industrial images with surface defects, recently, several GAN-based methods have been proposed. One of these methods is Mask2Defect GAN [30], which proposes a GAN model to generate a large volume of surface defect images with different features and shapes. The algorithm separates the generation process into two steps: the first step uses the mask-to-defect construction network (M2DCNet) to render the defect details according to the binary mask, and the second step uses the fake-to-real domain transformation GAN (F2RDT-GAN) to add background textures and transform the synthesized defects from the rendered domain to the real defect domain.

Another method is the surface defect generative adversarial network (SDGAN) [31], which utilizes D2 adversarial loss and cycle-consistency loss to generate high-quality and diverse defect datasets using a small number of defect images. SDGAN incorporates two diversity control discriminators and a cycle-consistency loss to generate defect images in a more efficient and effective way. The introduction of the diversity control discriminators allows one to control the diversity of the generated images, while the cycle-consistency loss helps to ensure that the generated images are consistent with the input images. Defect-GAN [32] is proposed to mimic defacement and restoration processes to generate realistic and diverse defect samples. Defect-GAN employs adaptive noise insertion to capture the stochastic variations within defects. Kaiqiong et al. [33] proposed an entirely multiscale GAN with a transformer to capture the intrinsic patterns of qualified samples of IC metal package images at multiple scales. The proposed GAN model is designed to improve the quality of the generated images by capturing the patterns of the images at multiple scales. The multiscale architecture is achieved by using a set of convolutional layers with different dilation rates to extract features at multiple scales. One of the key contributions of the proposed GAN model is the use of a Swin Transformer decoder, which is designed to strengthen the modeling ability of the GAN. The Swin Transformer decoder is a modified version of the transformer decoder that is designed to handle images with high resolution. Shuanlong et al. [34] proposed a novel approach for generating synthetic defects in metal surfaces that is based on the concept of image inpainting. The proposed method regards defect generation as a form of image inpainting, where defects are generated in nondefect images in regions specified by defect masks.

Our proposed method, Magna-Defect-GAN, is better than the abovementioned methods in several ways. One key advantage is that our method maps a given mask input to the sample space more efficiently by coupling the mask embedding vector, guide vector, and latent noise vector. This allows for the generation of various images that are realistic and captured under different thicknesses, brightness, and types of fasteners. In contrast, previous methods require significant manual effort to create masks with all possible combinations of defect parameters, such as defects with different thicknesses, brightness, etc. Our method involves learning a disentangled representation to separate the different elements of fastener images, such as the parameters of defects and the background. This makes it useful for creating a large number of different defects, with varying thicknesses, brightness, and backgrounds, by simply adjusting the guide vectors. Another advantage of our method is that we propose to utilize the latent noise vector in addition to the guide vector to improve the diversity of generated images. This allows for the generation of images with different variations with respect to defects of different thicknesses, brightness, and types of fasteners.

3. Materials and Methods

We require a method for pixel-level data augmentation to increase the intraclass variety of the training dataset and strengthen the robustness of defect detection models with limited data. The mask-to-image translation model, which creates images that resemble those obtained in a different setting (lighting, texture, etc.), is a key part of our suggested methodology. We first suggest the guide vector in the GAN model, which contains values that are understandable by humans and is hence controllable and explicable. Using input from the guide vector and mask, we then synthesize images with significant intraclass variance. Finally, we use synthetic data to supplement the training dataset and develop a reliable defect detection model. In this section, we first present a novel dataset of fastener defects. Next, we present the Magna-Defect-GAN model and the guiding vector.

3.1. Line Scan Defect Dataset

We collected a new fastener defect dataset using a DALSA Linea 2k 7.04 um 2048 × 2–26 kHz-Color line scan camera since the frame cameras have their limitations in resolution and high-speed imaging applications. Unlike a frame camera, which exposes the entire area of the sensor and gives an entire image, a line scan camera exposes just a single line of pixels. These single lines of pixels are stitched together to form a complete frame. As opposed to frame cameras, line scan cameras need special optic systems. We used 12 mm fixed focal length lenses and 600 mm field of view (FOV). In our study, the linear array camera was used to capture MPI images because it allows for a larger field of view and a higher resolution compared to an area array camera. Additionally, linear array cameras have higher sensitivity and signal-to-noise ratio, which is important for detecting small defects. Furthermore, linear array cameras can also provide a higher frame rate, which is useful in fast-moving production lines. Additionally, linear array cameras are more cost effective and have a smaller form factor as compared to area array cameras. Figure 2 compares the methods used by frame and line scan cameras for image capturing.

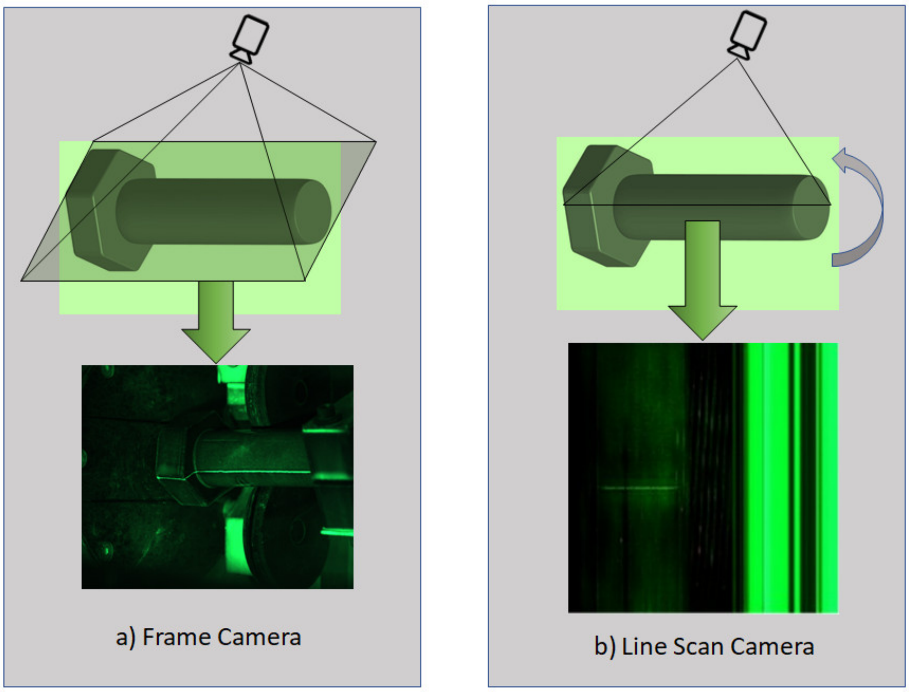


Figure 2. Comparison of (**a**) Frame and (**b**) Line Scan Cameras in terms of how they capture images.

The line scan camera must capture images at precisely the same rate that the fastener is being rotated—a too-fast scan rate, the image gets distorted; too slow, and some of the original slices are missed. We used an encoder to synchronize the rotational movement of the fastener as well as the triggering of the line scan camera to ensure that no unnecessary stretching or shrinking happens on the resultant image (Figure 3).

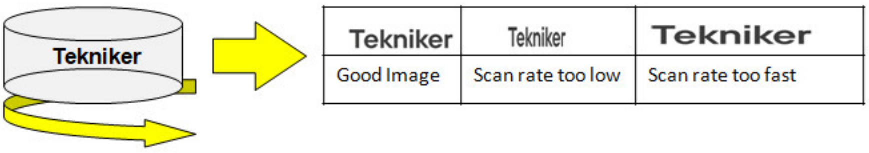


Figure 3. An illustration of scan rate not matching with moving speed.

The collected defect dataset consists of 1050 RGB images. The dataset was collected from a magnetic particle inspection apparatus located at Erreka Fastening solutions. The images were collected in different environments (background, lighting, thickness, and brightness). Ground truth masks and guide labels were labeled by experienced quality engineers. Specifically, three components of defect images were annotated so that not only defect shape, location, and numbers but also the thickness, brightness, and background of the defects can be controlled. At the right end of the line scan image, we can see a sizable dark green region that represents the background, which was stationary.

3.2. Fine-Grained Guide Label

In general, the thickness and brightness of the defect in an image depend on (1) particle concentration, (2) lighting, and (3) material type of the fasteners (background). Since our goal is, given a mask label, generating various images that are realistic and captured under different thicknesses, brightness, and types of fasteners, we propose to utilize thickness, brightness, and background (guide vector) to guide an image generation process. Therefore, given a mask label and a guide vector (e $\in R^3$), a corresponding image is generated.

3.3. Preliminaries

GANs were introduced to make a generative model by having two models (generative model G and discriminative model D) compete with each other. A generative model G turns noise into an imitation of the data to try to trick the discriminator, and a discriminative model D tries to identify real data from fakes created by the generator. Both G and D could be a convolutional neural network. To create a synthetic image x, the generator takes a noise vector from a prior noise distribution $p_z(z)$ and runs it through a differentiable function: $G(z) \rightarrow x$. The learning procedure of GAN is to train a discriminator D and a generator G in parallel. At each iteration, backpropagation is applied to adjust generator model parameters G to minimize $log(1 - D(G(z)))$ and adjust discriminator model parameters D to minimize $logD(x)$. Therefore, the loss function of GANs can be written as:

$$L_{GAN}(G,D) = E_{x\sim P_r(x)}[logD(x)] + E_{z\sim P_z(z)}\left[log\left(1 - D\left(G(z)\right)\right)\right] \quad (1)$$

To have a control on the kind of image being generated, in cGANs, both the generator G and discriminator D are conditioned on additional information such as class labels y. In pix2pix GAN, both the generator G and discriminator D are conditioned on an input image to generate a corresponding output image. In this case, adversarial loss can be formulated as:

$$\mathrm{L_{cGAN}(G,D) = E_{x\sim P_r(x)}[logD(x|y)] + E_{z\sim P_z(z)}\left[log\left(1 - D\left(G(z|y)\right)\right)\right]} \quad (2)$$

3.4. Proposed Architecture

3.4.1. Generator Architecture

The main challenge to training an image-to-image translation GAN without latent noise vector z is that the model would produce deterministic outputs. Wang et al. [35] used latent noise vector z as an input to the generator model in addition to the mask label. To improve the overall feature projection efficiency, our Magna-Defect-GAN model first performs mask embedding in the generator before the latent projection layer. Figure 4 represents the overall architecture of our Magna-Defect-GAN. The generator of our proposed GAN is based on a U-Net style design that can be decomposed into two branches, namely, the mask projection and the latent projection branch. First, the mask projection branch encodes the input mask into the mask embedding (32-dimensional vector). This mask projection branch consists of 7 convolution layers each with a stride of 2, each followed by a leaky rectified linear unit (Leaky ReLU). After that, we concatenate latent noise vector z (132-dimensional vector) and the guide label vector e with the mask embedding to improve sample space mapping and provide diverse texture detail in the synthetic images. Finally, the latent projection branch whose inputs are the latent noise vector z, which, in combination with the mask embedding and guide label, generates an output image. An image mask input provides the intended defect shape, position, and quantity, and a guide label provides the necessary defect background and thickness to generate a defect image.

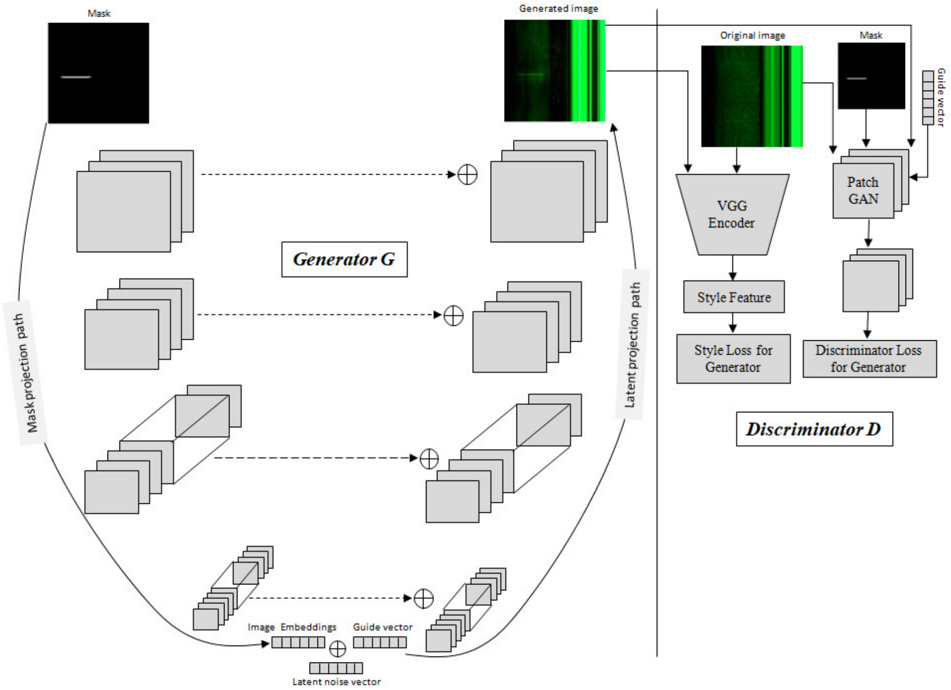


Figure 4. The Magna-Defect-GAN model comprises a U-net style generator network, a discriminator network, and a pretrained VGG feature extractor. The mask projection path in the Generator network G is tasked with mapping input masks into image embeddings through encoder blocks. The latent projection path is used to translate the combination of the image embeddings, guide vectors, and latent noise vectors into the output image. The discriminator network D is trained to distinguish real and generated images. The pretrained VGG network is used to extract features to calculate the style loss.

3.4.2. Discriminator Architecture

We employed a modified Patch-GAN architecture [26] for the discriminator. As opposed to classifying the output and target image as being real or fake, the Patch-GAN discriminator is designed to use a convolutional network that divides the input images into NxN patches of the image and outputs a matrix of values. Consequently, the discriminator gives feedback on each region or patch of the image, which enables high frequency and encourages detailed outputs by the generator. To avoid the common tiling artifacts with smaller patch sizes, 70 x 70 patches are typically used. However, we found that smaller patches in combination with style transfer losses yield sharper images while eliminating tiling artifacts. Consequently, we use a patch size of 16 x 16.

3.4.3. Loss Function

One of the key elements of GANs is the loss function that is used to train the GAN models. Different types of loss functions can be used depending on the specific GAN architecture and the desired properties of the generated samples.

The pix2pix loss function is commonly used in image-to-image translation tasks, such as converting a sketch to an image or converting a daytime image to a nighttime image. The main goal of the pix2pix loss function is to generate an output image that is as similar as possible to the target image. To achieve this, the pix2pix loss function uses two main components: the L1 loss and the adversarial loss.

The L1 loss, also known as the mean absolute error (MAE), compares the pixel-wise differences between the generated image and the target image. It calculates the absolute difference between each pixel in the generated image and the corresponding pixel in the target image and then takes the average of all these differences. The L1 loss is a popular choice for image-to-image translation tasks because it is less sensitive to outliers than the L2 loss (mean squared error) and has been shown to produce sharper images. The L1 loss is calculated as:

$$L_1 = \mathrm{E}_{\mathrm{x,y,z}} \| x - \mathrm{G(z, y)} \|_1 \quad (3)$$

where x is the target image and $\mathrm{G(z, y)}$ is the generated image. The L1 loss is a good choice for image-to-image translation tasks because it is able to capture the structural information of the image.

The pix2pix GAN loss is used in combination with the L1 loss to ensure that the generated image is not only similar to the target image but also visually realistic. The total loss function for the pix2pix method is a combination of the L1 loss and the adversarial loss. The total loss function is defined as:

$$L_{pix2pix} = \alpha * L_1 \text{ loss} + (1 - \alpha) * L_{adv} \quad (4)$$

where α is a hyperparameter that controls the balance between the L1 loss and the adversarial loss.

The CycleGAN loss function, on the other hand, uses a combination of the cycle-consistency loss and the adversarial loss. The cycle-consistency loss, also known as the cycle-consistency constraint, ensures that the generated image can be transformed back to the original image. The cycle-consistency loss is calculated as the difference between the original image and the transformed image. The cycle-consistency loss is defined as:

$$L_{Cycle} = \mathrm{E_x} \left\| \mathrm{x} - \mathrm{G}\left(\mathrm{F(x)}\right) \right\|_1 + \mathrm{E_y} \left\| \mathrm{y} - \mathrm{F}\left(\mathrm{G(y)}\right) \right\|_1 \quad (5)$$

where x is the input image, y is the target image, G is the generator for the input image, and F is the generator for the target image. The cycle-consistency loss ensures that the generated image preserves the characteristics of the input image.

Adversarial loss is used to ensure that the generated image looks like a real image and not a fake one. The total loss function for the CycleGAN method is a combination of the cycle-consistency loss and the adversarial loss. The total loss function is defined as:

$$L_{CycleGAN} = \lambda * L_{Cycle} + L_{adv} \quad (6)$$

where λ is a hyperparameter that controls the balance between the cycle-consistency loss and the adversarial loss.

We used a combination of three different losses in our proposed GAN model, i.e., adversarial loss, style loss, and reconstruction loss.

- Adversarial loss is used to ensure that the generated images are realistic and not easily distinguishable from the original images. This is done by training the generator to fool the discriminator, which is trained to distinguish between real and fake images.
- Style loss is used to ensure that the generated images have the same style as the original images. This is done by comparing the feature maps of the generated images to the feature maps of the original images.
- Reconstruction loss is used to ensure that the generated images are similar to the original images. This is done by comparing the generated images to the original images.

By combining these three types of losses, the GAN is able to generate high-quality images that have the same style and structure as the original images while also being realistic and difficult to distinguish from real images. This results in more realistic and visually appealing generated images.

The adversarial loss in the Magna-Defect-GAN is essential to ensure and guide the generator to generate synthetic images that look real and are able to fool the discriminator. The generator establishes a relationship between the source image mask y, guide vector e, and the random noise image z to the target image x, i.e., $y, z, e \rightarrow x$. The discriminator makes a distinction between original and fake $x| y, e$. The adversarial loss can be represented as:

$$L_{adv} = \mathrm{E}_{y,e,x}[\log D(y|e, x)] + \mathrm{E}_{z,y,e}\left[\log\left(1 - D\big(y, e, G(z, y|e)\big)\right)\right] \quad (7)$$

The conditional adversarial loss attempts to make the generated image look real. However, line scan industrial images, different from natural images that have higher diversity in texture, shape, and color, require intricate precision of internal structure. As a result, an additional constraint is necessary to ensure that the generated images are similar to the original. Therefore, we add a pixel-wise reconstruction loss to the adversarial loss that measures the pixel-wise distance between the generated images and the original image that is available at training time. Comparing the performance of utilizing L1 and L2 norms to ascertain the reconstruction loss, we observe that the L2 norm appears to perform better for our task. Our reconstruction loss is defined as below:

$$L_{rec} = \mathrm{E}_{y,e,x}\|x - G(z, y, e)\|_2^2 \quad (8)$$

The abovementioned reconstruction loss ensures structural consistency between the generated and original image. To make the training process more stable and minimize the total textural deviation between the generated and original image, style loss could be used as auxiliary regularization. We use the style loss to further improve the similarity between the generated and the original image in terms of intricate visual appearance such as texture and color. We employ style loss at multiple levels between the original and the generated image with a pretrained VGG model in a way similar to an earlier work [36]. The style information is measured as the degree of correlation between feature maps in a given layer. The style loss is then calculated by matching the mean and standard deviation between the feature maps computed by the generated image and the original image. We calculate the pair-wise correlation between all the feature vectors in the filters for each style layer in order to preserve similarity between the style image and the generated image based on the spatial information. These feature correlations are given by Gram matrix $G^l \in \mathbb{R}^{N_l \times N_l}$, the inner product between the vectorized feature maps in layer l:

$$G_{ij}^{l} = \sum_{k=1}^{M_l} F_{ik}^{l} F_{jk}^{l} \tag{9}$$

Assume that there are A_l filters in total, each with a feature map of size B_l, and that we have G_{ij}^l, H_{ij}^l gram matrices for the style image and the generated image. Thus, we can calculate the overall style loss as follows:

$$L_{style} = \sum_{l} w_l . \frac{1}{4A_l^2 B_l^2} \sum_{i,j} (G_{i,j}^l - H_{i,j}^l)^2 \tag{10}$$

where the weight given to layer l is w_l. Each w_l, in this case, contains the value $\frac{1}{\text{Total number of style layers}}$, i. e. , $\frac{1}{5}$.

Total generator loss is calculated as a weighted combination of reconstruction loss, style loss, and adversarial loss. Our final generator loss is formulated as:

$$L_{total} = \lambda_1 L_{adv} + \lambda_2 L_{rec} + \lambda_3 L_{style} \tag{11}$$

where λ_1, λ_2, and λ_3 regulate the relative weight of different loss terms. Although reconstruction should take precedence during the optimization phase, the adversarial loss plays a significant role in encouraging local realism of the synthesized output in our mask-to-image translation problem. We conducted our studies with the following settings: $\lambda_1 = 10$; $\lambda_2 = \lambda_3 = 0.1$.

4. Experiments

4.1. Training Details

All the experiments were run on Google-cloud infrastructure using a single Nvidia 12 GB Titan X GPU. We randomly divided our data into 80% training set and 20% test set. We trained the GAN model for an average of 200 epochs, and the generation of synthetic images took about 0.0265 ms per image. We kept the learning rate constant for the first 100 epochs, after which it exponentially declined to zero during the following epochs. All weights in the model were initialized from a Gaussian distribution with a mean of 0 and standard deviation of 0.01.

4.2. Evaluation Metrics

When evaluating the GAN model, it is important to consider not only fidelity, which quantifies the quality of the generated images, but also the diversity. In other words, fidelity measures image quality and diversity measures the variety of the generated images. A good generator must produce a good variety of images. For instance, of all images in the training dataset, the generator should model all types of defects, including defects in different positions, shapes, and sizes. We use the Inception score (IS) and Fréchet Inception Distance score (FID) to evaluate the performance of our Magna-Defect-GAN model. IS attempts to measure both the fidelity and diversity of the generated images. The Inception model pretrained on a fine-grained defect dataset is the backbone of the IS. Given image x and label y, for a high fidelity and diverse input, the posterior probability of a label $p(y|x)$ computed using the Inception model should have a low entropy, and the marginal class distribution $\int P(y|x = G(z))\, dz$ should have a high entropy. Mathematically, IS can be represented as:

$$IS(G) = \exp\left(E_{x\sim p_a} D_{KL}(p(y|x) \| p(y))\right) \tag{12}$$

FID is the frequently used method to measure the feature distance between real and generated images. Images from the training dataset and images generated by the generator are transformed into a feature space by FID using the output of the last hidden layer in Inception Net. Multivariate normal Fréchet Distance can be calculated as:

$$\mathrm{FID}(x, g) = \left\|\mu_x - \mu_g\right\|_2^2 + \mathrm{Tr}(\Sigma_x + \Sigma_g - 2(\Sigma_x \Sigma_g)^{\frac{1}{2}} \tag{13}$$

where (μ_x, μ_g) and (Σ_x, Σ_g) represent the mean and covariance of the true and generated features, respectively.

4.3. T-sne Visualization

To provide more powerful evidence that the generated synthetic images, indeed, contribute to the shape of the data manifold, we use a t-distributed stochastic neighbor embedding (t-SNE) algorithm to visualize the distribution of training and generated image samples by reducing high-dimensional data to a 2D plane. First, the t-SNE algorithm converts the similarities between data points to joint probabilities and then aims to minimize the KL divergence between the joint probability of the low-dimensional embedding and the high-dimensional data.

4.4. Evaluation of Defect Classification

In order to assess the performance benefits obtained by utilizing GAN-based synthetic images, we benchmark our approach with well-known existing CNN models, ResNet [37] and EfficientNet [38], which are frequently in a number of defect classification applications. The defect classification performances are compared for the following training approaches. (a) Model trained only with the original dataset, (b) model trained with the augmented dataset (traditional augmentation methods such as rotation, vertical/horizontal flips, zoom, shear, and channel shifts), (c) model pretrained with synthetic dataset and fine-tuned with the original dataset, (d) model pretrained with the ImageNet dataset and fine-tuned with the augmented dataset, (e) model pretrained with the ImageNet dataset and fine-tuned with the synthetic dataset, and (f) model pretrained with the ImageNet dataset and fine-tuned with a mix of augmented and synthesized datasets. The metrics employed for the performance comparison are precision, recall, F1 score, and binary accuracy.

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{14}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{15}$$

$$\mathrm{F1\ Score} = \frac{2.\,(\mathrm{precision}.\,\mathrm{Recall})}{(\mathrm{precision} + \mathrm{Recall})} \tag{16}$$

$$\mathrm{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}} \tag{17}$$

where TP, TN, FP, and FN denote true positive (correctly identified defects), true negative (correctly identified nondefect images), false positive (images erroneously classified as defect), and false negative (images erroneously classified as nondefect), respectively.

5. Results and Discussion

The Magna-Defect-GAN model was trained using 780 nondefective images and 270 defective images. After training the model, diverse and realistic synthetic images were generated by altering the input masks and guide vector. Because of the small training sample size, the augmented images by rotation, vertical/horizontal flips, zoom, and shear were additionally incorporated. We present the synthesized images given a mask and guiding vectors in Figure 5 to illustrate the controllability and explainability of the Magna-Defect-GAN model.

Figure 5 demonstrates how the proposed method can generate different fine-grained images from a given mask by altering the guide vectors. The first column in Figure 5 depicts the input masks, while the second column depicts the GAN-generated images given guide vectors. As illustrated in Figure 6, the generated images of the Magna-Defect-GAN are almost identical to the training dataset with high-fidelity image-specific details well preserved (e.g., defects, illumination, and background), while Pix2Pix and CycleGAN show poor perceptual quality.

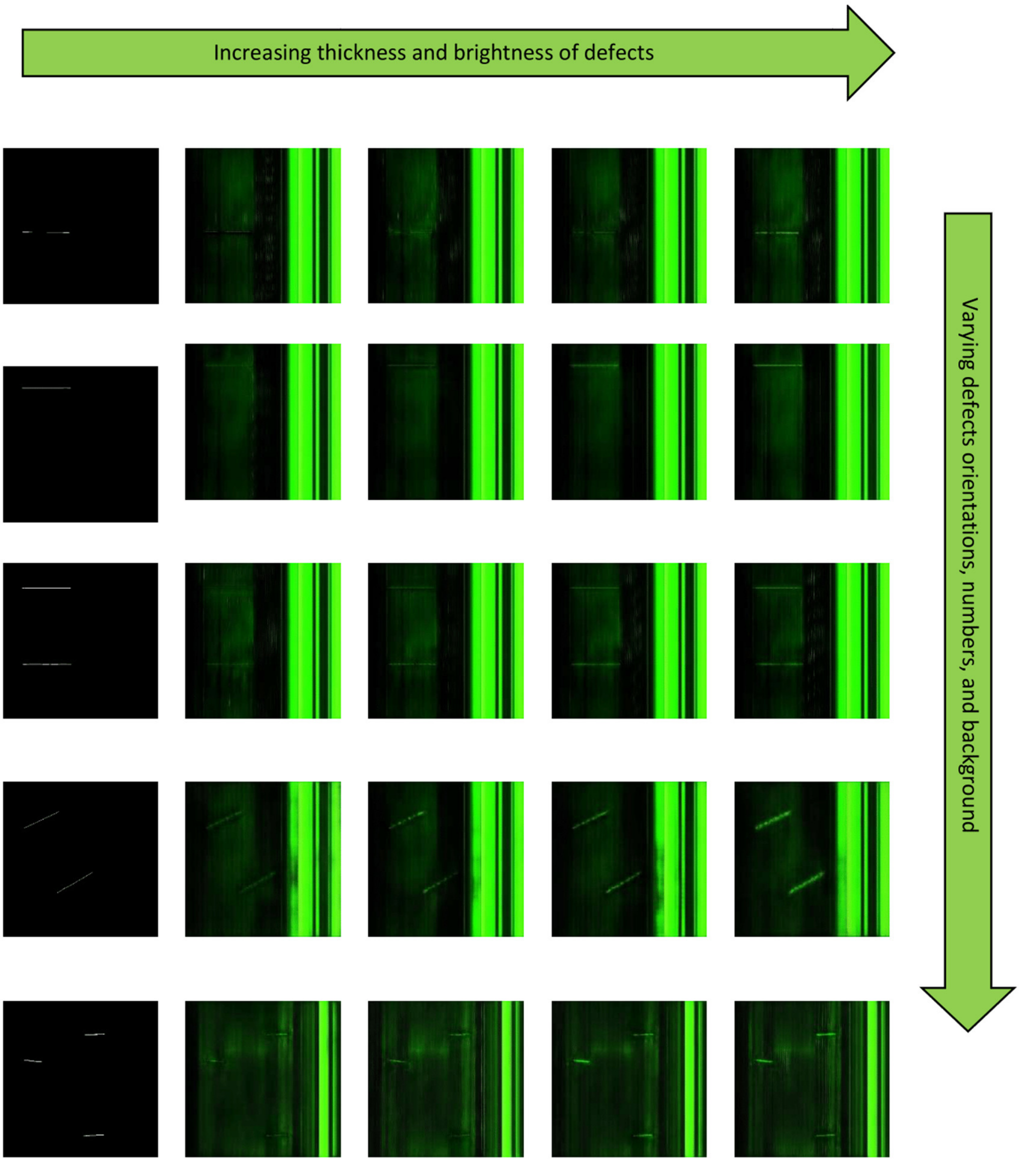

Increasing thickness and brightness of defects
Varying defects orientations, numbers, and background

Figure 5. Manipulating Guide vectors on defect dataset. The different rows correspond to different Guide vector settings of fixed mask input and noise.

It is worth noting that other GAN models would potentially generate a number of completely blank pixels on a defect region (Figure 6) in an effort to achieve greater diversity. This is harmful for training a defect detection model since the model would struggle to learn from the noisy images. However, our Magna-Defect-GAN model maintains both structural consistency and fine-grained background details, which is beneficial for a defect detection model to learn from different appearances of defects under different levels of ambient lighting.

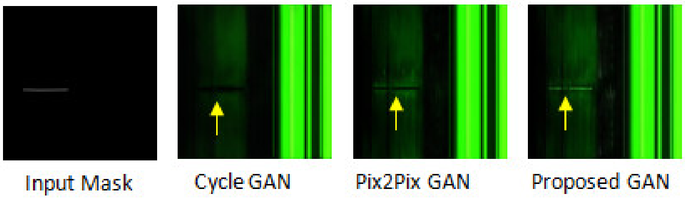


Figure 6. Comparison between the Magna-Defect-GAN and different image translation approaches.

Figure 7 shows the data distribution of original and synthetic images after dimensionality reduction by the t-SNE algorithm. As this figure illustrates, the proposed algorithm can generate data that not only overlap with the true data distribution but is also extremely close to the underlying distribution of the training data. All the generated data that mimic the real data distribution have not appeared in the training dataset. The proposed algorithm can provide an efficient way to close gaps in the discrete manifold distribution and supplement sources of variance that are difficult to augment in conventional methods.

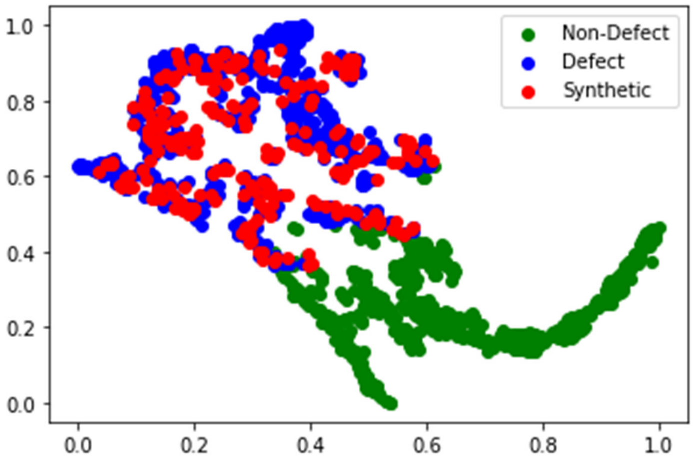


Figure 7. t-SNE visualization showing the effect of the Magna-Defect-GAN-based augmentation.

Our proposed method achieves better IS and FID scores (Table 1) compared to the rest of the methods. There could be a couple of reasons for the better scores. First, coupling the mask embedding vector, conditional label vector, and latent noise vector results in better sample space mapping, which leads to diverse textures of fine-grained details in the synthesized images. Moreover, the use of style loss in the Magna-Defect-GAN improves the fidelity of the generated image in terms of background attributes such as texture and color.

Table 1. Performance of the Magna-Defect-GAN Model.

Model	Inception Score ↑	FID ↓
Cycle [29]	2.88 ± 0.25	91.56
Pix2Pix [26]	3.08 ± 0.31	65.09
Magna-Defect-GAN	**3.88 ± 0.36**	**50.03**

We investigated the use of GAN-generated images to supplement the training dataset for the task of defect classification in the presence of a small number of training examples. Our Magna-Defect-GAN model was employed for generating photorealistic and high-resolution synthetic industrial images.

Using our proposed method, it is possible to generate images with large intraclass variations such as defects with different thicknesses, brightness, types of fasteners, etc. Furthermore, it is feasible to create realistic synthetic images that are similar to the training data by using simulated masks with different forms, locations, and orientations. Table 2 summarizes the six sets of experiments that were carried out to compare different training schemes. The defect classification accuracy and F1 score by the Resnet model trained with the original data from scratch were 80.8% and 0.801, respectively. The efficientnet-B7 model accuracy and F1 score were 89.8% and 0.893, respectively. When the training dataset was augmented by applying random but realistic lighter data augmentation schemes such as vertical/horizontal flips, zoom, and rotations, the model accuracy and F1 score of the Resnet model were enhanced to 83.5% and 0.868, respectively. Additionally, by incorporating the efficientnet-B7 model, the accuracy and F1 score further improved to 91.8% and 0.918, respectively. Because the training data size is too small and does not contain enough data samples to properly represent the greatest possible intraclass diversity, using the original data samples alone resulted in low and unstable training and validation accuracy. The training loss stabilized when the lighter data augmentation scheme was applied; however, the validation loss remained unstable. The accuracy and F1 score of the Resnet model were 87.5% and 0.88, respectively, using synthetic images for pretraining and the original dataset for fine-tuning. Additionally, the efficientnet-B7 model achieved an accuracy of 92.5% and an F1 score of 0.925 when using the same pretraining and fine-tuning methods. Compared with the results of the training model with the augmented dataset, the test accuracy and F1 score were comparable. The accuracy and AUC of the Resnet model increased from 83.5% to 89.8% and 0.868 to 0.919, respectively, when the ImageNet pretrained model was used and fine-tuned with the augmented dataset. Additionally, the efficientnet-B7 model showed an improvement in accuracy, increasing from 91.8% to 94.7%, and an increase in F1 score from 0.918 to 0.946. This is because the ImageNet dataset spans 21,000 object classes, the model is encouraged to learn more features than it requires when it is pretrained with fine-grain labels, and these excess features aid in network generalization, i.e., improving the testing accuracy. In other words, fine-grain labels help learn more features than coarse-grained labels (defect vs. nondefect).

When the synthetic images were used for fine-tuning, the accuracy and the F1 score of the Resnet model were 92.7% and 0.939, respectively, using the ImageNet pretrained model. The efficientnet-B7 model achieved an accuracy of 96.4% and an F1 score of 0.966 when fine-tuned with the synthetic images. The performance of defect detection models using synthetic images is on par with the outcome of regular data augmentation. It is observed that when the model was fine-tuned with a mix of synthetic and augmented data, the accuracy and the F1 score of Resnet were 93.5% and 0.947, respectively, which is clearly a better performance rate. Similarly, when using the efficientnet-B7 model, the accuracy and F1 score were even higher at 97.2% and 0.973, respectively, further demonstrating the effectiveness of using a mix of synthetic and augmented data in fine-tuning models. Both traditional augmentations and GAN-generated synthetic images are extremely beneficial to prevent overfitting when training a defect detection model with a limited dataset—the former extrapolates the training data distribution and the latter generates more diverse data by interpolating between the discrete data points in the manifold.

Table 2. Data Augmentation Experiments on Surface Defect Dataset.

Training Scheme	Model	Recall	Precision	F1-Score	Accuracy
a	ResNet [37]	0.684	0.969	0.801	0.808
	EfficientNet-B7 [38]	0.847	0.945	0.893	0.898
b	ResNet [37]	0.873	0.865	0.868	0.835
	EfficientNet-B7 [38]	0.895	0.943	0.918	0.918
c	ResNet [37]	0.863	0.907	0.88	0.875
	EfficientNet-B7 [38]	0.909	0.942	0.925	0.925

d	ResNet [37]	0.942	0.898	0.919	0.898
	EfficientNet-B7 [38]	0.958	0.935	0.946	0.947
e	ResNet [37]	0.945	0.934	0.939	0.927
	EfficientNet-B7 [38]	0.961	0.972	0.966	0.964
f	**ResNet** [37]	**0.955**	**0.94**	**0.947**	**0.935**
	EfficientNet-B7 [38]	**0.969**	**0.977**	**0.973**	**0.972**

6. Conclusions

In this work, we addressed the problem of defect detection with limited data. For that purpose, we proposed a GAN-based mask-to-image translation model for data augmentation. The main conclusions are listed below.

(1) By combining the mask embedding vector with the latent noise vector and the discrete fine-grained guide labels, an improved conditional mask-to-image translation GAN was proposed. Synthetic images with large intraclass diversity such as defect size, shape, position, thickness, brightness, and background can be generated conditionally.

(2) The proposed model training process was more stable, and the generated data sample was of higher quality when compared to the existing GAN models.

(3) GAN-based augmentation is a useful tool for bridging holes in the discrete training data distribution and enhancing sources of intraclass variation that are challenging to amplify in other ways, but they cannot expand the distribution beyond the training dataset extremes.

(4) When training a defect detection model with a small dataset, a mix of conventional augmentations and GAN-generated synthetic images are extremely helpful to avoid overfitting. The conventional data augmentation extrapolates the training data distribution, while the GAN-based synthetic images add more diversity by interpolating between the discrete data points in the manifold.

Author Contributions: Conceptualization, V.S. and I.M.; methodology, V.S. and J.J.A.M.; validation, V.S. and I.M.; formal analysis, V.S.; investigation, V.S. and G.A.; resources, V.S. and G.A.; data curation, V.S. and G.A.; writing—original draft preparation, V.S.; writing—review and editing, V.S. and A.I.; supervision, I.M. and J.J.A.M.; project administration, I.M., I.L., V.S. and G.A.; funding acquisition, I.M., V.S., I.L. and J.J.A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research work was undertaken in the context of the DIGIMAN4.0 project ("Digital Manufacturing Technologies for Zero-defect", https://www.digiman4-0.mek.dtu.dk/, accessed on 1 January 2023). DIGIMAN4.0 is a European Training Network supported by Horizon 2020, the EU Framework Program for Research and Innovation (project ID: 814225). This research was also partly supported by "5R- Red Cervera de Tecnologías robóticas en fabricación inteligente", contract number CER20211007, under the "Centros Tecnológicos de Excelencia Cervera" program funded by "The Centre for the Development of Industrial Technology (CDTI)".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dwivedi, S.K.; Vishwakarma, M.; Soni, P.A. Advances and Researches on Non Destructive Testing: A Review. *Mater. Today Proc.* **2018**, *5*, 3690–3698, 2018. https://doi.org/10.1016/j.matpr.2017.11.620.
2. Sampath, V.; Maurtua, I.; Martin, J.J.A.; Iriondo, A.; Lluvia, I.; Rivera, A. Vision Transformer based knowledge distillation for fasteners defect detection. In Proceedings of the 2022 International Conference on Electrical, Computer and Energy Technologies (ICECET), Prague, Czech Republic, 20–22 July 2022; pp. 1–6. https://doi.org/10.1109/ICECET55527.2022.9872566.

3. Sampath, V.; Maurtua, I.; Martin, J.J.A.; Rivera, A.; Molina, J.; Gutierrez, A. Attention Guided Multi-Task Learning for Surface defect identification. *IEEE Trans. Ind. Inform.* **2023**, *early access*. https://doi.org/10.1109/TII.2023.3234030.
4. Mirza, M.; Osindero, S. Conditional Generative Adversarial Nets. *arXiv* **2014**. https://doi.org/10.48550/arXiv.1411.1784.
5. Shorten, C.; Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 60. https://doi.org/10.1186/s40537-019-0197-0.
6. Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; Yang, Y. Random Erasing Data Augmentation. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 13001–13008. https://doi.org/10.1609/aaai.v34i07.7000.
7. FMoreno-Barea, J.; Strazzera, F.; Jerez, J.M.; Urda, D.; Franco, L. Forward Noise Adjustment Scheme for Data Augmentation. In Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India, 18–21 November 2018; pp. 728–734. https://doi.org/10.1109/SSCI.2018.8628917.
8. ECubuk, D.; Zoph, B.; Mane, D.; Vasudevan, V.; Le, Q.V. AutoAugment: Learning Augmentation Policies from Data. *arXiv* **2018**. https://doi.org/10.48550/arXiv.1805.09501.
9. Perez, L.; Wang, J. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *arXiv* **2017**. https://doi.org/10.48550/arXiv.1712.04621.
10. Sampath, V.; Maurtua, I.; Martín, J.J.A.; Gutierrez, A. A survey on generative adversarial networks for imbalance problems in computer vision tasks. *J. Big Data* **2021**, *8*, 27. https://doi.org/10.1186/s40537-021-00414-0.
11. Goodfellow, I.; Puget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. https://doi.org/10.1145/3422622.
12. Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv* **2015**. https://doi.org/10.48550/arXiv.1511.06434.
13. Karras, T.; Aila, T.; Laine, S.; Lehtinen, J. Progressive Growing of GANs for Improved Quality, Stability, and Variation. *arXiv* **2017**. https://doi.org/10.48550/arXiv.1710.10196.
14. Denton, E.; Chintala, S.; Szlam, A.; Fergus, R. Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. *arXiv* **2015**. https://doi.org/10.48550/arXiv.1506.05751.
15. Im, D.J.; Kim, C.D.; Jiang, H.; Memisevic, R. Generating images with recurrent adversarial networks. *arXiv* **2016**. https://doi.org/10.48550/arXiv.1602.05110.
16. Nguyen, T.D.; Le, T.; Vu, H.; Phung, D. Dual Discriminator Generative Adversarial Nets. *arXiv* **2017**. https://doi.org/10.48550/arXiv.1709.03831.
17. Shah, M.; Vakharia, V.; Chaudhari, R.; Vora, J.; Pimenov, D.Y.; Giasin, K. Tool wear prediction in face milling of stainless steel using singular generative adversarial network and LSTM deep learning models. *Int. J. Adv. Manuf. Technol.* **2022**, *121*, 723–736. https://doi.org/10.1007/s00170-022-09356-0.
18. Ghosh, A.; Kulharia, V.; Namboodiri, V.; Torr, P.H.S.; Dokania, P.K. Multi-Agent Diverse Generative Adversarial Networks. *arXiv* **2017**. https://doi.org/10.48550/arXiv.1704.02906.
19. Odena, A.; Olah, C.; Shlens, J. Conditional Image Synthesis with Auxiliary Classifier GANs. *arXiv* **2016**. https://doi.org/10.48550/arXiv.1610.09585.
20. Bazrafkan, S.; Corcoran, P. Versatile Auxiliary Classifier with Generative Adversarial Network (VAC+GAN), Multi Class Scenarios. *arXiv* **2018**. https://doi.org/10.48550/arXiv.1806.07751.
21. Chen, X.; Duan, Y.; Houthooft, R.; Schulman, J.; Sutskever, I.; Abbeel, P. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. *arXiv* **2016**. https://doi.org/10.48550/arXiv.1606.03657.
22. Li, X.; Chen, L.; Wang, L.; Wu, P.; Tong, W. SCGAN: Disentangled Representation Learning by Adding Similarity Constraint on Generative Adversarial Nets. *IEEE Access* **2019**, *7*, 147928–147938. https://doi.org/10.1109/ACCESS.2018.2872695.
23. Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X. Improved Techniques for Training GANs. *arXiv* **2016**. https://doi.org/10.48550/arXiv.1606.03498.
24. Chen, Z.; Zeng, Z.; Shen, H.; Zheng, X.; Dai, P.; Ouyang, P. DN-GAN: Denoising generative adversarial networks for speckle noise reduction in optical coherence tomography images. *Biomed. Signal Process. Control* **2020**, *55*, 101632. https://doi.org/10.1016/j.bspc.2019.101632.
25. Zhang, H.; Xu, Y.; Li, H.; Zhang, S.; Wang, X.; Huang, X.; Metaxas, D. StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks. *arXiv* **2016**. https://doi.org/10.48550/arXiv.1612.03242.
26. Isola, P.; Zhu, J.-Y.; Zhou, T.; Efros, A.A. Image-to-Image Translation with Conditional Adversarial Networks. *arXiv* **2016**. https://doi.org/10.48550/arXiv.1611.07004.
27. Wang, C.; Xu, C.; Wang, C.; Tao, D. Perceptual Adversarial Networks for Image-to-Image Transformation. *IEEE Trans. Image Process.* **2018**, *27*, 4066–4079. https://doi.org/10.1109/TIP.2018.2836316.
28. Kim, T.; Cha, M.; Kim, H.; Lee, J.K.; Kim, J. Learning to Discover Cross-Domain Relations with Generative Adversarial Networks. *arXiv* **2017**. https://doi.org/10.48550/arXiv.1703.05192.
29. Zhu, J.-Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *arXiv* **2017**. https://doi.org/10.48550/arXiv.1703.10593.
30. Yang, B.; Liu, Z.; Duan, G.; Tan, J. Mask2Defect: A Prior Knowledge-Based Data Augmentation Method for Metal Surface Defect Inspection. *IEEE Trans. Ind. Inform.* **2022**, *18*, 6743–6755. https://doi.org/10.1109/TII.2021.3126098.
31. Niu, S.; Li, B.; Wang, X.; Lin, H. Defect Image Sample Generation With GAN for Improving Defect Recognition. *IEEE Trans. Autom. Sci. Eng.* **2020**, *17*, 1611–1612. https://doi.org/10.1109/TASE.2020.2967415.

32. Zhang, G.; Cui, K.; Hung, T.-Y.; Lu, S. Defect-GAN: High-Fidelity Defect Synthesis for Automated Defect Inspection. In Proceedings of the 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), Virtual, 5–9 January 2021; pp. 2523–2533. https://doi.org/10.1109/WACV48630.2021.00257.
33. Chen, K.; Cai, N.; Wu, Z.; Xia, H.; Zhou, S.; Wang, H. Multi-scale GAN with transformer for surface defect inspection of IC metal packages. *Expert Syst. Appl.* **2023**, *212*, 118788. https://doi.org/10.1016/j.eswa.2022.118788.
34. Niu, S.; Li, B.; Wang, X.; Peng, Y. Region- and Strength-Controllable GAN for Defect Generation and Segmentation in Industrial Images. *IEEE Trans. Ind. Inform.* **2022**, *18*, 4531–4541. https://doi.org/10.1109/TII.2021.3127188.
35. Wang, X.; Gupta, A. Generative Image Modeling using Style and Structure Adversarial Networks. *arXiv* **2016**. https://doi.org/10.48550/arXiv.1603.05631.
36. Karras, T.; Laine, S.; Aila, T. A Style-Based Generator Architecture for Generative Adversarial Networks. *arXiv* **2018**. https://doi.org/10.48550/arXiv.1812.04948.
37. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. https://doi.org/10.1109/CVPR.2016.90.
38. Tan, M.; Le, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; Volume 97, pp. 6105–6114. Available online: https://proceedings.mlr.press/v97/tan19a.html (accessed on 24 May 2019).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

applied sciences

MDPI

Article

Machine Learning in Manufacturing towards Industry 4.0: From 'For Now' to 'Four-Know'

Tingting Chen [1,*], Vignesh Sampath [2], Marvin Carl May [3], Shuo Shan [1], Oliver Jonas Jorg [4], Juan José Aguilar Martín [5], Florian Stamer [3], Gualtiero Fantoni [4], Guido Tosello [1] and Matteo Calaon [1]

[1] Department of Civil and Mechanical Engineering, Technical University of Denmark, 2800 Kongens Lyngby, Denmark
[2] Autonomous and Intelligent Systems Unit, Tekniker, Member of Basque Research and Technology Alliance, 20600 Eibar, Spain
[3] wbk Institute of Production Science, Karlsruhe Institute of Technology (KIT), Kaiserstr. 12, 76131 Karlsruhe, Germany
[4] Department of Civil and Industrial Engineering, University of Pisa, 56122 Pisa, Italy
[5] Department of Design and Manufacturing Engineering, School of Engineering and Architecture, University of Zarazoga, 50009 Zaragoza, Spain
* Correspondence: tchen@dtu.dk; Tel.: +45-5028-1068

Abstract: While attracting increasing research attention in science and technology, Machine Learning (ML) is playing a critical role in the digitalization of manufacturing operations towards Industry 4.0. Recently, ML has been applied in several fields of production engineering to solve a variety of tasks with different levels of complexity and performance. However, in spite of the enormous number of ML use cases, there is no guidance or standard for developing ML solutions from ideation to deployment. This paper aims to address this problem by proposing an ML application roadmap for the manufacturing industry based on the state-of-the-art published research on the topic. First, this paper presents two dimensions for formulating ML tasks, namely, 'Four-Know' (Know-what, Know-why, Know-when, Know-how) and 'Four-Level' (Product, Process, Machine, System). These are used to analyze ML development trends in manufacturing. Then, the paper provides an implementation pipeline starting from the very early stages of ML solution development and summarizes the available ML methods, including supervised learning methods, semi-supervised methods, unsupervised methods, and reinforcement methods, along with their typical applications. Finally, the paper discusses the current challenges during ML applications and provides an outline of possible directions for future developments.

Keywords: machine learning; Industry 4.0; manufacturing; artificial intelligence; smart manufacturing; digitization

check for updates

Citation: Chen, T.; Sampath, V.; May, M.C.; Shan, S.; Jorg, O.J.; Aguilar Martín, J.J.; Stamer, F.; Fantoni, G.; Tosello, G.; Calaon, M. Machine Learning in Manufacturing towards Industry 4.0: From 'For Now' to 'Four-Know'. *Appl. Sci.* **2023**, *13*, 1903. https://doi.org/10.3390/app13031903

Academic Editors: Alexandre Carvalho and Richard (Chunhui) Yang

Received: 23 November 2022
Revised: 18 January 2023
Accepted: 27 January 2023
Published: 1 February 2023

1. Introduction

Within the fourth industrial revolution, coined as 'Industry 4.0', the way products are manufactured is changing dramatically [1]. Moreover, the way humans and machines interact with one another in manufacturing has seen enormous changes [2], developing towards an 'Industry 5.0' notion [3]. The digitalization of businesses and production companies, the inter-connection of their machines through embedded system and the Internet of Things (IoT) [4], the rise of cobots [5,6], and the use of individual workstations and matrix production [7] are disrupting conventional manufacturing paradigms [1,8]. The demand for individualized and customized products is continuously increasing. Consequently, order numbers are surging while batch sizes diminish, to the extremes of fully decentralized 'batch size one' production. The demand for a high level of variability in production and manufacturing through Mass Customization is inevitable. Mass Customization in turn requires manufacturing systems which are increasingly more flexible and adaptable [7–9].

Machine Learning (ML) is one of the cornerstones for making manufacturing (more) intelligent, and thereby providing it with the needed capabilities towards greater flexibility and adaptability [10]. These advances in ML are shifting the traditional manufacturing era into the smart manufacturing era of Industry 4.0 [11]. Therefore, ML plays an increasingly important role in manufacturing domain together with digital solutions and advanced technologies, including the Industrial Internet of Things (IIoT), additive manufacturing, digital twins, advanced robotics, cloud computing, and augmented/virtual reality [11]. ML refers to a field of Artificial Intelligence (AI) that covers algorithms learning directly from their input data [12]. Despite most researchers focusing on finding a single suitable ML solution for a specific problem, efforts have already been undertaken to reveal the entire scope of ML in manufacturing. Wang et al. presented frequently-used deep learning algorithms along with an assessment of their applications towards making manufacturing "smart" in their 2018 survey [13]. In particular, they discussed four learning models: Convolutional Neural Networks, Restricted Boltzmann Machines, Auto-Encoders, and Recurrent Neural Networks. In their recent literature review on "Machine Learning for Industrial Applications", Bertolini et al. [12] identified, classified, and analyzed 147 papers published during a twenty-year time span from Jan. 2000 to Jan. 2020. In addition, they provided a classification on the basis of application domains in terms of both industrial areas and processes, as well as their respective subareas. Within these domains, the authors analyzed the different trends concerning supervised, unsupervised, and reinforced learning techniques, including the most commonly used algorithms, Neural Networks (NNs), Support Vector Machine (SVM), and Tree-Based (TB) techniques. The goal of another literature review from Dogan and Birant [14] was to provide a sound comprehension of the major approaches and algorithms from the fields of ML and data mining (DM) that have been used to improve manufacturing in the recent past. Similarly, they investigated research articles from the period of the past two decades and grouped the identified articles under four main subjects: scheduling, monitoring, quality, and failure.

While these classifications and trend analyses provide an excellent overview of the extent of ML applications in manufacturing, they mainly focus on introducing ML algorithms; the implementation of ML solution for different tasks in an industrial environment from scratch has not yet been fully discussed. In general, a comprehensive formulation of industrial problems prior to the development of ML solutions seems lacking. Therefore, the issue we aim to address in this paper is how ML can be implemented to improve manufacturing in the transition towards Industry 4.0. From this issue, we derive the following research questions:

- RQ1: How does ML benefit manufacturing, and what are the typical ML application cases?
- RQ2: How are ML-based solutions developed for problems in manufacturing engineering?
- RQ3: What are the challenges and opportunities in applying ML in manufacturing contexts?

To answer these research questions, more than a thousand research articles retrieved from two well-known research databases were systematically identified, screened, and analyzed. Subsequently, the articles were classified within a two-dimensional framework, which takes value-based development stages into account on one axis and manufacturing levels on the other. The development stage concerns visibility, transparency, predictive capacity, and adaptability, whereas the four manufacturing levels are product, process, machine, and system.

The rest of this paper is structured as follows. Section 1 introduces the key concepts, research questions, and motivations. Section 2 proposes the methodology of 'Four-know' and 'Four-level' to establish a two-dimensional framework for helping to formulate industrial problems effectively. Based on the proposed framework, a systematic literature review is carried out and the identified articles are analysed and classified. Section 3 describes a six-step pipeline for the application of ML in manufacturing. Section 4 explains different ML methods, presenting where and how they have been applied in manufacturing according to the prior identified research articles. Section 5 formulates common challenges and potential future directions; finally, the paper concludes in Section 6 with a summary and discussion of the authors' findings.

2. Overview of Machine Learning in Manufacturing

Despite numerous ML studies and their promising performance, it remains very difficult for non-experts working in the manufacturing industry to begin developing ML solutions for their specific problems. The first challenging part of application is to formulate the actual problems to be solved [15]. Therefore, this section aims to overcome this problem by introducing the categories of Four-Know and Four-Level to help formulate ML tasks in manufacturing and describing the benefits of applying ML in manufacturing from ML use cases categorized using the Four-Know and Four-Level concepts (RQ1). Lastly, an overview and developing trends in recent ML studies are provided as formulated by Four-Know and Four-Level.

2.1. Introduction of Four-Know and Four-Level

According to the *Acatech* Industrie 4.0 Maturity levels [16], the development towards Industry 4.0 in manufacturing can be structured into the following six successive stages: computerization, connectivity, visibility, transparency, predictive capacity, and adaptability. The first two stages, computerization and connectivity, provide the basis for digitization, while the rest are analytic capabilities required for achieving Industry 4.0. ML, as powerful data analytics tools are normally applied in the last four stages. Inspired by the *Acatech* Industrie 4.0 Maturity levels, ML studies in manufacturing can be categorized into four subjects: Know-what, Know-why, Know-when, and Know-how, which to a degree overlap with visibility, transparency, predictive capacity, and adaptability, respectively. The Four-Know definitions are presented below:

- *Know-what* deals with understanding of the current states of machines, processes, or production systems, which can help in rapid decision-making. It should be noted that Know-what goes beyond visualization of real-time data. Instead, data should be processed, analyzed, and distilled into information which enables decision-making. For instance, typical examples of Know-what in manufacturing are defect detection in quality control [17,18], fault detection in process/machine monitoring [19,20], and soft sensor modelling [21,22].
- *Know-why*, based on the information from Know-what, aims to identify inner patterns from historical data, thereby discovering the reasons for a thing happening. Know-why includes the identification of interactions among different variables [23] and the discovery of cause-effect relationship between an event and other variables [24,25]. On one hand, Know-why can indicate most important factors for understanding Know-what. On the other hand, Know-why is the prerequisite for Know-when, as the reliability of predictions is heavily dependent upon the quality of casual inference.
- *Know-when*, built on Know-why, involves timely predictions of events or prediction of key variables based on historical data, allowing the decision-maker can take actions at early stages. For instance, Know-when in manufacturing includes quality prediction based on relevant variables [26,27], predictive maintenance via detection of incipient anomalies before break-down [28,29], and predicting Remaining Useful Life (RUL) [30,31].
- *Know-how*, on the foundation of Know-when, can recommend decisions that help adapt to expected disturbance and can aid in self-optimization. Examples in manufacturing include prediction-based process control [27,32], scheduling of predictive maintenance tasks [33,34], dynamic scheduling in the flexible production [35,36], and inventory control [34].

The aim of applying ML in manufacturing is to achieve production optimization across four different levels: product, process, machine, and system. Therefore, the use cases for applying ML can be further categorized by these different levels, as shown in Figure 1 and Table 1, which answer RQ1 in terms of ML typical use cases.

Table 1. Typical ML use cases categorized by Four-Level and Four-Know.

Level	Know-What	Know-Why	Know-When	Know-How
Product	Defect detection [37], Product design [38]	Correlation between process and quality [23]	Quality prediction [26]	Quality improvement [39]
Process	Process monitoring [40]	Root cause analysis of process failure [41], Process modelling [42]	Process fault prediction [43], Process characteristics prediction [44]	Self-optimizing process planning [45], Adaptive process control [46]
Machine	Machine tool monitoring [47]	Fault diagnosis [48], Downtime prediction [49]	RUL prediction [50], Tool wear prediction [51]	Adaptive compensation of errors [52,53],
System	Anomaly detection [54]	Root cause analysis of production disturbances or casual-relationship discovery [55]	Production performance prediction [56], Human behavior control [57]	Predictive scheduling [58], Adaptive production control [59]

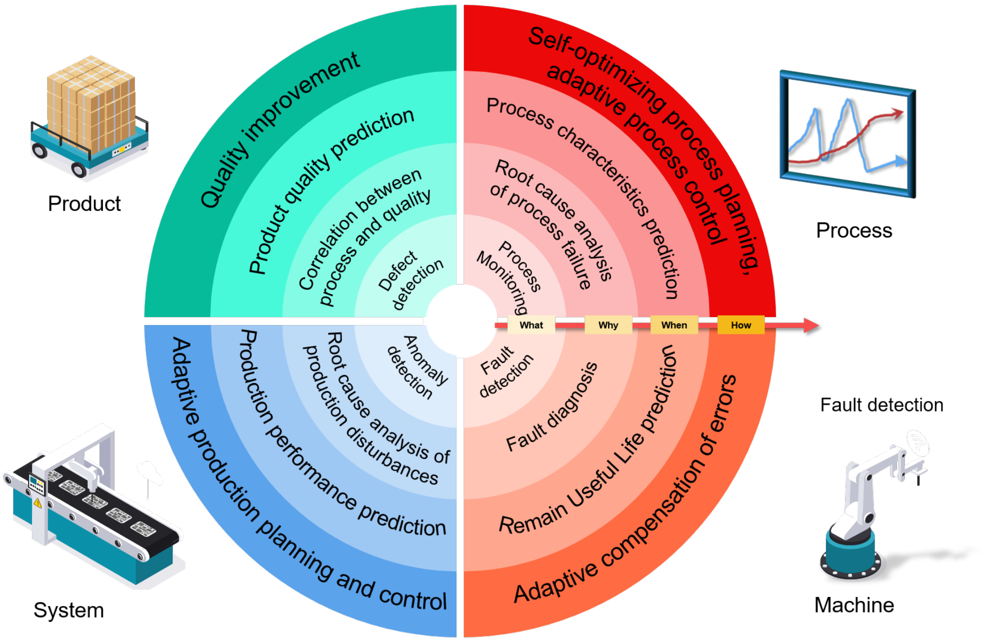

Product
Quality improvement
Product quality prediction
Correlation between process and quality
Defect detection
Self-optimizing process planning, adaptive process control
Process characteristics prediction
Root cause analysis of process failure
Process Monitoring
Process
What
Why
When
How
Anomaly detection
Root cause analysis of production disturbances
Production performance prediction
Adaptive production planning and control
Fault detection
Fault diagnosis
Remain Useful Life prediction
Adaptive compensation of errors
Fault detection
System
Machine

Figure 1. Four-Level and Four-Know categorization of ML applications. The Four-Know categories, from Know-what to Know-how, are respectively demonstrated by the four concentric circles, from the inner circle to the outer circle, with each circle divided into four quarters according to the Four Levels.

2.2. Literature Review Methodology

In order to address the research questions laid out in Section 1, a systematic literature review following the PRISMA methodology [60] was carried out. Two well-known research databases, Scopus (Elsevier) and Web of Science (WoS), were chosen for retrieving documents. The overall literature review process is shown in Figure 2.

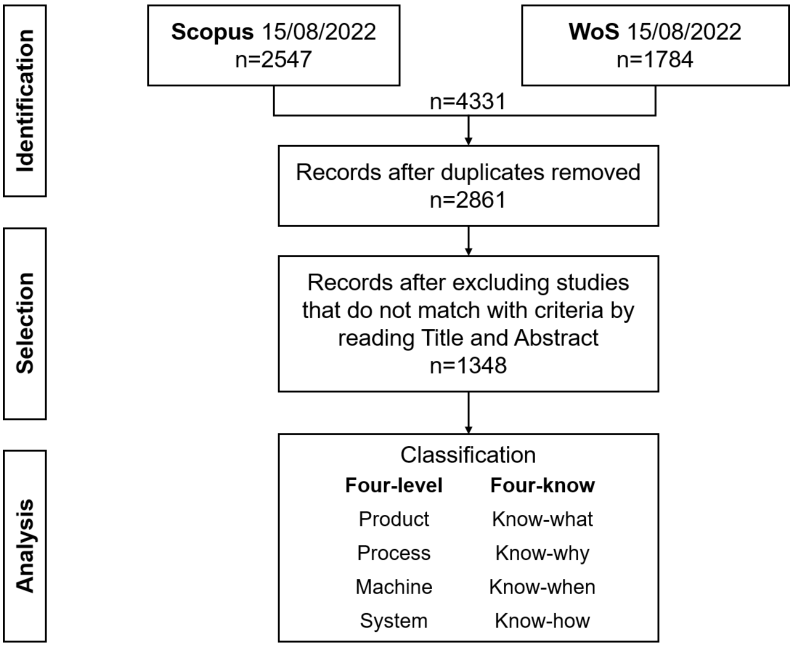


Figure 2. The overall literature review process following PRISMA. All identified documents were screened and assessed for eligibility, then subjected to Four-Level and Four-Know classification.

Table 2 shows the limitations used when performing the document search. It should be noted that the query strings were used for Title, Abstract, and Keywords as well as Keyword Plus (only in WoS).

Table 2. Limitations for document searching.

Item	**Description**
Query string	("manufacturing" OR "industry 4.0" OR "industrie 4.0") AND ("machine learning" OR "deep learning" OR "supervised learning" OR "semi-supervised learning" OR "unsupervised learning" OR "reinforcement learning")
Year	Published from 2018 to 2022
Language	English
Subject/Research area	Engineering
Document type	Article

Following the document search, 2547 documents were found from Scopus and 1784 from WoS. The identified publications from the two databases were merged and duplicates were removed, resulting in 2861 publications. The documents were then evaluated and selected by reading the Title and Abstract field, and articles that did not meet the following selection criteria were excluded:

- The study dealt with the context of manufacturing;
- The study dealt with ML applications in specific fields.

Therefore, conceptual models, frameworks, and studies that only focused on algorithm development were considered to be out of scope.

Finally, the remaining 1348 documents were analyzed and classified based on the Four-Level and Four-Know categories. Figure 3 shows the trend of ML applications in manufacturing over the past five years from the Four-Level perspective. Figure 4 reveals the detailed distribution of ML applications in Four-Know terms. It should be noted that because the literature review was conducted in August 2022, the actual numbers for the full year 2022 should be higher. As can be seen, there has been a gradual increase in the number of ML publications in manufacturing in all levels over the past five years. Typically, what stands out in this figure is the dominance of the product level. From Figure 4, it can be seen that recent ML applications in product level are mainly focused on Know-what and Know-when. A similar pattern can be found at the machine level. Interestingly, a considerable growth in Know-how is observed at the process and system levels compared to the others. The reason for this may be correlated with higher demand for adaptability with respect to changes on the process and system levels.

The identified documents were analyzed and classified according to their applied ML methods, providing examples for non-experts when dealing with similar tasks.

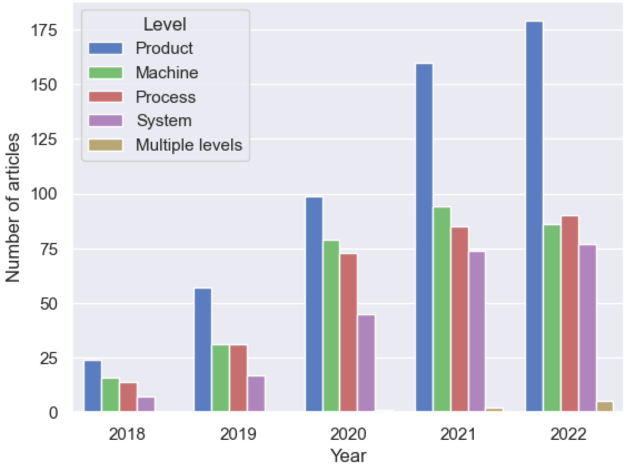


Figure 3. Trends in ML publications in manufacturing in the past five years by Four-Level grouping.

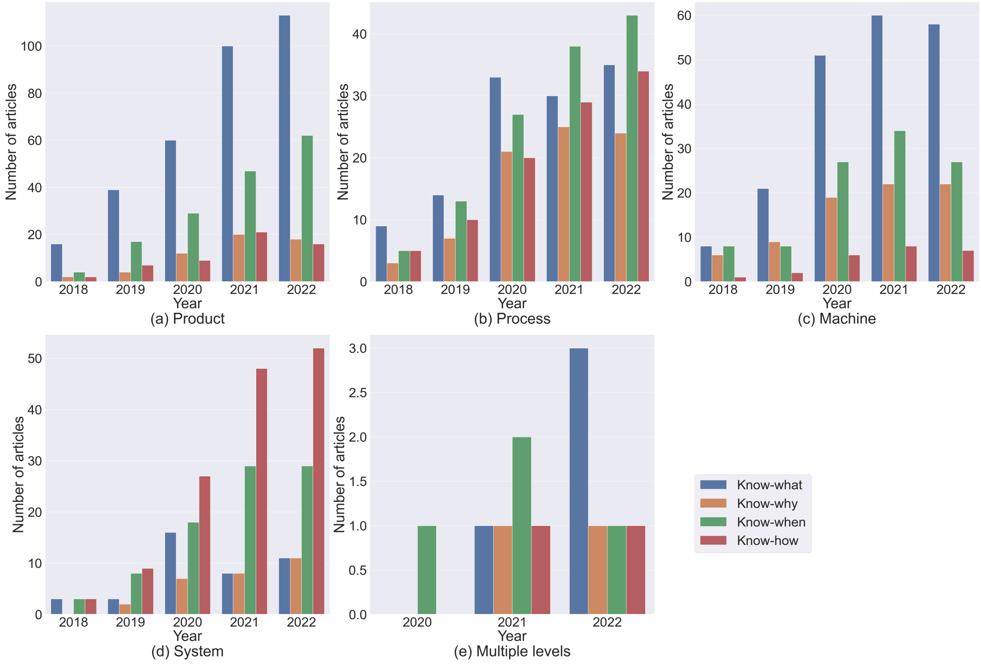


Figure 4. Four-Know development trends for each level over the past five years.

3. Pipeline of Applying Machine Learning in Manufacturing

ML is a technique capable of extracting knowledge from data automatically [12]. Increasing research on ML has shown that it is an appealing solution when tackling complex challenges. In recent years, more and more manufacturing industries have begun to leverage the benefits of ML by developing ML solutions in several industrial fields. However, despite plenty of off-the-shelf ML models, there are challenges when applying ML to real-world problems [61]. In particular, it is harder for small and medium-sized enterprises to develop in-house ML solutions, as commercial ML solutions are normally confidential and inaccessible. Therefore, this section aims to provide a pipeline for applying ML for those who are starting from scratch (RQ2). Applying machine learning in manufacturing normally involves the following six steps: (i) data collection, (ii) data cleaning, (iii) data transformation, (iv) model training, (v) model analysis, and (vi) model push, as shown in Figure 5.

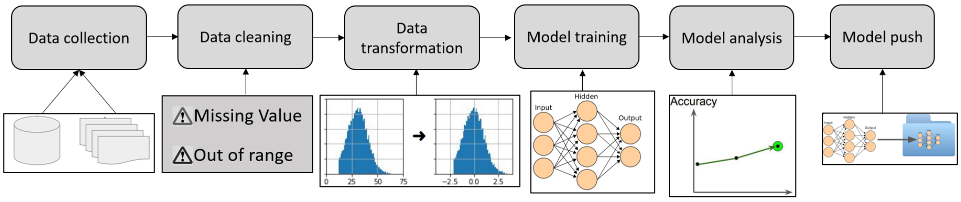


Figure 5. Pipeline of applying machine learning in manufacturing.

3.1. Data Collection

The lifeblood of any machine learning model is data. In order for an ML model to learn, clean data samples must be continuously fed into system throughout the training process. When the collected data are highly imbalanced or otherwise inadequate, the desired task may not be achievable. Data can be collected from different sources, including machines, processes, or production with the aid of sensors or external databases. In terms of data types, the data used in machine learning can be generally categorized as follows:

- *Image data*, matrices of pixels with two or more dimensions, such as gray-scale images or colored images. Image data can acquired by with vision systems, through data transformations such as simple concatenation of several one-dimensional vectors with same length, or by the transformation of images from the spatial domain to the frequency domain.
- *Tabular data* organized in a table, where normally one axis represents attributes and another axis represents observations. Tabular data are typically observed in production data, where the attributes of events of interest are collected. Though tabular data share a similar data structure with image data, the latter are more focused on one-dimensional interaction among attributes, while image data typically stress spatial interactions in both dimensions.
- *Time series data*, sequences of one or more attributes over time, with the former corresponding to univariate time series and the latter multivariate time series. In manufacturing, time series data are normally acquired with sensors whenever there is a need for monitoring time flow changes of data.
- *Text data*, including written documents with words, sentences or paragraphs. Examples of text data in manufacturing include maintenance reports on machines and descriptions of unexpected disturbances or events in production.

3.2. Data Cleaning

Real-world industrial data are highly susceptible to noisy, missing, and inconsistent data due to several factors. Low-quality noisy data can lead to less accurate ML models. Data cleaning [62] is a crucial step when organizing data into a consistent data structure across packages, and can improve the quality of the data, leading to more accurate ML models. It is usually performed as an iterative approach. Methods include filling in missing values, smoothing noisy data, removing outliers, resolving data inconsistencies, etc.

3.3. Data Transformation

Data transformation is the process of transforming unstructured raw data into data better suited for model construction. Data transformation can be broadly classified into mandatory transformations and optional quality transformations. Mandatory transformations must be carried out to convert the data into a usable format and then deliver the transformed data to the destination system. These include transforming non-numerical data into numerical data, resizing data to a fixed size, etc. It should be noted that data transformations are not always straightforward. Indeed, in certain situations data types can be interconvertible by leveraging specific processing techniques, as shown in Figure 6. For instance, univariate time series can be converted into image data using the Gramian Angular Field (GAF) or Markov Transition Field (MTF) [63] methods. Unstructured text data can be converted into tabular data via word embedding [64]. Tabular data can be transformed into image data by projecting data into a 2D space and assigning pixels, as in Deepinsight [65] or Image Generator for Tabular Data (IGTD) [66]. Image data are preferable for data analysis, as they allow the power of Convolutional Neural Networks (CNNs) [67] to be exploited.

In real-world applications, data are normally high-dimensional and redundant. When performing data modelling directly in the original high-dimensional space, the computational efficiency can be very low. Hence, it is necessary to reduce the dimensionality in order to obtain better representation for data modelling. This is achieved by feature selection, which selects the most informative feature subset from raw data, or feature extraction, which generates new lower-dimensional features. After feature engineering, features are either manually designed, so-called “handcrafted features” [68], or automatically learned from data, so-called “automatic features”. Handcrafted features are heavily dependent on domain knowledge, and normally have physical meaning. However, these features are highly subjective [69] and inevitably lack implicit key features [70,71].

By contrast, automatic features driven by data require no prior knowledge. Therefore, they have been gaining increasing research attention in recent years. Conventionally, automatic features are obtained by linear transformations such as Principle Component Analysis (PCA) [72] or Independent Component Analysis (ICA) [73]. However, with the development of Artificial Neural Networks (ANNs), direct learning of implicit features has become possible by optimizing the loss function. Thus, neural networks have gradually developed into an end-to-end solution where knowledge is directly learned from raw data without human effort. Typically, CNNs [74] and Recurrent Neural networks (RNNs) [75] are used for image data and time series data, respectively.

A summary of typical features for different data types can be seen in Table 3.

Table 3. Typical features for different data types.

Data Type	Handcrafted Features	Automatic Features
Image data	LBP [76], SIFT [77], HOG [78]	ICA, CNNs
Tabular data	feature selection	PCA, ICA, ANNs
Time series data	Time domain: mean, min, max, etc. Frequency domain: power spectrum [78] Time-frequency domain: DWT [79], STFT [80]	ICA, RNNs
Text data	Bag of Words (BoW) [81]	Word2vec [82]

3.4. Model Training

After selecting the features, it is necessary to form the correct data structure for each individual ML model used in the subsequent steps. Note that different ML algorithms might require different data models for the same task. Furthermore, results can be improved through normalization or standardization. Then, the ML models can be applied in the actual modelling phase. The first step in training a machine learning model typically involves selecting a model type that is appropriate for the nature of the data and the problem at hand. After a model has been chosen, it can be trained by providing it with the training data and using an optimization algorithm to find the set of parameters that provide the best performance on those data. Depending on the task, either unsupervised, semi-supervised, supervised, or reinforcement learning can be applied. These are individually introduced in the following section.

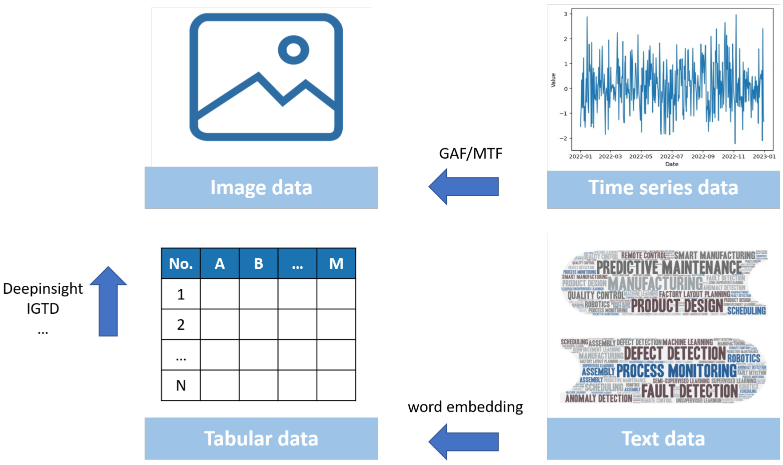
3.5. Model Analysis

Analysis of model performance is an important step in choosing the right model. This stage emphasizes how effective the selected model will perform in the future and helps to make the final decision with regard to model selection. Performance analysis evaluates models using different metrics, e.g., accuracy, precision, recall, and F1-score (the weighted average of precision and recall) for classification tasks and the root mean square error (RMSE) for regression tasks.

3.6. Model Push

Although state-of-the-art ML models improve predictive performance, they contain millions of parameters, and consequently require a large number of operations per inference. Such computationally intensive models make deployment in low-power or resource-constrained devices with strict latency requirements quite difficult. Several methods, including model pruning [83], model quantization [84], and knowledge distillation [85], have been suggested in the literature as ways to compress these dense models.

Overall, In the context of manufacturing applications, data collection, data cleaning, data transformation, model training, model analysis, and model push are key steps in the implementation of utilizing historical data with ML in order to optimize production and improve efficiency, quality, and productivity. For instance, data collection involves gathering data from various sources, such as sensor data, production logs, and quality control records. Data cleaning involves removing any errors, inconsistencies, or irrelevant information from the data. Data transformation involves preparing the data for analysis via formatting in a way that is suitable for the chosen model. Model training involves using the cleaned and transformed data to train a machine learning model. Model analysis involves evaluating the performance of the model and identifying any areas for improvement. Model push involves deploying the model in a production environment and making predictions or decisions based on the model. All of these steps are critical to ensuring that the results from ML models are accurate, reliable, and useful for manufacturing production.



No.	A	B	...	M
1				
2				
...				
N				

Figure 6. Data types used in ML and their convertibility.

4. Machine Learning Methods and Applications

Model development is the core of ML-based solutions, as the selection of an ML model plays a critical roles in the outcome. Therefore, this section aims to provide a comprehensive overview of ML methods and their potential possibilities in manufacturing applications, including supervised learning methods, semi-supervised learning methods, unsupervised learning methods, and reinforcement learning methods. In addition, example typical applications for each category of ML method are listed to support model selection.

4.1. Supervised Learning Methods

Supervised learning methods aim to learn an approximation function f that can map inputs x to outputs y with the guidance of annotations $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$. In supervised learning, the algorithm analyzes a labeled dataset and derives an inferred function which can be applied to unseen samples. It should be noted that labeled dataset is a necessity for supervised learning, and as such it requires a large amount of data and high labeling costs. Supervised learning methods are generally used for dealing with two problems, namely, regression and classification. The difference between regression and classification is in the data type of the output variables; regression predicts continuous numeric values ($y \in \mathbb{R}$), while classification predicts categorical values ($y \in \{0, 1\}$). In terms of principles, supervised learning methods can be further categorized into four groups: tree-based methods, probabilistic-based methods, kernel-based methods, and neural network-based methods.

Tree-based methods: Tree-based methods aim at partitioning the feature space into several regions until the datapoints in each region share a similar class or value, as depicted in Figure 7. After space partitioning, a series of if–then rules with a tree-like structure can be obtained and used to determine the target class or value. Compared with the black-box models in other supervised methods, Tree-based methods are easily understandable models that offer better model interpretability. Decision trees [86], in which only a single tree is established, are the most basic of tree-based methods. It is simple and effective to train a decision tree, and the results are intuitively understandable, though this approach is very prone to overfitting. A tree ensemble is an extension of the decision tree concept. Instead of establishing a single tree, multiple trees are established in parallel or in sequence, referred to as bagging [87] and boosting [88], respectively. Commonly used tree ensemble methods include Random Forest [89], Adaptive Boosting (AdaBoost) [88], and Extreme Gradient Boosting (XGBoost) [90].

Thanks to their better model interpretability, tree-based methods can be used to identify the most important factors leading up to events. Their possible applications in manufacturing are mainly in the Know-why and Know-when stages. For instance, examples of Know-why tasks with tree-based methods at the product and machine level include identifying the influencing factors that lead to quality defects [91] or machine failure [92], thereby allowing the manufacturer to diagnose problems effectively. In addition, the identified important factors when using tree-based methods can help in further predicting target values such as product quality [93](Know-when, product level) or events of interest before they happen, such as machine breakdown [31] (Know-when, machine level).

Probabilistic-based methods: For a given input, probabilistic-based methods provide probabilities for each class as the output. Probabilistic models are able to explain the uncertainties inherent to data, and can hierarchically build complex models. Widely used probabilistic-based methods include Bayesian Optimization (BO) [94] and Hidden Markov Models (HMM) [95].

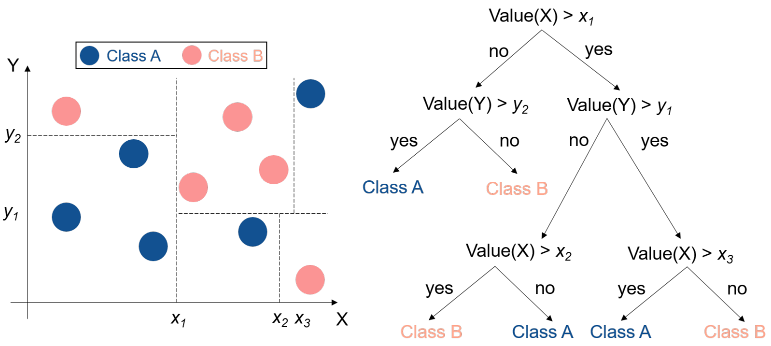


Figure 7. The principle of a decision tree. As shown, the feature space is partitioned into several rectangles in which the input point can find the corresponding class.

The dependencies among different variables can be well captured by Bayesian networks [94], enabling a greater likelihood of predicting the target. This can be potentially beneficial for manufacturing when it comes to Know-what and Know-when tasks, for instance, detection or prediction of events such as quality issues [96] (product level), machine failure [97] (machine level), or dynamic process modelling [98] (process level).

Markov chains [95], on the other hand, are a type of probabilistic model that describe a sequence of possible events in which the probability of each event depends only on the state attained in the previous event. Markov chains can be utilized in manufacturing to model and analyze the behavior of systems (Know-why, system level) such as production lines [99] or supply chains [100]. In addition, the capability of predicting future states with Markov chains enables applications predicting joint maintenance in production systems [101] (Know-when, system level) and optimizing production scheduling [102] (Know-how, system level).

Kernel-based methods: As depicted in Figure 8, kernel-based methods utilize a defined kernel function to map input data into a high-dimensional implicit feature space [103]. Instead of computing the targeted coordinates, kernel-based methods normally compute the inner product between a pair of data points in the feature space. However, kernel-based methods have low efficiency, especially with respect to large-scale input data. Due to the promising capability of kernel-based methods in classification and regression, they can be utilized in the Know-what and Know-when stages in manufacturing, such as defect detection [104] (Know-what, product level), quality prediction [105] (Know-when, product level), and wear prediction in machinery [106] (Know-when, machine level). There are different types of kernel-based methods in supervised learning, such as SVM [107] and Kernel–Fisher discriminant analysis (KFD) [108].

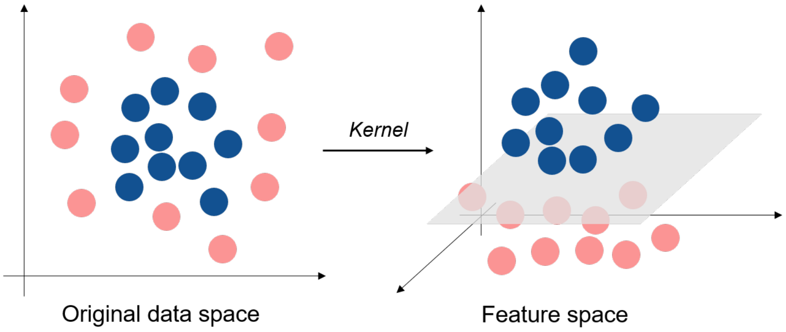


Figure 8. The principle of kernel-based methods. Using a kernel, the linearly inseparable input data are transformed to another feature space in which they become linearly separable.

Neural-network-based methods: Inspired by biological neurons and their ability to communicate with other connected cells, neural network-based methods employ artificial neurons. A typical neural network, such as ANNs, consists of an input layer, hidden layer, and output layer, as illustrated in Figure 9. Common ANNs types include CNNs [109], RNNs [110], and Deep Belief Network (DBN) [111].

Thanks to their powerful feature extraction capability when using matrix-like data, CNNs are widely used for image processing. In terms of possible applications in manufacturing, CNNs can be used in the Know-what stage to perform image-based quality control [112] (Know-what, product level) or image-based process monitoring [113] (Know-what, process level). In addition, by converting time series data from sensors to 2D images [114], CNNs can be used to detect and diagnosis machine failure as well.

RNNs are typically used to process sequential input data such as time series data or sequential images. Therefore, in terms of possible applications in manufacturing, RNNs are well-suited to the Know-when stage for analyzing sensor data or live images from machines, processes, or production systems. For instance, RNNs can enable the real-time performance prediction, such as the remaining useful life of machinery [115] (Know-when, machine level), process behavior prediction [116] (Know-when, process level), or the prediction of production indicators for real-time production scheduling [117] (Know-when, system level).

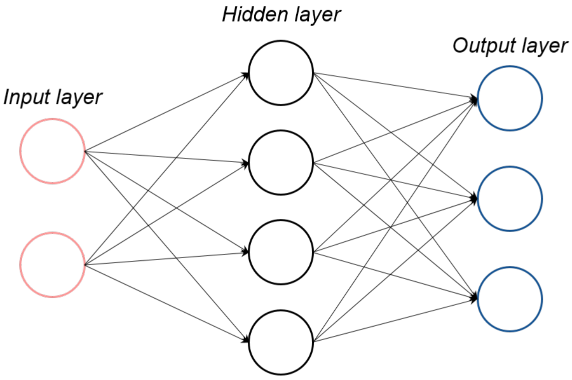


Figure 9. The scheme of an ANN, which normally consists of an input layer, hidden layer and output layer.

The typical supervised learning approaches applied in manufacturing are summarized in Table A1.

4.2. Unsupervised Learning Methods

Unsupervised learning algorithms aim to identify patterns in data sets containing data points that are not labeled. Unsupervised learning eliminates the need for labeled data and manual feature engineering, allowing for more general, flexible, and automated ML methods. As a result, unsupervised learning methods draw patterns and highlight areas of interest, revealing critical insight into the production process and opportunities for improvement. This can allow manufacturers to make better production-focused decisions, driving their business forward. The primary goal of unsupervised learning is to identify hidden and interesting patterns in unlabeled data. In terms of principles, there are three types of unsupervised tasks: Dimension Reduction [118,119], Clustering [120], and Association Rules [121]. Many aspects of unsupervised learning can be beneficial in manufacturing applications. First, clustering algorithms can be used to identify outliers in manufacturing data. Another aspect is to handle high dimensional data, e.g., for manufacturing cost estimation, quality improvement methodologies, production process optimization, better understanding of the customer’s data, etc. Usually, a dimensional reduction support algorithm is required to handle data complexity and high dimensionality. Finally, it is challenging to perform root cause analysis in large-scale process execution due to the complexity of services in data centers. Association rule-based learning can be employed to conduct root cause analysis and to identify correlations between variables in a dataset.

Dimensional reduction is the process of converting data from a high-dimensional space to a low-dimensional space while preserving important characteristics of the original data.

Principal component analysis (PCA) [118]: The main idea of PCA is to minimize the number of interrelated variables in a dataset while preserving as much of the dataset's inherent variance as possible. A new set of variables, called principal components (PCs), are generated; these are uncorrelated and sorted such that the first few variables retain the majority of the variance included in all of the original variables. A pictorial representation of PCA is shown in Figure 10.

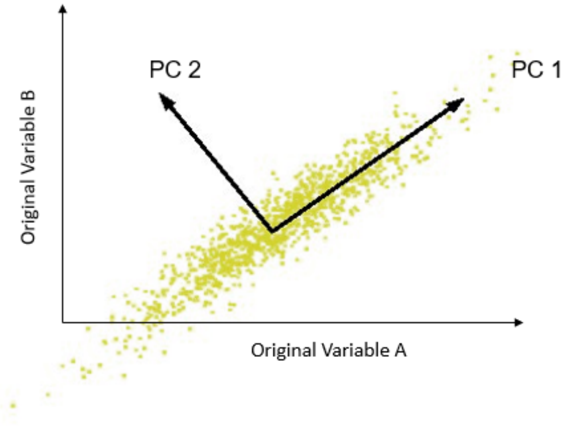


Figure 10. Principal Component Analysis.

The five steps below can be used to condense the entire process of extracting principal components from a raw dataset.

1. Say we wish to condense d features in our data matrix X to k features. The first step is to standardize the input data:
$$z = x - \mu / \sigma$$
where μ is the mean and σ is the standard deviation.
2. Next, it is necessary to find the covariance matrix of the standardized input data. The covariance of variables X and Y can be written as follows:
$$\operatorname{cov}(X, Y) = \frac{1}{n-1} \sum_{i=1}^{n} (Xi - \vec{x})(Yi - \bar{y}). \tag{1}$$
3. The third steps is to find all of the eigenvalues and eigenvectors of the covariance matrix:
$$A\vec{v} = \lambda\vec{v} \tag{2}$$
$$A\vec{v} - \lambda\vec{v} = 0 \tag{3}$$
$$\vec{v}(A - \lambda I) = 0. \tag{4}$$
4. Then, the eigenvector corresponding to the largest eigenvalue is the direction with the maximum variance, the eigenvector corresponding to second-largest eigenvalue is the direction with the second maximum variance, etc.
5. To obtain k features, it is necessary to multiply the original data matrix by the matrix of eigenvectors corresponding to the k largest eigenvalues.

PCA is particularly useful for processing manufacturing data, which typically have a large number of variables, making it difficult to identify patterns and trends. A variety of applications of PCA in manufacturing are listed below:

1. Quality improvement (Know-why, product level): by analyzing the variations of a product's features, PCA can be used to identify the causes of product defects [122].
2. Machine monitoring (Know-why, machine level): by analyzing sensor data from a machine, PCA can be used to detect incipient patterns in the data that indicate potential issues with the machinery, such as wear and tear [123].
3. Process optimization (Know-why, process level): by analyzing variations in the process data, PCA can be used to identify the most important factors that affect the process, allowing the manufacturer to optimize the process and thereby reduce costs [124].

Autoencoder (AE) [119] is another popular method for reducing the dimensionality of high-dimensional data. AE alone does not perform classification; instead, it provides a compressed feature representation of high-dimensional data. The typical structure of AE consists of an input layer, one hidden or encoding layer, one reconstruction or decoding layer, and an output layer. The training strategy of AE includes encoding input data into a latent representation that can reconstruct the input. To learn a compressed feature representation of input data, AE tries to reduce the reconstruction error, that is, to minimize the difference between the input and output data. An illustration of AE is shown in Figure 11.

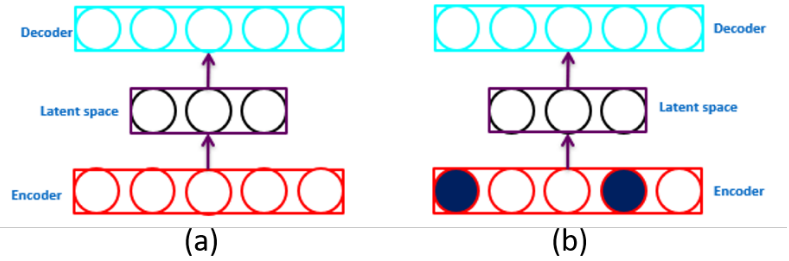


Figure 11. A pictorial representation of (**a**) an Autoencoder and (**b**) a Denoising Autoencoder. An autoencoder is trained to reconstruct its input, while a denoising autoencoder is trained to reconstruct a "clean" version of its input from a corrupted or "noisy" version of the input.

There are different types autoencoders that can be used for high-dimensional data. *Stacked Autoencoder* (SAE) [119] is built by stacking multiple layers of AEs in such a way that the output of one layer serves as the input of the subsequent layer. *Denoising autoencoder* (DAE) [125] is a variant of AE that has a similar structure except for the input data. In DAE, the input is corrupted by adding noise to it; however, the output is the original input signal without noise. Therefore, unlike AE, DAE has the ability to recover the original input from a noisy input signal. *Convolutional autoencoder* [126] is another interesting variant of AE, employing convolutional layers to encode and decode high-dimensional data.

AEs can be used for a variety of applications in manufacturing, such as:

1. Anomaly detection (Know-what): an AE can be trained to reconstruct normal data and detect abnormal data by measuring the reconstruction error, which allows the manufacturer to detect and address issues such as product defects [124] and machinery failure [127].
2. Feature selection (Know-why): an AE can be used to identify the most important features in the data and remove the noise and irrelevant information, which can be used for diagnosis of product defects or to detect events of interests [128].
3. Dimensionality reduction: an AE can be used to reduce the dimensionality of large and complex datasets, making it easier to identify patterns and trends [129].

Furthermore, AEs can be used in conjunction with other techniques, such as clustering or classification, to improve the accuracy of prediction and enhance the interpretability of the results [130]. Additionally, AEs can be used for data visualization. By reducing the dimensionality of the data, AEs allow high-dimensional data to be visualized clearly and interpretably [129] in a way that can be easily understood by non-technical stakeholders.

Clustering: The objective of clustering is to divide the set of datapoints into a number of groups, ensuring that the datapoints within each group are similar to one another and different from the datapoints in the other groups. Clustering methods are powerful tools, allowing manufacturers examine large and complex datasets and gain meaningful insights. There are different clustering methods available, each with their own strengths and weaknesses, and the choice of method depends on the characteristics of the data and the problem to be solved. Among the widely used clustering methods are *Centroid-based Clustering* [120], *Density-based Clustering* [131], *Distribution-based Clustering* [132], and *Hierarchical Clustering* [133]. Clustering algorithms have a wide range of applications in manufacturing. For instance, clustering can be used to group manufactured inventory parts according to different features [134] (Know-what). The obtained clusters can be used as a guideline for warehouse space optimization [135]. Clustering can be used for anomaly detection [136] (Know-what) and process optimization [137] (Know-how), and can be used in conjunction with other techniques to improve the interpretability of results.

Association rule-based learning [121]: Association rule-based learning is an unsupervised data-mining technique that finds important interactions among variables in a dataset. It is capable of identifying hidden correlations in datasets by measuring degrees of similarity. Hence, association rule-based learning is suitable in the Know-why stage in manufacturing. For instance, association rule-based learning can be utilized to accurately depict the relationship between quantifiable shop floor indicators and appropriate causes of action under various conditions of machine utilization (Know-why, system level), which can be used to establish an appropriate management strategy [138].

4.3. Semi-Supervised Learning Methods

Unsupervised learning methods do not have any input guidance during training, which reduces labeling costs; however, their performance is normally less accurate. Therefore, semi-supervised learning methods can be used to take advantage of the accuracy achieved by supervised learning while limiting costs thanks to the reduction in labeling effort. Therefore, researchers have turned to data augmentation [139,140] to enlarge dataset, with the inputs and labels generated massively based on the existing dataset in a controlled way while incurring no extra cost in the labeling phase. Taking an image with its label as an example, it can be enriched by basic transformations such as rotation, translation, flipping, noise injection, etc. It can be enriched by adversarial data augmentation, such as by generating synthetic dataset using generative models, e.g., Generative Adversarial Network (GAN) [141] and Variational AutoEncoder (VAE) [142], thereby obtaining new images for training ML models at low cost. However, the improvements obtainable with data augmentation are limited, and more real data are better than more synthetic data [143]. Therefore, increasing attention is being paid to the combination of supervised learning and unsupervised learning, namely, semi-supervised learning, in which both unlabeled data and labeled data are leveraged during training.

Semi-supervised learning methods can be generally divided into two groups: data augmentation-based methods and semi-supervised mechanism-based methods. An overview of semi-supervised methods is provided in Figure 12.

Data augmentation: through data augmentation, labeled data can be enlarged and augmented by adding model predictions of newly unlabeled data with high confidence as pseudo-labels, as shown in Figure 13. However, the model continues to be run in a fully supervised manner. In addition, the quality of the pseudo-labels can highly affect model performance, and incorrect pseudo-labels with high confidence are inevitable due to their nature. To improve the quality of pseudo-labels, there are hybrid methods combining pseudo-labels and consistency regularization, such as MixMatch [144] and FixMatch [145]. Nevertheless, data augmentation-based methods are simple, and there is no need to carefully design the loss. Therefore, data augmentation-based methods can be potentially useful for non-experts in manufacturing for enlarging labeled dataset when it is easy to collect massive amounts of unlabeled data.

Semi-supervised mechanisms: by contrast, semi-supervised mechanism-based methods are more focused on the mechanism of utilizing both labeled data and unlabeled data. The principle of semi-supervised mechanisms is illustrated in Figure 14, where both labeled data and unlabeled data can be model inputs while their losses are calculated in a different way. Semi-supervised mechanism-based methods can be further categorized into consistency-based methods, graph-based methods, and generative-based methods.

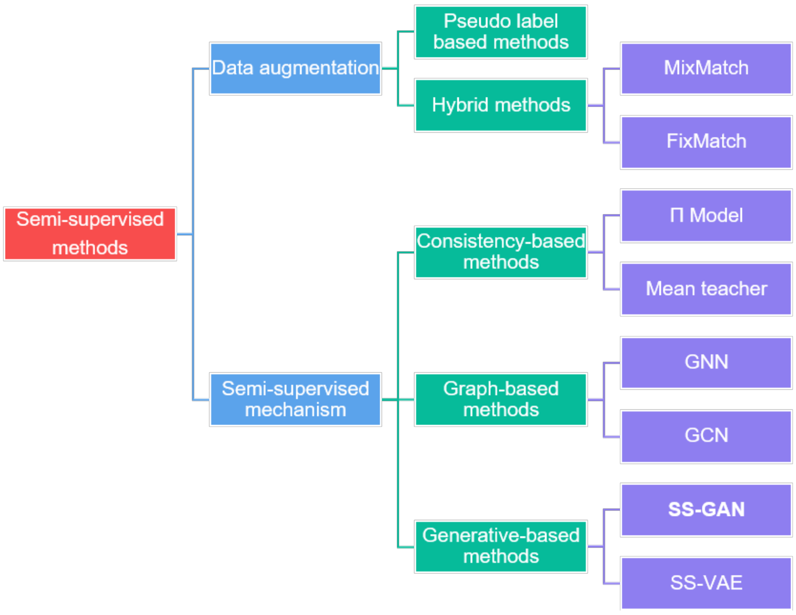


Figure 12. Overview of semi-supervised methods.

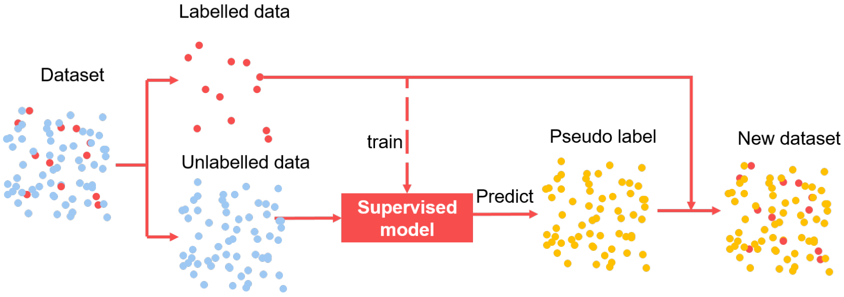


Figure 13. Data augmentation-based methods.

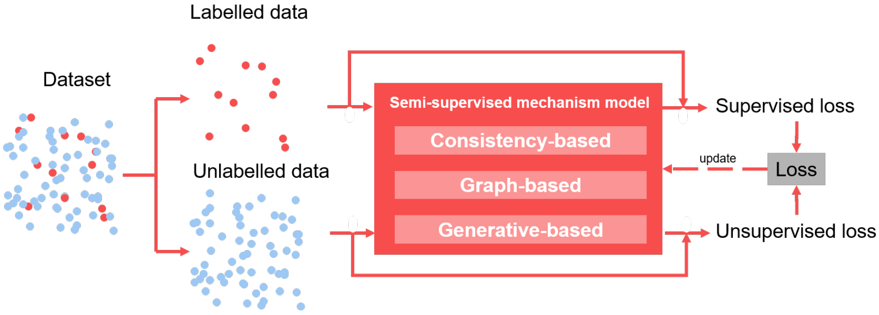


Figure 14. Semi-supervised mechanism-based methods.

Consistency-based methods take advantage of the consistency of model outputs after perturbations [146]; therefore, consistency regularization can be applied for unlabeled data. Consistency constraint can be either imposed between the predictions from perturbed inputs from the same sample, for instance, the π model [147], or between the predictions from two models with the same architecture, such as MeanTeacher [148]. Thanks to the perturbations in consistency-based methods, model generalization can be enhanced [149]. In terms of applications in manufacturing, depending on the output values consistency-based methods can be used in the Know-what and Know-when stages. For instance, consistency-based methods can be utilized in quality monitoring based on images (Know-what, product level).

Graph-based methods aim to establish a graph from a dataset by denoting each data point as a node, with the edge connecting two nodes representing the similarity between them. Label propagation is then performed on the established graph, with the information from labeled data used to infer the labels of the unlabeled data. Graph-based methods result in the connected nodes being closer in the feature space, while disconnected nodes repel each other. Therefore, graph-based methods can be used to address the problem of poor class separation due to intra-class variations and inter-class similarities [18]. Consequently, graph-based methods can be potentially useful for defect classification [18] (Know-what, product level) or machine health state monitoring [150] (Know-what, machine level) where there are problems with insufficient label information or poor class separation. However, it should be noted that graph-based methods are normally transductive methods, meaning that the constructed graph is only valid for the trained data and rebuilding the graph is necessary when it comes to new data. Typical examples of graph-based methods include Graph Neural Networks (GNNs) [151] and Graph Convolution Networks (GCNs) [152].

The main point of generative-based methods is to learn patterns from a dataset and to model data distributions, allowing the model to be used to generate new samples. Then during training, the model can be updated using the combination of the supervised loss (for existing data with labels) and unsupervised loss (for synthetic data). An inherent advantage of generative-based methods is that the labeled data can be enriched by a trained model which has learned the data distribution. Therefore, generative-based methods are well-suited for situations where it is difficult to collect labeled data, such as process fault detection [153] (Know-what, process level) and anomaly detection in machinery [154] (Know-what, machine level). Examples include the semi-supervised GAN series (SS-GANs), such as Categorical Generative Adversarial Network (CatGAN) [155], Improved GAN [156], and semi-supervised VAEs (SS-VAEs) [157].

Table A3 lists semi-supervised applications in manufacturing taken from the selected documents in Section 2.2.

4.4. Reinforcement Learning Methods

Reinforcement Learning (RL) algorithms consist of two elements, namely, an ***agent*** acting within an ***environment*** (see Figure 15). The agent is acting, and is therefore subject to the desired learning process by directly interacting with and manipulating the environment. Based on [158], the procedure of a learning cycle is as follows: first, the agent is presented with an observation of the environment state $s_t \in \mathbb{S}$; then, based on this observation (along with internal decision making), the selection of an action $a_t \in \mathbb{A}$. $\mathbb{S}$ refers to the state space, that is, the set of possible observations that could occur in the environment. The observation has to provide sufficient information on the current environment or system state in order for the agent to select actions in an ideal way to solve the control problem. For selecting the action, $\mathbb{A}$ refers to the action space, that is, the set of possible actions chosen by the agent. After a_t is performed (in a given state s_t), the environment moves to the resulting state s_{t+1} and the agent receives a reward r_{t+1}. Then, the reinforcement learning cycle continues to iterate as shown in Figure 15. The agent aims to maximize the (discounted) long-term cumulative reward by improving the selection of actions towards an optimum. In other words, the RL agent wants to learn an optimal control policy for the environment.

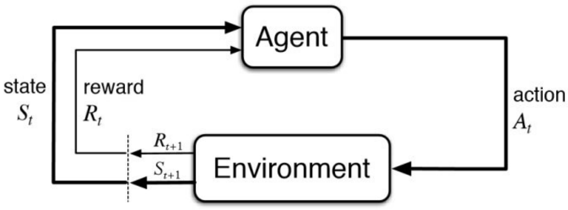


Figure 15. Overview of the Reinforcement Learning approach based on [158].

In general, RL approaches can be split into model-based, i.e., the agent has an internal model of how the environment works, and model-free. The latter is most common thanks to the advent of deep learning, and simplifies application, as feature selection can be applied. Model-free approaches themselves can be divided into short value-based or policy-based approaches by their approach to storing state-action value pairs, which are used to select the action for optimal value return; the latter directly optimize the action selection policy. In contrast to the other machine learning techniques, RL does not require large dataset, only a clearly specified environment. Typically, an RL agent is trained on a simulation or digital twin model [159]; after successful training, it can be implemented on the Know-how level for its original purpose. Otherwise, the agent starts with random non-optimal actions, leading to undesired system behavior.

Considering the aim of achieving the Know-how level for autonomous control in processes, machines, or systems, RL is extremely important for applications in future production. In addition, multi-agent RL is becoming of interest to the research community [33], and can even be applied for controlling products [160]. However, RL remains under-exploited in the industrial area, especially in respect to other machine learning techniques [161].

As of now, applied approaches can be summarized as shown in Table A4. Note that the applications reviewed here are implemented in a simulation or digital twin [159], and features are manually crafted from raw data.

5. Challenges and Future Directions

A large number of ML use cases have shown the great potential for addressing complex manufacturing problems, from knowing what is happening to knowing how employ self-adapting or self-optimizing systems. The data-driven mechanisms in ML enable broader applications in different fields as well as at different levels, from individual products to whole systems. However, in spite of the great potential and advantages offered by ML and numerous off-the-shelf ML models, there are critical challenges to overcome before the successful application of ML in manufacturing can be realized. The following demonstrate typical challenges that manufacturing industries might confront during the application and deployment of ML-based solutions, along with corresponding future directions for tackling these challenges (RQ3).

- *Lack of data*. Preparing the data used for ML is not a simple task, as the scale and the quality of data can greatly affect the performance of ML models. The most common challenge involves preparing a large amount of organized input data, and ensuring high-quality labels if labels are needed. Despite manufacturing data becoming increasingly more accessible due to the development of sensors and the Internet of Things, gathering meaningful data is time-consuming and costly in many cases, for example, fault detection and RUL prediction. This issue might be alleviated by the Synthetic Minority Over-sampling Technique (SMOTE) [162]. However, SMOTE cannot capture complex representative data, as it often relies on interpolation [163]. Data augmentation [139,164] or transfer learning [165] may address this problem. The aim of data augmentation is to enlarge dataset by means of transforming data [139], by transforming both data and labels, as with MixUp [166], or by generating synthetic data using generative models [167,168]. On the contrary, instead of focusing on expanding data, transfer learning aims to leverage knowledge from similar external datasets. A typically used method in transfer learning is parameter transfer, where a pretrained model from a similar dataset is employed for initialization [165]. Another situation involving lack of data is that certain data cannot be shared due to data privacy and security issues. In confronting this problem, Federated Learning (FL) [169] might be a potential opportunity to enable model training across multiple decentralized devices while holding local data privately.

- *Limited computing resources*. The high performance of ML models always comes with high computational complexity. In particular, obtaining high accuracy with a neural network requires on millions or even billions of parameters [170]. However, limited computing resources in industries makes it a challenge to deploy heavy ML models in real-time industrial environments. Possible approaches include model compression via pruning and sharing of model parameters [171] and knowledge distillation [172]. Parameter pruning aims to reduce the number of model parameters by removing redundant parameters without any effect on model performance. By contrast, seeking the same goal, knowledge distillation focuses on distilling knowledge from a cumbersome neural network to a lightweight network to allow it to be deployed more easily with limited computing resources.
- *Changing circumstances*. Most ML applications in manufacturing focus only on model development and verification in off-line environments. However, when deploying these models in running production, their performance may be degraded due to changing circumstances, leading to changes in data distribution, that is, drift [173,174]. Therefore, manual model adjustment over time, which is time-consuming, is usually unavoidable [175]. However, this could be addressed in the future by automatic model adaption [174], in which data drifts are automatically detected and handled with less resources.
- *Interpretability of results*. Many expectations have been placed on ML to overcome all types of problems without the need for prior knowledge. In particular, ML models are expected to directly learn higher level knowledge such as Know-when and Know-how, which is difficult for human beings to obtain in manufacturing. However, without the foundations of early-stage knowledge and an understanding of the data, the results inferred from big data by black-box ML models are meaningless and unreliable. For instance, predictions blindly obtained from all data, including both relevant and irrelevant data, might even degrade performance due to the GIGO (garbage in, garbage out) phenomenon [176]. To overcome this problem, future directions within ML development might include incorporating physical models into ML models [177] or obtaining Four-know knowledge successively.
- *Uncertainty of results*. Related to the challenge of interpretability is the challenge of uncertain results. The success of manufacturing depends heavily on the quality of the resulting products. As every manufacturing process has a degree of variability, almost all industrial manufacturers use statistical process control (SPC) to ensure a stable and defined quality of products [178]. A central element of statistical process control is the determination and handling of statistical uncertainty. The uncertainty of ML results often cannot be quantified reliably and efficiently, even with today's state-of-the-art [179–181]. Furthermore, model complexity and severe non-linearity in ML can hinder the evaluation of uncertainty [182]. Although there are promising approaches, e.g., Gaussian mixture models for NN [183,184] and Probabilistic Neural Network (PNN) [184], or the use of Baysian Networks [180], there are several limitations limiting potential applications, such as high computational cost and simplified assumptions [184]. Therefore, future research needs to make progress on the general theory of integrating uncertainty into ML methods to allow manufacturing in order to ensure high quality and stability in production.

To summarize, while ML is a fairly open tool which can be used to handle a variety of problems in manufacturing, it is necessary to have an understanding of the hidden challenges in ML application in order to provide more realistic and robust outcomes. For instance, early in ML application in manufacturing, one might face the problem of lacking data. During the deployment of ML-based solutions, one might confront challenges around integrating the solution into the industrial environment. After deployment, one might encounter the challenge of evaluating ML results on product and process in terms of interpretability and uncertainty. The future directions pointed out in this review can help to address the above-mentioned challenges and ensure reliable improvements in manufacturing contexts.

6. Conclusions

It is fully recognized that ML is playing an increasingly critical role in the digitization of manufacturing industries towards Industry 4.0, leading to improved quality, productivity, and efficiency. This review has paper aimed to address the issue of how ML can improve manufacturing, posing three research questions related to the above issue in the introduction. To address these research questions, we carried out a literature review assessing the state-of-the-art based on 1348 published scientific articles.

To answer RQ1, we first introduced the concepts of the 'Four-Know' (Know-what, Know-why, Know-when, Know-how) and 'Four-Level' (Product, Process, Machine, System) categories to help formulate ML tasks in manufacturing. By mapping ML use cases into the Four-Know and Four-Level matrix, we provide an understanding of typical ML use cases and their potential benefits for improving manufacturing. To further support RQ1, the identified ML studies were classified using the 'Four-Know' and 'Four-Level' perspective to provide an overview of ML publications in manufacturing. The results showed that current ML applications are mainly focused on the product level, in particular in terms of Know-what and Know-when. In addition, considerable growth in Know-how was observed at the process and system levels, which might be correlated to higher demand for adaptability to changes on these levels.

To fill the gap between academic research and manufacturing industries, we provided an actionable pipeline for the implementation of ML solutions by production engineers from ideation through to deployment, thereby answering RQ2. To further explain the 'model training' step, which is the core stage in the pipeline, a holistic review of ML methods was provided, including supervised, semi-supervised, unsupervised, and reinforcement learning methods along with their typical applications in manufacturing. We hope that this can provide support in method selection for decision-makers considering ML solutions.

Finally, to answer RQ3, we uncovered the current challenges that manufacturing industry is likely to encounter during application and deployment, and provided possible future directions for tackling these challenges as possible developments for ensuring more reliable and robust outcomes in manufacturing.

Author Contributions: Conceptualization, T.C., O.J.J., M.C.M., V.S. and G.F.; methodology, T.C. and S.S.; formal analysis, T.C. and S.S.; writing—original draft preparation, T.C., V.S., S.S., M.C.M., O.J.J. and F.S.; writing—review and editing, T.C., V.S., M.C.M., S.S., O.J.J., M.C., G.F., G.T., J.J.A.M. and F.S.; supervision, M.C., G.F., G.T., J.J.A.M. and F.S.; funding acquisition, G.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by a European Training Network supported by Horizon 2020, grant number 814225.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This research work was undertaken in the context of the DIGIMAN4.0 project ("DIGItal MANufacturing Technologies for Zero-defect Industry 4.0 Production", https://www.digiman4-0.mek.dtu.dk/, accessed on 1 January 2023). DIGIMAN4.0 is a European Training Network supported by Horizon 2020, the EU Framework Programme for Research and Innovation (Project ID: 814225).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Categories of supervised learning applications.

Ref.	Year	Level	Know-What	Know-Why	Know-When	Know-How	Data Type	Method Type	Case	Field
[104]	2018	Product	✓				Image	Kernel	Defect detection	Metallic powder bed fusion
[185]	2018	Product	✓				Tabular	Kernel	Product monitoring	Metal frame process in mobile device manufacturing
[186]	2020	Process	✓	✓	✓		Time series, Image	Kernel	Temperature prediction and potential anomaly detection	Additive manufacturing
[19]	2022	Product, Process	✓	✓			Tabular	Kernel	Fault detection and classification	Semiconductor Etch Equipment
[105]	2022	Product			✓		Tabular	Kernel	Quality prediction	Additive manufacturing
[106]	2022	Machine			✓		Time series	Kernel	Wear prediction	Metal forming
[187]	2022	System		✓	✓		Time series	Kernel	Content prediction	Steel making
[114]	2018	Machine	✓	✓			Time series (Image)	NN	Fault diagnosis	Motor bearning and pump
[188]	2020	System		✓	✓		Image	NN	Cost estimation	
[112]	2020	Product	✓				Image	NN	Defect detection	Battery manufacturing
[189]	2022	Product	✓				Time series	NN	Quality assurance	Fused deposition modeling
[190]	2022	Process	✓				Time series	NN	Process optimization	Wire arc additive manufacturing
[191]	2022	Process			✓	✓	Tabular	NN	Parameter optimization	Laser powder bed fusion
[192]	2022	Process	✓				Image	NN	Object detection	Robotic grasp
[193]	2022	Product	✓				Image	NN, kernel	Defect detection	Roller manufacturing
[194]	2022	Machine			✓		Time series (Image)	NN, kernel	Tool condition monitoring	Machining
[195]	2019	Product			✓		Time series	Tree	Material removal prediction	Robotic grinding
[196]	2022	Product			✓		Image (Tabular)	Tree	Porosity prediction	Powder-bed additive manufacturing
[197]	2019	Product	✓				Image	Probabilistic	Online quality inspection	Powder-bed additive manufacturing
[198]	2018	System				✓	Tabular	Hybrid	Scheduling	Flexible Manufacturing Systems (FMSs)

Table A2. Categories of unsupervised learning applications.

Ref.	Year	Level	Know-What	Know-Why	Know-When	Know-How	Data Type	Method Type	Case	Field
[120]	2021	Machine	✓				Time series	Clustering	Tool Condition clustering	Autonomous manufacturing
[131]	2021	Machine	✓				Time series	Clustering	Tool health monitoring	Machine tool health Monitoring
[132]	2019	Machine	✓	✓			Time series	Clustering	Defect Identification	Manufacturing systems
[199]	2021	Process	✓				Tabular, Time series	Clustering	Condition monitoring	Manufacturing Condition monitoring
[133]	2020	System	✓				Time series	Clustering	Condition monitoring	Manufacturing Condition monitoring
[125]	2018	Product	✓				Image, Text	Autoencoder	Defect Identification	Fabric industry
[119]	2019	Product	✓				Image	Autoencoder	Defect Identification	Automatic Optical Inspection
[200]	2021	Product	✓				Image	Autoencoder	Defect Identification	Printed circuit board manufacturing
[201]	2022	Machine	✓		✓		Tabular, Time series	Autoencoder	Anomaly detection	Steel rolling Process
[126]	2022	Process	✓		✓		Image	Autoencoder	Anomaly detection	Industrial Anomaly detection
[126]	2022	Process	✓		✓		Image, Text	Autoencoder	Anomaly detection	Semi conductor manufacturing
[202]	2022	Machine			✓		Time series	PCA	Predictive maintenance	Fan-motor system
[118]	2022	Machine	✓	✓			Time series	PCA	Anomaly detection	Programmable logic controllers
[121]	2015	Process	✓				Tabular	Association rule	Predictive maintenance	Wooden door manufacturing

Table A3. Categories of semi-supervised learning applications.

Ref.	Year	Level	Know-What	Know-Why	Know-When	Know-How	Data Type	Method Type	Case	Field
[203]	2020	Product	✓				Image	Data augmentation	Quality control	Automated Surface Inspection
[204]	2021	Process	✓				Image	Data augmentation	Measurement in process	Positioning of welding seams
[205]	2019	System			✓		Tabular	Data augmentation	Energy consumption modelling	Steel industry
[206]	2020	Product, System			✓		Time series	Data augmentation	Quality prediction	Continuous-flow manufacturing.
[207]	2021	Machine	✓				Time series	Consistency-based	Predictive quality control	Semiconductor manufacturing
[149]	2020	Product	✓				Image	Consistency-based	Quality monitoring	Metal additive manufacturing
[18]	2021	Product	✓				Image	Graph-based	Quality control	Automated Surface Inspection
[150]	2022	Machine	✓	✓			Time series	Graph-based	Machine health state diagnosis	Manipulator
[208]	2022	Machine		✓	✓		Tabular	Graph-based	Predict tool tip dynamics	Machine tool
[209]	2021	Product			✓		Image	Generative-based	Assessing manufacturability of cellular structures	Direct metal laser sintering process
[210]	2019	Product	✓				Time series	Generative-based	Quality inferred from process	laser powder-bed fusion
[211]	2020	Product	✓				Image	Generative-based	Quality diagnosis	Wafer fabrication
[212]	2021	Product	✓				Image	Generative-based	Quality control	Automated Surface Inspection
[213]	2020	Machine			✓		Time series	Generative-based	Remaining useful life prognostics	Turbofan engine and rolling bearing
[214]	2021	Machine	✓	✓			Tabular	Generative-based	Machine condition monitoring	Vacuum system in styrene petrochemical plant
[153]	2021	Machine	✓	✓			Time series	Generative-based	Anomaly detection for predictive maintenance	Press machine
[154]	2022	Process	✓				Time series (image)	Generative-based	Process fault detection	Die casting process

Table A4. Categories of reinforcement learning applications.

Ref.	Year	Level	Know-What	Know-Why	Know-When	Know-How	Data Type	Method Type	Case	Field
[32]	2021	Process	✓			✓	Tabular	Value-based	Quality control	Statistical Process Control
[215]	2022	System			✓	✓	Tabular	Value-based	Scheduling	Semiconductor fab
[216]	2021	System				✓	Tabular	Value-based	Throughput control	Flow shop
[217]	2021	Machine				✓	Tabular	Value-based	Scheduling & Maintenance	Multi-state single machine
[34]	2020	System				✓	Tabular	Value-based	Quality Control & Maintenance	Production system
[218]	2022	System				✓	Tabular	Value-based	Lead time management	Flow shop
[219]	2020	Process				✓	Tabular	Value-based	Robotic arm control	Soft fabric manufacturing
[220]	2021	System			✓	✓	Tabular	Value-based	Layout planning	Greenfield factories
[221]	2020	Machine				✓	Tabular	Value-based	Maintenance scheduling	Preventive maintenance
[33]	2022	Machine			✓	✓	Tabular	Policy-based	Maintenance scheduling	Parallel machines
[222]	2021	Process				✓	Tabular	Policy-based	Improving efficiency	Automated product disassembly
[223]	2021	System				✓	Tabular	Policy-based	Dispatching	Job shop
[224]	2022	System				✓	Tabular	Policy-based	Scheduling & maintenance	Semiconductor fab
[225]	2022	System				✓	Tabular	Policy-based	Yield optimization	Multi-agent RL
[57]	2022	System				✓	Tabular	Policy-based	Human Worker Control	Flow shop
[59]	2022	System				✓	Tabular	Policy-based	Scheduling & dispatching	Disassembly job shop
[160]	2021	Product				✓	Tabular	Policy-based	Multi-agent production control	Job shop
[226]	2022	Process				✓	Tabular	Both	Parameter optimisation	Manufacturing processes
[227]	2019	Process				✓	Tabular	Both	Online parameter optimisation	Injection molding
[228]	2022	System				✓	Tabular	Both	scheduling	Matrix production system

References

1. Abele, E.; Reinhart, G. *Zukunft der Produktion: Herausforderungen, Forschungsfelder, Chancen*; Hanser: München, Germany, 2011.
2. Zizic, M.C.; Mladineo, M.; Gjeldum, N.; Celent, L. From industry 4.0 towards industry 5.0: A review and analysis of paradigm shift for the people, organization and technology. *Energies* **2022**, *15*, 5221. [CrossRef]
3. Huang, S.; Wang, B.; Li, X.; Zheng, P.; Mourtzis, D.; Wang, L. Industry 5.0 and Society 5.0—Comparison, complementation and co-evolution. *J. Manuf. Syst.* **2022**, *64*, 424–428. [CrossRef]
4. Vukovic, M.; Mazzei, D.; Chessa, S.; Fantoni, G. Digital Twins in Industrial IoT: A survey of the state of the art and of relevant standards. In Proceedings of the 2021 IEEE International Conference on Communications Workshops (ICC Workshops), Montreal, QC, Canada, 14–23 June 2021. [CrossRef]
5. Mourtzis, D.; Fotia, S.; Boli, N.; Vlachou, E. Modelling and quantification of industry 4.0 manufacturing complexity based on information theory: A robotics case study. *Int. J. Prod. Res.* **2019**, *57*, 6908–6921. [CrossRef]
6. Galin, R.; Meshcheryakov, R.; Kamesheva, S.; Samoshina, A. Cobots and the benefits of their implementation in intelligent manufacturing. *IOP Conf. Ser. Mater. Sci. Eng.* **2020**, *862*, 032075. [CrossRef]
7. May, M.C.; Schmidt, S.; Kuhnle, A.; Stricker, N.; Lanza, G. Product Generation Module: Automated Production Planning for optimized workload and increased efficiency in Matrix Production Systems. *Procedia CIRP* **2020**, *96*, 45–50. [CrossRef]
8. Lu, Y. Industry 4.0: A survey on technologies, applications and open research issues. *J. Ind. Inf. Integr.* **2017**, *6*, 1–10. [CrossRef]
9. Miqueo, A.; Torralba, M.; Yagüe-Fabra, J.A. Lean manual assembly 4.0: A systematic review. *Appl. Sci.* **2020**, *10*, 8555. [CrossRef]
10. Wuest, T.; Weimer, D.; Irgens, C.; Thoben, K.D. Machine learning in manufacturing: Advantages, challenges, and applications. *Prod. Manuf. Res.* **2016**, *4*, 23–45. [CrossRef]
11. Rai, R.; Tiwari, M.K.; Ivanov, D.; Dolgui, A. Machine learning in manufacturing and industry 4.0 applications. *Int. J. Prod. Res.* **2021**, *59*, 4773–4778. [CrossRef]
12. Bertolini, M.; Mezzogori, D.; Neroni, M.; Zammori, F. Machine Learning for industrial applications: A comprehensive literature review. *Expert Syst. Appl.* **2021**, *175*, 114820. [CrossRef]
13. Wang, J.; Ma, Y.; Zhang, L.; Gao, R.X.; Wu, D. Deep learning for smart manufacturing : Methods and applications. *J. Manuf. Syst.* **2018**, *48*, 144–156. [CrossRef]
14. Dogan, A.; Birant, D. Machine learning and data mining in manufacturing. *Expert Syst. Appl.* **2021**, *166*, 114060. [CrossRef]
15. Alshangiti, M.; Sapkota, H.; Murukannaiah, P.K.; Liu, X.; Yu, Q. Why is developing machine learning applications challenging? a study on stack overflow posts. In Proceedings of the 2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), Porto de Galinhas, Brazil, 19–20 September 2019; pp. 1–11.
16. Zeller, V.; Hocken, C.; Stich, V. Acatech Industrie 4.0 maturity index—A multidimensional maturity model. In Proceedings of the IFIP International Conference on Advances in Production Management Systems, Seoul, Republic of Korea, 26–30 August 2018; Springer: Cham, Switzerland, 2018; pp. 105–113.
17. Yang, L.; Fan, J.; Huo, B.; Li, E.; Liu, Y. A nondestructive automatic defect detection method with pixelwise segmentation. *Knowl.-Based Syst.* **2022**, *242*, 108338. [CrossRef]
18. Wang, Y.; Gao, L.; Gao, Y.; Li, X. A new graph-based semi-supervised method for surface defect classification. *Robot. Comput. Integr. Manuf.* **2021**, *68*, 102083. [CrossRef]
19. Kim, S.H.; Kim, C.Y.; Seol, D.H.; Choi, J.E.; Hong, S.J. Machine Learning-Based Process-Level Fault Detection and Part-Level Fault Classification in Semiconductor Etch Equipment. *IEEE Trans. Semicond. Manuf.* **2022**, *35*, 174–185. [CrossRef]
20. Peng, S.; Feng, Q.M. Reinforcement learning with Gaussian processes for condition-based maintenance. *Comput. Ind. Eng.* **2021**, *158*, 107321. [CrossRef]
21. Zheng, W.; Liu, Y.; Gao, Z.; Yang, J. Just-in-time semi-supervised soft sensor for quality prediction in industrial rubber mixers. *Chemom. Intell. Lab. Syst.* **2018**, *180*, 36–41. [CrossRef]
22. Kang, P.; Kim, D.; Cho, S. Semi-supervised support vector regression based on self-training with label uncertainty: An application to virtual metrology in semiconductor manufacturing. *Expert Syst. Appl.* **2016**, *51*, 85–106. [CrossRef]
23. Srivastava, A.K.; Patra, P.K.; Jha, R. AHSS applications in Industry 4.0: Determination of optimum processing parameters during coiling process through unsupervised machine learning approach. *Mater. Today Commun.* **2022**, *31*, 103625. [CrossRef]
24. Antomarioni, S.; Ciarapica, F.E.; Bevilacqua, M. Association rules and social network analysis for supporting failure mode effects and criticality analysis : Framework development and insights from an onshore platform. *Saf. Sci.* **2022**, *150*, 105711. [CrossRef]
25. Pan, R.; Li, X.; Chakrabarty, K. Semi-Supervised Root-Cause Analysis with Co-Training for Integrated Systems. In Proceedings of the 2022 IEEE 40th VLSI Test Symposium (VTS), San Diego, CA, USA, 25–27 April 2022. [CrossRef]
26. Chen, R.; Lu, Y.; Witherell, P.; Simpson, T.W.; Kumara, S.; Yang, H. Ontology-Driven Learning of Bayesian Network for Causal Inference and Quality Assurance in Additive Manufacturing. *IEEE Robot. Autom. Lett.* **2021**, *6*, 6032–6038. [CrossRef]
27. Sikder, S.; Mukherjee, I.; Panja, S.C. A synergistic Mahalanobis–Taguchi system and support vector regression based predictive multivariate manufacturing process quality control approach. *J. Manuf. Syst.* **2020**, *57*, 323–337. [CrossRef]
28. Cerquitelli, T.; Ventura, F.; Apiletti, D.; Baralis, E.; Macii, E.; Poncino, M. Enhancing manufacturing intelligence through an unsupervised data-driven methodology for cyclic industrial processes. *Expert Syst. Appl.* **2021**, *182*, 115269. [CrossRef]
29. Kolokas, N.; Vafeiadis, T.; Ioannidis, D.; Tzovaras, D. A generic fault prognostics algorithm for manufacturing industries using unsupervised machine learning classifiers. *Simul. Model. Pract. Theory* **2020**, *103*, 102109. [CrossRef]

30. Verstraete, D.; Droguett, E.; Modarres, M. A deep adversarial approach based on multisensor fusion for remaining useful life prognostics. In Proceedings of the 29th European Safety and Reliability Conference (ESREL 2019), Hannover, Germany, 22–26 September 2020; pp. 1072–1077. [CrossRef]
31. Wu, D.; Jennings, C.; Terpenny, J.; Gao, R.X.; Kumara, S. A Comparative Study on Machine Learning Algorithms for Smart Manufacturing: Tool Wear Prediction Using Random Forests. *J. Manuf. Sci. Eng. Trans. ASME* **2017**, *139*, 071018. [CrossRef]
32. Viharos, Z.J.; Jakab, R. Reinforcement Learning for Statistical Process Control in Manufacturing. *Meas. J. Int. Meas. Confed.* **2021**, *182*, 109616. [CrossRef]
33. Luis, M.; Rodríguez, R.; Kubler, S.; Giorgio, A.D.; Cordy, M.; Robert, J.; Le, Y. Multi-agent deep reinforcement learning based Predictive Maintenance on parallel machines. *Robot. Comput. Integr. Manuf.* **2022**, *78*, 102406.
34. Paraschos, P.D.; Koulinas, G.K.; Koulouriotis, D.E. Reinforcement learning for combined production-maintenance and quality control of a manufacturing system with deterioration failures. *J. Manuf. Syst.* **2020**, *56*, 470–483. [CrossRef]
35. Liu, Y.H.; Huang, H.P.; Lin, Y.S. Dynamic scheduling of flexible manufacturing system using support vector machines. In Proceedings of the 2005 IEEE Conference on Automation Science and Engineering, IEEE-CASE 2005, Edmonton, AB, Canada, 1–2 August 2005; Volume 2005, pp. 387–392. [CrossRef]
36. Zhou, G.; Chen, Z.; Zhang, C.; Chang, F. An adaptive ensemble deep forest based dynamic scheduling strategy for low carbon flexible job shop under recessive disturbance. *J. Clean. Prod.* **2022**, *337*, 130541. [CrossRef]
37. de la Rosa, F.L.; Gómez-Sirvent, J.L.; Sánchez-Reolid, R.; Morales, R.; Fernández-Caballero, A. Geometric transformation-based data augmentation on defect classification of segmented images of semiconductor materials using a ResNet50 convolutional neural network. *Expert Syst. Appl.* **2022**, *206*, 117731. [CrossRef]
38. Krahe, C.; Marinov, M.; Schmutz, T.; Hermann, Y.; Bonny, M.; May, M.; Lanza, G. AI based geometric similarity search supporting component reuse in engineering design. *Procedia CIRP* **2022**, *109*, 275–280. [CrossRef]
39. Onler, R.; Koca, A.S.; Kirim, B.; Soylemez, E. Multi-objective optimization of binder jet additive manufacturing of Co-Cr-Mo using machine learning. *Int. J. Adv. Manuf. Technol.* **2022**, *119*, 1091–1108. [CrossRef]
40. Jadidi, A.; Mi, Y.; Sikström, F.; Nilsen, M.; Ancona, A. Beam Offset Detection in Laser Stake Welding of Tee Joints Using Machine Learning and Spectrometer Measurements. *Sensors* **2022**, *22*, 3881. [CrossRef]
41. Sanchez, S.; Rengasamy, D.; Hyde, C.J.; Figueredo, G.P.; Rothwell, B. Machine learning to determine the main factors affecting creep rates in laser powder bed fusion. *J. Intell. Manuf.* **2021**, *32*, 2353–2373. [CrossRef]
42. Verma, S.; Misra, J.P.; Popli, D. Modeling of friction stir welding of aviation grade aluminium alloy using machine learning approaches. *Int. J. Model. Simul.* **2022**, *42*, 1–8. [CrossRef]
43. Gerling, A.; Ziekow, H.; Hess, A.; Schreier, U.; Seiffer, C.; Abdeslam, D.O. Comparison of algorithms for error prediction in manufacturing with automl and a cost-based metric. *J. Intell. Manuf.* **2022**, *33*, 555–573. [CrossRef]
44. Akbari, P.; Ogoke, F.; Kao, N.Y.; Meidani, K.; Yeh, C.Y.; Lee, W.; Farimani, A.B. MeltpoolNet: Melt pool characteristic prediction in Metal Additive Manufacturing using machine learning. *Addit. Manuf.* **2022**, *55*, 102817. [CrossRef]
45. Dittrich, M.A.; Uhlich, F.; Denkena, B. Self-optimizing tool path generation for 5-axis machining processes. *CIRP J. Manuf. Sci. Technol.* **2019**, *24*, 49–54. [CrossRef]
46. Xi, Z. Model predictive control of melt pool size for the laser powder bed fusion process under process uncertainty. *ASCE-ASME J. Risk Uncertain. Eng. Syst. Part B Mech. Eng.* **2022**, *8*, 011103. [CrossRef]
47. Li, X.; Liu, X.; Yue, C.; Liu, S.; Zhang, B.; Li, R.; Liang, S.Y.; Wang, L. A data-driven approach for tool wear recognition and quantitative prediction based on radar map feature fusion. *Measurement* **2021**, *185*, 110072. [CrossRef]
48. Xia, B.; Wang, K.; Xu, A.; Zeng, P.; Yang, N.; Li, B. Intelligent Fault Diagnosis for Bearings of Industrial Robot Joints Under Varying Working Conditions Based on Deep Adversarial Domain Adaptation. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–13. [CrossRef]
49. May, M.C.; Neidhöfer, J.; Körner, T.; Schäfer, L.; Lanza, G. Applying Natural Language Processing in Manufacturing. *Procedia CIRP* **2022**, *115*, 184–189. [CrossRef]
50. Xu, X.; Li, X.; Ming, W.; Chen, M. A novel multi-scale CNN and attention mechanism method with multi-sensor signal for remaining useful life prediction. *Comput. Ind. Eng.* **2022**, *169*, 108204. [CrossRef]
51. Shah, M.; Vakharia, V.; Chaudhari, R.; Vora, J.; Pimenov, D.Y.; Giasin, K. Tool wear prediction in face milling of stainless steel using singular generative adversarial network and LSTM deep learning models. *Int. J. Adv. Manuf. Technol.* **2022**, *121*, 723–736. [CrossRef]
52. Verl, A.; Steinle, L. Adaptive compensation of the transmission errors in rack-and-pinion drives. *CIRP Ann.* **2022**, *71*, 345–348. [CrossRef]
53. Frigerio, N.; Cornaggia, C.F.; Matta, A. An adaptive policy for on-line Energy-Efficient Control of machine tools under throughput constraint. *J. Clean. Prod.* **2021**, *287*, 125367. [CrossRef]
54. Bozcan, I.; Korndorfer, C.; Madsen, M.W.; Kayacan, E. Score-Based Anomaly Detection for Smart Manufacturing Systems. *IEEE/ASME Trans. Mechatron.* **2022**, *27*, 5233–5242. [CrossRef]
55. Bokrantz, J.; Skoogh, A.; Nawcki, M.; Ito, A.; Hagstr, M.; Gandhi, K.; Bergsj, D. Improved root cause analysis supporting resilient production systems. *J. Manuf. Syst.* **2022**, *64*, 468–478. [CrossRef]
56. Long, T.; Li, Y.; Chen, J. Productivity prediction in aircraft final assembly lines: Comparisons and insights in different productivity ranges. *J. Manuf. Syst.* **2022**, *62*, 377–389. [CrossRef]

57. Overbeck, L.; Hugues, A.; May, M.C.; Kuhnle, A.; Lanza, G. Reinforcement Learning Based Production Control of Semi-automated Manufacturing Systems. *Procedia CIRP* **2021**, *103*, 170–175. [CrossRef]
58. May, M.C.; Behnen, L.; Holzer, A.; Kuhnle, A.; Lanza, G. Multi-variate time-series for time constraint adherence prediction in complex job shops. *Procedia CIRP* **2021**, *103*, 55–60. [CrossRef]
59. Wurster, M.; Michel, M.; May, M.C.; Kuhnle, A.; Stricker, N.; Lanza, G. Modelling and condition-based control of a flexible and hybrid disassembly system with manual and autonomous workstations using reinforcement learning. *J. Intell. Manuf.* **2022**, *33*, 575–591. [CrossRef]
60. Liberati, A.; Altman, D.G.; Tetzlaff, J.; Mulrow, C.; Gøtzsche, P.C.; Ioannidis, J.P.; Clarke, M.; Devereaux, P.J.; Kleijnen, J.; Moher, D. The PRISMA statement for reporting systematic reviews and meta-analyses of studies that evaluate health care interventions: Explanation and elaboration. *J. Clin. Epidemiol.* **2009**, *62*, e1–e34. [CrossRef]
61. Sampath, V.; Maurtua, I.; Aguilar Martín, J.J.; Gutierrez, A. A survey on generative adversarial networks for imbalance problems in computer vision tasks. *J. Big Data* **2021**, *8*, 27. [CrossRef]
62. Polyzotis, N.; Roy, S.; Whang, S.E.; Zinkevich, M. Data lifecycle challenges in production machine learning: A survey. *ACM Sigmod Rec.* **2018**, *47*, 17–28. [CrossRef]
63. Wang, Z.; Oates, T. Imaging time-series to improve classification and imputation. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015.
64. Lee, G.; Flowers, M.; Dyer, M. Learning distributed representations of conceptual knowledge. In Proceedings of the International 1989 Joint Conference on Neural Networks, Washington, DC, USA, 18–22 June 1989. [CrossRef]
65. Zhu, Y.; Brettin, T.; Xia, F.; Partin, A.; Shukla, M.; Yoo, H.; Evrard, Y.A.; Doroshow, J.H.; Stevens, R.L. Converting tabular data into images for deep learning with convolutional neural networks. *Sci. Rep.* **2021**, *11*, 11325. [CrossRef]
66. Sharma, A.; Vans, E.; Shigemizu, D.; Boroevich, K.A.; Tsunoda, T. DeepInsight: A methodology to transform a non-image data to an image for convolution neural network architecture. *Sci. Rep.* **2019**, *9*, 11399. [CrossRef]
67. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
68. Nanni, L.; Ghidoni, S.; Brahnam, S. Handcrafted vs. non-handcrafted features for computer vision classification. *Pattern Recognit.* **2017**, *71*, 158–172. [CrossRef]
69. Alkinani, M.H.; Khan, W.Z.; Arshad, Q.; Raza, M. HSDDD: A Hybrid Scheme for the Detection of Distracted Driving through Fusion of Deep Learning and Handcrafted Features. *Sensors* **2022**, *22*, 1864. [CrossRef]
70. Chen, Z.; Zhang, L.; Cao, Z.; Guo, J. Distilling the Knowledge from Handcrafted Features for Human Activity Recognition. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4334–4342. [CrossRef]
71. Albawi, S.; Mohammed, T.A.; Al-Zawi, S. Understanding of a convolutional neural network. In Proceedings of the 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 21–23 August 2017; pp. 1–6.
72. Pearson, K. LIII. On lines and planes of closest fit to systems of points in space. *Lond. Edinb. Dublin Philos. Mag. J. Sci.* **1901**, *2*, 559–572. [CrossRef]
73. Comon, P. Independent component analysis, a new concept? *Signal Process.* **1994**, *36*, 287–314. [CrossRef]
74. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation applied to handwritten zip code recognition. *Neural Comput.* **1989**, *1*, 541–551. [CrossRef]
75. Mikolov, T.; Karafiát, M.; Burget, L.; Cernockỳ, J.; Khudanpur, S. Recurrent neural network based language model. In *Interspeech*; Makuhari: Chiba-city, Japan, 2010; Volume 2, pp. 1045–1048.
76. Ojala, T.; Pietikainen, M.; Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 971–987. [CrossRef]
77. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
78. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–26 June 2005; Volume 1, pp. 886–893.
79. Shensa, M.J. The discrete wavelet transform: Wedding the a trous and Mallat algorithms. *IEEE Trans. Signal Process.* **1992**, *40*, 2464–2482. [CrossRef]
80. Gröchenig, K. The short-time Fourier transform. In *Foundations of Time-Frequency Analysis*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2001; pp. 37–58.
81. Harris, Z.S. Distributional structure. *Word* **1954**, *10*, 146–162. [CrossRef]
82. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
83. Liu, D.; Kong, H.; Luo, X.; Liu, W.; Subramaniam, R. Bringing AI to edge: From deep learning's perspective. *Neurocomputing* **2021**, *485*, 297–320. [CrossRef]
84. Gray, R.M.; Neuhoff, D.L. Quantization. *IEEE Trans. Inf. Theory* **1998**, *44*, 2325–2383. [CrossRef]
85. Sampath, V.; Maurtua, I.; Aguilar Martín, J.J.; Iriondo, A.; Lluvia, I.; Rivera, A. Vision Transformer based knowledge distillation for fasteners defect detection. In Proceedings of the 2022 International Conference on Electrical, Computer and Energy Technologies (ICECET), Prague, Czech Republic, 20–22 July 2022; pp. 1–6.
86. Shelden, R. Decision Tree. *Chem. Eng. Prog.* **1970**, *66*, 8.
87. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]

88. Freund, Y.; Schapire, R.E.; others. Experiments with a new boosting algorithm.. In Proceedings of the Thirteenth International Conference on International Conference on Machine Learning (ICML'96), Bari, Italy, 3–6 July 1996; Volume 96, pp. 148–156.
89. Ho, T.K. Random decision forests. In Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, Canada, 14–16 August 1995; Volume 1, pp. 278–282.
90. Chen, T.; Guestrin, C. XGBoost. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016. [CrossRef]
91. Choi, S.; Battulga, L.; Nasridinov, A.; Yoo, K.H. A decision tree approach for identifying defective products in the manufacturing process. *Int. J. Contents* **2017**, *13*, 57–65.
92. Sugumaran, V.; Muralidharan, V.; Ramachandran, K. Feature selection using decision tree and classification through proximal support vector machine for fault diagnostics of roller bearing. *Mech. Syst. Signal Process.* **2007**, *21*, 930–942. [CrossRef]
93. Hung, Y.H. Improved ensemble-learning algorithm for predictive maintenance in the manufacturing process. *Appl. Sci.* **2021**, *11*, 6832. [CrossRef]
94. Močkus, J. On bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference Novosibirsk, Novosibirsk, Russia, 1–7 July 1974*; Marchuk, G.I., Ed.; Springer: Berlin/Heidelberg, Germany, 1975; pp. 400–404.
95. Baum, L.E.; Petrie, T. Statistical inference for probabilistic functions of finite state Markov chains. *Ann. Math. Stat.* **1966**, *37*, 1554–1563. [CrossRef]
96. Papananias, M.; McLeay, T.E.; Mahfouf, M.; Kadirkamanathan, V. A Bayesian framework to estimate part quality and associated uncertainties in multistage manufacturing. *Comput. Ind.* **2019**, *105*, 35–47. [CrossRef]
97. Patange, A.D.; Jegadeeshwaran, R. Application of bayesian family classifiers for cutting tool inserts health monitoring on CNC milling. *Int. J. Progn. Health Manag.* **2020**, *11*. [CrossRef]
98. Pandita, P.; Ghosh, S.; Gupta, V.K.; Meshkov, A.; Wang, L. Application of Deep Transfer Learning and Uncertainty Quantification for Process Identification in Powder Bed Fusion. *ASME J. Risk Uncertain. Part B Mech. Eng.* **2022**, *8*, 011106. [CrossRef]
99. Farahani, A.; Tohidi, H.; Shoja, A. An integrated optimization of quality control chart parameters and preventive maintenance using Markov chain. *Adv. Prod. Eng. Manag.* **2019**, *14*, 5–14. [CrossRef]
100. El Haoud, N.; Bachiri, Z. Stochastic artificial intelligence benefits and supply chain management inventory prediction. In Proceedings of the 2019 International Colloquium on Logistics and Supply Chain Management (LOGISTIQUA), Paris, France, 12–14 June 2019; pp. 1–5.
101. Feng, M.; Li, Y. Predictive Maintenance Decision Making Based on Reinforcement Learning in Multistage Production Systems. *IEEE Access* **2022**, *10*, 18910–18921. [CrossRef]
102. Sobaszek, Ł.; Gola, A.; Kozłowski, E. Predictive scheduling with Markov chains and ARIMA models. *Appl. Sci.* **2020**, *10*, 6121. [CrossRef]
103. Hofmann, T.; Schölkopf, B.; Smola, A.J. Kernel methods in machine learning. *Ann. Stat.* **2008**, *36*, 1171–1220. [CrossRef]
104. Gobert, C.; Reutzel, E.W.; Petrich, J.; Nassar, A.R.; Phoha, S. Application of supervised machine learning for defect detection during metallic powder bed fusion additive manufacturing using high resolution imaging. *Addit. Manuf.* **2018**, *21*, 517–528. [CrossRef]
105. McGregor, D.J.; Bimrose, M.V.; Shao, C.; Tawfick, S.; King, W.P. Using machine learning to predict dimensions and qualify diverse part designs across multiple additive machines and materials. *Addit. Manuf.* **2022**, *55*, 102848. [CrossRef]
106. Kubik, C.; Knauer, S.M.; Groche, P. Smart sheet metal forming: Importance of data acquisition, preprocessing and transformation on the performance of a multiclass support vector machine for predicting wear states during blanking. *J. Intell. Manuf.* **2022**, *33*, 259–282. [CrossRef]
107. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
108. Mika, S.; Ratsch, G.; Weston, J.; Scholkopf, B.; Mullers, K. Fisher discriminant analysis with kernels. In Proceedings of the Neural Networks for Signal Processing IX: Proceedings of the 1999 IEEE Signal Processing Society Workshop (Cat. No.98TH8468), Madison, WI, USA, 25 August 1999; pp. 41–48. [CrossRef]
109. Fukushima, K.; Miyake, S. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and Cooperation in Neural Nets*; Springer: Berlin/Heidelberg, Germany, 1982; pp. 267–285.
110. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
111. Hinton, G.E. Deep belief networks. *Scholarpedia* **2009**, *4*, 5947. [CrossRef]
112. Badmos, O.; Kopp, A.; Bernthaler, T.; Schneider, G. Image-based defect detection in lithium-ion battery electrode using convolutional neural networks. *J. Intell. Manuf.* **2020**, *31*, 885–897. [CrossRef]
113. Ho, S.; Zhang, W.; Young, W.; Buchholz, M.; Al Jufout, S.; Dajani, K.; Bian, L.; Mozumdar, M. DLAM: Deep Learning Based Real-Time Porosity Prediction for Additive Manufacturing Using Thermal Images of the Melt Pool. *IEEE Access* **2021**, *9*, 115100–115114. [CrossRef]
114. Wen, L.; Li, X.; Gao, L.; Zhang, Y. A new convolutional neural network-based data-driven fault diagnosis method. *IEEE Trans. Ind. Electron.* **2017**, *65*, 5990–5998. [CrossRef]
115. Al-Dulaimi, A.; Zabihi, S.; Asif, A.; Mohammadi, A. A multimodal and hybrid deep neural network model for remaining useful life estimation. *Comput. Ind.* **2019**, *108*, 186–196. [CrossRef]

116. Huang, J.; Segura, L.J.; Wang, T.; Zhao, G.; Sun, H.; Zhou, C. Unsupervised learning for the droplet evolution prediction and process dynamics understanding in inkjet printing. *Addit. Manuf.* **2020**, *35*, 101197. [CrossRef]
117. Huang, J.; Chang, Q.; Arinez, J. Product completion time prediction using a hybrid approach combining deep learning and system model. *J. Manuf. Syst.* **2020**, *57*, 311–322. [CrossRef]
118. Cohen, J.; Jiang, B.; Ni, J. Machine Learning for Diagnosis of Event Synchronization Faults in Discrete Manufacturing Systems. *J. Manuf. Sci. Eng.* **2022**, *144*, 071006. [CrossRef]
119. Mujeeb, A.; Dai, W.; Erdt, M.; Sourin, A. One class based feature learning approach for defect detection using deep autoencoders. *Adv. Eng. Inform.* **2019**, *42*, 100933. [CrossRef]
120. Kasim, N.; Nuawi, M.; Ghani, J.; Rizal, M.; Ngatiman, N.; Haron, C. Enhancing Clustering Algorithm with Initial Centroids in Tool Wear Region Recognition. *Int. J. Precis. Eng. Manuf.* **2021**, *22*, 843–863. [CrossRef]
121. Djatna, T.; Alitu, I.M. An application of association rule mining in total productive maintenance strategy: An analysis and modelling in wooden door manufacturing industry. *Procedia Manuf.* **2015**, *4*, 336–343. [CrossRef]
122. Chiang, L.H.; Colegrove, L.F. Industrial implementation of on-line multivariate quality control. *Chemom. Intell. Lab. Syst.* **2007**, *88*, 143–153. [CrossRef]
123. You, D.; Gao, X.; Katayama, S. WPD-PCA-based laser welding process monitoring and defects diagnosis by using FNN and SVM. *IEEE Trans. Ind. Electron.* **2014**, *62*, 628–636. [CrossRef]
124. Moshat, S.; Datta, S.; Bandyopadhyay, A.; Pal, P. Optimization of CNC end milling process parameters using PCA-based Taguchi method. *Int. J. Eng. Sci. Technol.* **2010**, *2*, 95–102. [CrossRef]
125. Mei, S.; Wang, Y.; Wen, G. Automatic fabric defect detection with a multi-scale convolutional denoising autoencoder network model. *Sensors* **2018**, *18*, 1064. [CrossRef] [PubMed]
126. Maggipinto, M.; Beghi, A.; Susto, G.A. A Deep Convolutional Autoencoder-Based Approach for Anomaly Detection With Industrial, Non-Images, 2-Dimensional Data: A Semiconductor Manufacturing Case Study. *IEEE Trans. Autom. Sci. Eng.* **2022**. [CrossRef]
127. Yang, Z.; Gjorgjevikj, D.; Long, J.; Zi, Y.; Zhang, S.; Li, C. Sparse autoencoder-based multi-head deep neural networks for machinery fault diagnostics with detection of novelties. *Chin. J. Mech. Eng.* **2021**, *34*, 54. [CrossRef]
128. Cheng, R.C.; Chen, K.S. Ball bearing multiple failure diagnosis using feature-selected autoencoder model. *Int. J. Adv. Manuf. Technol.* **2022**, *120*, 4803–4819. [CrossRef]
129. Ramamurthy, M.; Robinson, Y.H.; Vimal, S.; Suresh, A. Auto encoder based dimensionality reduction and classification using convolutional neural networks for hyperspectral images. *Microprocess. Microsyst.* **2020**, *79*, 103280. [CrossRef]
130. Angelopoulos, A.; Michailidis, E.T.; Nomikos, N.; Trakadas, P.; Hatziefremidis, A.; Voliotis, S.; Zahariadis, T. Tackling faults in the industry 4.0 era—A survey of machine-learning solutions and key aspects. *Sensors* **2019**, *20*, 109. [CrossRef]
131. de Lima, M.J.; Crovato, C.D.P.; Mejia, R.I.G.; da Rosa Righi, R.; de Oliveira Ramos, G.; da Costa, C.A.; Pesenti, G. HealthMon: An approach for monitoring machines degradation using time-series decomposition, clustering, and metaheuristics. *Comput. Ind. Eng.* **2021**, *162*, 107709. [CrossRef]
132. Song, W.; Wen, L.; Gao, L.; Li, X. Unsupervised fault diagnosis method based on iterative multi-manifold spectral clustering. *IET Collab. Intell. Manuf.* **2019**, *1*, 48–55. [CrossRef]
133. Subramaniyan, M.; Skoogh, A.; Muhammad, A.S.; Bokrantz, J.; Johansson, B.; Roser, C. A generic hierarchical clustering approach for detecting bottlenecks in manufacturing. *J. Manuf. Syst.* **2020**, *55*, 143–158. [CrossRef]
134. Srinivasan, M.; Moon, Y.B. A comprehensive clustering algorithm for strategic analysis of supply chain networks. *Comput. Ind. Eng.* **1999**, *36*, 615–633. [CrossRef]
135. Das, J.N.; Tiwari, M.K.; Sinha, A.K.; Khanzode, V. Integrated warehouse assignment and carton configuration optimization using deep clustering-based evolutionary algorithms. *Expert Syst. Appl.* **2023**, *212*, 118680. [CrossRef]
136. Stojanovic, L.; Dinic, M.; Stojanovic, N.; Stojadinovic, A. Big-data-driven anomaly detection in industry (4.0): An approach and a case study. In Proceedings of the 2016 IEEE International Conference on Big Data (Big Data), Washington, DC, USA, 5–8 December 2016; pp. 1647–1652.
137. Saldivar, A.A.F.; Goh, C.; Li, Y.; Chen, Y.; Yu, H. Identifying smart design attributes for Industry 4.0 customization using a clustering Genetic Algorithm. In Proceedings of the 2016 22nd International Conference on Automation and Computing (ICAC), Colchester, UK, 7–8 September 2016; pp. 408–414.
138. Chen, W.C.; Tseng, S.S.; Wang, C.Y. A novel manufacturing defect detection method using association rule mining techniques. *Expert Syst. Appl.* **2005**, *29*, 807–815. [CrossRef]
139. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]
140. Iwana, B.K.; Uchida, S. An empirical survey of data augmentation for time series classification with neural networks. *PLoS ONE* **2021**, *16*, e0254841. [CrossRef]
141. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]
142. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114.
143. Wong, S.C.; Gatt, A.; Stamatescu, V.; McDonnell, M.D. Understanding data augmentation for classification: When to warp? In Proceedings of the 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA), Gold Coast, Australia, 30 November–2 December 2016; pp. 1–6.

144. Berthelot, D.; Carlini, N.; Goodfellow, I.; Papernot, N.; Oliver, A.; Raffel, C.A. Mixmatch: A holistic approach to semi-supervised learning. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 5049–5059.
145. Sohn, K.; Berthelot, D.; Carlini, N.; Zhang, Z.; Zhang, H.; Raffel, C.A.; Cubuk, E.D.; Kurakin, A.; Li, C.L. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 596–608.
146. Yang, X.; Song, Z.; King, I.; Xu, Z. A Survey on Deep Semi-supervised Learning. *arXiv* **2021**, arXiv:2103.00550.
147. Sajjadi, M.; Javanmardi, M.; Tasdizen, T. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 1171–1179.
148. Tarvainen, A.; Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1195–1204.
149. Li, X.; Jia, X.; Yang, Q.; Lee, J. Quality analysis in metal additive manufacturing with deep learning. *J. Intell. Manuf.* **2020**, *31*, 2003–2017. [CrossRef]
150. Zhao, B.; Zhang, X.; Zhan, Z.; Wu, Q.; Zhang, H. A Novel Semi-Supervised Graph-Guided Approach for Intelligent Health State Diagnosis of a 3-PRR Planar Parallel Manipulator. *IEEE/ASME Trans. Mechatron.* **2022**, *27*, 4786–4797. [CrossRef]
151. Gilmer, J.; Schoenholz, S.S.; Riley, P.F.; Vinyals, O.; Dahl, G.E. Neural message passing for quantum chemistry. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1263–1272.
152. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
153. Serradilla, O.; Zugasti, E.; Ramirez de Okariz, J.; Rodriguez, J.; Zurutuza, U. Adaptable and explainable predictive maintenance: Semi-supervised deep learning for anomaly detection and diagnosis in press machine data. *Appl. Sci.* **2021**, *11*, 7376. [CrossRef]
154. Song, J.; Lee, Y.C.; Lee, J. Deep generative model with time series-image encoding for manufacturing fault detection in die casting process. *J. Intell. Manuf.* **2022**, 1–14. [CrossRef]
155. Springenberg, J.T. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *arXiv* **2015**, arXiv:1511.06390.
156. Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X. Improved techniques for training gans. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 2234–2242.
157. Kingma, D.P.; Mohamed, S.; Jimenez Rezende, D.; Welling, M. Semi-supervised learning with deep generative models. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 3581–3589.
158. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
159. May, M.C.; Overbeck, L.; Wurster, M.; Kuhnle, A.; Lanza, G. Foresighted digital twin for situational agent selection in production control. *Procedia CIRP* **2021**, *99*, 27–32. [CrossRef]
160. May, M.C.; Kiefer, L.; Kuhnle, A.; Stricker, N.; Lanza, G. Decentralized multi-agent production control through economic model bidding for matrix production systems. *Procedia Cirp* **2021**, *96*, 3–8. [CrossRef]
161. Yao, M. Breakthrough Research In Reinforcement Learning From 2019. 2019. Available online: https://www.topbots.com/top-ai-reinforcement-learning-research-papers-2019 (accessed on 1 September 2022).
162. Gao, R.X.; Wang, L.; Helu, M.; Teti, R. Big data analytics for smart factories of the future. *CIRP Ann.* **2020**, *69*, 668–692. [CrossRef]
163. Kozjek, D.; Vrabič, R.; Kralj, D.; Butala, P. Interpretative identification of the faulty conditions in a cyclic manufacturing process. *J. Manuf. Syst.* **2017**, *43*, 214–224. [CrossRef]
164. Wen, Q.; Sun, L.; Yang, F.; Song, X.; Gao, J.; Wang, X.; Xu, H. Time series data augmentation for deep learning: A survey. *arXiv* **2020**, arXiv:2002.12478.
165. Pan, S.J.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2009**, *22*, 1345–1359. [CrossRef]
166. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv* **2017**, arXiv:1710.09412.
167. Bao, J.; Chen, D.; Wen, F.; Li, H.; Hua, G. CVAE-GAN: Fine-grained image generation through asymmetric training. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2745–2754.
168. Yoon, J.; Jarrett, D.; Van der Schaar, M. Time-series generative adversarial networks. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 5508–5518.
169. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
170. Cheng, Y.; Wang, D.; Zhou, P.; Zhang, T. Model compression and acceleration for deep neural networks: The principles, progress, and challenges. *IEEE Signal Process. Mag.* **2018**, *35*, 126–136. [CrossRef]
171. Gou, J.; Yu, B.; Maybank, S.J.; Tao, D. Knowledge distillation: A survey. *Int. J. Comput. Vis.* **2021**, *129*, 1789–1819. [CrossRef]
172. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531.
173. Schlimmer, J.C.; Granger, R.H. Incremental learning from noisy data. *Mach. Learn.* **1986**, *1*, 317–354. [CrossRef]
174. Gama, J.; Žliobaitė, I.; Bifet, A.; Pechenizkiy, M.; Bouchachia, A. A survey on concept drift adaptation. *ACM Comput. Surv.* **2014**, *46*, 44. [CrossRef]
175. Baier, L.; Jöhren, F.; Seebacher, S. Challenges in the Deployment and Operation of Machine Learning in Practice. In Proceedings of the ECIS 2019 27th European Conference on Information Systems, Stockholm, Sweden, 8–14 June 2019.
176. Canbek, G. Gaining insights in datasets in the shade of "garbage in, garbage out" rationale: Feature space distribution fitting. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2022**, *12*, e1456. [CrossRef]

177. Moges, T.; Yang, Z.; Jones, K.; Feng, S.; Witherell, P.; Lu, Y. Hybrid modeling approach for melt-pool prediction in laser powder bed fusion additive manufacturing. *J. Comput. Inf. Sci. Eng.* **2021**, *21*, 050902. [CrossRef]
178. Colledani, M., Statistical Process Control. In *CIRP Encyclopedia of Production Engineering*; Laperrière, L., Reinhart, G., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; pp. 1150–1157. [CrossRef]
179. Abdar, M.; Pourpanah, F.; Hussain, S.; Rezazadegan, D.; Liu, L.; Ghavamzadeh, M.; Fieguth, P.; Cao, X.; Khosravi, A.; Acharya, U.R.; et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Inf. Fusion* **2021**, *76*, 243–297. [CrossRef]
180. Yong, B.X.; Brintrup, A. Multi Agent System for Machine Learning Under Uncertainty in Cyber Physical Manufacturing System. In *Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future*; Borangiu, T., Trentesaux, D., Leitão, P., Giret Boggino, A., Botti, V., Eds.; Studies in Computational Intelligence; Springer International Publishing: Cham, Switzerland, 2020; Volume 853, pp. 244–257. [CrossRef]
181. Tavazza, F.; DeCost, B.; Choudhary, K. Uncertainty Prediction for Machine Learning Models of Material Properties. *ACS Omega* **2021**, *6*, 32431–32440. [CrossRef]
182. Arkov, V. Uncertainty Estimation in Machine Learning. *arXiv* **2022**. [CrossRef]
183. Zhang, B. Data-Driven Uncertainty Analysis in Neural Networks with Applications to Manufacturing Process Monitoring. Ph.D. Thesis, Purdue University Graduate School, West Lafayette, IN, USA, 2021. [CrossRef]
184. Zhang, B.; Shin, Y.C. A probabilistic neural network for uncertainty prediction with applications to manufacturing process monitoring. *Appl. Soft Comput.* **2022**, *124*, 108995. [CrossRef]
185. Lee, S.; Kim, S.B. Time-adaptive support vector data description for nonstationary process monitoring. *Eng. Appl. Artif. Intell.* **2018**, *68*, 18–31. [CrossRef]
186. Gaikwad, A.; Yavari, R.; Montazeri, M.; Cole, K.; Bian, L.; Rao, P. Toward the digital twin of additive manufacturing: Integrating thermal simulations, sensing, and analytics to detect process faults. *IISE Trans.* **2020**, *52*, 1204–1217. [CrossRef]
187. Zhang, C.J.; Zhang, Y.C.; Han, Y. Industrial cyber-physical system driven intelligent prediction model for converter end carbon content in steelmaking plants. *J. Ind. Inf. Integr.* **2022**, *28*, 100356. [CrossRef]
188. Ning, F.; Shi, Y.; Cai, M.; Xu, W.; Zhang, X. Manufacturing cost estimation based on the machining process and deep-learning method. *J. Manuf. Syst.* **2020**, *56*, 11–22. [CrossRef]
189. Westphal, E.; Seitz, H. Machine learning for the intelligent analysis of 3D printing conditions using environmental sensor data to support quality assurance. *Addit. Manuf.* **2022**, *50*, 102535. [CrossRef]
190. Qin, J.; Wang, Y.; Ding, J.; Williams, S. Optimal droplet transfer mode maintenance for wire+ arc additive manufacturing (WAAM) based on deep learning. *J. Intell. Manuf.* **2022**, *33*, 2179–2191. [CrossRef]
191. Lapointe, S.; Guss, G.; Reese, Z.; Strantza, M.; Matthews, M.; Druzgalski, C. Photodiode-based machine learning for optimization of laser powder bed fusion parameters in complex geometries. *Addit. Manuf.* **2022**, *53*, 102687. [CrossRef]
192. Zhang, T.; Zhang, C.; Hu, T. A robotic grasp detection method based on auto-annotated dataset in disordered manufacturing scenarios. *Robot. Comput. Integr. Manuf.* **2022**, *76*, 102329. [CrossRef]
193. Singh, S.A.; Desai, K. Automated surface defect detection framework using machine vision and convolutional neural networks. *J. Intell. Manuf.* **2022**, 1–17. [CrossRef]
194. Duan, J.; Hu, C.; Zhan, X.; Zhou, H.; Liao, G.; Shi, T. MS-SSPCANet: A powerful deep learning framework for tool wear prediction. *Robot. Comput. Integr. Manuf.* **2022**, *78*, 102391. [CrossRef]
195. Gao, K.; Chen, H.; Zhang, X.; Ren, X.; Chen, J.; Chen, X. A novel material removal prediction method based on acoustic sensing and ensemble XGBoost learning algorithm for robotic belt grinding of Inconel 718. *Int. J. Adv. Manuf. Technol.* **2019**, *105*, 217–232. [CrossRef]
196. Gawade, V.; Singh, V.; Guo, W. Leveraging simulated and empirical data-driven insight to supervised-learning for porosity prediction in laser metal deposition. *J. Manuf. Syst.* **2022**, *62*, 875–885. [CrossRef]
197. Aminzadeh, M.; Kurfess, T.R. Online quality inspection using Bayesian classification in powder-bed additive manufacturing from high-resolution visual camera images. *J. Intell. Manuf.* **2019**, *30*, 2505–2523. [CrossRef]
198. Priore, P.; Ponte, B.; Puente, J.; Gómez, A. Learning-based scheduling of flexible manufacturing systems using ensemble methods. *Comput. Ind. Eng.* **2018**, *126*, 282–291. [CrossRef]
199. Guo, S.; Chen, M.; Abolhassani, A.; Kalamdani, R.; Guo, W.G. Identifying manufacturing operational conditions by physics-based feature extraction and ensemble clustering. *J. Manuf. Syst.* **2021**, *60*, 162–175. [CrossRef]
200. Kim, J.; Ko, J.; Choi, H.; Kim, H. Printed circuit board defect detection using deep learning via a skip-connected convolutional autoencoder. *Sensors* **2021**, *21*, 4968. [CrossRef]
201. Jakubowski, J.; Stanisz, P.; Bobek, S.; Nalepa, G.J. Anomaly Detection in Asset Degradation Process Using Variational Autoencoder and Explanations. *Sensors* **2021**, *22*, 291. [CrossRef]
202. Sarita, K.; Devarapalli, R.; Kumar, S.; Malik, H.; Garcia Marquez, F.P.; Rai, P. Principal component analysis technique for early fault detection. *J. Intell. Fuzzy Syst.* **2022**, *42*, 861–872. [CrossRef]
203. Zheng, X.; Wang, H.; Chen, J.; Kong, Y.; Zheng, S. A generic semi-supervised deep learning-based approach for automated surface inspection. *IEEE Access* **2020**, *8*, 114088–114099. [CrossRef]
204. Zhang, W.; Lang, J. Semi-supervised training for positioning of welding seams. *Sensors* **2021**, *21*, 7309. [CrossRef]

205. Chen, C.; Liu, Y.; Kumar, M.; Qin, J.; Ren, Y. Energy consumption modelling using deep learning embedded semi-supervised learning. *Comput. Ind. Eng.* **2019**, *135*, 757–765. [CrossRef]
206. Jun, J.h.; Chang, T.W.; Jun, S. Quality prediction and yield improvement in process manufacturing based on data analytics. *Processes* **2020**, *8*, 1068. [CrossRef]
207. Shim, J.; Cho, S.; Kum, E.; Jeong, S. Adaptive fault detection framework for recipe transition in semiconductor manufacturing. *Comput. Ind. Eng.* **2021**, *161*, 107632. [CrossRef]
208. Qiu, C.; Li, K.; Li, B.; Mao, X.; He, S.; Hao, C.; Yin, L. Semi-supervised graph convolutional network to predict position-and speed-dependent tool tip dynamics with limited labeled data. *Mech. Syst. Signal Process.* **2022**, *164*, 108225. [CrossRef]
209. Guo, Y.; Lu, W.F.; Fuh, J.Y.H. Semi-supervised deep learning based framework for assessing manufacturability of cellular structures in direct metal laser sintering process. *J. Intell. Manuf.* **2021**, *32*, 347–359. [CrossRef]
210. Okaro, I.A.; Jayasinghe, S.; Sutcliffe, C.; Black, K.; Paoletti, P.; Green, P.L. Automatic fault detection for laser powder-bed fusion using semi-supervised machine learning. *Addit. Manuf.* **2019**, *27*, 42–53. [CrossRef]
211. Lee, H.; Kim, H. Semi-supervised multi-label learning for classification of wafer bin maps with mixed-type defect patterns. *IEEE Trans. Semicond. Manuf.* **2020**, *33*, 653–662. [CrossRef]
212. Liu, J.; Song, K.; Feng, M.; Yan, Y.; Tu, Z.; Zhu, L. Semi-supervised anomaly detection with dual prototypes autoencoder for industrial surface inspection. *Opt. Lasers Eng.* **2021**, *136*, 106324. [CrossRef]
213. Verstraete, D.; Droguett, E.; Modarres, M. A deep adversarial approach based on multi-sensor fusion for semi-supervised remaining useful life prognostics. *Sensors* **2019**, *20*, 176. [CrossRef]
214. Souza, M.L.H.; da Costa, C.A.; de Oliveira Ramos, G.; da Rosa Righi, R. A feature identification method to explain anomalies in condition monitoring. *Comput. Ind.* **2021**, *133*, 103528. [CrossRef]
215. Lee, Y.H.; Lee, S. Deep reinforcement learning based scheduling within production plan in semiconductor fabrication. *Expert Syst. Appl.* **2022**, *191*, 116222. [CrossRef]
216. Marchesano, M.G.; Guizzi, G.; Santillo, L.C.; Vespoli, S. A deep reinforcement learning approach for the throughput control of a flow-shop production system. *IFAC-PapersOnLine* **2021**, *54*, 61–66. [CrossRef]
217. Yang, H.; Li, W.; Wang, B. Joint optimization of preventive maintenance and production scheduling for multi-state production systems based on reinforcement learning. *Reliab. Eng. Syst. Saf.* **2021**, *214*, 107713. [CrossRef]
218. Schneckenreither, M.; Haeussler, S.; Peiró, J. Average reward adjusted deep reinforcement learning for order release planning in manufacturing. *Knowl.-Based Syst.* **2022**, *247*, 108765. [CrossRef]
219. Tsai, Y.T.; Lee, C.H.; Liu, T.Y.; Chang, T.J.; Wang, C.S.; Pawar, S.J.; Huang, P.H.; Huang, J.H. Utilization of a reinforcement learning algorithm for the accurate alignment of a robotic arm in a complete soft fabric shoe tongues automation process. *J. Manuf. Syst.* **2020**, *56*, 501–513. [CrossRef]
220. Klar, M.; Glatt, M.; Aurich, J.C. An implementation of a reinforcement learning based algorithm for factory layout planning. *Manuf. Lett.* **2021**, *30*, 1–4. [CrossRef]
221. Huang, J.; Chang, Q.; Arinez, J. Deep reinforcement learning based preventive maintenance policy for serial production lines. *Expert Syst. Appl.* **2020**, *160*, 113701. [CrossRef]
222. Zhang, H.; Peng, Q.; Zhang, J.; Gu, P. Planning for automatic product assembly using reinforcement learning. *Comput. Ind.* **2021**, *130*, 103471. [CrossRef]
223. Kuhnle, A.; May, M.C.; Schaefer, L.; Lanza, G. Explainable reinforcement learning in production control of job shop manufacturing system. *Int. J. Prod. Res.* **2021**, *60*, 5812–5834. [CrossRef]
224. Valet, A.; Altenmüller, T.; Waschneck, B.; May, M.C.; Kuhnle, A.; Lanza, G. Opportunistic maintenance scheduling with deep reinforcement learning. *J. Manuf. Syst.* **2022**, *64*, 518–534. [CrossRef]
225. Huang, J.; Su, J.; Chang, Q. Graph neural network and multi-agent reinforcement learning for machine-process-system integrated control to optimize production yield. *J. Manuf. Syst.* **2022**, *64*, 81–93. [CrossRef]
226. Zimmerling, C.; Poppe, C.; Stein, O.; Kärger, L. Optimisation of manufacturing process parameters for variable component geometries using reinforcement learning. *Mater. Des.* **2022**, *214*, 110423. [CrossRef]
227. Guo, F.; Zhou, X.; Liu, J.; Zhang, Y.; Li, D.; Zhou, H. A reinforcement learning decision model for online process parameters optimization from offline data in injection molding. *Appl. Soft Comput. J.* **2019**, *85*, 105828. [CrossRef]
228. Hofmann, C.; Liu, X.; May, M.; Lanza, G. Hybrid Monte Carlo tree search based multi-objective scheduling. *Prod. Eng.* **2022**, *17*, 133–144. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

This article has been accepted for publication in IEEE Transactions on Industrial Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TII.2023.3234030

Attention Guided Multi-Task Learning for Surface defect identification

Vignesh Sampath*, Iñaki Maurtua, Juan José Aguilar Martín, Andoni Rivera, Jorge Molina and Aitor Gutierrez

***Abstract*—Surface defect identification is an essential task in the industrial quality control process, in which visual checks are conducted on a manufactured product to ensure that it meets quality standards. Convolutional Neural Network (CNN) based surface defect identification method has proven to outperform traditional image processing techniques. However, the real-world surface defect datasets are limited in size due to the expensive data generation process and the rare occurrence of defects. To address this issue, this paper presents a method for exploiting auxiliary information beyond the primary labels to improve the generalization ability of surface defect identification tasks. Considering the correlation between pixel level segmentation masks, object level bounding boxes and global image level classification labels, we argue that jointly learning features of the related tasks can improve the performance of surface defect identification tasks. This paper proposes a framework named Defect-Aux-Net, based on multi-task learning with attention mechanisms that exploit the rich additional information from related tasks with the goal of simultaneously improving robustness and accuracy of the CNN based surface defect identification. We conducted a series of experiments with the proposed framework. The experimental results showed that the proposed method can significantly improve the performance of state-of-the-art models while achieving an overall accuracy of 97.1%, Dice score of 0.926 and mAP of 0.762 on defect classification, segmentation and detection tasks.**

***Index Terms*—Deep learning, defect classification, defect detection, defect segmentation, machine vision, multi-task-learning, quality control, surface defect detection.**

I. INTRODUCTION

AUTOMATED visual inspection plays an important role in industrial informatics based decision-making systems in various industries, including steel manufacturing companies, automotive industries, electronic manufacturing, and pharmaceutical companies. The correct, consistent, and early detection of surface defects can make it possible to detect defective products early in the manufacturing process, which leads to time and cost savings. Inspection procedures for detecting such defects are usually performed using non-destructive testing (NDT) methods. NDT procedure is a combination of various inspection steps used to identify discontinuities or defects in a product without causing damage to its usability. The most frequently used industrial NDT methods are Visual optic testing, Radiography, X-ray vision, Ultrasonic imaging, Dye penetrant testing, Magnetic particle testing, and Infrared thermal imaging. The testing procedure for each of these methods involves several steps, all of which can be easily automated. However, the final step of visual inspection is more complex in terms of automation and remains primarily a manual process performed by operators.

The traditional machine-vision system relies on a hand-crafted features such as color, contrast, texture, edges, foreground background statistics, etc. followed by machine learning classifiers such as support vector machines, decision tree or K-Nearest Neighbors. Consequently, hand-crafted features extraction plays an important role in classical approaches. However, these features are not robust and suited for different tasks, which lead to long development cycles. Deep learning methods, on the other hand, learn the relevant features directly from the raw data, without the need for handcrafted feature representations. In recent years, Convolutional Neural Network (CNN) has achieved and even surpassed human-level performance on computer vision tasks such as image classification. The key difference between CNN and traditional machine-vision algorithms is that CNN automatically detects significant features without any human supervision which made it the most widely used. A fascinating feature of CNN is its ability to take advantage of the spatial or temporal correlation of image data. There are three main problem categories for image recognition tasks using CNN: classification, segmentation, and object detection. Classification task aims to classify an image into a certain category. Starting with the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winning architecture of AlexNet [1], a series of increasingly complex architectures including ResNet [2], Inception [3], Densenet [4], and EfficientNet [5] have been proposed in the literature for the classification task. Object detection is a task that localizes an

This research work was undertaken in the context of DIGIMAN4.0 project ("Digital Manufacturing Technologies for Zero-defect", https://www.digiman4-0.mek.dtu.dk/). DIGIMAN4.0 is a European Training Network supported by Horizon 2020, the EU Framework Programme for Research and Innovation (Project ID: 814225). This research was also partly supported by the ELKARTEK project KK-2020/00049 3KIA of the Basque Government.

S. Vignesh, M. Iñaki, R. Andoni, M. Jorge and G. Aitor are with Tekniker, Autonomous and Intelligent Systems unit, Gipuzkoa 20600, Spain (e-mail: vignesh.sampath@tekniker.es; inaki.maurtua@tekniker.es; andoni.rivera@tekniker.es; jmolina@tekniker.es; aitor.gutierrez@tekniker.es).

A.M. Juan José is with the Department of Design and Manufacturing Engineering, University of Zaragoza, Zaragoza 50009, Spain (e-mail: jaguilar@unizar.es).

object using a bounding box. Some of the notable object detection algorithms include Fast R-CNN [6], Faster R-CNN, Mask R-CNN [7], Single Shot Detection (SSD) [8], You Only Look Once (YOLO) [9], etc. Segmentation is the task of performing pixel-by-pixel classification. Several segmentation algorithms have been proposed in the literature including fully convolutional networks, encoder-decoder based approaches [10], multi-scale and pyramid architectures [11], etc.

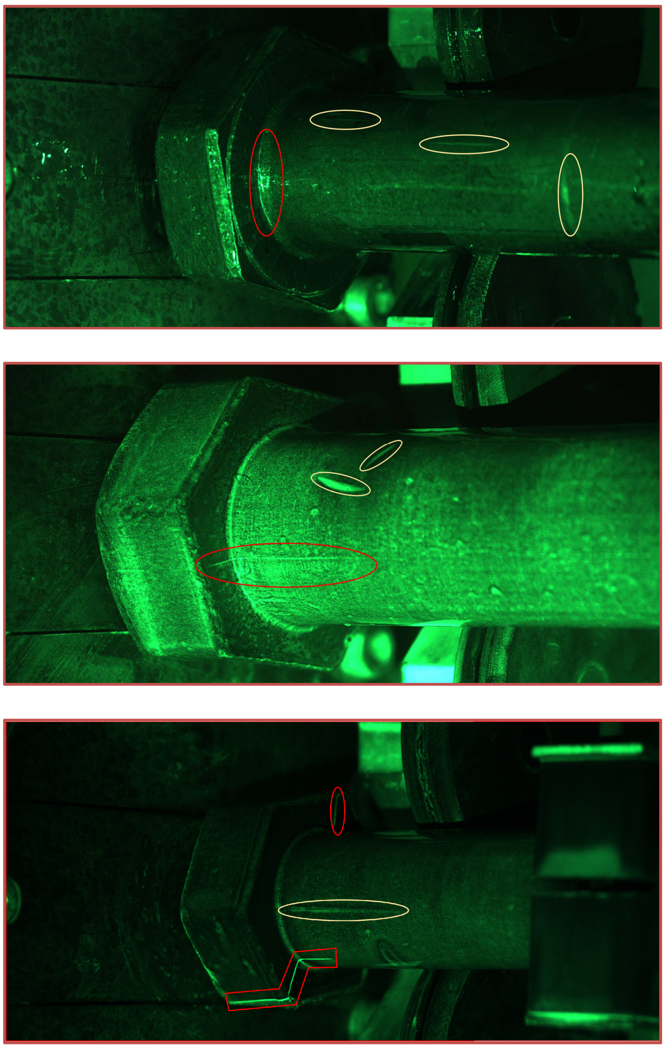


Fig. 1. Magnetic particle inspection on threaded fasteners of different surface finish (TekErreka dataset). Surface defects are marked by red circles and noise due to magnetic particle depositions are marked in yellow.

However, industrial visual inspection systems barely utilized the potential of those complex architectures due to several reasons [12]. One of the main reasons is that the continuous improvement in industrial processes has resulted in fewer and fewer defective samples, or the number of defective samples is very limited [13]. This problem of learning from a limited number of samples is usually referred to as the small sample problem, which can easily lead to poor generalization ability of the trained model [14]. In addition, the target surface defects have different scales, making the deep learning models even more challenging to identify the small sized defects. On the one hand, visual appearance of the real-world surfaces defects varies with type of materials, imaging conditions, and camera position. On the other hand, it is challenging to distinguish tiny defects from the noise or non-defect components within an image (as shown in Fig. 1). Hence the appearance of false positives in a defect free image is an inevitable circumstance. Furthermore, real time applications of complex CNN models are extremely limited due to the long inference time and the resulting higher computational resource and power consumption.

To address these limitations, we present a novel universal architecture that integrates classification, segmentation, and detection of surface defects in a single network. Our architecture, Defect-Aux-Net, is primarily motivated by a multi-task learning (MTL) scheme that exploits useful information from related learning tasks to help mitigate the problem of data scarcity. The proposed architecture is based on FPN-semantic-segmentation [11] with the additional tasks of defect classification and detection to improve the generalization ability by utilizing the image level information as an inductive bias. Specifically, we developed a new multi-task learning network based on FPN, where the classification task is carried out in the bottom-up pathway of the network and segmentation is performed in the top-down pathway of the network. To create a bounding box we employ two sub-networks in the top-down pathway, where one subnet determines the class associated with bounding box and the other performs the regression to adjust the bounding box position.

The FPN-based Feature Extractor in the proposed network allows surface defects to be recognized at vastly different scales by efficiently sharing features between image regions. We further introduce the positional and the channel attention mechanisms that focus on learning the features of small surface defects to improve the robustness of detecting small defects surrounded by complex background.

We evaluate our model on TekErreka, and Severstal [15] surface defect datasets, with defect classification, segmentation, and detection tasks. Experimental results demonstrate that jointly learning features of related tasks can improve the performance of all tasks.

Overall, the contributions of our work are as follows:

1) Firstly, we propose a Defect-Aux-Net model architecture, which can perform classification, segmentation, and detection of surface defects in a single network. Compared with the existing state-of-the-art CNN models, this architecture is lightweight and compact in terms of model parameters. From the model training point of view employing fewer parameters in the architecture enables model to efficiently learn potential surface defects from a smaller number of labelled examples.
2) In contrast to existing single task learning, our proposed multi-task learning in surface defect detection facilitates the model to learn useful representations of the data by exploiting shared information from related tasks.

3) Considering surface defect detection with complex background, the positional and the channel attention mechanisms are incorporated to amplify target features and to reduce the influence of background noise.
4) The proposed model is compact and efficient with state-of-the-art performance that meets the computational resource requirements of the real-time inference speed.

II. Related Work

A large and growing body of literature has explored the use of CNN for surface defect identification. Kim et al. [16] adopted few-shot learning technique with Siamese Neural Network using CNN, which aims to classify surface defects with a limited number of training images. Lin et al. [17] employed class activation mapping technique in CNN to simultaneously achieve defect classification and localization tasks in LED chip defect inspection process. Tao et al. [18] designed cascaded autoencoder (CASAE) architecture to segment and localize defect region. The proposed architecture transforms the input image into a mask prediction and then defect regions of segmented mask is classified to their specific classes. Jing et al. [19] combined autoencoder with fully connected network (FCN) to detect keyboard light leakage defect from mere dust. Jian et al. [20] leveraged Generative adversarial network (GAN) to exaggerate the tiny defects within the images to improve the accuracy of different classifiers. Zheng et al. [21] proposed a 3-stage model for rail surface and fastener defect detection. At the first stage, YOLOV5 framework is employed to localize the rail and fasteners. Then, an object detection model based on Mask-RCNN is used to detect the surface defect of the rail surface. At the final stage, the Resnet architecture is utilized to classify defects of the fasteners. To detect defects at different scale, Xu et al. [22] used a pre-trained ResNet model to extract the multi-scale features and fuse them using a multilevel feature fusion network (MFN). In [23], U-Net and residual U-Net architectures were used for the fine-grained segmentation of surface defects on a steel sheet. The main drawback of these methods is that the model needs a large amount of annotated data and hence the localization of defect is very coarse in the real-time scenario.

III. Proposed Method

A. Network architecture

Our proposed network is inspired by two deep learning architectures that are widely used: Feature pyramid Network (FPN) and ResNet-50. Recognizing surface defects at vastly different scales is a fundamental challenge in industrial machine vision system. For this reason, we use FPN that uses a pyramidal hierarchy of convolutional filters to extract feature pyramids at different scales. FPN consists of two pathways: bottom-up and top-down. The bottom-up pathway also known as encoder, is the typical convolutional neural network, which can be any image classifier for feature extraction. As we go up, the encoder gradually decreases the spatial resolution, while building high level feature maps. The top-down pathway is connected to the bottom-up pathway through lateral connections for efficient multi-scale feature fusion. It is designed to enhance the feature maps from the bottom-up pathway and build semantically strong feature maps at multiple scales by double upscaling. As a result, the feature pyramid has rich semantics at all levels because the lower semantic features are interconnected to the higher semantics.

1) Bottom-up pathway

We tested several standard image classification architectures to select the core model, and finally chose ResNet-50 as the backbone. ResNet-50 has shown great performance for surface defect classification, segmentation and detection tasks. ResNet-50 architecture has the advantage of using a stride of two for each scale reduction, which makes it easier to incorporate ResNet-50 into FPNs when we need to upscale feature maps in top-down pathway. Furthermore, Resnet-50 is a relatively small network based on modern standards; therefore, it is suitable for our limited labeled data problem. However, existing ResNet-50 feature pyramids have two problems in the way they apply convolution operations to the input features. Firstly, the receptive field of the encoder has the information only about the local region, so the global information is lost. Secondly, the feature maps constructed from the learned weights are given equal magnitude of importance, but some feature maps are more important for the next layers than others. For instance, a feature map that contains edge information of the defects might be more important than another feature map that has background texture information (as shown in Fig. 3.). Thus, to incorporate channel attention we adopt Squeeze-and-Excitation (SE) module [24] in the encoder. SE module consists of three components 1. Squeeze, 2. Excite and 3.Scale components.

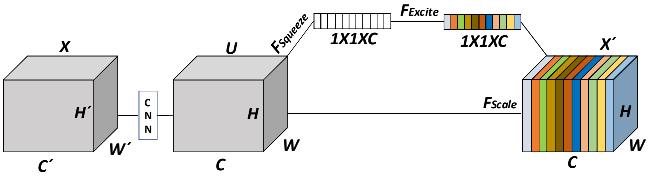


Fig. 2. Structure of Squeeze and Excite module.

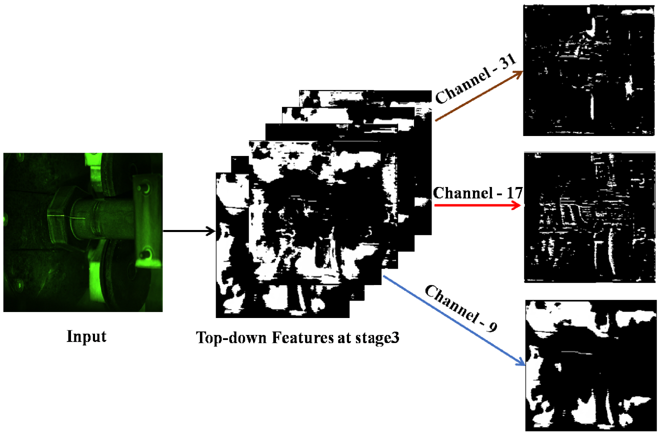


Fig. 3. Sample features in different channels of top-down pathway at stage 3.

This article has been accepted for publication in IEEE Transactions on Industrial Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TII.2023.3234030

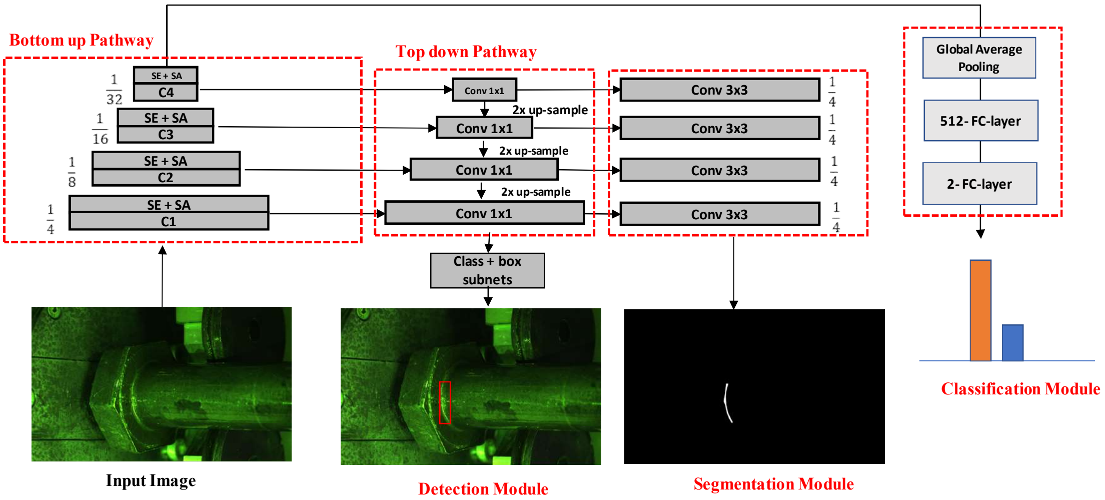


Fig. 4. An overview of proposed Defect-Aux-Net architecture. It mainly composed of classification, segmentation and detection module that incorporates multi-task loss function.

The main goal of the squeeze component is to extract global information from each of the channels c in a feature block U. The global information is acquired by applying a global average pooling operation across their spatial dimensions (H × W) for each channel U_c of U to obtain global statistics (1 × 1 × C). Mathematically, squeeze operation can be represented as:

$$z_c = F_{squeeze}(U_c) = \frac{1}{H\times W}\sum_{m=1}^{H}\sum_{n=1}^{W} U_c(m,n) \qquad (1)$$

After obtaining global information from the squeeze component, the excite component generate a set of weights for each channel. It uses a fully connected Multi-Layer Perceptron (MLP) bottleneck structure to dynamically calibrate the weights. This MLP bottleneck has two fully connected layers with sigmoid activation as the output layer. Output of the excitation component can formally be represented by the following equation:

$$s = F_{excite}(z, W) = \sigma\big(g(z, W)\big) = \sigma(W_2\rho(W_1, z)) \qquad (2)$$

Where σ is a Sigmoid operation, ρ is ReLU operation, z is the output from the squeeze component, W_1 and W_2 refers to weights of the two fully connected layers. Subsequently each channel in the feature map is scaled by a simple element-wise multiplication of the input feature map and weights obtained from the excite component (as shown in Fig. 2).

Surface defects only appear in some parts of the image but not the whole image. Unlike the conventional Resnet-50 architecture, which gives equal importance to each region in an image, the spatial attention reduces background interferences by assigning a weight to each pixel in the feature map.

The spatial attention focuses on the most relevant parts of the feature maps in the spatial dimension. The working principle of our spatial attention mechanism is as follows. Given feature block U, we use average and max-pooling operations along the channel axis and concatenate them to generate an efficient feature map summary M. A convolutional layer followed by sigmoid operation is then performed on the feature M to produce spatial attention map (as shown in Fig. 5).

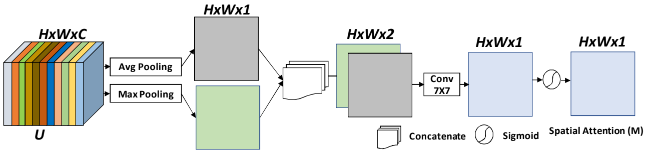


Fig. 5. Structure of Spatial Attention module.

Resnet uses four modules consisting of residual blocks, each of which uses two blocks, Identity (ID) blocks and convolution blocks, depending on whether the input / output dimensions are the same or different. We arrange SE and SA module in series and integrate into residual block (as shown in Fig. 6)

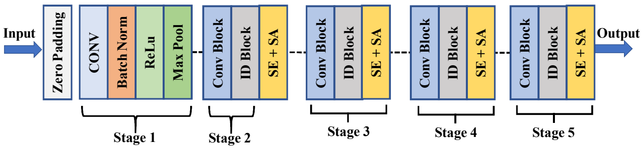


Fig. 6. FPN Bottom-Up structure with attention module

2) Top-down pathway

Deep features from bottom-up pathway are upsampled by convolutions and bilinear up-sampling operations until all the feature maps reach ¼ scale. Attention module outputs from bottom-up pathway $\{C_2, C_3, C_4, C_5\}$ are fused to top-down pathway through lateral connections for an efficient multi-scale feature fusion. Firstly, 1 x 1 convolutional filter is applied to the feature maps $\{C_2, C_3, C_4, C_5\}$ to get a fixed number of channels and then merged with the corresponding top-down feature map by element-wise addition. Finally, the outputs are summed and then transformed into a pixel-wise output (as shown in Fig. 4).

3) Segmentation branch

The segmentation branch from top-down pathway aims at classifying pixels into a set of pre-defined classes. The pixels corresponding to background are far numerous than pixels of surface defects in the real-world dataset, which causes the model to be biased toward the background element. To address the pixel wise class imbalance, we employ Dice loss, which uses Dice coefficient to calculate overlapping of the pixels of the predicted mask with the ground truth label. Mathematically Dice loss function is defined as:

$$L_{seg} = 1 - \frac{2y\hat{y}+1}{y+\hat{y}+1} \tag{3}$$

Where, $\mathrm{y_i}$ is the ground truth label, $\mathrm{\hat{y}_i}$ is the predicted label. The value of Dice coefficient ranges from 0 to 1, where 1 indicates the perfect and complete overlap of pixels.

4) Classification branch

The output of the bottom-up pathway encodes the rich abstract feature representations of the input image. Hence, we utilize the spatial average of the feature maps from the bottom-up pathway via a global average pooling layer and then the resulting feature vector is fed into the sigmoid or softmax layer depending on classification type. We employ binary cross-entropy (BCE) as classification loss function. Mathematically our classification loss is defined as:

$$L_{class} = \frac{1}{k}\sum_{1}^{k} CE(y_i, \hat{y}_i) \tag{4}$$

Where, $\mathrm{y_i}$ is the ground truth label, $\mathrm{\hat{y}_i}$ is the predicted label of $\mathrm{i^{th}}$ sample, k is the total number of samples. CE is the binary cross entropy function.

5) Object Detection branch

We extract bounding boxes and its associated classes by employing box regression and classification subnets at each level of top-down pathway. The classification subnet predicts the probability of defect presence at each spatial location of an input image. The box regression subnet is attached to top-down pathway in parallel to classification subnet for the purpose of regressing offset from each anchor box to the ground truth bounding boxes. To handle class imbalance problem, we adopt focal loss [25], an improved version of cross entropy to focus learning on hard negative examples. It is defined as:

$$L_{detection} = -\alpha_t(1-p_t)^{\gamma}\log(p_t) \tag{5}$$

Where, α_t is the weight parameter per class and γ is the hyper parameter focuses on hard negative samples. We choose α_t=0.25 and γ= 4 as suggested in [26].

B. Loss Function

Our proposed method combines three loss functions from the classification, segmentation and detection tasks which provide mutual sources of inductive bias for each task. Specifically, the segmentation and detection loss functions signal back to the entire model (bottom-up and top-down pathway), while the classification loss signals back only to bottom-up pathway. We combine and weight the three losses into a multi-task loss $\mathrm{L_M}$ to leverage the heterogeneous annotations and jointly optimize multiple tasks as follows:

$$L_M = \beta L_{class} + \beta_1 L_{seg} + \beta_2 L_{detection} \tag{6}$$

Where, β, β_1, and β_2 are weight parameters. We tested with different combinations of weight parameters and found that $\beta = \beta_1 = \beta_2 = 1$ yields the best result for all the tasks.

IV. Experiments

A. Datasets

In this paper, we evaluate our framework on real-world surface defect identification problems. We use two challenging datasets with increasing resolutions and complexities, Severstal steel sheet [15] and TekErreka steel fastener defect datasets. Severstal, the largest steel and steel-related mining company, has recently published the largest industrial steel sheet surface defect dataset, which contains pixel-wise masks annotated by their technical experts. The dataset contains 12568 grayscale images of size of 1600×256. Each image in the dataset has the possibility of having either no defects, a single defect, or multiple defects divided into four classes. Fig. 7 show the example of steel defect images on Severstal datasets. We randomly select 10% and 20% of the 12,568 original images as the validation and test data. The main challenge with this dataset is that the inter-class similarities between defective and defect-free examples are very high.

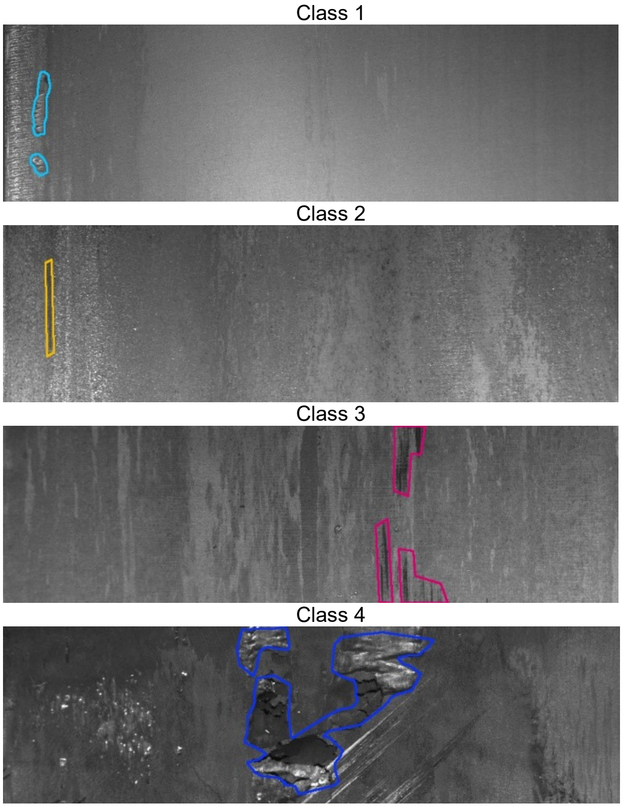


Fig. 7. Sample images of Severstal steel with 4 classes of defect.

The TekErreka dataset is a self-collected steel fastener surface defect dataset based on magnetic particle inspection procedure. The magnetic particle inspection is an excellent method to investigate near surface defects in steel fasteners. The basic principle is to magnetize a steel fastener parallel to its surface. If the fastener is free from defects the magnetic field lines run within the fastener and parallel to its surface. In case of magnetic inhomogeneity, for instance, near cracks, the magnetic field lines will locally leave the surface and a leakage field occurs. When a suspension of ferromagnetic particles is applied onto the test piece surface the magnetic particles will run off at defect free areas. In the places of leakage fields the magnetic particles are attracted and clustered together thus indicating the location of the defect. The surface defects can be visible under ultra violet light. We acquired TekErreka dataset from a magnetic particle inspection apparatus located at the Erreka Fastening solutions. The defects in the TekErreka dataset differ in their size, shape, location and materials type and thus cover several scenarios in real time defect detection. The difficulty in this dataset lies in the similarity of defects and noise due to magnetic particles deposition on defect free surface of the fasteners. There are many factors responsible for the noise component, which include magnetic particle size, the amount of magnetic particle used, ultra-violet light present, etc. The original examples are directly stored in a database as RGB images of size 2464 x 2056. It has 450 positive and 1200 negative examples. We split TekErreka dataset into training and testing sets: 80% for training and 20% for evaluation of the model performance.

B. Preprocessing

We resized the images of Severstal dataset to 128x800 and TekErreka dataset to 600x600. To keep the pixel values in same scale, we normalized the images using min-max standardization. It rescales raw pixel values to range of 0 and 1. This helps the optimizer not get stuck taking steps that are too large in one dimension, or too small in another.

C. Data Augmentation

To improve the diversity of the training set we apply random but realistic data augmentation such as rotation, vertical/horizontal flips, zoom, shear and channel shifts.

D. Training details

The Defect-Aux-Net is implemented using the Tensorflow framework. All the experiments are run on Google-cloud TPU V2 infrastructure which contains 8 cores with 64 GB memory. The network is optimized with the Adam optimizer and trained with a batch size of 128 for 50 epochs. We adopt one cycle policy [27] to find an optimal learning rate.

E. Evaluation Metrics

The classification results are evaluated using precision, recall, F1-score and binary accuracy.

$$Recall = \frac{TP}{TP+FN} \tag{7}$$

$$Precision = \frac{TP}{TP+FP} \tag{8}$$

$$F1\,Score = \frac{2.(Precision.Recall)}{(Precision+Recall)} \tag{9}$$

$$Accuracy = \frac{TP+TN}{TP+FP+TN+F} \tag{10}$$

Where TP, TN, FP and FN denote true positive (correctly identified surface defects), true negative (correctly identified non defect images), false positive (erroneously classified images as surface defect) and false negative (erroneously classified images as non defect). Precision measures the percentage of images with surface defect that are correctly classified, while recall is the ratio of correctly classified images with surface defect to all images with surface defect. F1- score can be interpreted as harmonic mean of precision and recall. The overall performance of the classification task is measured by its accuracy.

The segmentation results are evaluated using Dice score and Intersection-over-Union (IoU), which quantify the percentage overlap between the predicted and target binary masks. To evaluate defect detection results, we used the mean average precision (mAP) that compares the detected bounding box to the ground truth bounding box and returns a score.

F. Experiments on Defect Segmentation

We performed series of experiments on TekErreka dataset to test the effectiveness of different loss functions. First, we trained Defect-Aux-Net using BCE, and Dice loss alone as the segmentation loss. Then it was trained using a combination of loss functions. The results are shown in TABLE I.

TABLE I
PERFORMANCE OF THE PROPOSED APPROACH ON LOSS VARIANTS FOR THE DEFECT SEGMENTATION TASK

Loss Function	IoU	Dice
BCE	0.892	0.911
Dice	**0.903**	**0.926**
Jaccard	0.900	0.913
Dice + BCE	0.901	0.920
Jaccard + BCE	0.899	0.912

Using Dice loss alone yielded more accurate results than using combination of losses. Additionally, Dice loss function assisted our model to converge faster. We use Dice loss function throughout rest of the experiments.

To verify the effectiveness of segmentation task using multi-task learning strategy, we compared the proposed multi-task learning network (Defect-Aux-Net) against the following network with same bottom-up backbone (Resnet50 + SE + SA attention module):

1. FPN [11]: This is the original FPN architecture without multi-task learning strategy and serves as our baseline.
2. UNet [10]: This network uses an encoder for multi-level feature extraction and a decoder that scales them up and combines multi-level feature through stacking.
3. LinkNet [28]: This is similar to UNet with the difference of replacing stacking operation with addition in skip connections.

4. PSPNet [28]: Pyramid scene parsing Network uses pyramid pooling module for multi-scale feature extraction

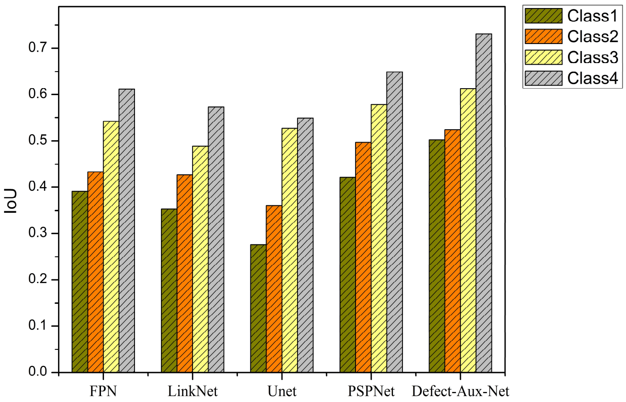


Fig. 8. IOU comparison between the state-of-the-art segmentation methods and the proposed approach on each type of defect classification.

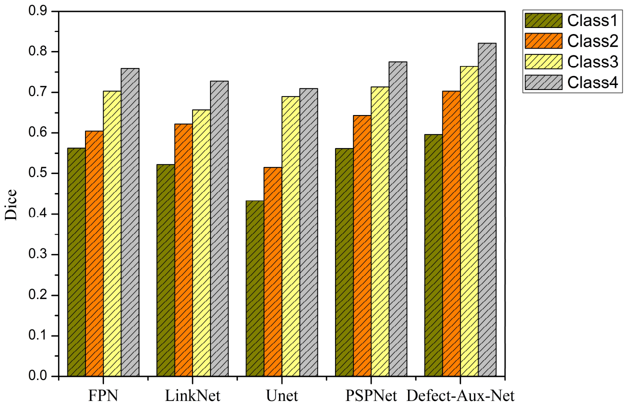


Fig. 9. Dice score comparison between the state-of-the-art segmentation methods and the proposed approach on each type of defect classification.

Based on the experimental results, we observed that the proposed multi task learning strategy achieves better segmentation performance as compared to the state-of-the-art segmentation models. The Dice and IoU scores of the various segmentation models on Severstal dataset are depicted in Fig. 8 and Fig. 9.

TABLE II
PERFORMANCE OF THE COMPETING MODELS ON THE TEKERREKA DATASET

Model	Iou	Dice
FPN [11]	0.881	0.902
LinkNet [28]	0.876	0.895
Unet [10]	0.832	0.856
PSPNet [29]	0.885	0.917
Defect-Aux-Net	**0.903**	**0.926**

We observe that Defect-Aux-Net is able to achieve higher scores for all classes as compare to the other segmentation models. TABLE II shows the performance of the various networks on TekErreka dataset. Experimental results from TABLE II showed that the proposed multi-task-learning can improve the performance of its corresponding single task model. Taking advantages of the classification-guidance module, Defect-Aux-Net avoids the over-segmentation of defects in complex background.

G. Experiments on Defect Classification

We evaluated and compared the classification task performance of proposed approach with the state-of-the-art deep learning architectures. While evaluating classification task, other two modules: segmentation and detection are removed from the network. Results of the experiments are summarized in TABLE III. It can be noted that the most errors are due to false positives. The visual similarity between defects and surface noise leads to false positive errors. Notably, Defect-Aux-Net obtains overall accuracy of at least 92.9% and at most 99.4% across all defect types on Severstal dataset. Based on the experimental results, we observe that the proposed multi-task learning approach achieves a surpassing performance over the other models. Also, it is evident that incorporating segmentation task improves the performance of classification task and vice-versa.

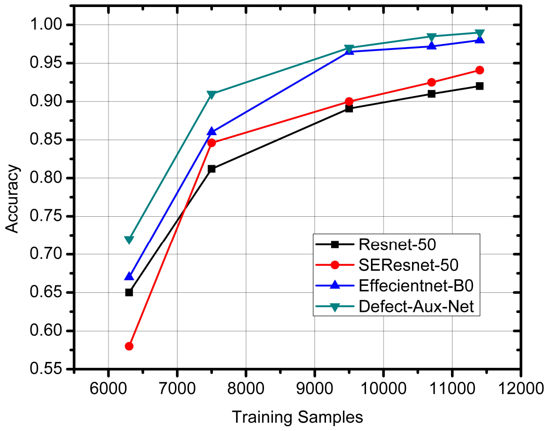


Fig. 10. Training data size vs. classification accuracy of Severstal dataset.

To assess the effectiveness of the proposed approach against limited data problem, we removed part of the training data and conducted series of experiments leaving 90%, 75%, and 50% from the training data. The effect of training data size on its accuracy is shown in Fig. 10. The proposed Defect-Aux-Net showed a consistent performance even when only 50% of the original training data is used in training. As seen, the proposed multi-task loss function greatly improves performance of the classification task by talking image, pixel, and map level optimization into the consideration.

To verify the importance of the attention mechanisms in Defect-Aux-Net, we compared accuracy the network with and without spatial and channel attention mechanism (squeeze and excite) on TekErreka dataset, as shown in TABLE IV. Further, we experimented with inserting combination of both spatial and channel attention mechanisms.

TABLE III
COMPARISION OF PERFORMANCE OF DEFECT-AUX-NET AND STATE-OF-THE-ART CLASSIFICATION MODELS

Model	Dataset	Class	Recall	Precision	F1-Score	Accuracy
Resnet-50 [2]	Severstal	Class1	0.454	0.403	0.427	0.831
		Class2	0.591	0.533	0.561	0.958
		Class3	0.918	0.847	0.881	0.811
		Class4	0.857	0.852	0.854	0.963
	TekErreka	Class1	0.759	0.979	0.855	0.949
SEResnet-50 [24]	Severstal	Class1	0.508	0.556	0.531	0.875
		Class2	0.617	0.580	0.598	0.970
		Class3	0.980	0.816	0.891	0.817
		Class4	0.559	0.940	0.701	0.940
	TekErreka	Class1	0.803	0.968	0.878	0.955
Effecient net-B0 [5]	Severstal	Class1	0.891	0.859	0.875	0.964
		Class2	0.872	0.732	0.796	0.984
		Class3	0.943	0.963	0.953	0.929
		Class4	0.946	0.924	0.935	0.983
	TekErreka	Class1	0.858	0.928	0.892	0.958
Defect-Aux-Net (ours)	Severstal	Class1	0.891	0.926	0.908	0.975
		Class2	0.957	0.900	0.928	0.994
		Class3	0.982	0.929	0.955	0.929
		Class4	0.946	0.940	0.943	0.985
	TekErreka	Class1	0.887	0.939	0.912	0.971

TABLE IV
EFFECT OF USING ATTENTION MECHANISMS ON TEKERREKA DATASET

Model	Accuracy	Parameters (M)
Defect-Aux-Net (without attentions)	0.962	33.2
Defect-Aux-Net (with SE attention)	0.968	35.7
Defect-Aux-Net (Spatial attention)	0.963	33.5
Defect-Aux-Net (with SE + Spatial attention)	**0.971**	**36.2**

H. Experiments on Defect Detection

The proposed is compared with other object detection algorithms on the TekErreka dataset. The comparative models include SSD [8], RetinaNet [25], and cascade R-CNN [30]. Fig. 11 shows the mAP scores of the various detection models for the TekErreka dataset. We observe that Defect-Aux-Net is able to achieve higher mAP score as compared to the alternative networks. The mAP of the proposed algorithm is 17.95%, 43.77%, and 26.03% higher than that of RetinaNet, SSD and Cascade RCNN.

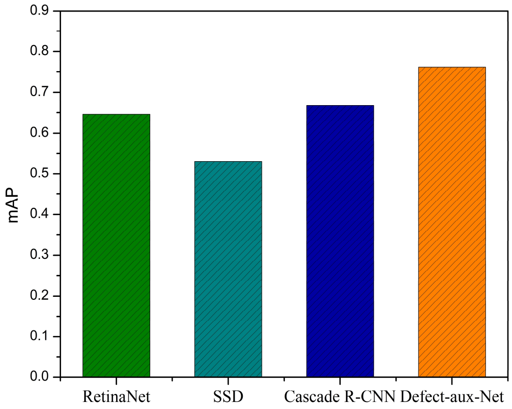


Fig. 11. mAP comparison between the state-of-the-art detection models and the proposed.

I. Inference Time

In addition to the model performance, we attempt to determine the effectiveness of multi-task learning framework on the inference time. We compared inference time of the proposed approach with conventional single task network where each task requires a separate pass through the network during inference. All the inference time was measured using a computer with an Intel Core processor. The CPU specification is summarized in TABLE V.

TABLE V
SYSTEM SPECIFICATION

CPU Specification	
CPU Processor type	Intel(R) Xeon(R)
Processor Base Frequency	2.20 GHz
Total Cores	1

From the TABLE VI, we can see that our proposed framework allows for a 57.1% reduction in the model size by solving different tasks jointly rather than independently. Compared to the single task network, the inference time of our proposed network reduce by 45.5%.

TABLE VI
COMPARISION OF INFERENCE TIME OF DEFECT-AUX-NET AND BASELINE MODEL

Model	Task	Task Name	Inference time CPU (s)	Parameters (M)
Single Task Networks	Task 1	Classification (ResNet-50)	0.0654	23.5
	Task 2	Segmentation (ResNet-50 FPN)	0.1106	26.9
	Task 3	Detection (ResNet-50 RetinaNet)	0.1780	34.0
	Total	Classification + Segmentation + Detection	0.3540	84.4
Multitask Network	Multitask	Classification + Segmentation + Detection (Defect- Aux-Net)	**0.1927**	**36.2**

V. Discussion

By incorporating multi-task learning strategy, our proposed Defect-Aux-Net improves the performance of defect classification, segmentation and detection tasks. Intuitively multi-task deep learning system can provide regularization effects to the multi-scale feature learning and thus improve the performance as opposed to the single task algorithms. Also, the multi-task learning framework can save computational inference time as only single network needs to be evaluated for three different tasks. The experimental results show that our proposed algorithm greatly improves the performance of the surface defect identification tasks compared to other state-of-the-art deep learning algorithms.

VI. Conclusion

In this work, we described an attention guided multi-task learning scheme which combines classification, segmentation and defection for automated surface defect detection. Specifically, we proposed an extended FPN architecture with Resnet-50 incorporated as the encoder section of the model. The hybrid loss function is introduced to enhance the performance of the model. An overall accuracy of 97.1%, Dice score of 0.926 and mAP of 0.762 on classification, segmentation and detection tasks of TekErreka dataset were achieved with Defect-Aux-Net.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, vol. 2016-Decem, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[3] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826, doi: 10.1109/CVPR.2016.308.

[4] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2261–2269, doi: 10.1109/CVPR.2017.243.

[5] M. Tan and Q. Le, "{E}fficient{N}et: Rethinking Model Scaling for Convolutional Neural Networks," in *Proceedings of the 36th International Conference on Machine Learning*, 2019, vol. 97, pp. 6105–6114.

[6] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[7] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, Feb. 2020, doi: 10.1109/TPAMI.2018.2844175.

[8] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 21–37.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[10] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," 2015, pp. 234–241.

[11] S. Seferbekov, V. Iglovikov, A. Buslaev, and A. Shvets, "Feature Pyramid Network for Multi-class Land Segmentation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 272–2723, doi: 10.1109/CVPRW.2018.00051.

[12] X. Ni, Z. Ma, J. Liu, B. Shi, and H. Liu, "Attention Network for Rail Surface Defect Detection via Consistency of Intersection-over-Union(IoU)-Guided Center-Point Estimation," *IEEE Trans. Ind. Informatics*, vol. 18, no. 3, pp. 1694–1705, Mar. 2022, doi: 10.1109/TII.2021.3085848.

[13] D. Zhang, K. Song, Q. Wang, Y. He, X. Wen, and Y. Yan, "Two Deep Learning Networks for Rail Surface Defect Inspection of Limited Samples With Line-Level Label," *IEEE Trans. Ind. Informatics*, vol. 17, no. 10, pp. 6731–6741, Oct. 2021, doi: 10.1109/TII.2020.3045196.

[14] L. Wen, Y. Wang, and X. Li, "A New Cycle-consistent Adversarial Networks with Attention Mechanism for Surface Defect Classification with Small Samples," *IEEE Trans. Ind. Informatics*, pp. 1–1, 2022, doi: 10.1109/TII.2022.3168432.

[15] "Kaggle. Severstal: Steel Defect Detection. Can You Detect and Classify Defects in Steel?," 2019. .

[16] M. S. Kim, T. Park, and P. Park, "Classification of Steel Surface Defect Using Convolutional Neural Network with Few Images," in *2019 12th Asian Control Conference (ASCC)*, 2019, pp. 1398–1401.

[17] H. Lin, B. Li, X. Wang, Y. Shu, and S. Niu, "Automated defect inspection of LED chip using deep convolutional neural network," *J. Intell. Manuf.*, vol. 30, no. 6, pp. 2525–2534, Aug. 2019, doi: 10.1007/s10845-018-1415-x.

[18] X. Tao, D. Zhang, W. Ma, X. Liu, and De Xu, "Automatic metallic surface defect detection and recognition with convolutional neural networks," *Appl. Sci.*, vol. 8, no. 9, pp. 1–15, 2018, doi: 10.3390/app8091575.

[19] J. Ren and X. Huang, "Defect Detection Using Combined Deep Autoencoder and Classifier for Small Sample Size," in *2020 IEEE 6th International Conference on Control Science and Systems Engineering (ICCSSE)*, 2020, pp. 32–35, doi: 10.1109/ICCSSE50399.2020.9171953.

[20] J. Lian *et al.*, "Deep-Learning-Based Small Surface Defect Detection via an Exaggerated Local Variation-Based Generative Adversarial Network," *IEEE Trans. Ind. Informatics*, vol. 16, no. 2, pp. 1343–1351, Feb. 2020, doi: 10.1109/TII.2019.2945403.

[21] D. Zheng *et al.*, "A Defect Detection Method for Rail Surface and Fasteners Based on Deep Convolutional Neural Network," *Comput. Intell. Neurosci.*, vol. 2021, pp. 1–15, Jul. 2021, doi: 10.1155/2021/2565500.

[22] P. Xu, Z. Guo, L. Liang, and X. Xu, "MSF-Net: Multi-Scale Feature Learning Network for Classification of Surface Defects of Multifarious Sizes," *Sensors*, vol. 21, no. 15, p. 5125, Jul. 2021, doi: 10.3390/s21155125.

[23] D. Amin and S. Akhter, "Deep Learning-Based Defect Detection System in Steel Sheet Surfaces," in *2020 IEEE Region 10 Symposium (TENSYMP)*, 2020, pp. 444–448, doi: 10.1109/TENSYMP50017.2020.9230863.

[24] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141, doi: 10.1109/CVPR.2018.00745.

[25] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020, doi: 10.1109/TPAMI.2018.2858826.

[26] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2999–3007, doi: 10.1109/ICCV.2017.324.

[27] L. Smith, "A disciplined approach to neural network hyper-parameters: Part 1 -- learning rate, batch size, momentum, and weight decay," 2018.

[28] A. Chaurasia and E. Culurciello, "LinkNet: Exploiting encoder representations for efficient semantic segmentation," in *2017 IEEE Visual Communications and Image Processing (VCIP)*, 2017, pp. 1–4, doi: 10.1109/VCIP.2017.8305148.

[29] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid Scene Parsing Network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6230–6239, doi: 10.1109/CVPR.2017.660.

[30] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving Into High Quality Object Detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6154–6162, doi: 10.1109/CVPR.2018.00644.

Vision Transformer based knowledge distillation for fasteners defect detection

Vignesh Sampath
Smart and Autonomous System Unit
Tekniker, Member of Basque Research & Technology alliance
Eibar, Spain
vignesh.sampath@tekniker.es

Iñaki Maurtua
Smart and Autonomous System Unit
Tekniker, Member of Basque Research & Technology alliance
Eibar, Spain
inaki.maurtua@tekniker.es

Juan José Aguilar Martín
Department of Design and Manufacturing Engineering
University of Zarazoga, School of Engineering and Architecture
Zarazoga,Spain
jaguilar@unizar.es

Ander Iriondo
Smart and Autonomous System Unit
Tekniker, Member of Basque Research & Technology alliance
Eibar, Spain
airiondo@tekniker.es

Iker Lluvia
Smart and Autonomous System Unit
Tekniker, Member of Basque Research & Technology alliance
Eibar, Spain
iker.lluvia@tekniker.es

Andoni Rivera
Smart and Autonomous System Unit
Tekniker, Member of Basque Research & Technology alliance
Eibar, Spain
andoni.rivera@tekniker.es

***Abstract*— Computer vision based visual inspection systems are gaining enormous importance for manufacturing quality control in recent years due to the advent of Convolutional neural networks (CNN) and transformer-based (vision) models. CNN based models attempt to extract global features by gradually increasing the receptive field, while long-range dependencies are ignored. Therefore, CNN recognizes objects based on the texture instead of the shape. Transformer models, on the other hand, enable modeling long range dependencies using self-attention mechanism. But learning ability of spatial information inside each patch is limited, which means it can disregard a significant spatial local pattern, such as texture. In this work, we propose to combine transformer-based and CNN-based models to take advantage of the strengths of both methods. To meet inference time constraints of real time defect classification tasks, we exploit knowledge distillation method (KD) using softened logits of ensemble model as supervision to train a lightweight CNN model (Resnet18). The study showed that the proposed vision transformer-based KD approach overcome the requirements of limited computational resources and can be deployed on low-power and resource limited devices. The experimental results also showed that proposed framework outperforms in terms of mean accuracy on the test datasets compared to stand-alone CNN methods.**

Keywords—Convolutional Neural Networks, Vision transformer, Defect classification, Machine Vision, Deep learning

I. Introduction

Threaded fasteners are among the most standard fundamental components in the manufacturing industry, providing the functions of sealing and connecting two or more pieces together [1]. With the advent of industry 4.0, the quality requirements of threaded fasteners are also getting higher. Quenching and drawing (tempering) are the most widely used heat treatment processes for steel fastener manufacturing. The threaded fasteners should be tempered as soon as they are being removed from the quench and before they reach a room temperature. Failure to do so could result in quench crack and premature failure. A survey showed that 23% of the service problems in automotive industries could be attributed to the threaded fastener failure [2]. Occasional failure of fasteners in the aerospace industry could cause a fatal consequence.

Magnetic particle inspection (MPI) is the most widely used non-destructive testing (NDT) method to identify detects in the threaded fasteners. In the case of ferromagnetic fasteners production, magnetic particle inspection allows detecting some surface and near-surface defects which are not otherwise visible to the human eye. The identification of these defects is done by qualified operators by visual inspection of the parts once the magnetic particles are applied. However, manual visual inspection requires a great deal of concentration from the operators, so that good production quality is continuously guaranteed. On the other hand, due to fatigue of the operators, small parts, small details, hazardous inspection conditions, poor lighting conditions and process complexity result in uncertainty and reduced precision during inspection.

In recent days, computer vision-based defect detection models based on CNN and transformer models have achieved state of the art performance in terms of its accuracy [3]. CNNs are composed of successive convolutional and pooling layers, followed by fully connected layers, and excel at capturing local features. Yet CNN models are inferior at modeling long-range dependencies compared to transformer-based models. Transformer models, on the hand, are better in capturing long-term dependencies using the self-attention mechanisms. However, learning ability of spatial information inside each patch is limited, which means it can disregard a significant spatial local pattern, such as texture.In this work, we propose to combine transformer-based and CNN-based models to take

978-1-6654-7087-2/22/$31.00 ©2022 IEEE

advantage of the strengths of both methods using ensemble technique.

Although ensemble models improve predictive performance, they contain millions of parameters and operations per inference, making them memory and computation intensive networks. Such computationally heavy networks hinder their deployment in low power or resource limited devices with strict latency requirements. To compress these heavy networks, several techniques such as model pruning [4], model quantization [5], and knowledge distillation [6] have been proposed in the literature.

- **Pruning** method aims at removing filters and their respective connections with weights close to zero. The main advantage of this method is that it does not introduce any sparsity in the model weight matrices.
- **Quantization:** Models are typically trained in a higher precision with 32-bit floating point operations and hence weights and activations are represented in FP32. Quantization technique aims at reducing the model size by representing weights and activations in lower-precision numerical formats.
- **Knowledge distillation:** KD involves training a smaller compact student network under the supervision of a larger pretrained teacher model or an ensemble of larger models (teachers) in an interactive manner.

Our study focuses on utilizing the advantages KD and quantization techniques to build fast and lightweight CNN models that can be deployed in low-power and resource limited devices for the real-time fastener defect detection system. Considering above mentioned limitations, we employ the compact smaller CNN model (Restnet18) as a student network and larger pretrained transformer/CNN model as a teacher network in KD framework. To study effect of combining CNN and transformer-based model we choose one of the modern transformer-based models, Swin-transformer [7], and one of the cutting-edge CNN-based models, Efficientnet [8]. The main article contributions are summarized below.

1. Collection of magnetic particle inspection-based fasteners defect classification dataset consisting of thousands of images captured from a high-resolution industrial camera.
2. We present a KD framework which improves defect classification performance and generalization ability of small and compact convolutional neural networks by distilling knowledge from vision transformer models.
3. We perform ablation study of our framework on self-collected fasteners (Tek-Erreka) defect dataset, in terms of different combinations of CNN and transformer models and ensemble methods. Experiments show that by employing multiple teacher models using heterogeneous combinations of CNN and Transformer models in feature learning, our KD based compact student model produce superior performance compared to the model without using knowledge distillation.

II. Fasteners defect identification system

A. Overview

Our defect detection system consists of a stepper motor, magnetic suspension spray, magnetization system, ultra violet lambs, cameras and computer. The system is designed to guarantee the inspection of the whole surface of the fasteners in a real time conditions. Multiple cameras are used to ensure that defect is vible and captures in at least one acquired images of the fastener. MPI is an extermely reliable method that uses the behaviour of the magnetic field distribution to identify defects in the ferro-magnetic materials. In a component without defects, the magnetic field lines are undistrubed and drawn in to the object. In the case of componets with defects, the magnetic field lines can only run straight within the undamaged area. They cannot bride the air-grap formed by the defects or cracks, and therefore bend away from it back into the material. This phenomenon is known as magnetic-flux leakage. When the fine iron particles are applied to the component under test, the particles are attracted and clustered together thus indicating the location of the defect. The fine iron particles can be applied either dry or suspended within a fluid. In this work, the wet flourasent magnetic particle inspection (suspended within a fluid) method is considered. MPI procedure consists of four main steps: The first step is to clean the test pieces before inspection with a solvent degreaser to remove all contaminations. Then, the second step is to magnetize the fastener under inspection. When the fastener is magnetized, the magnetic field tends to distribute themselves evenly through the material. The third step is to apply fine iron particles suspended within the fluid. The iron particles stick to the region of the flux leakage thus indicating the exact position of the surface crack. The fluorescent dye is added to the iron particle and hence it glows brightly under ultraviolet light. Finally, after the inspection is completed, fasteners are run through a demagnetizer to remove or reduce the residual magnetism to within the allowable limits of the applicable specification. They are also post cleaned in a solvent degreaser and coated with a light rust preventative oil.

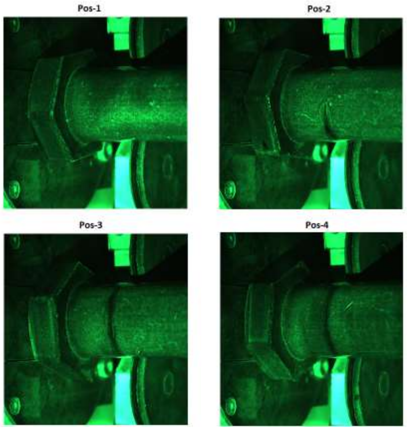


Fig 1. Images acquired from the four different rotational positions for the same Fastener

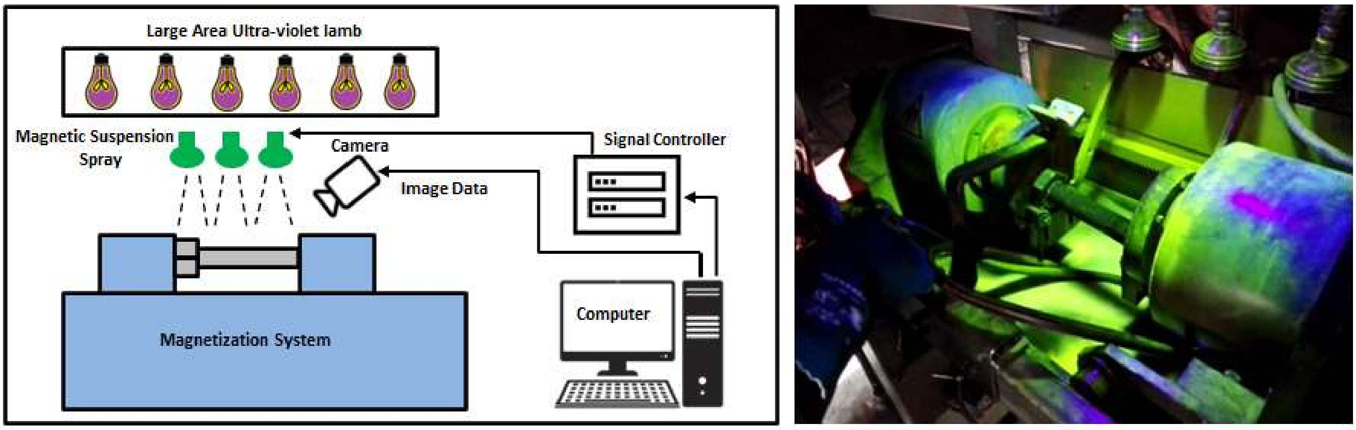


Fig 2. Overall image acquistion system description of wet flouresent magnetic paticle inspection method

B. Image acquisition system design

In MPI, for parts with complex structures such as fasteners, it is difficult to capture all the key information based on a single image. In order to enable inspection of the whole surface of the fasteners, the stepper motor is used to perform one complete 360^0 rotation around the fastener axis. During the rotation of fastener, a signal from the controller is triggered to acquire images at different rotational positions. A total of 4 rotational positions was found to be sufficient to inspect the whole surface of the fasteners.

There is a large variety of defects that are located on the surface of manufactured fasteners. This variety is generally studied by the size of the defect, its shape, and its probable cause. For instance, some of these defects can be identified as marks, scratches, geometrical deformation, etc. Hence, the image acquisition system should be able capture all different kinds of defects.

III. Proposed Method

Fasteners' defect images are characterized by complex spatial layout and background objects. The appearance of defect varies with material types, camera position, magnetic particle size and lighting condition. Using CNN model alone can perform automatic feature extraction, which improves performance compared to manual feature extraction. However due to the nature of convolution operation, the extracted feature maps are locally sensitive; that is, CNNs lack modeling long-range dependencies. So, it is an efficient approach to further boost performance of CNN models via exploring long-range dependencies. To solve this problem, our proposed method employs transformer-based and CNN-based model ensemble, which effectively model both local and global features. KD framework is used to distill the knowledge from larger ensemble models to smaller network. The framework of our proposed method is shown in Fig 3. As shown in the Fig 3, the proposed method consists of four main components, including teacher models, student model, pruning and quantization. Due to the high capacity to learn long range spatial relations and spatial local patterns, the state-of-the-art Swin transformer [7] and Efficientnet-B7 [8] models are introduced as a teacher model. Resnet18 [9] is adapted as student model to match the average probability distribution of the predictions of the teacher models.

A. Teacher Network:

Swin transformer stands for Shifted window and is based on visual transformer (ViT) [3]. In VIT the image is decomposed into 16x16 pixel patches, and then these patches are transformed into patch vectors by a linear transformation. These patch vectors combined with the positional embeddings, are processed by a transformer. But the computational complexity of ViT increases quadratic to the image size as it computes self-attention globally. To overcome this problem, Swin transformer uses the concept of shifted window attention and hierarchical feature maps that adds a linear computation complexity to the input image size. Attention mechanism in the transformer is complementary to convolution layers in CNN. It constructs hierarchical feature maps by merging image patches into deeper layers. Swin transformer computes self-attention only within the local window. They can model long-range dependency relations between sequences through shifted window attention. Swin transformer backbone is capable of modeling pixel to pixel, object to pixel, object to object relationships.

Efficientnet-B7 (CNN-based model) combines low level features to increasingly complex shapes until the defects in the fasteners can be classified. It can reach impressive performance on defect detection tasks. Overall, the family of EfficientNet models achieves both higher accuracy and better efficiency over existing CNNs. Convolution operation used in Efficientnet is being local that the convolution layer has difficulty extracting long range pixel to pixel dependencies. But they are strongly biased towards recognizing textures of the defects rather than its shapes [10]. We train both Swin transformer and Efficientnet B7 using identical training data, and the ensemble prediction results of both the models.

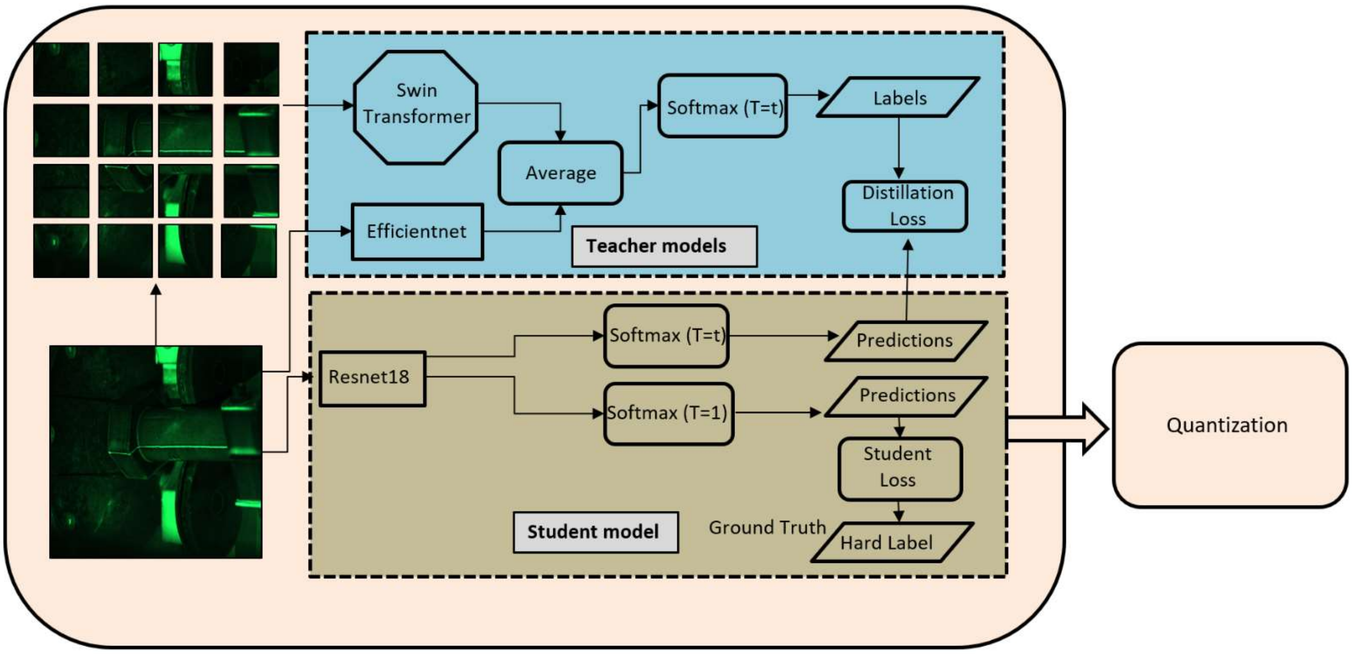


Fig 3. An overview of proposed Knowledege distillation framework. It is mainly composed of Ensemble teacher models, Student model and quantization.

B. Student Network:

The state-of-the-art CNN models have very dense layers that are costly in terms of memory and computation. It is not possible to deploy these complex networks because of high computational cost and energy usage on edge devices. Therefore, we use light weight Resnet-18 models as student network. Resnet-18 is a relatively small network based on modern standards; therefore, it is suitable for low power edge device

C. Knowledge distillation:

The main idea behind KD is that the probability distribution of the predictions of the "teacher" network contains a lot more information about a data than the original class labels. Therefore, the teacher models are trained first using a standard cross entropy loss ($L_{CE}(P_t, y)$) that seeks to minimize the overall loss. Where $P_t = f_t(x)$ represent the ensemble logits of the teacher models and y is the ground truth label.In most of the cases, probability distribution of the teacher models has the correct ground truth class at a very high probability. Therefore, it may not provide much additional information beyond the ground truth labels. To overcome this problem, Hinton et al [6] presented the concept of "SoftMax temperature" that soften this probability distribution using temperature scaling.The softened probability P_i from the logit z can be calculated as

$$P_i = \frac{exp\,(\frac{z_i}{T})}{\Sigma_j(\frac{z_j}{T})} \qquad (1)$$

Where the term T is the temperature parameter which controls the softening of the teacher probability distribution. As T starts increasing, the probability distribution generated by the SoftMax function becomes softer.The student model seeks transferable knowledge from the teacher. Therefore, it is trained on both the softened teacher logits and the target label, using Kullback-Leibler (KL) divergence (Distillation loss) and standard cross entropy (Student loss). Our KD loss (L_{KD}) is weighted sum of KL loss and cross entropy $L_{CE}(P_s, y)$). Where $P_s = f_s(x)$ represent the logits of the student model. Mathematically, L_{KD} can be written as

$$L_{KD} = \alpha * KL(P_s, \frac{P_t}{T}) + (1-\alpha)L_{CE}(P_s, y)) \qquad (2)$$

Where, α is the weight parameter. We tested with different combinations of weight parameters and found that $\alpha = 1$ yields the best result for the defect detection task.

D. Quantization

As our focus is to reduce the model size and speed up the real-time inference on CPUs,we have employed quantization method on trained student network as a post model optimization.The idea is to reduce the model size by representing weights and activations in lower-precision numerical formats. The student model is trained in FP32 and then the model is compressed to INT8 using quantization method. By doing this, it is possible gain up to three times lower latency without taking a major hit on accuracy. Post-training quantization can result in a loss of accuracy, particularly for smaller student networks, but it is often fairly negligible. On the plus side, this will speed up execution of the heaviest computations by using lower precision and the most sensitive computations with higher precision, thus typically resulting in little or no final loss of accuracy.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

The original images of the fasteners are directly stored in a database as RGB images of size 2464 x 2056.The datasetcontains 450 positive and 1200 negative examples. In order to inflate the training sample, we include augmented, and GAN based synthetic images to the original dataset. We split TekErreka dataset into training and testing sets: 80% for training and 20% for evaluation of the model performance.

A. Experimental setup:

We resize the collected defect dataset to 224x224 and normalize the pixel values using min-max standardization. Both the teacher and student models are optimized with Adam optimizer and trained with the batch size of 64. The initial learning rate of the optimizer is set to 0.001 with the weight decay of 0.05.

B. Hardware and software:

For training our models, we use the Tensorflow framework. All the experiments are run using a computer with an Intel Core i7-8700 K processor, GPU GTX-1080Ti, and 12GB RAM.

Hardware Environment	Software Environment
GPU: GTX-1080Ti CPU: Intel Core i7-8700, 3.7 GHz, six-core twelve thread, RAM 12GB	Framework: Tensorflow and Python 3.7

Table 1 System specification

C. Evaluation metrics:

The classification results are evaluated using overall accuracy.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FP} \quad (3)$$

Where TP, TN, FP and FN refer to true positive (correctly identified fastener defect), true negative (correctly identified defect free fastener), false positive (erroneously classified as fastener defect) and false negative (erroneously classified defect free fastener).

D. Results and discussion:

Our hybrid ensemble of transformer and CNN-based models are able to identify complex defects, i.e., small, dense and overlapped defects.Using the hybrid combination, it is shown that the ensemble of transformer and CNN perform better than each single model in terms of its accuracy as reported in Table 2. From the Table 2, We can see that KD framework using both CNN-Swin Transformer and CNN-CNN ensemble teacher model outperforms standalone CNN student model in terms of accuracy. The results show that there is a significant increase in accuracy when using the combination of swin transformer and Efficientnet-B7 model. For instance, a student network trained with ensemble combinations of swin transformer and Efficientnet-B7 models improves accuracy by 2.9%, whereas the Efficientnet-B0 and Efficientnet-B7 ensemble improves accuracy only by 1.6%. These experiments show that training a student network using transformer and CNN ensemble models enable the student to learn spatial local patterns, such as texture, and long-range spatial relationship.

Model	Resolution	Accuracy	Parameter
Resnet-18 (Baseline)	224x224	0.905	11.174 M
Efficientnet-B5	224x224	0.947	30 M
Efficientnet-B7	224x224	0.968	66 M
Swin Transformer-Large	224x224	0.971	197 M
Resnet-18 KD (EFF-B5+EFF-B7)	224x224	0.920	11.174 M
Resnet-18 KD (EFF-B7+SwinTransformer teacher)(Ours)	**224x224**	**0.932**	**11.174 M**

Table 2 Performance of the student and teacher models on TEK-Erekka dataset.

To visualize the prediction of the trained student model, we use class activation maps on the training examples. As shown in Fig 4, we discover that the student network is basing predictions not on background or noise components within the fasteners, but on the defects.

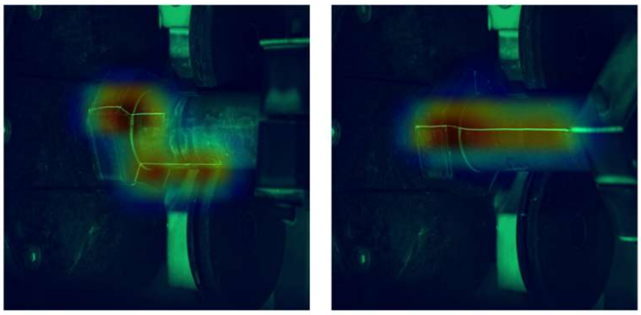


Fig 4. GRAD-CAM visualization.

Our post quantization operation on student model can reduce the compute resources required to serve the model such as mobile and IoT, where the capabilities of the device are extremely limited compared to running on a server or in the Cloud.Table 3 shows that the post quantization method allow for a 4x reduction in the model size and a 4x reduction in memory bandwidth requirements. Compared to the original full precision model (FP32), the inference time of quantized (INT8) student model was reduced by 75%.

Model	Quantization	Accuracy	Inference time CPU (ms)	Size (MB)
Resnet-18 (Baseline)	FP32	0.905	15.05	44.75
Resnet-18 (Baseline)	INT8	0.890	3.64	11.20
Resnet-18 (CNN+Swin teacher)	FP32	0.932	15.25	44.75
Resnet-18 (CNN+Swin teacher) (Ours)	**INT8**	**0.929**	**3.60**	**11.20**

Table 3 Performance of the student model before and after quantization

V. Conclusion

In this paper, we proposed fusion of CNN-based and transformer-based models for fastener defect identification using KD framework. To reduce the model size and speed up the real-time inference on edge devices, we used light weight Resnet 18 in KD framework, and also quantization method used as a post model optimization to further reduce model size. By combining CNN-based and transformer-based models, our experiments showed that our proposed KD framework outperformed the respective baseline model. We also showed that the post quantization operation on student model can reduction 4x in the model size and a 4x reduction in memory bandwidth requirements.

Acknowledgment

This research work was undertaken in the context of DIGIMAN4.0 project ("Digital Manufacturing Technologies for Zero- defect", https ://www.digim an4-0.mek.dtu.dk/). DIGIMAN4.0 is a European Training Network supported by Horizon 2020, the EU Framework Programme for Research and Innovation (Project ID: 814225). This research was also partly supported by "5R- Red Cervera de Tecnologías robóticas en fabricación inteligente", contract number CER-20211007, under "Centros Tecnológicos de Excelencia Cervera" programme funded by "The Centre for the Development of Industrial Technology (CDTI)".

References

[1] C. Matthews, "Fasteners and couplings – better design," in *Case Studies in Engineering Design*, Elsevier, 1998, pp. 86–100.

[2] H. GONG, J. LIU, and H. FENG, "Review on anti-loosening methods for threaded fasteners," *Chinese Journal of Aeronautics*, vol. 35, no. 2, pp. 47–61, Feb. 2022, doi: 10.1016/j.cja.2020.12.038.

[3] A. Dosovitskiy *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," Oct. 2020, doi: https://doi.org/10.48550/.

[4] D. Liu, H. Kong, X. Luo, W. Liu, and R. Subramaniam, "Bringing AI to edge: From deep learning's perspective," *Neurocomputing*, vol. 485, pp. 297–320, May 2022, doi: 10.1016/j.neucom.2021.04.141.

[5] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2325–2383, 1998, doi: 10.1109/18.720541.

[6] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," Mar. 2015, [Online]. Available: http://arxiv.org/abs/1503.02531.

[7] Z. Liu *et al.*, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9992–10002, 2021.

[8] M. Tan and Q. Le, "{E}fficient{N}et: Rethinking Model Scaling for Convolutional Neural Networks," in *Proceedings of the 36th International Conference on Machine Learning*, 2019, vol. 97, pp. 6105–6114, [Online]. Available: https://proceedings.mlr.press/v97/tan19a.html.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[10] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness," Nov. 2018, [Online]. Available: http://arxiv.org/abs/1811.12231.

4. SUMMARY

Summary 4

The objectives, contributions, and methodology used in this thesis are all described in this section.

4.1 Objectives

In the case of fasteners production process, magnetic particle inspection allows detecting some surface and near-surface defects which are not otherwise visible to the human eye. Fasteners of poorer quality can be produced due to a variety of production faults, including the use of poor-quality raw materials, improper plating, or heat treatment, wrong machining, irresponsible handling, and more. The identification of these defects is done by qualified operators by visual inspection of the parts once the magnetic particles are applied. Visual inspection of manufactured products has always been one of the common and most important applications of quality control in any industry. However, Visual inspection requires a great deal of concentration from the operators, so that good production quality is continuously guaranteed. On the other hand, due to fatigue of the operators, small parts, small details, hazardous inspection conditions and process complexity result in uncertainty and reduced precision during inspection.

In this line of research, the overall purpose of this PhD study is to develop a fully automated AI based vision system to inspect the whole surface of fasteners, based on the magnetic particle testing technique. An AI based visual inspection system is generally composed of three main subsystems. First, the hardware-based image acquisition subsystem. It has a role of collecting data from cameras, or other devices and provides a digital output of what is seen in the real world. Usually, suitable cameras are used to acquire images. Granular information in image data, such as color, brightness, intensity, and light scatter, can assist deep learning models in learning to comprehend what they see under all circumstances. Second, a software-based image processing subsystem based on deep learning algorithms. It mainly consists of deep learning models that are designed to analyze the acquired data and provide the final inspection result. Third, hardware-software based Pick and place robot. It makes it possible to apply automated methods for moving objects from one place to another.

This project can be divided in the following specific aims:

1. The first objective of our thesis consists in designing the image acquisition subsystem. The design of an image acquisition system for machine vision applications involves choosing the appropriate sensors and cameras, as well as the necessary hardware and software, to capture high-quality images for machine vision tasks. The design of the system should take into account the specific requirements of the application, such as the type of fasteners and scenes to be imaged, the lighting conditions, and the distance and resolution of the images. The system should also be able to capture images at a sufficient rate to enable real-time processing and analysis. Additionally, the system should be robust and reliable, and should be able to handle noise and other sources of uncertainty. There is a large variety of defect types that are located on the surface of manufactured fasteners. This variety is generally studied by the size of the defect, its shape, its location and its probable cause. For instance, some of these defects can be identified as marks, scratches, geometrical deformation, etc. Hence, the image acquisition system should capture all different kinds of defects. In order to capture all different variations of defects in the fasteners, this PhD thesis intends to create an effective image acquisition system that must be tailored to the demands of MPI. There are several difficulties with real-time surface inspection of finished fasteners. Some are imposed by characteristics of MPI, while others are imposed by the industry. The designed image acquisition system must therefore overcome these challenges.
2. The efficiency of MPI is influenced by a number of variables. 1) Particle concentration is a major consideration, 2) Contamination of suspension, 3) lightings, 4) size, shape, type, orientation, and depth of the defects. Due to the low likelihood of defects occurring, collecting defective images with various combinations of these characteristics (intra-class variances) on a wide scale is expensive. It results in a number of challenges when gathering defect data with a wide range of variability, which negatively affects the defect detection model's capacity to generalize. Furthermore, in most cases collecting an equal number of images for each class is infeasible (class imbalance), e.g., the shank portion of the fasteners may have more defects than the thread region. As a result, the second goal of this thesis is to make sure that the data accurately reflects what the defect detection must learn. The proposed approach must be capable of resolving various issues related to the data acquired. The main issues with

data include: 1. label inconsistency, 2. Poor Image quality, 3. Low variations of the acquired images (intra-class imbalance) 4. Inter-class imbalance.

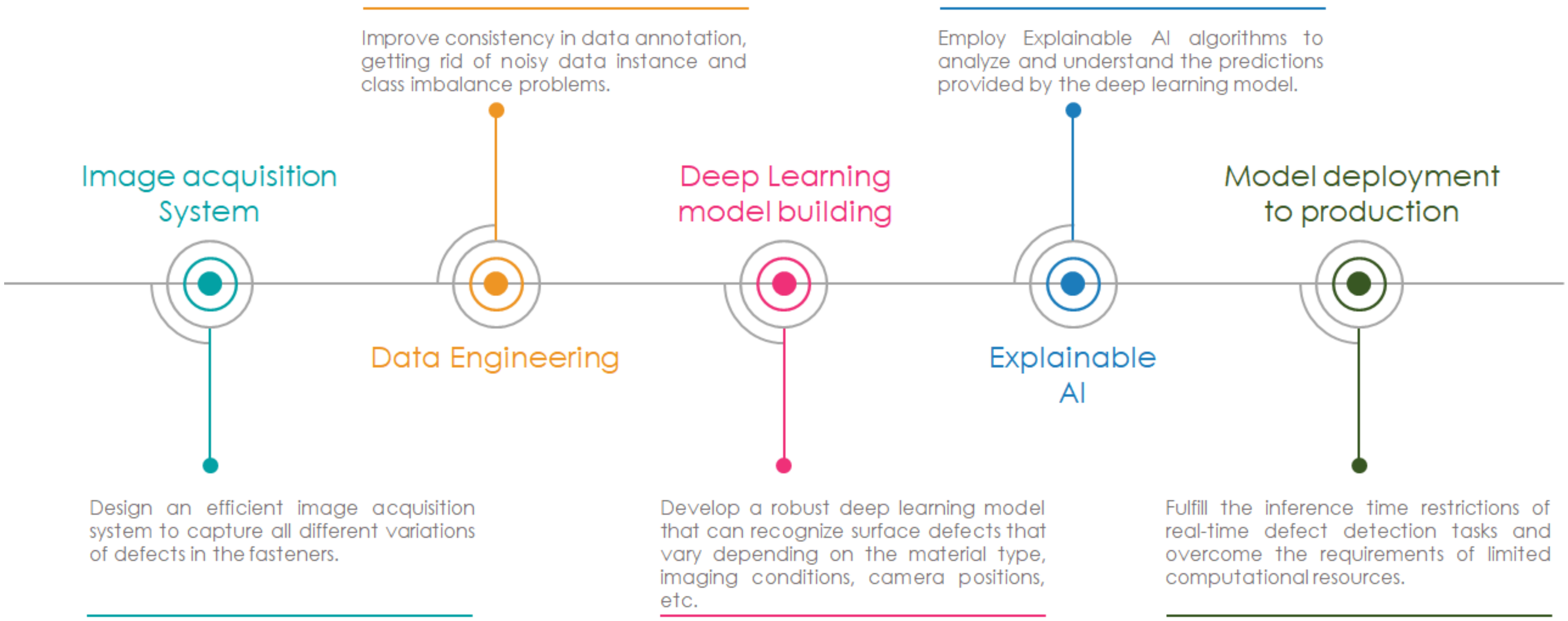
3. The third objective is to develop a surface defect inspection model based on deep learning. The target surface defects usually have different scales, making the deep learning models even more challenging to identify the small sized defects. On the one hand, visual appearance of the real-world surfaces defects varies with type of materials, imaging conditions, and camera position. The model must be general enough for the inspection of a wide range of surface defects. On the other hand, it is challenging to distinguish tiny defects from the noise or non-defect within an image. Hence the appearance of false positives in a defect free image is an inevitable circumstance. In addition, from an industrial preceptive, it is important to control the false alarm rate to prevent unnecessary production stoppage, while ensuring the highest defect detection performance. Designing the robust defect defection algorithm can help to achieve process improvement by monitoring any increase of faults and take remedial action in good time. Also, achieve increased raw material utilization and yield whilst reducing overall wastage.
4. Due to their multilayer nonlinear structure, modern machine learning models like deep neural networks are frequently referred to as "black box" models. It has millions of parameters that need to be learned, and to classify a single image, millions of operations per inference must be performed. It is challenging to comprehend how each individual neuron interacts with the others to produce the outcome. As a result, it is even unclear what a particular neuron is doing on its own. The fourth objective of this thesis is to employ Explainable AI algorithms to analyze and understand the predictions provided by deep learning algorithms. Deep learning models can be checked for sanity using explainable AI techniques. If the deep learning models can make its reasoning understandable to the quality inspectors, they can use the domain knowledge to make sure that the AI model is focusing on correct predictors for the task at hand.
5. The final objective consists in building a defect detection application that can be used on production lines. This application should be capable of manipulating and inspecting several parts per minute and provides more consistent and reliable inspection results than human inspectors. During the quality inspection, fasteners are handled and placed on a test bed by a pick-and-place robot, a type of industrial robot. They are frequently utilized in high-volume industrial environments where they can quickly and accurately perform quality checks. The developed model must have a high throughput and latency in order for the

application to integrate with a pick-and-place robot. A pictorial representation of the objectives of our thesis is shown in Figure 4.1.

4.2 Scientific Contributions

The main contributions of this PhD study to the scientific community are:

- An updated review of GAN for imbalance problems in the computer vision tasks. It provides necessary material to inform research communities about the latest development and essential technical components in the field of GAN based synthetic image generation. Furthermore, It highlights real world applications where GAN based synthetic images are used to alleviate imbalances and fills a research gap in the use of synthetic images for the imbalance problems in visual recognition tasks.
- A systematic approach to classify ML tasks from both 'Four-Know' (Know-what, Know-why, Know-when, Know-how) and 'Four-Level' (Product, Process, Machine, System) perspectives. To bridge the gap between academic research and the manufacturing industries, the study provides a practical pipeline for production engineers to use when implementing ML solutions, from conception to deployment.
- One of the scientific contributions of this thesis is the development of a collection of magnetic particle inspection-based fasteners defect classification dataset consisting of thousands of images captured from a high-resolution industrial camera. We refer to this dataset as the "TekErreka dataset". This dataset is particularly useful for training deep learning algorithms for the purpose of identifying and classifying defects in fasteners using magnetic particle inspection. The high-resolution industrial camera ensures that the images are of sufficient quality for accurate defect classification, and the large number of images in the dataset allows for robust training of deep learning models. This dataset is a valuable resource for industrial practitioners in the field of magnetic particle inspection and defect classification.
- It develops a novel Pixel level image augmentation method that is based on mask-to-image translation with GAN conditioned on fine grained labels. The proposed Magna-Defect-GAN model can gain control over the image generation process and generate image samples with photorealistic variations. Experimental results demonstrate that the proposed Magna-Defect-GAN model can generate realistic and high-resolution surface defect images up to the resolution of 512 × 512 in a controlled manner. It also shows that the proposed



Design an efficient image acquisition system to capture all different variations of defects in the fasteners.

Improve consistency in data annotation, getting rid of noisy data instance and class imbalance problems.

Develop a robust deep learning model that can recognize surface defects that vary depending on the material type, imaging conditions, camera positions, etc.

Employ Explainable AI algorithms to analyze and understand the predictions provided by the deep learning model.

Fulfill the inference time restrictions of real-time defect detection tasks and overcome the requirements of limited computational resources.

Fig. 4.1: A pictorial representation of the five distinct goals that make up this thesis objective.

augmentation method can boost accuracy and be easily adopted to any other surface defect identification models.

- It proposes a novel deep learning model named "Defect-Aux-Net" based on multi-task learning with attention mechanisms that take advantage of the rich additional information from related tasks in order to simultaneously improve the robustness and accuracy of the CNN-based surface defect identification. According to the experimental findings, the Defect-Aux-Net is capable of enhancing the performance of state-of-the art models while obtaining an overall accuracy of 97.1 %, a Dice score of 0.926, and a mAP of 0.762 on defect classification, segmentation, and detection tasks.
- It studies the combination of knowledge distillation and post quantization methods to significantly reduce the size and improve the speed of real-time inference for machine learning models on edge devices. The proposed method first use knowledge distillation to train a smaller model, and then apply post quantization to this smaller model. The experimental result shows that the proposed method can achieve 4x reduction in the model size and a 4x reduction in memory bandwidth requirements.

4.3 Methodology

Our proposed solution methodology involves the combination of several approaches in order to create a robust computer vision application. A pictorial illustration of our proposed methodology is shown in Figure 4.2. These approaches include:

1. Image Acquisition System: The first step in our proposed methodology is to design a reliable image acquisition system for acquiring images from the MPI process. We proposed to use the combination of both frame and line scan cameras in an image acquisition system to capture the head and shank portion of rotating fasteners respectively. One of the main advantages of this combination is the ability to capture high resolution images of both the head and shank of the fasteners, which is important for ensuring the quality and integrity of the fasteners. The use of a line scan camera to capture high resolution images of the shank portion of the fasteners allows for more detailed analysis of the surface finish, and dimensional tolerances, which can help identify potential defects or issues that may affect the performance of the fasteners. Line scan cameras also enable capturing images of the complete surface of a fastener of revolution while it is rotating. Another benefit of this methodology is the ability to capture images of the fasteners at high speeds,

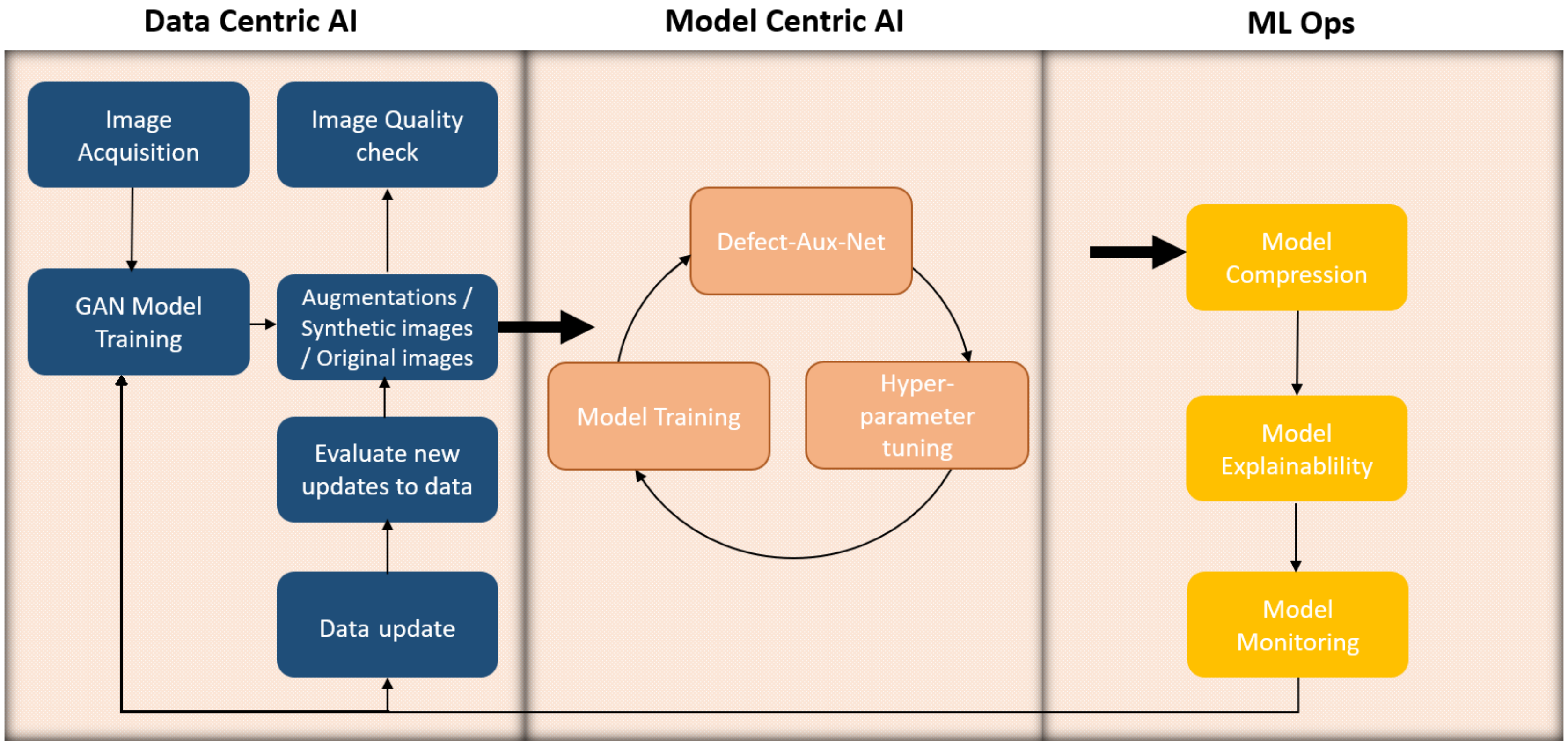


Fig. 4.2: A pictorial illustration of our proposed methodology.

which is important for applications where fasteners are being produced or assembled at a high rate. The use of a frame camera to capture a wide field of view of the fasteners allows for efficient image acquisition, while the use of a line scan camera allows for high resolution images to be captured even at high speeds. In addition to these benefits, the proposed methodology also offers the potential for improved accuracy and repeatability in the image acquisition process. By using both frame and line scan cameras, it is possible to capture images of the fasteners from multiple angles and viewpoints, which can help to improve the accuracy and reliability of the image analysis. To sum up, the proposed methodology of combining both frame and line scan cameras in an image acquisition system to capture the head and shank portion of rotating fasteners represents a significant advancement in the field of fastener inspection and quality control. By offering the ability to capture high resolution images at high speeds, as well as improved accuracy and repeatability, this methodology has the potential to significantly improve the efficiency and effectiveness of fastener inspection processes.

2. Data centric deep learning approach: In the data-centric approach, we used combination of both traditional data augmentation and GAN based synthetic images to expand the size and diversity of the training dataset. Data augmentation is used to artificially expand the size of the training dataset by generating new, synthesized samples that are variations of the original samples. This can help to improve the generalization ability of the model and reduce overfitting. There are several ways in which data augmentation can be used in the context of data-centric approach. One common method is to apply transformations to the existing images to create new, synthetic images. For example, an image dataset can be augmented by applying different rotations, translations, or scaling to the existing images, or by adding noise or blur to the images. We also used GAN-based synthetic images to generate additional samples for the training dataset. GANs are a type of neural network that can learn to generate synthetic images that are indistinguishable from real images. By using GANs to generate synthetic images of defects, it is possible to greatly expand the size and diversity of the training dataset, which can improve the accuracy of the model. By combining these two approaches, we leverage the strengths of both approaches to achieve improved performance in defect detection. The data-centric approach helps to increase the size and diversity of the training data, while the model-centric approach allows the model to learn from multiple sources of information and to generalize better to new tasks.

3. Model centric deep learning approach: Our proposed model-centric approach centered on the use of multi-task learning, which involves training a single model to perform multiple related tasks simultaneously. This can help to improve the performance of the defect detection model by allowing it to learn from multiple sources of information and to generalize better to new tasks. By learning from multiple types of data simultaneously, our proposed multi-task learning model performs better than a single-task model. This is because the model can learn from the shared features of the different tasks, as well as task-specific features, which can improve its overall performance. Additionally, by using a multi-task learning model, it is possible train a single model that can handle multiple tasks simultaneously, which is more efficient. Also, our proposed multi-task learning model learns to generalize better to new data, as it has been trained on multiple tasks and has learned to identify shared features between them. This can be especially useful in defect detection, where the model may need to identify defects in new or unseen contexts. Furthermore, our proposed multi-task learning model can be more interpretable than a single-task model, as it can provide insights into the shared features that are important for multiple tasks. This can be useful for identifying common patterns or features that may be indicative of defects. To sum up, by combining both model and data centric approaches, it is possible to leverage the strengths of both approaches to achieve improved performance in defect detection. The data-centric approach helps to increase the size and diversity of the training data, while the model-centric approach allows the model to learn from multiple sources of information and to generalize better to new tasks.
4. Explainable AI: In order to make the defect detection model more interpretable and explainable, we studied and compared heatmaps of four different explainable deep learning methods including GradCAM, GradCAM++, Score-CAM, and Guided GradCAM that indicate the regions in the input image that are most relevant to the model's prediction. GradCAM is a gradient-based technique that uses the gradients of the target class with respect to the activations of the final convolutional layer to compute a heatmap. GradCAM++ is an extension of GradCAM that takes into account activations from multiple layers. Score-CAM uses scores of the target class to compute the heatmap while Guided GradCAM uses a guided backpropagation technique to highlight the regions in the input image that are most important for the model's prediction. After comparing the heatmaps generated by these methods on a defect detection dataset, we found that GradCAM generated the most interpretable and explainable heatmaps for our case and thus, chose it to use for our defect detection application.

5. We employed the combination of knowledge distillation and post quantization methods to reduce the size and improve the speed of real-time inference for deep learning based defect detection models on edge devices. To implement this method, the larger model is first used to generate soft labels for the training data. The smaller model is then trained using these soft labels, with the aim of replicating the behavior of the larger model as closely as possible. Once the smaller model has been trained, it is quantized using post quantization techniques to further reduce its size and improve its performance on edge devices.

4.3.1 Image acquisition system

Image acquisition is crucial in computer vision applications because it provides the raw sensory data that is used to perceive and understand the environment. Unquestionably, it is the most important step in the computer vision application workflow because a bad image will make the entire process useless. Acquiring an image with the proper clarity and contrast is essential since the computer vision systems only evaluate the digital image of the fasteners that has been captured, not the actual fasteners. The quality and resolution of the captured images directly impact the performance and accuracy of the machine vision algorithms and models. Poor quality or low resolution images may result in inaccurate or unreliable results, whereas high quality and high resolution images can enable the algorithms and models to extract more detailed and discriminative features from the images. Additionally, the image acquisition system needs to be efficient and scalable, to ensure the captured images can be processed and analyzed in real-time, without causing delays or bottlenecks.

The design of an image acquisition system for machine vision applications involves selecting and configuring the hardware and software components that are used to capture and process images from the environment. This may involve selecting the type and resolution of the cameras, as well as the lighting and illumination conditions. The image acquisition system may also include preprocessing and data preparation steps, such as image scaling, normalization, and noise reduction, to ensure the input data is in the correct format and ready for further processing. Additionally, the design of the image acquisition system may involve choosing the type and configuration of the storage and transmission systems, to ensure the captured images can be stored and accessed efficiently.

General requirements: Understanding and defining the system and magnetic particle inspection requirements is the first stage in designing an image acquisition system. After that, begin with a conceptual design that offers some first suggestions for how to satisfy the previously established criteria, and eventually, lay down the fundamental elements of the entire system. There are several general requirements that an image acquisition system should meet in order to be effective.

- First and foremost, the image acquisition system should be able to capture images with a high level of detail and resolution. This is necessary in order to accurately identify and classify defects, as well as to measure the size and shape of defects in some cases. The system should also be able to capture images at a high frame rate, in order to accurately capture fast-moving objects or processes.
- In addition to these technical requirements, the image acquisition system should also be easy to use and operate. This may include features such as an intuitive user interface, the ability to adjust image capture settings quickly and easily, and the ability to store and retrieve images for further analysis.
- The system should also be able to handle a wide range of lighting conditions, including both natural and artificial light sources. This is important in order to ensure that images can be captured accurately in a variety of different environments.
- Finally, the image acquisition system should be robust and reliable, with a high level of uptime and minimal maintenance requirements. This is particularly important in industrial or manufacturing environments, where downtime can be costly.

In summary, an effective image acquisition system for defect detection should be able to capture high-quality, high-resolution images at a high frame rate, be easy to use and operate, be able to handle a wide range of lighting conditions, and be robust and reliable.

Magnetic particle Inspection requirements and conditions: In addition to general requirements, image acquisition system must also be tailored to satisfy the magnetic particle inspection requirements. Real-time surface defect inspection of finished fasteners faces a number of challenges. Most of them are imposed by the characteristics of MPI. Some of the key factors that can impact MPI results include:

- Part geometry and surface finish: The shape and size of the part being inspected, as well as its surface finish, can affect the distribution and intensity of the magnetic field. Parts with complex geometry or rough surface finish may have uneven magnetic field distribution, which can make it difficult to detect small or shallow defects.

- Magnetization method and intensity: The method and intensity of magnetization can also impact the accuracy of MPI results. Different magnetization methods, such as direct current (DC), alternating current (AC), and pulsed magnetization, can be used to create the magnetic field. The intensity of the magnetization determines the sensitivity of the inspection process. Higher intensity can improve sensitivity, but it may also increase the likelihood of false indications.
- Particle size and type: The size and type of magnetic particles used in the inspection process can affect the sensitivity and contrast of the indications. Larger particles are more easily attracted to defects, but they may also be more difficult to remove from the surface of the part. Different types of particles, such as iron oxide or cobalt, may have different magnetic properties and may be more or less suitable for certain applications.
- Inspection environment: The inspection environment can also affect MPI results. Factors such as temperature, humidity, and the presence of ferromagnetic contaminants can impact the accuracy of the inspection. For example, high humidity can cause the particles to clump together and reduce the sensitivity of the inspection.

Image Acquisition System layout:

We designed an image acquisition system that is tailored to capture images of fasteners during MPI. It consists of several different components, including: Cameras, Lens, Image sensor, Triggering system, Lighting, stepper motor, magnetic suspension spray, magnetization system, ultra violet lambs and computer (Figure 4.3 and Figure 4.4).

The first step in the image acquistion based on MPI process is to prepare the part for inspection. This includes cleaning the surface of the part to remove any contaminants or debris that may interfere with the inspection. The part should also be magnetized to create a magnetic field. This is done using a magnetization system with DC magnetization method. After the part is magnetized, a suspension of magnetic particles is applied to the surface of the part by magnetic suspension spray. Using the suspension spray, the particles can be evenly distributed across the surface of the part.

For fasteners with complex shapes, it is difficult to capture all the key information based on a single camera. In order to enable inspection of the whole surface of the head of the fasteners, the stepper motor is used to perform one complete 360

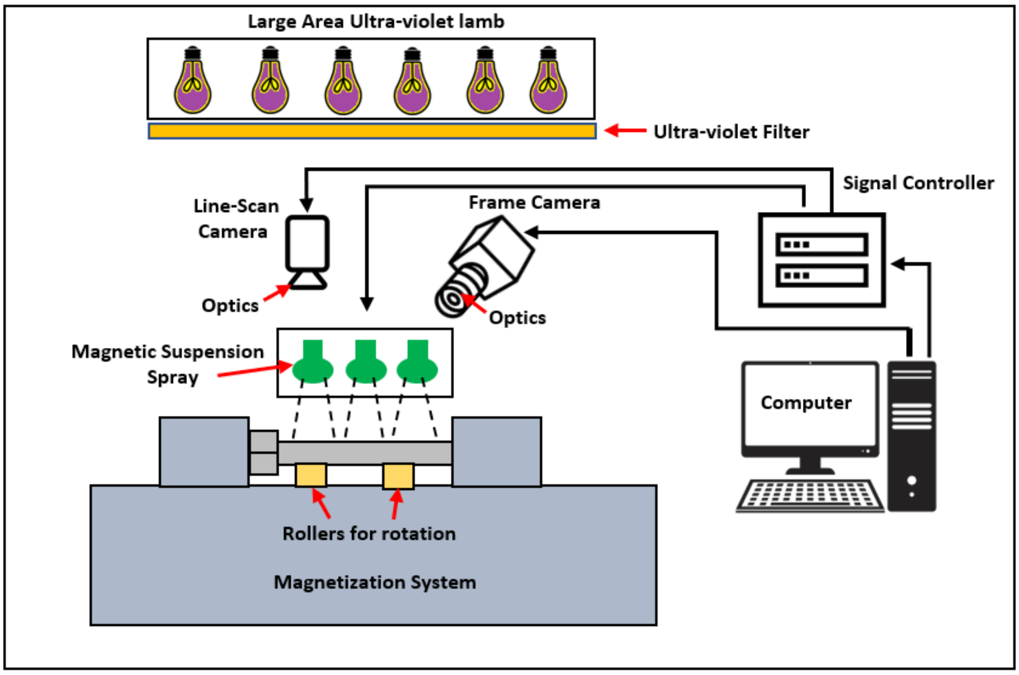


Fig. 4.3: The design of our image acquisition system.

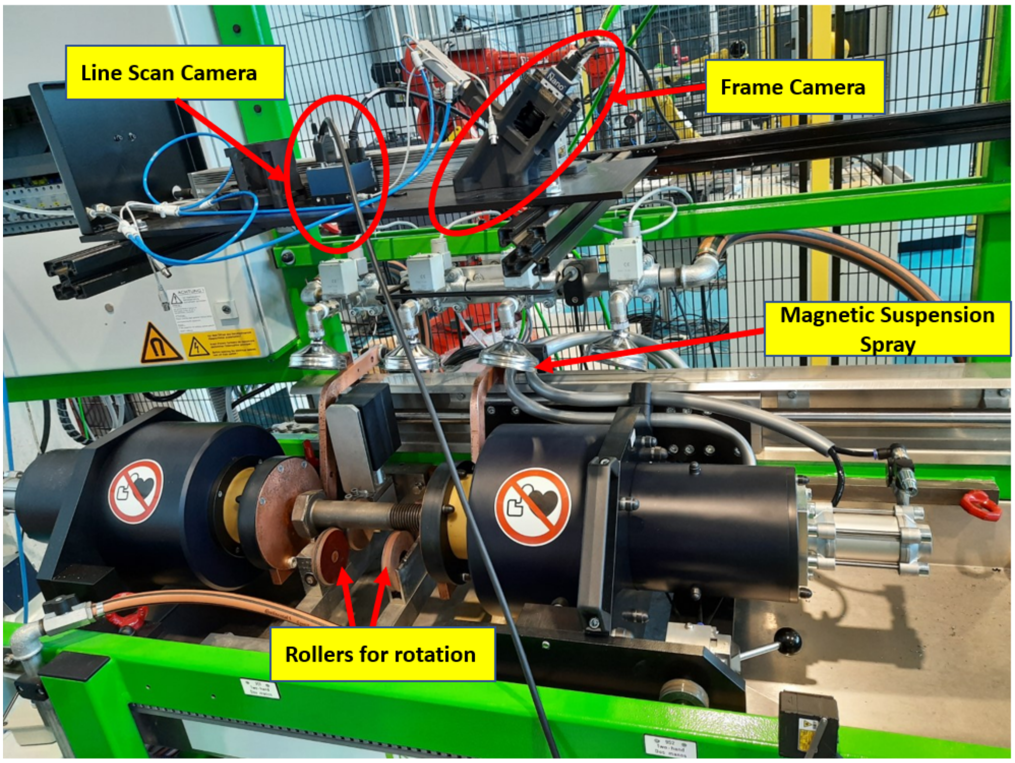


Fig. 4.4: A Picture of our image acquisition system that utilizes the wet fluorescent magnetic particle inspection method.

degree rotation around the fastener axis. During the rotation of fastener, a signal from the controller is triggered to acquire images at different rotational positions. It was found that a total of 4 rotational positions were sufficient to inspect the whole surface of the fasteners head. We used industrial Frame camera to capture all 4 rotational positions of the fasteners head. In addition to Frame camera, we employed line scan camera to capture images of shank and threaded region of the fasteners. Images captured by the cameras can be finally stored in a hard drive and computer.

Cameras:

In this work we developed a novel methodology that makes use of both frame and line scan cameras in an image acquisition system aims to improve the accuracy and efficiency of capturing images of rotating fasteners. The frame camera is used to capture the head portion of the fastener, while the line scan camera is used to capture the shank portion. The main difference between the Line scan cameras and frame cameras is the way that they acquire and process the image data as shown in Figure 4.5. Line scan cameras are specialized cameras that are designed to capture images of moving objects by scanning a single line of pixels across the object. They are typically used in industrial and scientific applications where high-speed image capture is required. Line scan cameras are capable of capturing images at very high frame rates, typically in the range of thousands to tens of thousands of frames per second. Frame cameras, on the other hand, capture images of an entire scene or object at once by sampling a two-dimensional array of pixels. They are more common than line scan cameras and are used in a wide range of applications, including photography, and machine vision. Frame cameras are typically slower than line scan cameras, with frame rates in the range of a few dozen to a few hundred frames per second.

One potential advantage of this approach is that it allows for the capture of high-resolution images of both the head and shank portions of the fastener. This is especially useful for applications where detailed analysis of the fastener's features is required, such as in defect detection. We addressed several challenges in order to effectively combine both cameras in an image acquisition system. One challenge is ensuring that the the lighting conditions are consistent for both cameras. This can be achieved through the use of specialized lighting equipment and careful control of the lighting environment. The use of a frame camera to capture a wide field of view of the fasteners head allows for efficient image acquisition, while the use of a line scan camera allows for high resolution images of the fasteners shank to be captured even at high speeds.

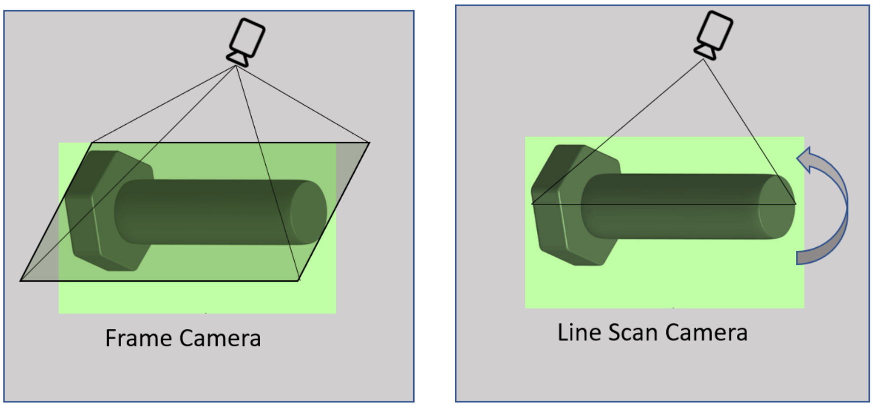


Fig. 4.5: The differences between (a) Frame Camera and (b) Line Scan Camera in terms of how they capture images.

The proposed methodology of combining both frame and line scan cameras in an image acquisition system has the potential to significantly improve the accuracy and efficiency of capturing images of rotating fasteners.

Line Scan Cameras: We selected the Dalsa Linea Color 4K GigE vision camera as our line scan camera. This camera is affordable and offers high speed, responsiveness, and color capabilities at a competitive price. It is suitable for various applications such as materials grading and inspection, transportation safety, automated optical inspection, and general purpose machine vision. It combines standard gigabit Ethernet technology (supporting GigE Vision 1.2) with Teledyne DALSA Trigger-to-Image-Reliability and has a reliable system for capturing and transferring color images from the camera to the host computer. The specifications and dimensions of Dalsa Linea Color 4K GigE are shown in a Table 4.1 and Figure 4.6.

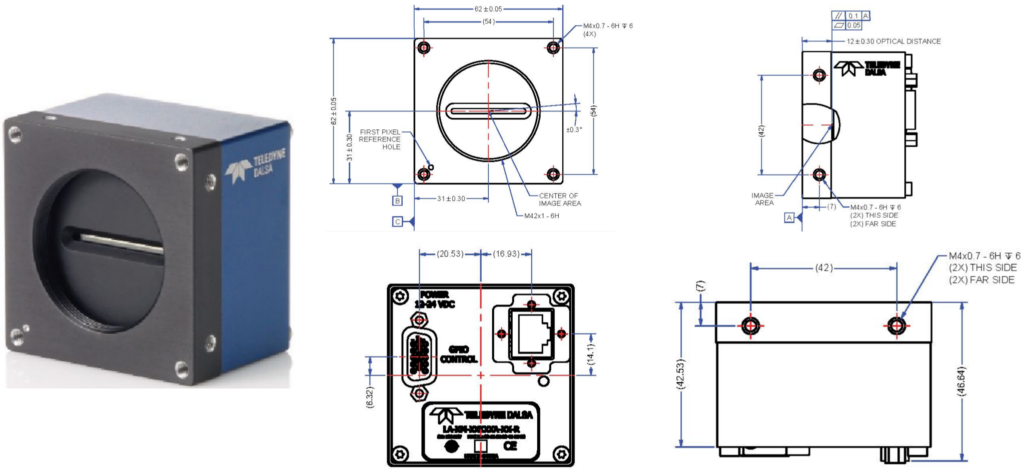


Fig. 4.6: Dalsa Linea Color 4K GigE vision camera

Specification	
Resolution	2048 x 2
Line Rate	45 KHz
Pixel Size	7.04 μm x 7.04 μm
Data Format	8 bit
Output	Gigabit Ethernet
Lens Mount	M42 x 1, C and F-mount adapters
Dynamic Range	>60 dB
Nominal Gain Range	10x
Size	62 mm x 62 mm x 46.64 mm
Mass	<280 g
Operating Temperature	0° C to 65° C
Power	+12 V to +24 V DC, HD15 connector
Power Dissipation	<8 W
I/O	HD15 connector
Software Platform	GigE Vision v1.2 compliant Teledyne DALSA Sepera LT

Tab. 4.1: Dalsa Linea Color 4K GigE Specifications

The line scan camera needs to capture images at exactly the same speed as the fastener is rotating. If the scan rate is too fast, the image becomes distorted. If it's too slow, some of the slices will not be captured, as demonstrated in the Figure 4.7. To prevent any stretching or shrinking of the final image, an encoder was used to synchronize both the rotation of the fastener and the activation of the line scan camera.

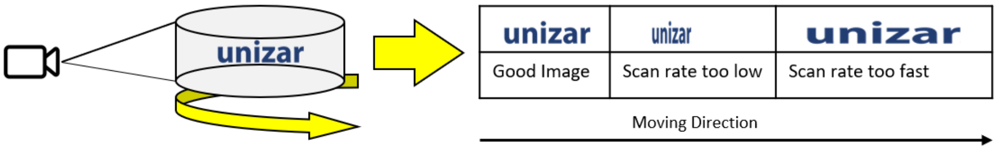


Fig. 4.7: An illustration of the correlation between the rate of scanning and the spinning speed of a product.

Frame Camera: We selected the Genie Nano C2420 color Ethernet camera with a 2464x 2056 pixel sensor as our frame camera. This camera is designed to use the GigE Vision v1.2 Standard, which is a standard for using the Gigabyte Ethernet communication protocol to transfer images quickly and efficiently using low cost cables. It is widely used in the vision industry and many companies make cameras that use this standard. It also features a user-friendly design, with a compact form factor that makes it easy to integrate into a variety of systems. It is built to be rugged and durable, with a rugged aluminum housing that protects the internal components from harsh industrial environments. The frame camera is particularly suited to capturing the head portion of fasteners, which typically have a large surface area and require a wide view to capture all details. The frame camera allows us to take images with a high level of detail, capturing the precise shape and structure of the head of the fastener. This type of camera is also very flexible and versatile, offering the ability to adjust the focus, exposure time, and other parameters to optimize the image capture process. Additionally, the images captured by a frame camera can be easily processed and analyzed using computer software, making it a valuable tool in a variety of industrial and scientific applications. The use of a frame camera in conjunction with a line camera provides a comprehensive and accurate representation of the fastener, capturing both its head and shank portions in detail. Overall, the frame camera's ability to capture high-resolution images of the head of the fastener allows for the accurate detection and analysis of any defects, helping to ensure the quality and reliability of fasteners used in industrial processes. The specifications and dimensions of Genie Nano C2420 are shown in a Table 4.2 and Figure 4.8.

Specification	
Resolution	2464 x 2056
Sensor	Sony IMX264 (5.1M)
Pixel Size	3.45 μm x 3.45 μm
Data Format	RGB.24- bit
Output	Gigabit Ethernet
Lens Mount	C-mount adapters, Optional Tripod Mount
Dynamic Range	76.8 dB
Nominal Gain Range	16x
Size	21.2 mm x 29 mm x 44 mm (without lens mount or Ethernet connector) 38.9 mm x 29 mm x 44 mm (with C-mount and Ethernet connector)
Mass	~46g
Operating Temperature	-20° C to +65° C
Power	+12 V to +36 V DC, HD15 connector
Power Dissipation	From 3.8W to 4.9W
I/O	HD15 connector
Software Platform	DALSA Software

Tab. 4.2: Genie Nano C2420 Specifications

Optics: Optics lenses are essential components in both frame and line cameras. They are used to control the way light enters the camera and to focus the image onto the image sensor. The quality of the lens greatly affects the clarity and sharpness of the image captured, making it an essential part of the camera system.

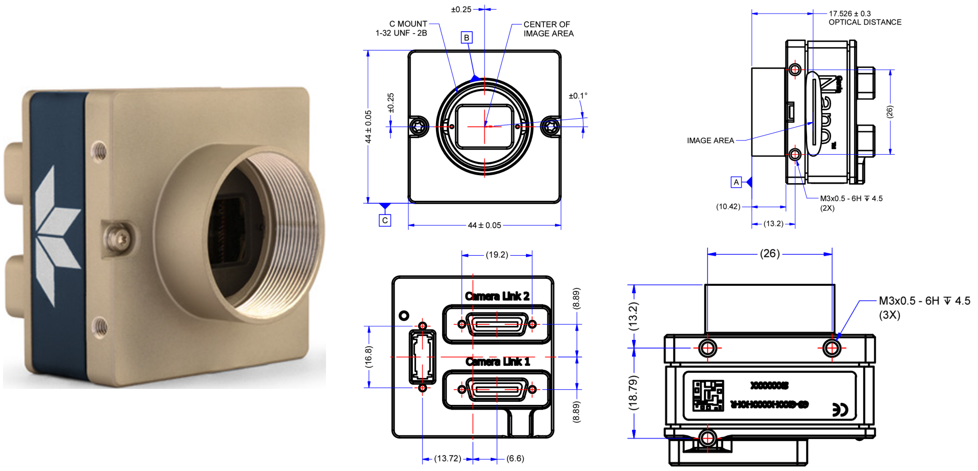


Fig. 4.8: Genie Nano C2420

In a frame camera, the optics lens is used to capture a wide and clear image of the object being photographed. It typically has a wider field of view than a line camera, and the lens must be carefully designed to allow for a high level of detail and clarity in the image. The lens must also be able to control the amount of light entering the camera, as well as adjust the focus to capture objects at different distances from the camera.

In a line camera, the optics lens is used to focus light onto the image sensor in a linear fashion. This is essential in capturing the shank portion of the rotating fasteners, as the fastener is moving in a linear direction and the lens must follow the movement to capture a clear image. The optics lens in a line camera must be designed to be very fast, allowing for rapid image capture as the fastener rotates. Additionally, it must be able to control the amount of light entering the camera and to focus the image onto the image sensor, even as the fastener rotates at high speeds.

There are several factors to be considered when selecting an optic lens for both linear and Frame cameras:

- Field of view: The field of view (FOV) determines the area that the camera can capture. A wider FOV is useful for capturing a larger area, while a narrower FOV is better for focusing on specific details.
- Resolution: The resolution of the lens determines how much detail it can capture. A lens with a higher resolution will be able to capture more detail, but it may also require a more powerful processor to process the data.

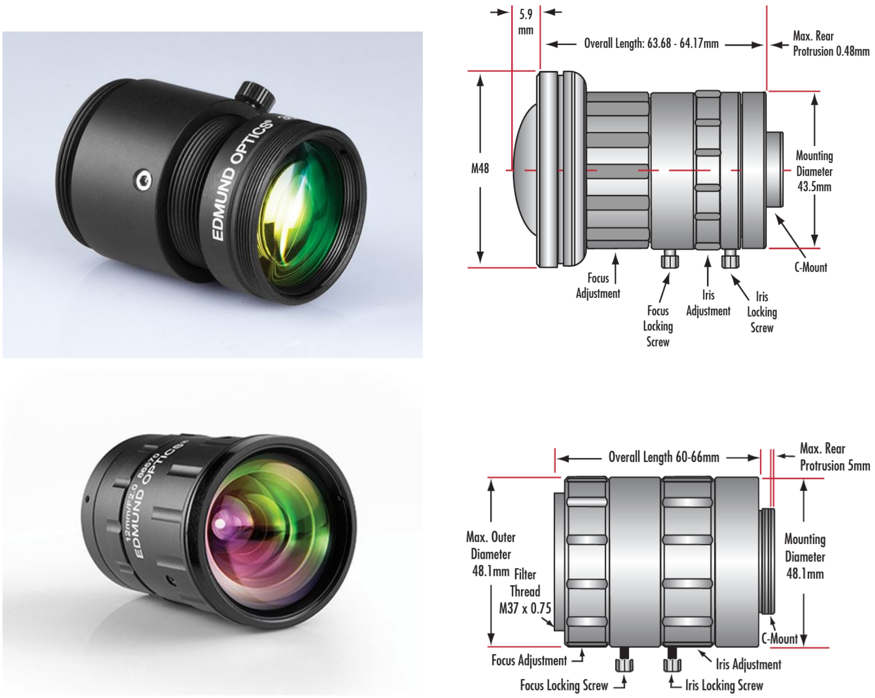


Fig. 4.9: Edmund Optics of Line scan and Frame cameras

- Focal length: The focal length of the lens determines how much the lens will magnify the image. A shorter focal length will produce a wider angle of view, while a longer focal length will produce a narrower angle of view.
- Aperture: The aperture of the lens determines how much light is allowed to pass through the lens. A larger aperture allows more light to pass through, which is useful in low light conditions.
- Distortion: Distortion refers to the degree to which the lens distorts the image. A lens with low distortion will produce a more accurate representation of the scene, while a lens with high distortion may produce an image that is stretched or distorted.
- Depth of field: The depth of field refers to the range of distance in which objects appear in focus. A lens with a shallow depth of field will only focus on objects at a specific distance, while a lens with a deep depth of field will be able to focus on objects at a range of distances.

We have chosen two different optical lenses for our cameras, one for a linear camera and one for a frame camera. The lens for the linear camera has a focal length of 12 mm and a horizontal field of view of 129.6 mm, while the lens for the frame camera has a focal length of 35 mm and a horizontal field of view of 101 mm. The specifications and dimensions of Optics lenses used for line scan and frame cameras are shown in a Table 4.3 and Figure 4.9.

Specification	**Line-scan Camera**	**Frame Camera**
Focal Length	12mm	35mm
Max. Camera Sensor Format	1”	1”
Horizontal FOV	129.6mm - 52.4°	101mm - 26°
Working Distance	100mm - inf	200mm - inf
Aperture	f/1.8 - f/16	f/1.8 - f/16
Filter Thread	M62 x 0.75	M37 x 0.75
Distortion	<-4.2%	<-1.5%
Weight (g)	260	252
Outer Diameter (mm)	48.0	48.1
Max. Length (mm)	63.6	66
Mount	C-Mount	C-Mount

Tab. 4.3: Optics Specifications of Line scan and Frame cameras

Band pass filters: A band pass filter is a type of optical filter that allows light within a specific wavelength range to pass through while blocking light outside of that range. We used band pass filters-BP 525 Figure 4.10 in both frame and line scan cameras to filter out unwanted light and improve the visibility of magnetic particle indications. By filtering out unwanted light with a band pass filter, it is possible to reduce the influence of sunlight, ambient light, and fluorescent light. This results in images with improved contrast and clarity, making it easier to identify and analyze the magnetic particle indications. It helps to improve the accuracy of the inspection and reduce the risk of false readings. We placed the band pass filter in front of the camera lens and it acts as a selective barrier to light. By selecting the correct wavelength range, the filter can effectively block light outside of that range and allow only the desired light to pass through to the camera sensor. This results in images with improved contrast and clarity, making it easier to identify and analyze the magnetic particle indications.

Illuminations: To ensure accurate and reliable results, it is important to use an appropriate illumination system during MPI.One of the main benefits of an illumination system is that it helps to enhance the visibility of magnetic particle indications. Magnetic particle indications are small areas of accumulated magnetic particles that can be difficult to capture with the cameras, especially in low-light conditions. By providing adequate illumination, it is possible to highlight these indications and make them easier to capture, which helps to improve the accuracy and reliability of the inspection. Another important benefit of an illumination system is that it helps to reduce the influence of ambient light on the inspection results.

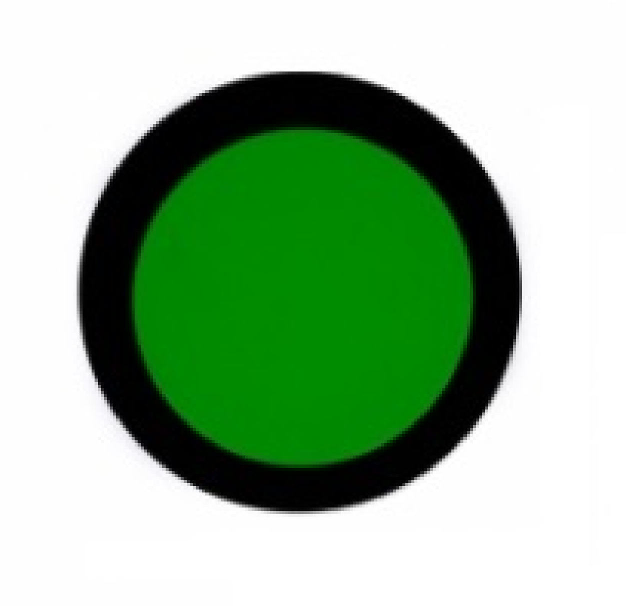


Fig. 4.10: Band pass filters-BP 525

Ambient light can interfere with the visibility of magnetic particle indications, making it difficult to see them accurately. By using a specialized illumination system that is optimized for MPI, it is possible to reduce the impact of ambient light and improve the quality of the inspection results. A third benefit of an illumination system is that it helps to increase the contrast of the magnetic particle indications. The more contrast that is present in the images, the easier it is to see and analyze the magnetic particle indications. An illumination system that provides high-contrast images can help to improve the accuracy and reliability of the inspection, reducing the risk of false readings.

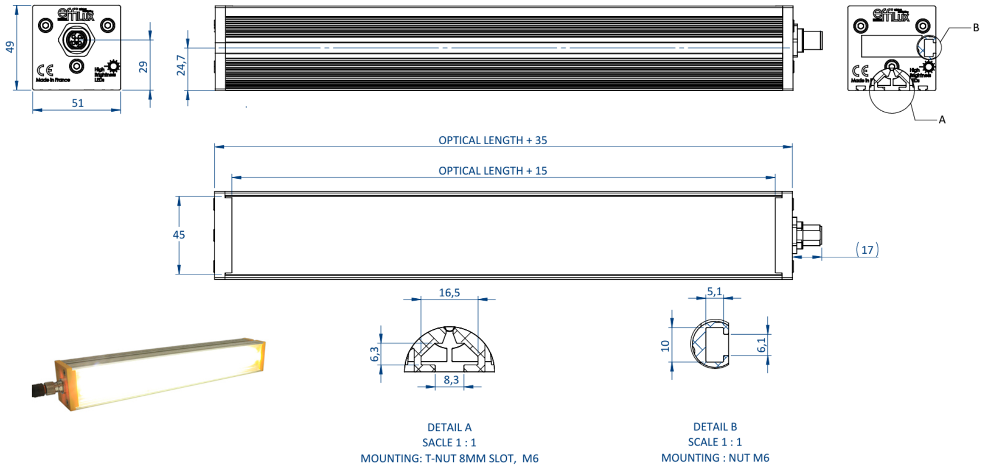


Fig. 4.11: EFFI-Flex-25-365-TR-P0 illumination

We chose to use the EFFI-Flex-25-365-TR-P0 LED bar for our MPI system. This EFFI-Flex LED bar is equipped with 25 high-power LEDs, each with a wavelength of 365 nm. This wavelength is specifically chosen to provide maximum visibility of magnetic particle indications, as it is known to be highly effective in illuminating these particles. Furthermore, the LED bar is equipped with transparent windows and a 90 degree lens position, which provides a clear and uniform illumination of the metal component being inspected. One of the key advantages of the EFFI-Flex LED bar is its flexibility in terms of diffuser and emission angle. The bar is designed to offer multiple solutions in a single light, allowing users to choose the right diffuser and emission angle to find the right compromise between power, illuminated area, and uniformity. This means that the bar can be easily customized to meet the specific requirements of each individual inspection, providing the most accurate and reliable results possible. In addition to its flexibility, the EFFI-Flex LED bar is also equipped with an integrated controller that includes an Auto-Strobe feature. This feature allows the bar to increase its intensity by 300% when strobed compared to continuous mode, providing even greater visibility of magnetic particle indications. Furthermore, the bar is designed to be robust and durable, making it ideal for use in harsh industrial environments where it may be subjected to high levels of vibration and shock. The specifications and dimensions of the EFFI-Flex-25-365-TR-P0 are shown in a Table 4.4 and Figure 4.11.

Specification		
Mechanics	*Number of LED*	25
	Optical Length	500mm
	Product Length	535mm
	Weight (KG)	0.6
	Width x Height	51mm x 49mm
	Fastener	One T-slot on the back for M6 T-nut and, one slot on the side for M6 hex nut
	Material	Device body: Aluminum alloy & ABS
Electronics	*Connectors*	M12 – 4 Pins
	Power supply	24V DC
	Illumination mode	Continuous or strobe mode
	Electronic mode	Auto-Strobe
Environment	*Working Temperature*	0°C to 50°C

Tab. 4.4: EFFI-Flex-25-365-TR-P0 illumination Specifications.

Acquired Images: Our image acquisition system is designed to capture images of fasteners at a high speed, with precision and accuracy, making it an ideal solution for a wide range of applications, particularly in the field of fastener defect detection. The frame camera in the system is used to capture the head portion of the fasteners and has an image size of 2464x2056 pixels, while the line camera is used to capture the shank portion and has an image size of 2048x2048 pixels. Both cameras are specifically designed to capture images of fasteners as they rotate, allowing for a complete and comprehensive analysis of the fastener's structure and surface. By using both cameras, we are able to obtain a more detailed and accurate image of the fastener, which can be used to identify and classify various types of defects that may be present.

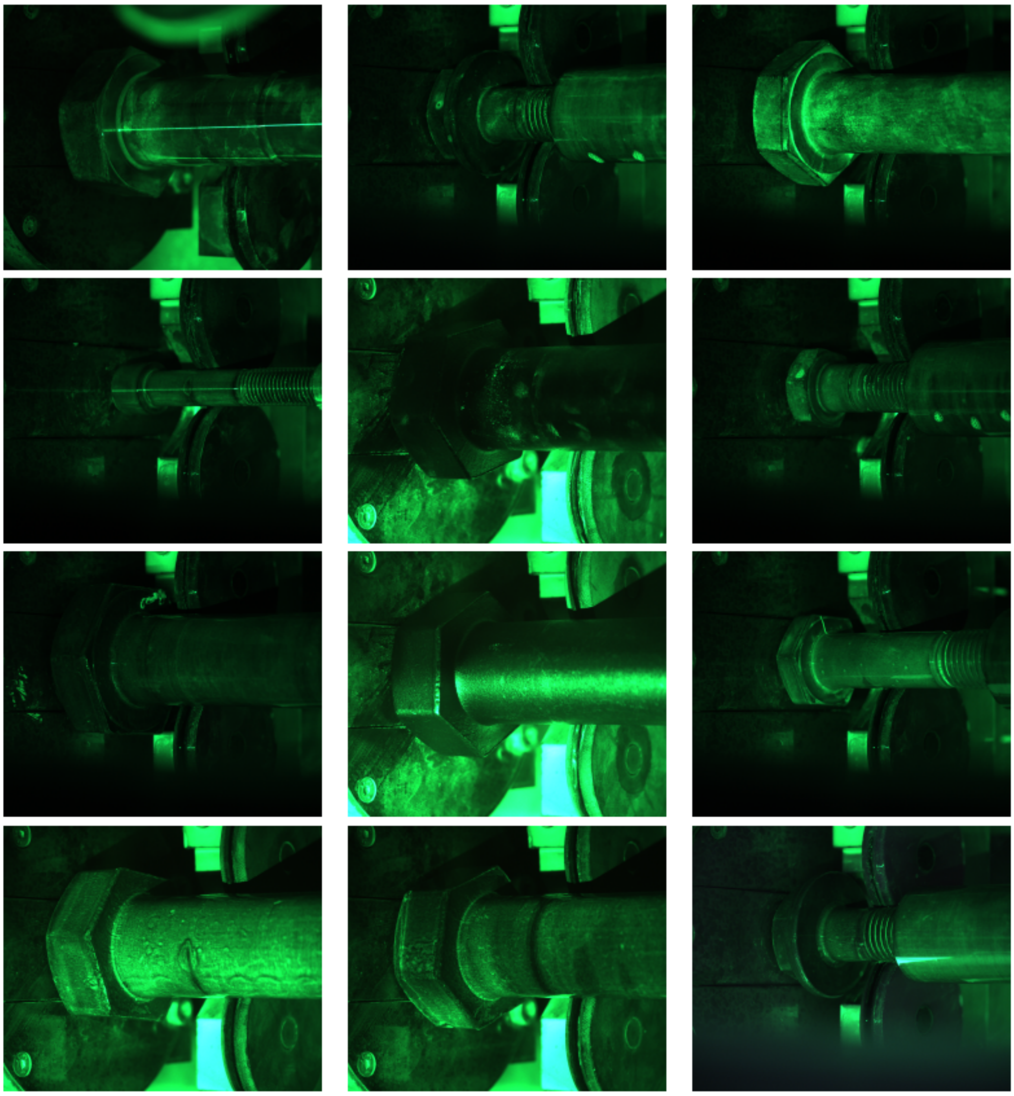


Fig. 4.12: Sample images acquired from a Frame camera.

The images captured by the cameras are stored in a database, where they can be easily retrieved, analyzed, and compared to other images. This allows us to build up a comprehensive picture of the fastener's characteristics, including its size, shape, surface conditions, and any defects that may be present. This information is then used to make informed decisions regarding the quality and safety of the fastener, helping to ensure that it is fit for purpose and suitable for use in the intended application.

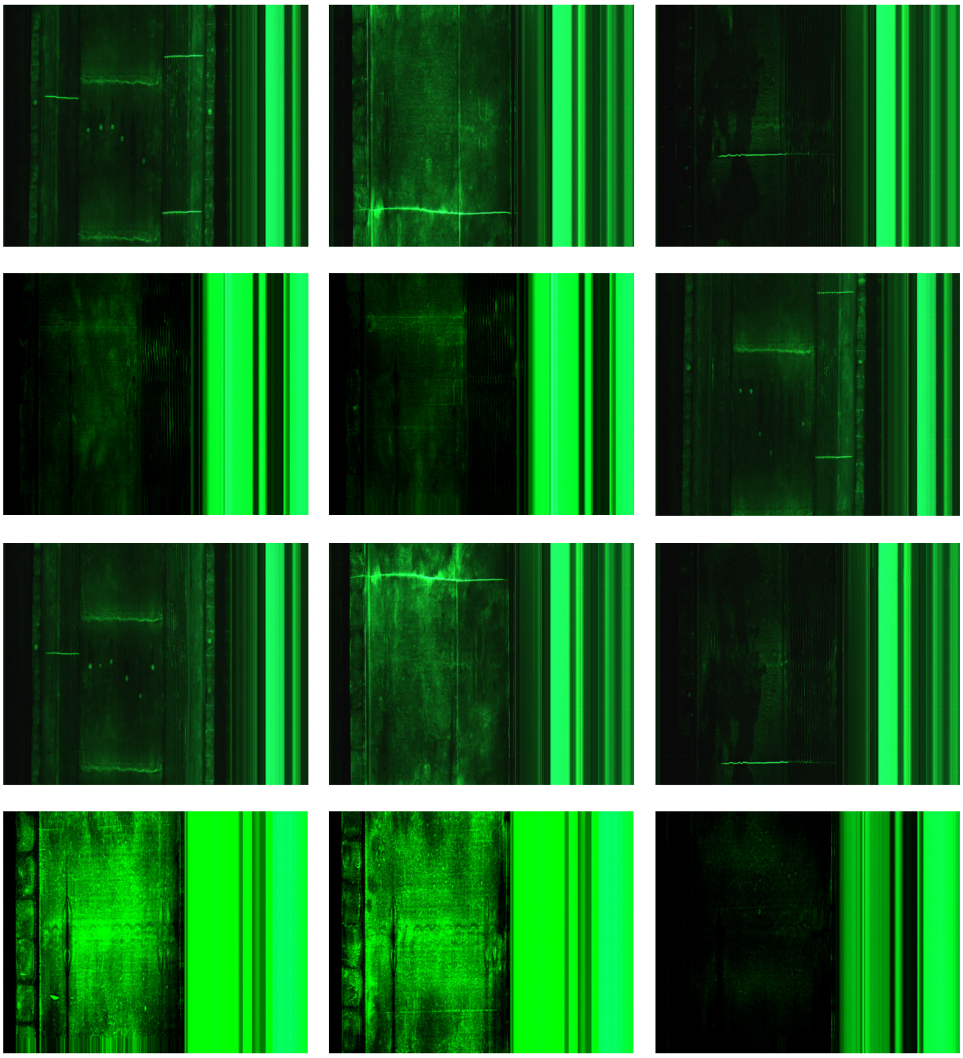


Fig. 4.13: Sample images acquired from a line scan camera.

One of the key benefits of our image acquisition system is its ability to capture as many images as possible in a short period of time. This allows us to quickly and easily inspect large numbers of fasteners, improving the efficiency and speed of the inspection process. Furthermore, by storing the images in a database, we can easily compare and analyze the images over time, allowing us to track any changes or developments in the fastener's structure and surface. Another important aspect of our image acquisition system is the use of advanced optical lens technology, which provides high-resolution images and improved image quality. The use of high-quality optics ensures that the images captured are sharp and clear, allowing for precise and accurate analysis. Furthermore, the use of specialized filters, such as band pass filters, can be used to reduce the influence of ambient light, improve image contrast, and increase the accuracy of the inspection process. Sample images acquired from a Frame and line scan cameras are shown in Figure 4.12 and Figure 4.13.

We took into consideration the varying sizes and shapes of fasteners, surface finishes, heat treatments, and materials that can affect the texture of the images. Additionally, factors such as magnetizing force, suspension particle size and type, and light intensity can also play a role in the acquired images. Given the low probability of defect occurrence and the practical challenges in collecting a large scale of defective images with different combinations of these factors, we sought to utilize data centric approaches to solve this problem.

4.3.2 Data centric deep learning approach

Data-centric deep learning approach is a rapidly growing field that is revolutionizing the way AI systems are designed, developed and deployed. The discipline of data-centric deep learning focuses on the systematic engineering of the data needed to build a successful AI system. This is a critical aspect of AI development as the quality and quantity of data has a significant impact on the performance of AI systems. This is because our deep learning algorithms rely heavily on the quality of the data that is fed into the model. Any errors or inconsistencies in the data can significantly impact the accuracy of the model and its ability to generalize to new data. The idea of data-centric deep learning approach is that by fixing the model and improving the data, the model will be able to perform better and generalize to new situations. We encountered three significant difficulties that are specifically linked to data-centric deep learning.

1. Human bias in labelling: First, human bias in labelling posed a significant challenge in data collection. Human bias can occur when quality inspectors make subjective interpretations of the data they are labelling, such as the size, shape, and location of defects (Figure 4.14). This can result in inconsistencies in the labelled data and compromise the accuracy of the model. To mitigate the impact of human bias, it is essential to have a clear and consistent protocol for labelling the data and to ensure that multiple quality inspectors are involved in the labelling process.
2. Dynamic manufacturing environments: Second, the dynamic nature of the environment in which magnetic particle inspection is performed also created

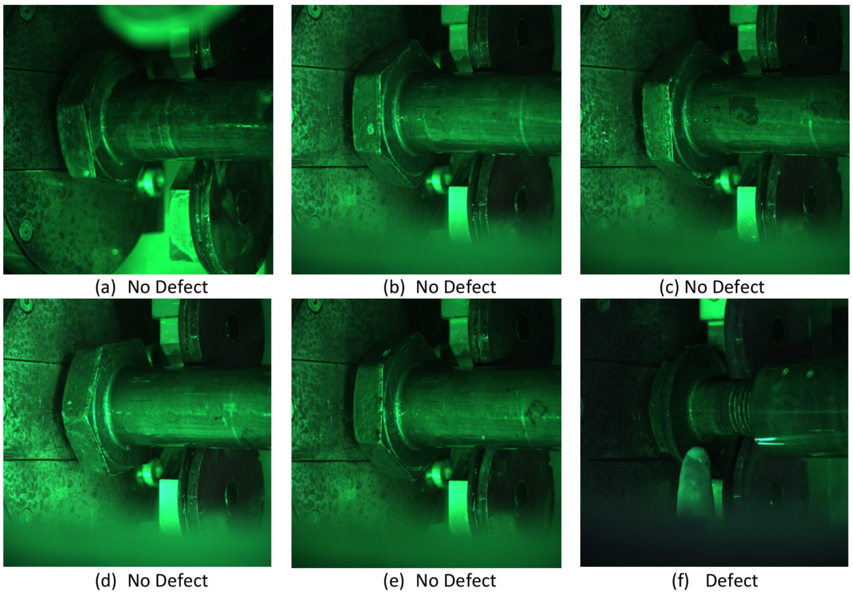


Fig. 4.14: Examples of label inconsistencies

challenges for data collection. For instance, introducing new or custom parts or working in an environment that constantly changes, such as lighting conditions in a plant with large windows, lead to variation in the images of the fasteners. This variability affected the accuracy of our deep learning model and make it difficult to generalize to new data. To overcome this challenge, it is essential to have control over the environment in which the data is collected and to ensure that all relevant parameters, such as lighting conditions, are recorded and consistent across all images.

3. Complex data variations: Finally, fasteners come in a wide range of sizes and shapes, and their surface finishes, heat treatments, and materials can create different textures in the images of the fasteners. Additionally, several other magnetic particle inspection factors, such as magnetizing force, suspension particle size and type, and light intensity, can also create different variations in the acquired images. To ensure that the data collected for magnetic particle inspection is representative and diverse, it is critical to collect images of fasteners with different sizes, shapes, textures, and surface finishes. Moreover, it is essential to control and standardize the parameters of magnetic particle inspection, such as magnetizing force and light intensity, to ensure that all images are consistent.

To address the challenges, we employed a systematic approach to design and organize the data in order to fully tap into the potential of data-centric deep learning.

Human bias in labelling: In order to overcome this challenge, we employed a data centric approach to ensure the quality and consistency of the labeled data.

- Multiple labelers to spot inconsistencies: One of the key steps in this approach was the use of multiple quality inspectors to identify inconsistencies in the annotations. This was done by having more than two quality inspectors label the same example, and then taking the maximum vote to determine the correct label. This approach allowed us to identify discrepancies in the annotations, such as a defect being annotated as a defect by one labeler and as a non-defect by another. In our application, inconsistencies can occur in various forms, such as bounding box size, number of bounding boxes, label names, and so on. By using multiple quality inspectors to identify these inconsistencies, we were able to clean up the dataset and significantly improve the performance of the defect detection model.
- Making sure the instructions for labeling are clear by finding and fixing any unclear labels: The first step in solving the label inconsistency problem using this method was to identify examples that were ambiguous or unclear. These examples were then used to clarify the instructions for how to label edge cases. The labeling instructions, along with examples of the concepts, were created as a single source of truth to ensure consistency in the labeling process. In the instructions, obvious examples of defects were shown along with borderline cases, near-misses and any other confusing examples. This was done to provide a clear understanding of the concepts to the labelers and avoid any confusion or misunderstandings. The instructions were also actively reviewed for any labels that were ambiguous or inconsistent and a definitive labeling decision was documented to provide a clear reference for future labelers. By creating a well-structured labeling instructions document, the quality of the labels was significantly improved. This ensured that the deep learning models trained on these labels would perform optimally and produce accurate results.
- Tossing out bad examples: This approach is based on the assumption that removing examples with incorrect labels will improve the overall quality of the data set and result in a better model. This approach is simple and straightforward, and we used this approach when the number of examples with incorrect labels is relatively small.
- Error Analysis: We randomly selected a certain percentage of the labeled data for spot-checking. This was done to ensure that the data quality was consistently high, and that any errors or inconsistencies were caught before they can cause problems for the deep learning model. The data was then thoroughly reviewed and any final changes were made to the selected data.

This approach was iteratively improved, with the goal of ensuring that the data is of the highest quality and free from inconsistencies.

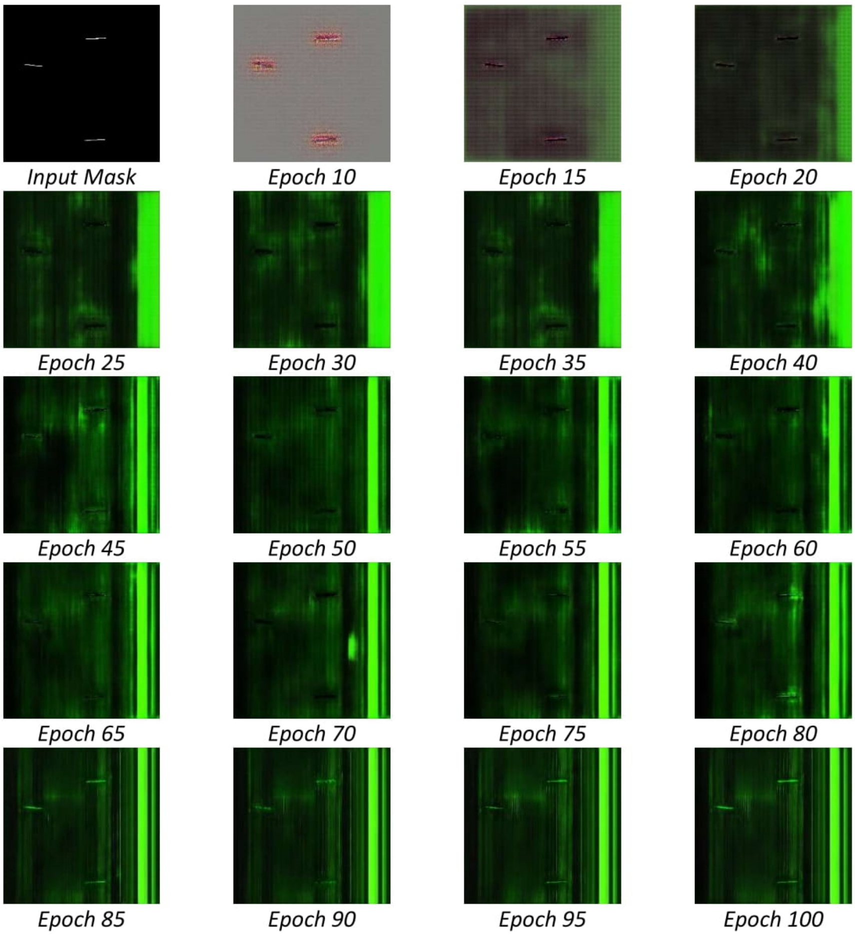
Dynamic manufacturing environment and complex data variations:

The training data plays a critical role in building robust defect detection models for real-world applications. Ideally, the training data should be representative of the data that we expect to see in deployment, covering all variations that deployment data will present. This is important because if the training data does not include all the variations that we expect to see in the deployment data, the model will likely not be able to generalize well and accurately detect defects in real-world scenarios. For example, in the context of manufacturing, the data collected during training should represent different camera positions, lighting conditions, and the position, shape, and size of the defects. This is crucial because the model must be invariant to these factors, meaning it should be able to detect defects regardless of the variations in these factors.

One solution to ensuring that the training data is representative of the deployment data is to collect new data with more variation. However, collecting new data with all possible variations in real-time manufacturing environments is infeasible. This is because collecting such data requires significant effort and resources, and it may also not be possible to recreate the exact conditions under which the deployment data will be collected. Our proposed approach to overcome this challenge is to employ a combination of data augmentation and synthetic image generation through GAN methods.

Data augmentation: One of the most important invariances in defect detection models is camera position invariance. This refers to the ability of the model to detect defects regardless of the position of the camera when the image was captured. This is important because in real-world scenarios, the camera may be positioned at different angles and distances from the object being inspected. To address this issue, we applied augmentation techniques such as center crop, horizontal flip, rotation, shear, vertical flip, and translation to the training data. These techniques artificially introduce variations in camera position to the training data, allowing the model to learn to detect defects regardless of camera position. A thorough examination of various data augmentation techniques are outlined in subsection 2.2.1.

Another important invariance in defect detection models is camera lighting invariance. This refers to the ability of the model to detect defects regardless of the lighting conditions when the image was captured. In real-world scenarios, lighting conditions can vary greatly, making it difficult for the model to accurately detect defects. To address this issue, we applied augmentation techniques such as noise injection, color space transformations, mixing images, random erasing, sharpness, brightness, contrast, and Gaussian blur to the training data. These techniques artificially introduce variations in lighting to the training data, allowing the model to learn to detect defects regardless of lighting conditions.



Input Mask, Epoch 10, Epoch 15, Epoch 20

Epoch 25, Epoch 30, Epoch 35, Epoch 40

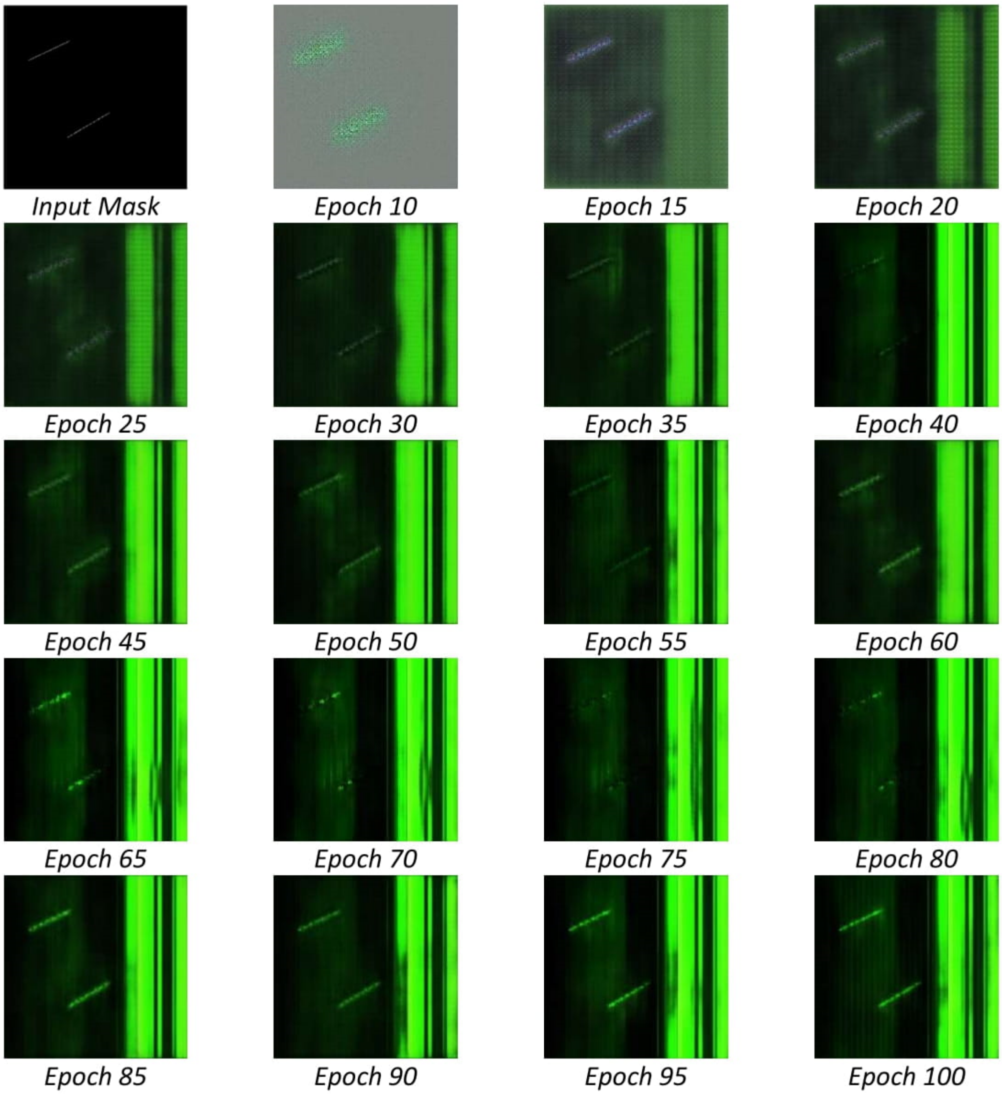
Epoch 45, Epoch 50, Epoch 55, Epoch 60

Epoch 65, Epoch 70, Epoch 75, Epoch 80

Epoch 85, Epoch 90, Epoch 95, Epoch 100

Fig. 4.15: Training of Magna-Defect-GAN over a span of epochs for generating an image with multiple defects

GAN-Based augmentation: Finally, it is crucial for the defect detection models to have invariance in defect position, shape, orientation and size. This means that the model should be able to detect defects regardless of the position, shape, orientation and size of the defects in the image. To address this issue, we have employed the use of Magana-Defect-GAN. A thorough architectural details of Magana-Defect-GAN is outlined in subsection 2.2.2.This method encodes prior knowledge in the form of binary masks and guidance vectors during the generation process. The binary masks allow us to incorporate industrial knowledge into the generation process, enabling



Input Mask

Epoch 10

Epoch 15

Epoch 20

Epoch 25

Epoch 30

Epoch 35

Epoch 40

Epoch 45

Epoch 50

Epoch 55

Epoch 60

Epoch 65

Epoch 70

Epoch 75

Epoch 80

Epoch 85

Epoch 90

Epoch 95

Epoch 100

Fig. 4.16: Training of Magna-Defect-GAN over a span of epochs for generating an image with defects at different orientation.

the creation of additional defects with different shapes, severities, scales, rotation angles, spatial locations, and part numbers. The Magana-Defect-GAN model also includes strategies to enhance the quality of the generated images and stabilize the training process. The GAN architecture maps the given mask input to the sample space more efficiently by combining a mask embedding vector, conditional label vector, and latent noise vector. This results in generated samples that are more diverse compared to traditional image-to-image translation GANs. The image generation process of Magna-Defect-GAN given the input masks with different variations are shown in Figure 4.15 and Figure 4.16. By using Magana-Defect-GAN, we ensure that the generated data has a wide range of diverse features, which is critical for improving the generalization ability of the defect inspection model. In other words, by having a more diverse set of features in the training data, the model will be better equipped to detect defects in real-world scenarios, even if they differ in position, shape, and size from the training data.

4.3.3 Model centric deep learning approach

The model-centric deep learning approach is a strategy where the focus is placed on improving the Defect-aux-net model itself, rather than the quality and diversity of the training data. We used a combination of augmentation, GAN based synthetic, and original image data to train the Defect-aux-net model. The first step in this approach is splitting the data into separate train, validation, and test sets. This is done to ensure that the model is trained on a diverse set of data, evaluated on a representative set of validation data, and finally tested on a set of unseen data to measure its performance. Once the data is split, the next step is to convert the train, validation, and test data into tfrecord format. Tfrecord is a binary file format that is optimized for storing and serving large datasets. It is a popular format for training large machine learning models, as it allows for efficient data loading and parallel processing. Once the data is converted to tfrecord format, the next step is to create an efficient data pipeline using the tf.data library. This allows us to easily and efficiently load, shuffle, and batch the data, making it easier to train the model. To train the Defect-aux-net model, we used Google Cloud Platform (GCP) and its compute engine. The compute engine provides us with a powerful and scalable platform for training large machine learning models, while the Google Cloud Storage (GCS) platform allows us to store the tfrecord files in a highly scalable and secure manner. Finally, to ensure that the model is trained effectively, we performed a hyperparameter search. This involves exploring different hyperparameters, such as the learning rate, batch size, and number of epochs, to find the best combination of parameters that gives us the best performance.

Data Split: Splitting image data is an important aspect of training our Defect-aux-net model. The purpose of splitting the data is to divide the images into training, validation, and testing sets in order to evaluate the model's performance. Different methods of data split can be used, including random, stratified, and k-fold cross-validation. Stratified data split is a method of dividing the data into subsets that are representative of the distribution of the classes in the data. This is particularly important for surface defect detection, where the images may be imbalanced in terms of the presence or absence of defects. By using stratified data split, it is feasible to ensure that each subset of the data contains an equal proportion of positive and negative samples, making it possible to train the model to accurately detect the fastener surface defects. Therefore we employed Stratified data split method, in order to split the synthetic, original, and augmented images in a stratified way. We first assigned each image to a class based on the presence or absence of surface defects. Next, we used stratified sampling to create the training, validation, and testing sets, ensuring that each set contains an equal proportion of positive and negative samples. This was accomplished by using a random number generator to select a random subset of the images, while taking into account the distribution of the classes in the data.

TensorFlow records: We created TensorFlow records (TF-records) data format for our training, validation, and test datasets. The TFrecord format is a common data format used in deep learning frameworks, particularly in TensorFlow, to efficiently store and access large datasets. This format serializes the data into a binary format, which enables faster and more efficient data loading, as well as reduces the memory requirements. To convert the data into the TFrecord format, we used the TensorFlow APIs to write the data into the TFrecord format, using a specified schema that defines the data structure and the data type of each feature.

Data pipeline using tf.data: Once the data was in the TFrecord format as illustrated in Figure 4.18, we created an efficient data pipeline using the TensorFlow tf.data API, which enables parallel and efficient data processing. The learning process of Defect-Aux-Net involves inputting a large volumes of images into the model so that it can gain knowledge. The input pipeline, which is responsible for reading images, applying transformations such as data augmentations and normalizations, and transferring data to hardware accelerators, is a critical aspect of Defect-Aux-Net training. If the input pipeline is not implemented efficiently, it can significantly impact the performance and accuracy of the trained model.

Implementing an efficient input pipeline for Defect-Aux-Net is a challenging task due to several factors. Firstly, the large volume of images that need to be read and processed can cause significant overhead in terms of memory usage and computation time. Secondly, the complexity of the data augmentation and normalization transformations that need to be applied to the images can also increase the overhead of the input pipeline. Finally, transferring data to hardware accelerators, such as TPUs, requires a high-bandwidth communication channel and must be done efficiently in order to achieve optimal performance. To overcome these challenges, we used the tf.data framework for building and executing efficient input pipelines for Defect-Aux-Net. The tf.data framework offers several benefits for implementing the input pipeline for Defect-Aux-Net. Firstly, the framework provides a high-level interface for reading and processing large volumes of images, making it easy to implement the data augmentation and normalization transformations. Secondly, the framework supports parallel processing of images, allowing for faster data transfer to hardware accelerators. Finally, the tf.data runtime optimizes the input pipeline by overlapping computation and communication to achieve optimal performance.

Our data pipeline for training Defect-Aux-Net with TensorFlow's tf.data module involves several important steps to ensure efficient and effective processing of data. First, we cache the input data to improve training speed. Then, we shuffle the data to provide a random order of samples to the model, improving training stability. The data is repeated to enable multiple epochs of training. The map method is used to apply a set of data preprocessing operations to the data. The filter method is used to remove any samples that do not meet specific criteria. Finally, the data is batched and prefetched to ensure the TPU has a continual supply of data to process during training.

The following are the TensorFlow tf.data methods utilized for conducting the ETL (Extract, Transform, Load) operations within our data pipeline.

1. Cache: When training Defect-Aux-net, it's important to have fast access to the training data. The cache method in TensorFlow's tf.data API helps to achieve this by saving the processed data in binary format on disk, allowing for quick retrieval during the training process. This is useful because it reduces the amount of time spent preprocessing the data each time the model is trained.
2. Shuffle: The shuffle method shuffles the elements of the dataset randomly. This is important in learning process because it helps to reduce overfitting and increase the model's ability to generalize to unseen data.
3. Repeat: The repeat method repeats the elements of the dataset a specified number of times. This is useful in cases where the amount of data is limited

and the model needs to be trained multiple times on the same data in order to improve its accuracy.

4. Map: The map method applies a function to each element of the dataset, allowing for data preprocessing or augmentation. This is important in order to prepare the data for training and make it suitable for the model to learn from.
5. Filter: The filter method filters the elements of the dataset based on a specified condition. This can be useful for removing examples from the dataset that might negatively impact the model's performance.
6. Batch: The batch method groups the elements of the dataset into batches, allowing for more efficient processing. This is important because it reduces the amount of memory required to store the data and allows the model to make more effective use of the TPU.
7. Prefetch: The prefetch method prepares the next batch of data while the TPU is working on the current batch, improving the overall efficiency of the training process. This helps to reduce the amount of time spent waiting for the next batch of data and enables the model to make more effective use of the TPU.

The python pseudocode for our input pipeline is presented below:

```python
# Load the dataset
dataset = tf.data.Dataset.from_tensor_slices((data, labels))
# Cache the dataset to memory for faster retrieval during training
dataset = dataset.cache()
# Shuffle the data to ensure randomness in training
dataset = dataset.shuffle(buffer_size=10000)
# Repeat the dataset to enable multiple epochs in training
dataset = dataset.repeat()
# Map the data to perform pre-processing operations (if any)
def preprocess_data(data, labels):
    # Perform data pre-processing operations here
    return data, labels
dataset = dataset.map(preprocess_data)
# Filter the data to exclude outlier instances
def filter_data(data, labels):
    # Filter data based on certain conditions here
    return data, labels
dataset = dataset.filter(filter_data)
# Batch the data to create mini-batches for training
dataset = dataset.batch(batch_size=32)
# Prefetch the data to improve training performance
dataset = dataset.prefetch(buffer_size=tf.data.AUTOTUNE)
```

GCP cloud training environment:

To train the Defect-aux-net model, we used the GCP cloud platform, which provides a range of services for machine learning, including data storage, computing power, and high-performance hardware accelerators.

GCS platform: The GCS platform was used to store the data for the model in the form of TF-records. This is a format for storing large amounts of data in a efficient and scalable manner, making it ideal for deep learning models. GCS is the primary object storage service in GCP and provides a highly scalable and durable storage solution for unstructured data. One of the key benefits of the GCS platform is its scalability. The platform uses a pay-as-you-go pricing model, allowing us to only pay for the storage they actually use, rather than having to invest in expensive hardware and infrastructure upfront. Another advantage of the GCS platform is its reliability. The platform uses state-of-the-art data centers and advanced technologies, such as multiple levels of redundancy and automatic replication, to ensure that data is always available and protected against data loss. Additionally, GCS provides various backup and disaster recovery options to ensure that data is always recoverable in the event of an outage or other issue. GCS platform also provides strong security features to protect customer data. Data is encrypted both in transit and at rest, and the platform provides various access controls and authentication mechanisms to ensure that only authorized users have access to data. The platform also complies with various industry standards and regulations, such as the General Data Protection Regulation (GDPR) and the Payment Card Industry Data Security Standard (PCI DSS).

GCP Compute Engine: The GCP Compute Engine-Tensorflow TPU was used to run the Defect-aux-net model during the training process. TPUs are custom-built chips designed by Google specifically for accelerating deep learning computations (Figure 4.17). They provide a significant speed boost for large-scale machine learning tasks and are integrated into our Tekniker's computer CPU. TPUs are designed to handle the massive amounts of matrix and vector computations required for deep learning. These computations are the backbone of neural networks, and the TPUs can perform them many times faster than traditional processors such as CPUs and GPUs. TPUs are designed to perform matrix and vector operations in parallel, and they use a unique architecture that allows them to perform these operations much faster than traditional processors. This is because TPUs have a large number of cores, each of which can perform independent computations. The cores are organized into multiple clusters, and each cluster can communicate with other clusters through a high-bandwidth interconnect. This allows the TPUs to perform many operations simultaneously, which increases the overall processing power. The TPU architecture also includes specialized circuits for performing deep learning operations such as convolutions and activations. This is because these operations are commonly used in neural networks, and they can be performed much faster on TPUs than on traditional processors. Additionally, TPUs are designed to handle the large amounts of data required for deep learning, which is often too much for traditional processors to handle efficiently. TPUs are also designed to be energy-efficient, which is important for large-scale ML tasks that can require a lot of power. TPUs are able to perform many operations per second while consuming less power than traditional processors, which makes them more cost-effective for large-scale computations.

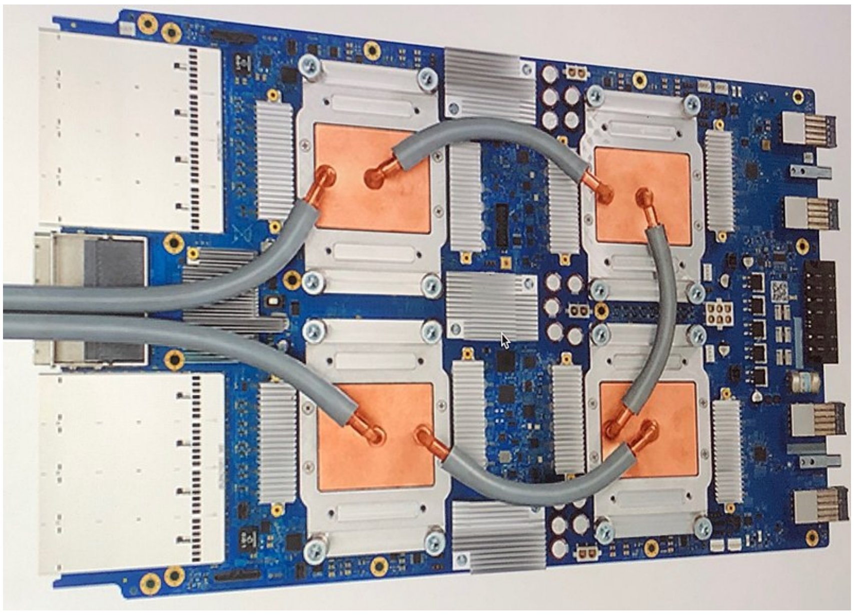


Fig. 4.17: Google's Tensor Processing Unit 3.0 [186]

Defect-aux-Net Training: Training Defect-Aux-Net requires careful consideration of various hyperparameters to achieve optimal performance. In addition to hyperparameter tuning, implementing an efficient input pipeline is also crucial for the Defect-Aux-Net. The input pipeline is responsible for loading and preprocessing the data and can have a significant impact on the training time and overall performance of the model. The pipeline should be optimized for both speed and memory usage, and it should be scalable to handle large amounts of data.

To achieve optimal performance with the Defect-Aux-Net, it is important to give careful consideration to the various hyperparameters involved in the training process. Finding the best hyperparameters involves experimenting with different combinations of values to identify the ones that result in the best accuracy and robustness of the model. This process is known as hyperparameter tuning or hyperparameter search, and it is critical for the success of the model training. By adjusting the values of hyperparameters, the model can be optimized to improve its ability to generalize to new data and to achieve better performance on the task of surface defect identification.

Hyperparameters are configuration parameters that are set before training a model and determine the model's behavior and performance. In Defect-Aux-Net model, some of the important hyperparameters include the learning rate, batch size, Number of neurons in fully connected layers, Weight initialization, L1/L2 Regularization, Dropout rate and epochs.

1. Learning rate: The learning rate determines the size of the steps the model takes to adjust its weights and biases during training. A high learning rate can cause the model to overshoot the optimal solution, while a low learning rate can cause slow convergence.
2. Batch size: The batch size determines the number of training samples used in each iteration of training. A large batch size can lead to stable gradient updates, while a small batch size can introduce more noise and instability.
3. Number of neurons in fully connected layers: The number of neurons in each fully connected layer determines the capacity of the model to learn complex functions. Too few neurons can result in under-fitting, while too many neurons can lead to over-fitting.
4. Weight initialization: The weight initialization determines the initial values of the weights in the model. Poor weight initialization can result in slow convergence or poor performance, while good weight initialization can improve the performance of the model.
5. L1/L2 Regularization: Regularization can prevent over-fitting by adding a penalty term to the loss function. L1 regularization adds a penalty term proportional to the absolute value of the weights, while L2 regularization adds a penalty term proportional to the square of the weights.
6. Dropout rate: The dropout rate determines the fraction of neurons that are dropped out during training. Dropout can prevent over-fitting by reducing the complexity of the model.
7. Epochs: The number of epochs determines the number of times the model is trained on the entire training dataset. Too few epochs can result in under-fitting, while too many epochs can lead to over-fitting.

The performance of the Defect-Aux-Net depends heavily on these hyperparameters. A model with poorly chosen hyperparameters can under-perform or fail to converge, while a model with well-chosen hyperparameters can achieve higher accuracy and faster convergence.

Hyperparameter search can be performed using several methods, including grid search, random search, and Bayesian optimization. Grid search involves testing all combinations of hyperparameters within a specified range. This method is simple and straightforward, but can be computationally expensive and time-consuming. Random search involves randomly sampling hyperparameters from a specified range, which is faster and more efficient than grid search but may not result in the best set of hyperparameters. Bayesian optimization is a more sophisticated method that uses a probabilistic model to guide the search and select the most promising hyperparameters. This method is more efficient than grid search or random search and often results in the best set of hyperparameters. Therefore, we incorporated Bayesian optimization method for choosing the best combination of hyperparameters.

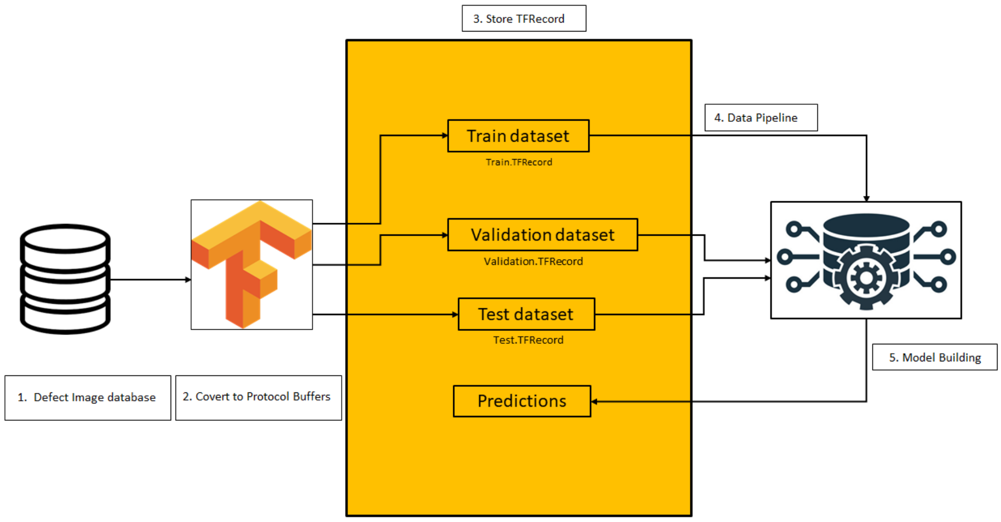


Fig. 4.18: Defect-Aux-net training pipeline using tf.data module.

After training the Defect-Aux-Net model, we utilized tf.saved_model API to save and export the model. The tf.saved_model API is a flexible and efficient way to save TensorFlow models for serving, as it allows us to save the entire model, including its architecture, weights, and training configurations, in a single file. The saved model can be loaded and served as a RESTful API using TensorFlow serving or any other serving tool, such as Flask or Django, giving us the flexibility to choose the serving tool that best meets our needs and requirements. The use of the tf.saved_model API provided us with a simple and efficient way to save the Defect-Aux-Net model for serving, making it easier for us to build and deploy our model for fastener defect detection.

4.3.4 Explainable deep learning algorithms for quality inspection

Once the Defect-Aux-net has been trained, it can be used to automatically inspect new MPI images for defects. The Defect-Aux-net would process the input image and output a prediction for each region of the image, indicating whether it contains a defect or not. This prediction could then be used to flag potential defects for further inspection by a human operator. In order to make the trained Defect-Aux-net model more interpretable and explainable. We compared heatmap of four different XAI methods that indicates the regions in the input image that are most relevant to the model's prediction. A heatmap is a graphical representation of data that uses color-coding to show the relative intensity of values in a matrix to visualize the contribution of each input feature to the model's output.

GradCAM: GradCAM [117] works by computing the gradient of the target class with respect to the feature maps generated by the convolutional layers of the deep learning model. This gradient information is then used to generate a coarse localization map highlighting the regions in the input image that are most important for the model's prediction. The localization map is generated by weighting the gradient information by the importance of each feature map and summing the weighted gradients across all feature maps. The resulting localization map is a coarse heatmap that indicates the regions in the input image that are most relevant to the model's prediction. This visualization can help to understand the model's decision-making process and identify potential errors or biases in the model's predictions. The activation maps generated by Grad-CAM is illustrated in Figure 4.19.

GradCAM++: GradCAM++ [187] is an improved version of GradCAM that addresses some of the limitations of the original method. Some of the advantages of GradCAM++ over GradCAM are:

1. GradCAM++ uses a weighted combination of the gradients from multiple layers, instead of using only the gradients from the final convolutional layer. This allows for a more fine-grained localization and a more detailed visualization of the model's decision-making process.
2. GradCAM++ uses a guided backpropagation technique to compute the gradient information, which reduces the impact of gradients that are not relevant to the target class. This allows for a more accurate and robust localization map, which is less susceptible to noise and irrelevant details.
3. GradCAM++ uses an upsampling and interpolation step to generate the localization map, which ensures that the map is of the same size and resolution as the input image. This allows for a more direct and intuitive comparison

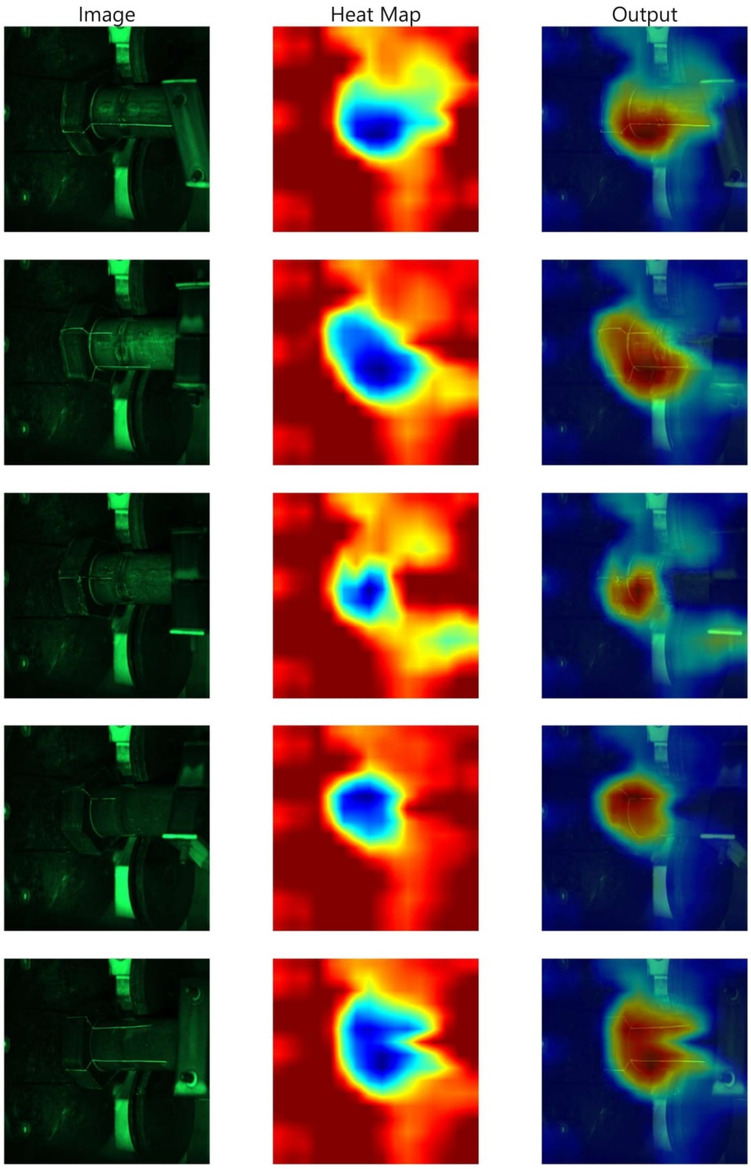


Fig. 4.19: Visualization results of Grad-CAM.

between the input image and the localization map. Overall, GradCAM++ provides a more detailed, accurate, and intuitive explanation of the model's decision-making process, which can help to improve the model's performance and trustworthiness. The activation maps generated by Grad-CAM++ is illustrated in Figure 4.20.

Score-CAM: Unlike previous class activation mapping-based approaches, Score-CAM [188] gets rid of the dependence on gradients by obtaining the weight of each activation map through its forward passing score on target class, the final result

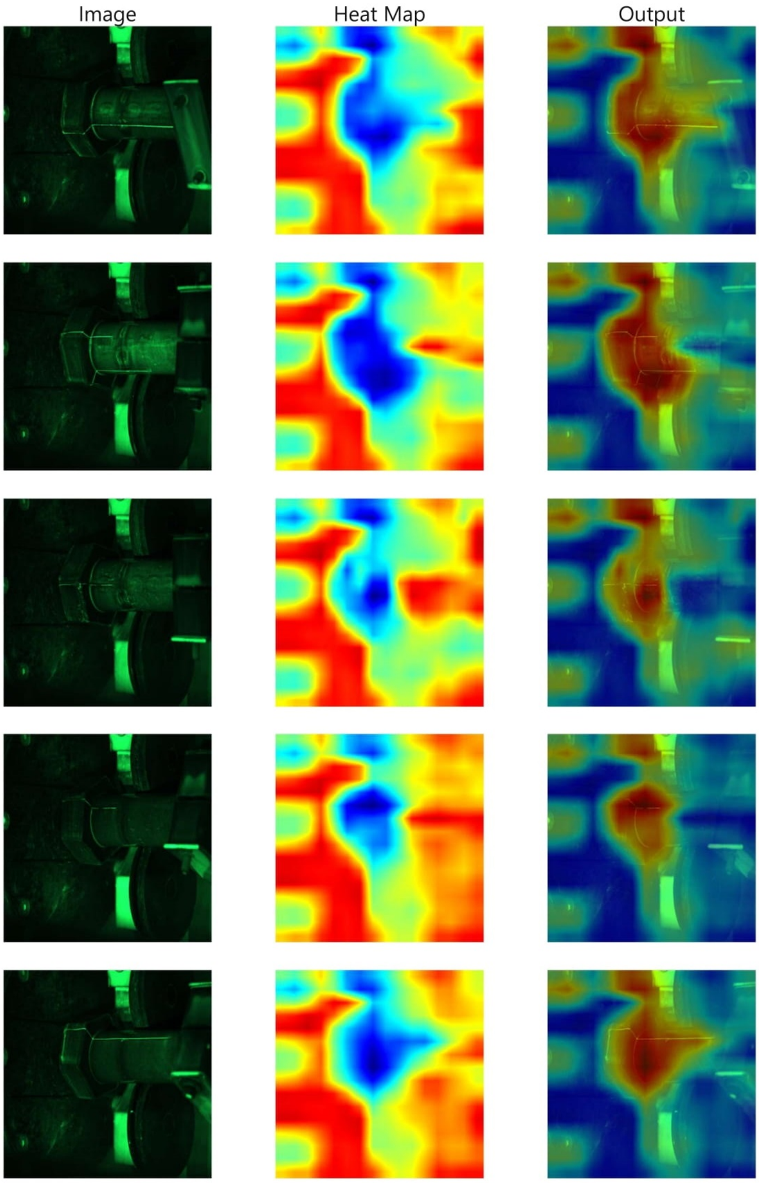


Fig. 4.20: Visualization results of Grad-Cam++.

is obtained by a linear combination of weights and activation maps. Therefore, Score-CAM achieves better visual performance and fairness for interpreting the decision-making process. The activation maps generated by Score-CAM is illustrated in Figure 4.21.

Guided-GradCAM: Guided GradCAM [189] is a combination of GradCAM's map and Guided Backpropagation (GBP) attribution. To compute the Guided GradCAM, the Hadamard product (also known as element-wise multiplication) of the attribution from GBP with a map from GradCAM is computed. Combining GBP and GradCAM

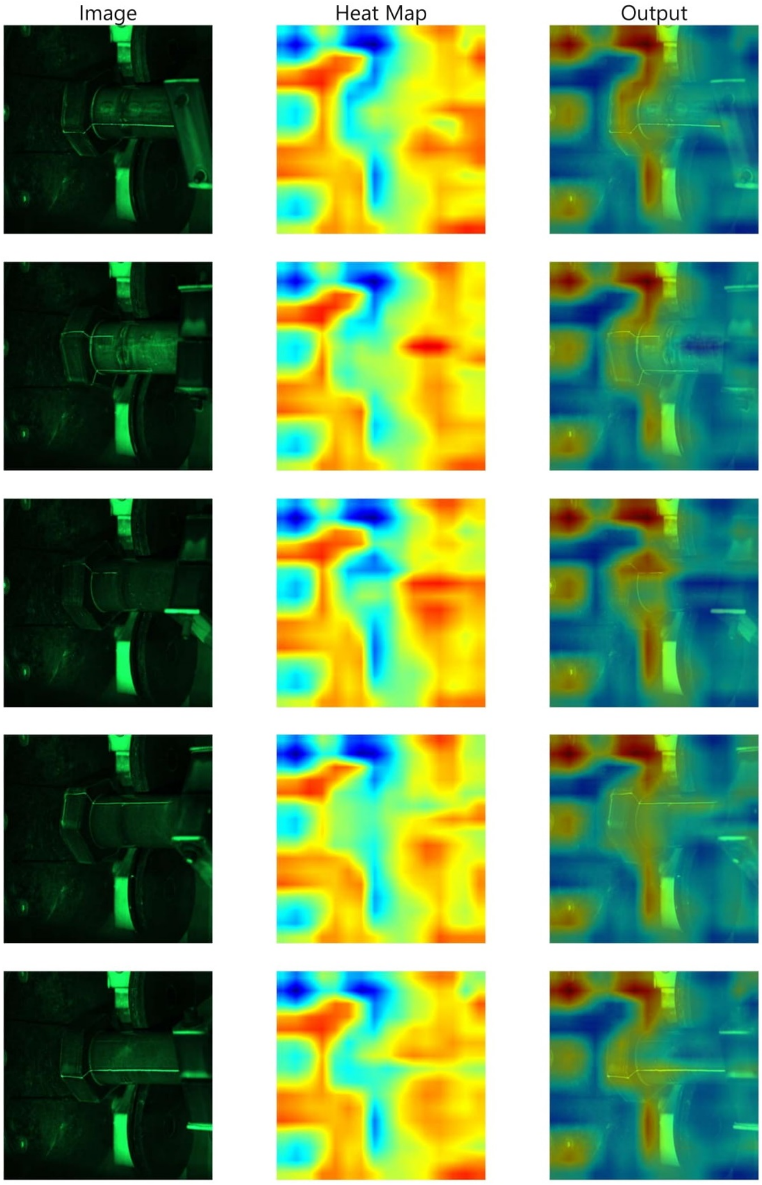


Fig. 4.21: Visualization results of Score-CAM.

allows us to generate sharp attributions. The activation maps generated by Guided Grad-CAM is illustrated in Figure 4.22.

After comparing the heatmaps generated by these methods on a defect detection dataset, we found that GradCAM generated the most interpretable and explainable heatmaps and thus, chose it for our defect detection application. The reason we chose GradCAM for our defect detection application is due to its ability to generate the most interpretable and explainable heatmaps. The heatmaps generated by GradCAM were able to clearly highlight the regions in the input image that were

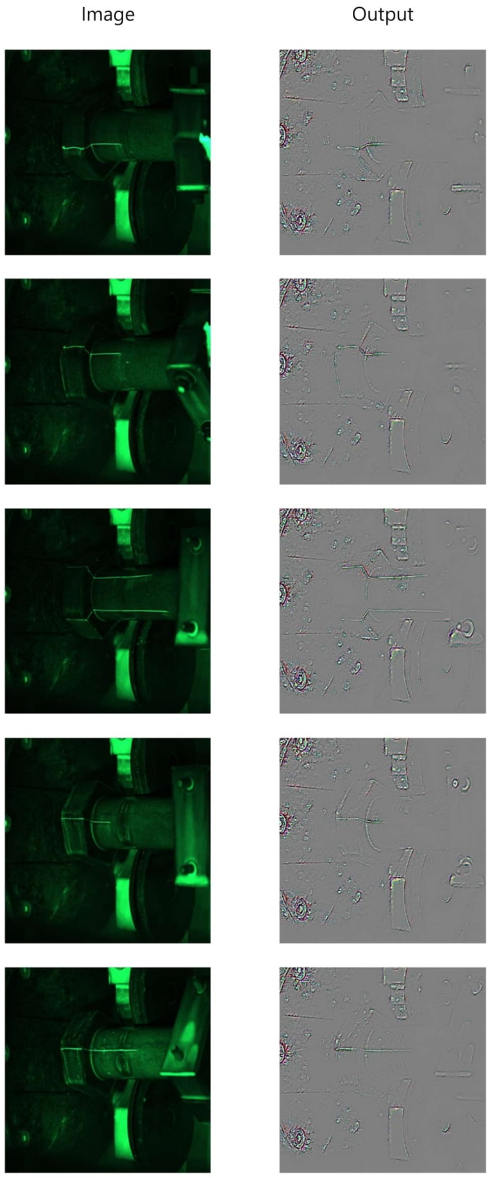


Fig. 4.22: Visualization results of Guided-GradCAM.

most important for the model's prediction, which is crucial in the context of defect detection. By clearly indicating the regions in the input image that are most relevant to the model's prediction, GradCAM provides a better understanding of the model's decision-making process. This increased transparency and accessibility of the model's prediction can help to improve the overall accuracy of the model and make the end-user more confident in its predictions. Furthermore, GradCAM provides a more fine-grained explanation of the model's prediction compared to the other methods. By taking into account the activations of the final convolutional layer, GradCAM is able to provide a more comprehensive explanation of the model's prediction. This helps the end-user to understand why the model is predicting a particular defect, which is crucial in the field of defect detection.

4.3.5 Deploying deep learning models to production

In our effort to make the Defect-Aux-Net model more practical and accessible, we have taken the initiative to convert our trained model as described in subsection 4.3.3 into a smaller, more compact form. The goal was to make the Defect-Aux-Net model easier to deploy and integrate with other systems, particularly the pick and place robotic application. In order to achieve this, we utilized the compression techniques as described in subsection 2.2.5, which allowed us to take the complex Defect-Aux-Net model and reduce its size while still preserving its accuracy. One of the key aspects of this process was the use of knowledge distillation, which allowed us to transfer knowledge from the larger Defect-Aux-Net model to the smaller (Resnet18), more manageable model. This was done by training the smaller model to mimic the behavior of the larger model, using the outputs from the larger model as ground truth during the training process. The result was a smaller, more efficient model that still had the same accuracy as the larger model. Once the smaller model (Resnet18) was created, we then used post quantization techniques to further optimize its performance. This involved converting the model from a floating-point representation to a fixed-point representation, which reduced the memory requirements of the model and allowed for faster computation. Finally, we deployed the model as a REST API using Flask Restful. This allowed us to make the model available to other systems, and to integrate it with the pick and place robotic application. The REST API also provides a flexible and scalable platform for us to modify and update the model as needed. By separating the model from the application, we can now make changes to either one independently, without having to worry about dependencies or compatibility issues.

KUKA KR 60 HA pick and place robotic arm: We have utilized the advanced capabilities of KUKA KR 60 HA pick and place robotic arm to perform crucial manipulation tasks in the MPI process. KUKA KR 60 HA is a robotic arm produced by KUKA, a German manufacturer of industrial robots. It is a 6-axis robot, meaning that it has six degrees of freedom, which allows it to move in a wide range of directions and perform a variety of tasks. The KR 60 HA has a reach of 600 mm and is capable of handling payloads of up to 6 kg. A 3D CAD model of KUKA KR 60 HA is shown in Figure 4.23. It is designed for use in a variety of industries, including automotive, aerospace, and electronics. It can be used for tasks such as welding, assembly, painting, and material handling. The robot is equipped with a controller that allows it to be programmed to perform a variety of tasks. It can be programmed using a variety of programming languages, including KUKA's own KRL (KUKA Robot Language) or more general-purpose languages such as C++ or Python. In terms of safety, the KR 60 HA is equipped with a number of safety features to protect operators and other personnel. It has an emergency stop button, as well as safety sensors that can detect obstacles and stop the robot if necessary.

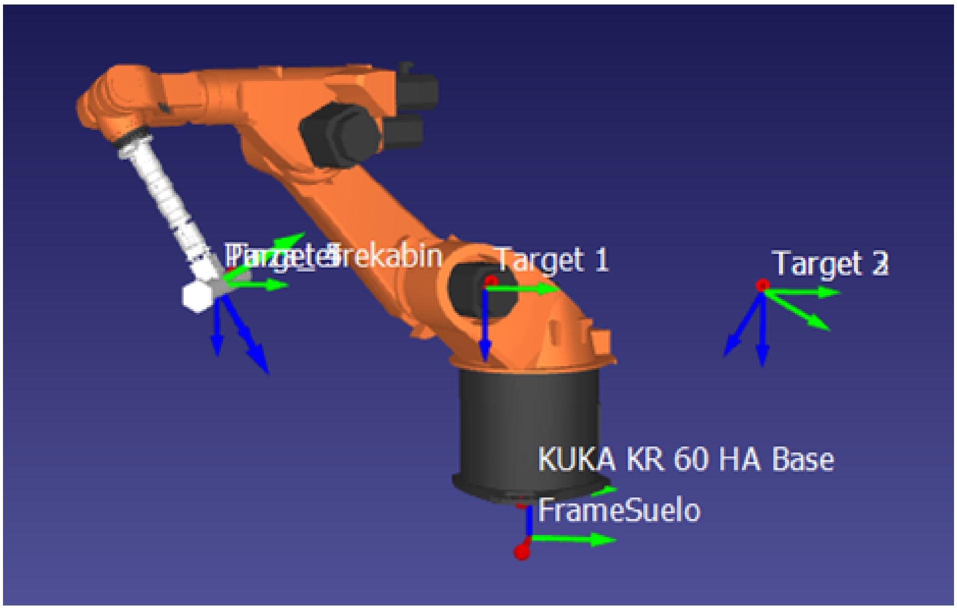


Fig. 4.23: 3D CAD model of KUKA KR 60 HA.

Quality control system: The working principle behind our system is an innovative combination of advanced robotics and computer vision technologies. This system has been designed to provide a complete end-to-end solution for fastener inspection and correction, from the pick and place of fasteners to the detection and correction of defects. The KUKA KR 60 HA robotic arm is the backbone of this system, providing the ability to perform various manupulation tasks such as picking fasteners from a bin and placing them into the magnaflux equipment. The arm is equipped with a highly sophisticated control system that allows it to perform precise movements with a high degree of accuracy and efficiency. The magnaflux equipment, on the other hand, is designed to perform magnetic particle inspection tasks. This process involves magnetizing the fasteners and then spraying iron particle liquid to highlight any defects. The encoder in the magnaflux equipment triggers the cameras to capture images, which are then analyzed by the computer vision-based AI inference engine. The AI inference engine is built on Intel Edge AI processors and has been designed specifically for fastener inspection and defect correction. This engine is capable of analyzing the captured images in real-time, detecting any fastener defects and providing commands to the robotic arm to place the fastener in a separate bin for corrective action. By combining the power of advanced robotics and computer vision technologies, the KUKA KR 60 HA robotic arm and magnaflux equipment provide a highly automated and efficient solution for fastener inspection and correction as shown in Figure 4.24. This system reduces the potential for manual errors and increases production quality, making it an ideal solution for the manufacturing industry.

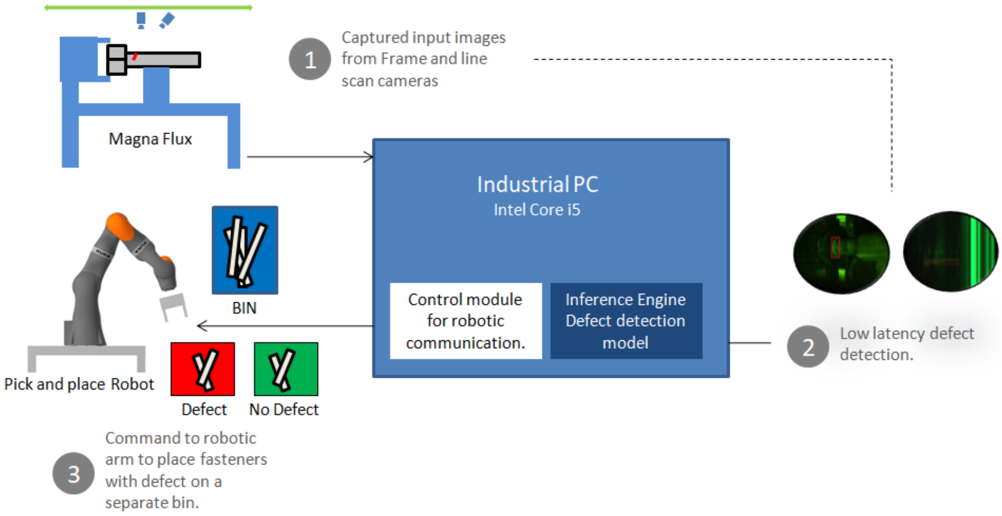

Magna Flux
1
Captured input images from Frame and line scan cameras
Industrial PC
Intel Core i5
Control module for robotic communication.
Inference Engine Defect detection model
2
Low latency defect detection.
BIN
Pick and place Robot
Defect
No Defect
3
Command to robotic arm to place fasteners with defect on a separate bin.

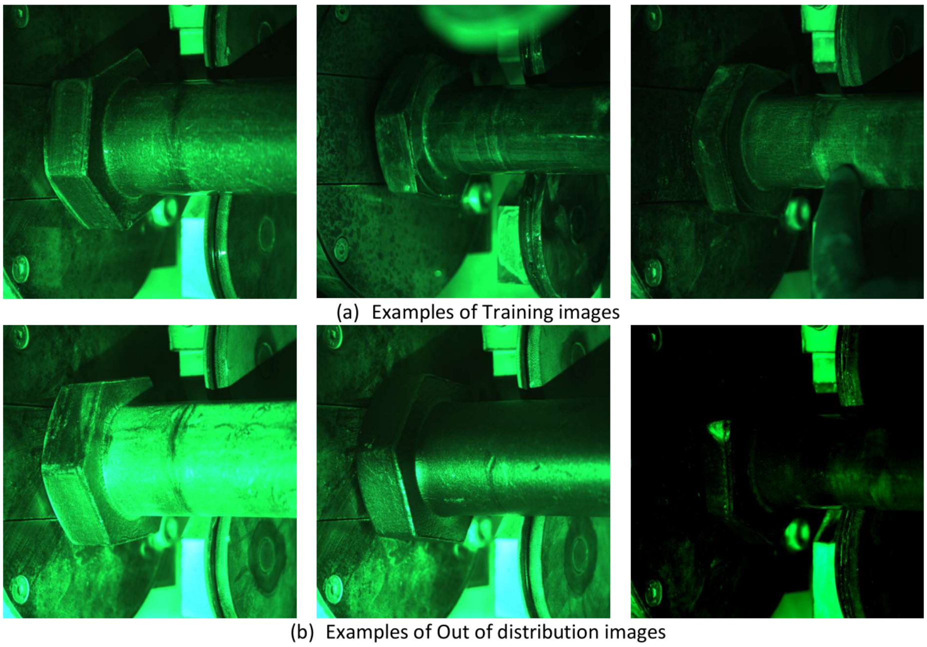
Fig. 4.24: Fasteners Quality inspection system.

Continuous Monitoring: The deployment of the Defect-Aux-Net model was just the first step of the aim towards a fully automated and optimized system. In order to assure that the model performs as expected, we employed MLops to monitor its performance. MLops is a crucial aspect of the development and deployment of machine learning models, and it helps to monitor various aspects of the model to ensure its robustness and accuracy.

- Monitoring Data Invariants: The first step in our MLops process is to monitor data invariants in training and serving inputs. We use data validation mechanisms to alert us if the data being used does not match the schema specified in the training step. This helps to prevent data quality issues that can impact the performance of the model.
- Numerical Stability of the Model: Another important aspect of our MLops process is monitoring the numerical stability of the Defect-Aux-Net model. We continuously monitor the model to ensure that there are no occurrences of NaNs or infinities. These numerical instability issues can impact the model's

performance, and it is crucial to catch them early to prevent any potential failures.

- Computational Performance: In addition to monitoring the numerical stability of the model, we also collect computational performance metrics to monitor the overall performance of the system. These metrics include CPU memory allocation, network traffic, and disk usage, which are crucial for estimating cloud costs and optimizing the system for optimal performance.
- Predictive Quality of the Model: To ensure that the model remains accurate and continues to perform well, we also monitor the degradation of the predictive quality of the model over time. Both slow and dramatic degradations in prediction quality should be notified to ensure that corrective actions can be taken to maintain the model's performance.
- Out of Distribution Data: Finally, we also monitor out of distribution data to identify any unexpected data inputs that might impact the model's performance. Comparison of training images and out of distribution images are demonstrated in Figure 4.25. Out of distribution images appear more contrasted, with brighter and darker pixels, compared to the more uniform brightness in the training images.When out of distribution data is identified, it is continuously collected in a database to be used for future retraining of the model.


(a) Examples of Training images

(b) Examples of Out of distribution images

Fig. 4.25: Examples of (a) Training (b) Out of distribution images.

The next step after monitoring the performance of the Defect Detection model is to improve its accuracy and efficiency. This is done through a combination of data-centric and model-centric approaches. The data-centric approach involves collecting more out-of-distribution data from the database or Magna-Defect-GAN model and using it to retrain the model. This helps to improve the performance of the model by increasing its exposure to different variations of data. On the other hand, the model-centric approach involves modifying the architecture of the model or using more advanced techniques such as knowledge distillation and post-quantization to improve its performance. These approaches are applied iteratively until the model is able to perform optimally on the given data. By combining both data-centric and model-centric approaches, the performance of the Defect Detection model can be greatly improved.

4.4 Conclusions

In conclusion, this PhD study has made several important contributions to the scientific community, specifically in the area of surface defect inspection and quality control using magnetic particle inspection. The first contribution is the design of a reliable image acquisition system that combines both frame and line scan cameras to capture high resolution images of both the head and shank portion of fasteners at high speeds. This methodology offers improved accuracy and repeatability, which has the potential to significantly improve the efficiency and effectiveness of fastener inspection processes.

Another significant contribution is the creation of the TekErreka dataset, a collection of magnetic particle inspection-based fasteners defect classification images that is of sufficient quality and quantity for training deep learning algorithms. This dataset is a valuable resource for industrial practitioners in the field of magnetic particle inspection and defect classification.

The study also presents a novel Pixel level image augmentation method based on mask-to-image translation with GAN conditioned on fine grained labels, referred to as the Magna-Defect-GAN model. This method has the ability to generate realistic and high-resolution surface defect images and has been shown to improve the accuracy of other surface defect identification models.

Additionally, the study proposes a novel deep learning model called the Defect-Aux-Net, which takes advantage of related tasks to simultaneously improve the robustness and accuracy of the CNN-based surface defect identification. The model achieved outstanding results in terms of accuracy, dice score, and mean average precision on defect classification, segmentation, and detection tasks.

Moreover, the study explored the combination of knowledge distillation and post quantization methods to reduce the size and improve the speed of real-time inference for machine learning models on edge devices. The results showed that this method can achieve a 4x reduction in model size and memory bandwidth requirements. The iterative training of the Defect-Aux-Net model using both data-centric and model-centric approaches, as well as the implementation of MLops to monitor the model's performance, demonstrated a commitment to ensuring that the model performed as expected and was of the highest quality.

Finally, we designed a comprehensive and innovative system for fastener inspection and correction that utilizes advanced robotics and Defect-Aux-net model. The system is centered around the KUKA KR 60 HA pick and place robotic arm, which provides the ability to perform crucial manipulation tasks with high accuracy and efficiency. The magnaflux equipment performs magnetic particle inspection, while the AI inference engine built on Intel Edge AI processors analyzes the images captured by the cameras and detects any fastener defects. This system provides a complete end-to-end solution for fastener inspection and correction and has the potential to revolutionize the MPI process. The use of advanced technologies such as robotics and computer vision not only increases the efficiency and accuracy of the inspection process but also reduces human intervention, thereby minimizing the chances of human error. This cutting edge system was developed at Tekniker and has been put into use at Erreka Fastening Solutions. The system is a testament to the possibilities of combining advanced technologies for practical and impactful applications.

4.5 Future Work

In the field of AI-based surface defect detection, immediate future research should focus on certifying the Defect-Aux-Net model as per the standards established by the European Federation for Non-Destructive Testing (EFNDT) or the American Society for Non-Destructive Testing (ASNT). The manufacturing industry heavily relies on certified quality inspectors to perform non-destructive testing and ensure the quality of their products. While the Defect-Aux-Net model has shown promising results in detecting surface defects, it cannot replace the expertise of certified quality inspectors yet.

To gain the trust of industries and to be widely adopted, Defect-Aux-Net needs to undergo rigorous testing and certification processes. These tests should assess the accuracy and reliability of the model in detecting various types of surface defects and its ability to meet the standards set by EFNDT and ASNT. Additionally, efforts should be made to further improve the model's robustness and reliability. Once certified, Defect-Aux-Net can be a valuable tool for certified quality inspectors, helping them speed up their tasks and reducing the risk of human error. In the long run, Defect-Aux-Net has the potential to replace certified quality inspectors, provided that it is thoroughly tested and meets the standards set by industry regulators. Moreover, future work can also focus on expanding the scope of the Defect-Aux-Net model to detect other types of surface defects, as well as incorporating additional data sources such as temperature and humidity data, to improve its accuracy and robustness.

The development of an active learning system for defect detection models represents an another important area of future research. The current defect detection models often struggle to adapt to new environments, particularly those that are significantly different from the training data. This is because the models lack the ability to learn on the fly and to retrain themselves when faced with new data. An active learning system for defect detection models would address this limitation by providing the model with the ability to continuously learn from new data, and to adapt to changing environments. This system would be able to actively seek out new data, select the most informative samples, and incorporate them into the model's training process. The result would be a model that is able to continuously improve its performance over time, even in the presence of new or changing data.

There are several key challenges associated with the development of an active learning system for defect detection models. Firstly, the system must be able to effectively identify the most informative samples for retraining. This requires the development of new selection algorithms that can accurately identify samples that are likely to improve the model's performance. Secondly, the system must be able to efficiently retrain the model with the selected samples. This requires the optimization of the training process to ensure that it is fast and computationally efficient. Finally, the system must be able to effectively integrate the newly learned information into the model in a way that preserves its overall performance.

Another promising future work in the area of end-to-end manufacturing process of fasteners involves utilizing the latest advances in artificial intelligence and robotics to create a fully automated and optimized system. This can be achieved by integrating multiple pick place robots and AI-based models in each step of the manufacturing process, from the production of fasteners to their final shipment as shown in

Figure 4.26. For example, the production process can be enhanced by using AI-based models to optimize the manufacturing process, such as optimizing the parameters of the manufacturing equipment to increase the production efficiency and quality. In the palletizing process, pick place robots can be utilized to automate the palletizing of fasteners and ensure that the pallets are loaded correctly and efficiently. AI-based models can also be used in the quality inspection process to accurately detect any defects and ensure that only high-quality fasteners are packaged and shipped. The integration of these technologies in the manufacturing process can provide numerous benefits, such as increased efficiency, improved quality, reduced manual labor, and decreased waste. By implementing these advances in technology, the future work in the end-to-end manufacturing process of fasteners will add significant value to the manufacturing industries and help to create a more sustainable and efficient production system.

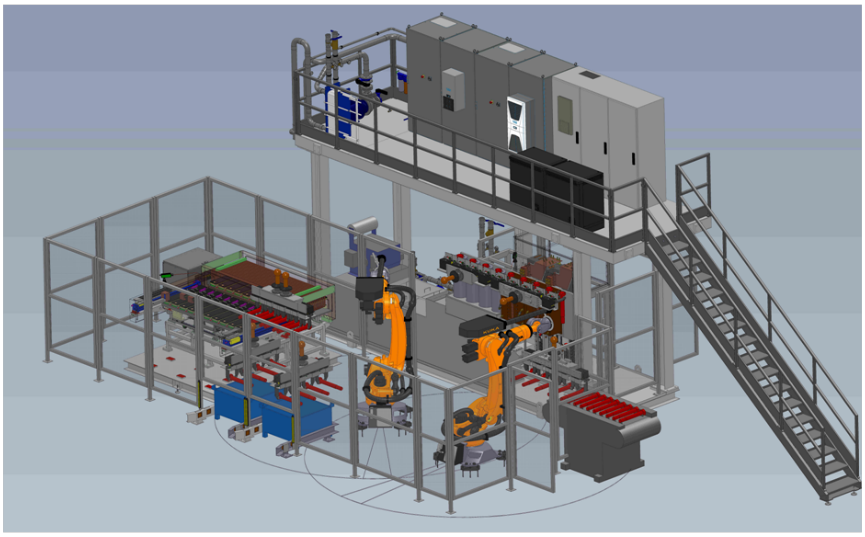


Fig. 4.26: Conceptual 3D model of end-to-end manufacturing process of fasteners.

5. SPANISH VERSION

Spanish version 5

5.1 Resumen

En esta sección se describen los objetivos, las contribuciones y la metodología utilizados en esta tesis.

Objetivos: En el proceso de producción de elementos de fijación, la inspección por partículas magnéticas permite detectar algunos defectos superficiales y cercanos a la superficie que, de otro modo, no serían visibles para el ojo humano.
Los elementos de fijación pueden presentar una variedad de defectos de producción debidos a múltiples causas: el uso de materias primas de baja calidad, procesado o tratamiento térmico inadecuado, mecanizado incorrecto, manipulación irresponsable, etc. La identificación de esos defectos la realizan operarios cualificados mediante la inspección visual de las piezas una vez aplicadas las partículas magnéticas.
La inspección visual de los productos fabricados ha sido una de las técnicas de inspección más comunes e importantes en las aplicaciones de control de calidad industrial. Sin embargo, la inspección visual presenta diversas características que pueden hacer disminuir la calidad del resultado del proceso de control: requiere una elevada concentración por parte de los operarios, que unido a una posible fatiga de los mismos, pueden dar lugar a incertidumbre y reducción de la precisión, en particular cuando se realiza sobre piezas pequeñas o con detalles sutiles. Y no menos relevante, está sujeto a la subjetividad humana.
El objetivo general de este trabajo de doctorado es desarrollar un sistema de visión basado en IA totalmente automatizado para inspeccionar toda la superficie de elementos de fijación, basado en la técnica de ensayo de partículas magnéticas.
Un sistema automatizado de inspección visual basado en IA se compone generalmente de tres subsistemas principales. En primer lugar, el subsistema de adquisición de imágenes cuya función es recoger datos de cámaras u otros dispositivos y proporcionar su representación digital. La información de la imagen, como el color, el brillo, la intensidad y la dispersión de la luz, puede ayudar a los modelos de aprendizaje profundo. En segundo lugar, un subsistema de procesamiento de imágenes por software basado en algoritmos de aprendizaje

profundo compuesto principalmente de modelos de aprendizaje profundo diseñados para analizar los datos adquiridos y proporcionar el resultado final de la inspección. En tercer lugar, un sistema robótico para la manipulación de las piezas, cogiéndolas de un contenedor, introduciéndolas en el sistema de inspección y finalmente dejándolas en el contenedor correspondiente de pieza buena o mala en función del resultado del análisis.

Este proyecto puede dividirse en los siguientes objetivos específicos:

1. El primer objetivo de la tesis consiste en diseñar el subsistema de adquisición de imágenes que implica elegir los sensores y cámaras adecuados, así como el hardware y software necesarios para capturar imágenes de alta calidad. El diseño del sistema debe tener en cuenta los requisitos específicos de la aplicación, como el tipo de elementos a inspeccionar, las escenas que se van a capturar, las condiciones de iluminación y la distancia y resolución de las imágenes. El sistema también debe ser capaz de capturar imágenes a una frecuencia que permita su procesamiento y análisis en tiempo real. Además, el sistema debe ser robusto y fiable, y debe ser capaz de manejar el ruido y otras fuentes de incertidumbre.

   Existe una gran variedad de tipos de defectos que se localizan en la superficie de los elementos de fijación objeto de este trabajo, caracterizada por el tamaño de los defectos, su forma, su ubicación y su causa probable. Por ejemplo, algunos de estos defectos pueden identificarse como marcas, arañazos, deformaciones geométricas, etc. El sistema de adquisición propuesto en este trabajo, basado en la técnica de inspección de partículas magnéticas (MPI), debe capturar todos esos tipos de defectos.
2. La eficiencia de un sistema de inspección basado en MPI está influenciada por diferentes factores. 1) La concentración de partículas, 2) la contaminación de la suspensión, 3) la iluminación, 4) el tamaño, la forma, el tipo, la orientación y la profundidad de los defectos.

   Debido a la baja probabilidad de que se produzcan defectos, la recogida de un elevado número de imágenes defectuosas con diversas combinaciones de estas características (varianza intraclase) es muy costosa. Como consecuencia de esa falta de variabilidad, la capacidad de generalización del modelo de detección de defectos se ve afectada negativamente. Además, en la mayoría de los casos no es factible recopilar el mismo número de imágenes para cada clase (desequilibrio de clases), por ejemplo, la parte del vástago de los elementos de fijación puede presentar más defectos que la región de la rosca. En consecuencia, el segundo objetivo de esta tesis es asegurarse de que los datos reflejan adecuadamente el espectro que el sistema detección de defectos debe aprender.

El enfoque propuesto debe ser capaz de resolver varios problemas relacionados con los datos adquiridos: 1. Incoherencia de las etiquetas, 2. Mala calidad de las imágenes, 3. Baja calidad de los datos. 4. Escasas variaciones de las imágenes adquiridas (desequilibrio intraclase) 5. Desequilibrio interclase.

3. El tercer objetivo es desarrollar un modelo de inspección de defectos superficiales basado en el aprendizaje profundo.

   Los defectos superficiales suelen tener diferentes escalas, lo que dificulta la identificación de los defectos de pequeño tamaño mediante los modelos de aprendizaje profundo. Por un lado, la apariencia visual de los defectos superficiales varía con el tipo de materiales, las condiciones de la imagen y la posición de la cámara. Por otro lado, es difícil distinguir los defectos más pequeños del ruido o de los que no lo son en una imagen, dando lugar a la presencia de falsos positivos que, desde un punto de vista industrial, es importante limitarlos para evitar paradas innecesarias de la producción, al tiempo que se garantiza el máximo rendimiento en la detección de defectos y la adopción temprana de medidas correctoras en la producción. Se consigue así optimizar la utilización de las materias primas y el rendimiento del conjunto del proceso.

4. Debido a su estructura multicapa no lineal, los modelos modernos de aprendizaje automático, como las redes neuronales profundas, suelen denominarse modelos de "caja negra". Tienen millones de parámetros que deben aprenderse y, para clasificar una sola imagen, deben realizarse millones de operaciones por inferencia. Es difícil comprender cómo interactúa cada neurona individual con las demás para producir el resultado, e incluso lo que hace cada neurona concreta. El cuarto objetivo de esta tesis es emplear algoritmos de IA explicable para analizar y comprender las predicciones proporcionadas por los algoritmos de aprendizaje profundo.

   Si los modelos de aprendizaje profundo pueden hacer que su razonamiento sea comprensible para los inspectores de calidad, estos pueden utilizar el conocimiento del dominio para asegurarse de que el modelo de IA se centra en los predictores correctos.

5. El objetivo final consiste en implementar una aplicación de detección de defectos que pueda utilizarse en líneas de producción. Esta aplicación debe ser capaz de manipular e inspeccionar un elevado número de piezas por minuto y proporcionar resultados de inspección más consistentes y fiables que los inspectores humanos.

Durante la inspección de calidad, los elementos de fijación son manipulados y colocados en el banco de partículas magnéticas por medio de un robot industrial de modo desatendido.

Contribuciones científicas: Las principales contribuciones de este estudio de doctorado a la comunidad científica son:

1. Una revisión actualizada de las técnicas de GAN para problemas de desequilibrio en las tareas de visión por computador que proporciona información de los últimos desarrollos y componentes técnicos esenciales en el campo de la generación de imágenes sintéticas basadas en GAN. Además, recoge las aplicaciones del mundo real donde las imágenes sintéticas basadas en GAN se utilizan para aliviar los desequilibrios y llena un vacío de investigación en el uso de imágenes sintéticas para los problemas de desequilibrio en las tareas de reconocimiento visual.

   Presenta un enfoque sistemático para clasificar las tareas de ML tanto desde la perspectiva de los "cuatro conocimientos" (saber qué, saber por qué, saber cuándo, saber cómo) como de los "cuatro niveles" (producto, proceso, máquina, sistema). El estudio proporciona un procedimiento práctico para que los ingenieros de producción lo utilicen a la hora de implementar soluciones de ML, desde la concepción hasta el despliegue.
2. Una segunda contribución científica de esta tesis es el desarrollo de un conjunto de datos para la clasificación de defectos en elementos de fijación basado en la inspección por partículas magnéticas, que consta de miles de imágenes capturadas con cámaras industriales de alta resolución. Este conjunto de datos, al que denominamos TekErreka, es especialmente útil para entrenar algoritmos de aprendizaje profundo con el fin de identificar y clasificar defectos en elementos de fijación utilizando la inspección por partículas magnéticas.
3. Desarrolla un novedoso método de aumentación de imagen a nivel de pixel que se basa en la conversión de máscara a imagen con GAN condicionado a etiquetas de grano fino. El modelo Magna-Defect-GAN propuesto puede controlar el proceso de generación de imágenes y generar muestras de imágenes con variaciones fotorrealistas. Los resultados experimentales demuestran que el modelo Magna-Defect-GAN propuesto puede generar imágenes de defectos superficiales realistas y de alta resolución hasta la resolución de 512×512 de forma controlada. También muestra que el método de aumentación propuesto puede aumentar la precisión y adaptarse fácilmente a cualquier otro modelo de identificación de defectos superficiales.
4. Propone un nuevo modelo de aprendizaje profundo llamado "Defect-Aux-Net" basado en el aprendizaje multitarea con mecanismos de atención que

aprovechan la rica información adicional de tareas relacionadas con el fin de mejorar simultáneamente la robustez y la precisión de la identificación de defectos superficiales basada en CNN.

Según los resultados experimentales, la Defect-Aux-Net es capaz de mejorar el rendimiento de los modelos más avanzados, obteniendo una precisión global del 97.1%, una puntuación Sørensen–Dice coefficient (Dice) de 0.926 y un Promedio de precisión media (mAP) de 0,762 en tareas de clasificación, segmentación y detección de defectos.

5. Estudia la combinación de métodos de destilación de conocimiento y postcuantización para reducir significativamente el tamaño y mejorar la velocidad de inferencia en tiempo real de modelos de aprendizaje automático en dispositivos de recursos computacionales limitados.

   El método propuesto utiliza inicialmente la destilación de conocimiento para entrenar un modelo más pequeño y, a continuación, aplica la postcuantización a este modelo. Los resultados experimentales muestran que el método propuesto puede reducir 4 veces el tamaño del modelo y 4 veces los requisitos de ancho de banda de memoria.

Metodología: La metodología de solución que proponemos implica la combinación de varios enfoques para crear una aplicación robusta de visión por ordenador. Estos enfoques incluyen:

1. Sistema de adquisición de imágenes: El primer paso en nuestra metodología propuesta es diseñar un sistema de adquisición de imágenes fiable para adquirir imágenes del proceso MPI.

   Se propone utilizar una combinación de cámaras matriciales y de barrido lineal para capturar imágenes de la zona de la cabeza y el vástago de los elementos de fijación, respectivamente. Una de las principales ventajas de esta combinación es la capacidad de capturar imágenes de alta resolución.

   El uso de una cámara matricial para capturar un amplio campo de visión de las fijaciones, junto con el uso de una cámara de barrido lineal para capturar imágenes de alta resolución de la parte del vástago de los elementos de fijación, permite un análisis más detallado del acabado de la totalidad de la superficie y de las tolerancias dimensionales de los elementos de fijación. Otra ventaja de esta metodología es la capacidad de capturar imágenes a alta velocidad para dar respuesta a la alta cadencia de producción.

   Mediante la combinación de cámaras matriciales y de barrido lineal, es posible capturar imágenes de los elementos de fijación desde múltiples ángulos y puntos de vista, lo que puede contribuir a mejorar la precisión y fiabilidad del análisis de imágenes.

2. Enfoque de aprendizaje profundo centrado en los datos utilizando una combinación de aumentación de datos tradicionales e imágenes sintéticas basadas en GAN para ampliar el tamaño y la diversidad del conjunto de datos de entrenamiento.

   La aumentación de datos se utiliza para ampliar artificialmente el tamaño del conjunto de datos de entrenamiento generando nuevas muestras sintetizadas que son variaciones de las muestras originales. Esto ayuda a mejorar la capacidad de generalización del modelo y reducir el sobreajuste.

   Un método de aumentación habitual consiste en aplicar transformaciones a las imágenes existentes para crear nuevas imágenes sintéticas. Por ejemplo, aplicando diferentes rotaciones, traslaciones o escalados a las imágenes existentes, o añadiendo ruido o desenfoque a las imágenes.

   También utilizamos imágenes sintéticas basadas en GAN para generar muestras adicionales para el conjunto de datos de entrenamiento. Las GAN son un tipo de red neuronal que puede aprender a generar imágenes sintéticas indistinguibles de las reales. Mediante su utilización es posible ampliar enormemente el tamaño y la diversidad del conjunto de datos de entrenamiento, lo que puede mejorar la precisión del modelo. Al combinar estos dos enfoques, aprovechamos los puntos fuertes de ambos para lograr un mejor rendimiento en la detección de defectos.

   El enfoque centrado en los datos ayuda a aumentar el tamaño y la diversidad de los datos de entrenamiento, mientras que el enfoque centrado en el modelo permite a éste aprender de múltiples fuentes de información y generalizar mejor a nuevas tareas.

3. Aprendizaje profundo centrado en el modelo mediante el uso del aprendizaje multitarea que implica el entrenamiento de un único modelo para realizar múltiples tareas relacionadas simultáneamente.

   Esto puede ayudar a mejorar el rendimiento del modelo de detección de defectos al permitirle aprender de múltiples fuentes de información y generalizar mejor a nuevas tareas. Esto se debe a que el modelo puede aprender de las características compartidas de las distintas tareas, así como de las características específicas de cada una de ellas. Además, al utilizar un modelo de aprendizaje multitarea, se consigue un proceso más eficiente.

   Estas características son especialmente útiles en la detección de defectos, donde el modelo puede necesitar identificar defectos en contextos nuevos. Además, el modelo de aprendizaje multitarea propuesto puede ser más interpretable que un modelo de una sola tarea, ya que puede proporcionar información sobre las características compartidas que son relevantes para múltiples tareas, permitiendo identificar patrones o características comunes que pueden ser indicativos de defectos.

En resumen, combinando los enfoques centrados en el modelo y en los datos, es posible aprovechar los puntos fuertes de ambos enfoques para mejorar el rendimiento en la detección de defectos. El enfoque centrado en los datos ayuda a aumentar el tamaño y la diversidad de los datos de entrenamiento, mientras que el enfoque centrado en el modelo permite aprender de múltiples fuentes de información y generalizar mejor a nuevas tareas.

4. IA explicable: Para que el modelo de detección de defectos sea más interpretable y explicable, estudiamos y comparamos mapas térmicos de cuatro métodos de aprendizaje profundo explicables, incluidos GradCAM, GradCAM++, Score-CAM y Guided GradCAM, que indican las regiones de la imagen de entrada más relevantes para la predicción del modelo.

   GradCAM es una técnica que utiliza los gradientes de la clase objetivo con respecto a las activaciones de la capa convolucional final para calcular un mapa de calor. GradCAM++ es una extensión de GradCAM que tiene en cuenta las activaciones de múltiples capas. Score-CAM utiliza puntuaciones de la clase objetivo para calcular el mapa de calor, mientras que Guided GradCAM utiliza una técnica de retropropagación guiada para resaltar las regiones de la imagen de entrada que son más importantes para la predicción del modelo.

   Tras comparar los mapas de calor generados por estos métodos en un conjunto de datos de detección de defectos, se concluye que GradCAM genera los mapas de calor más interpretables y explicables para nuestro caso y, por tanto, ha sido el elegido para la aplicación en la detección de defectos.

5. Se ha utilizado la combinación de métodos de destilación de conocimiento y postcuantización para reducir el tamaño y mejorar la velocidad de inferencia en tiempo real para modelos de detección de defectos basados en aprendizaje profundo en dispositivos de borde.

   Para implementar este método, el modelo más grande se utiliza primero para generar etiquetas blandas para los datos de entrenamiento. A continuación, el modelo más pequeño se entrena utilizando estas etiquetas blandas, con el objetivo de replicar el comportamiento del modelo más grande lo más fielmente posible. Una vez entrenado el modelo más pequeño, se cuantiza mediante técnicas de poscuantización para reducir aún más su tamaño y mejorar su rendimiento en los dispositivos de menores capacidades computacionales ("edge”).

5.2 Conclusiones

En conclusión, este estudio de doctorado ha realizado varias contribuciones importantes a la comunidad científica en el ámbito de la inspección de defectos superficiales y el control de calidad mediante inspección por partículas magnéticas. La primera contribución es el diseño de un sistema fiable de adquisición de imágenes que combina cámaras de barrido lineal y matriciales para capturar imágenes de alta resolución tanto de la parte de la cabeza como del vástago de los elementos de fijación a altas velocidades. Esta metodología ofrece una mayor precisión y repetibilidad, lo que tiene el potencial de mejorar significativamente la eficiencia y eficacia de los procesos de inspección.

Otra contribución significativa es la creación del conjunto de datos TekErreka, un conjunto amplio de imágenes de alta calidad para la clasificación de defectos de elementos de fijación basada en la inspección por partículas magnéticas mediante algoritmos de aprendizaje profundo. Este conjunto de datos es un recurso valioso para los profesionales de la industria en el campo de la inspección por partículas magnéticas y la clasificación de defectos.

El estudio también presenta un novedoso método de aumentación de imágenes a nivel de píxel basado en la conversión de máscara a imagen con GAN condicionado a etiquetas de grano fino, denominado modelo Magna-Defect-GAN. Este método tiene la capacidad de generar imágenes de defectos superficiales realistas y de alta resolución y ha demostrado mejorar la precisión de otros modelos de identificación de defectos superficiales.

Además, el estudio propone un novedoso modelo de aprendizaje profundo llamado Defect-Aux-Net, que aprovecha las tareas relacionadas para mejorar simultáneamente la robustez y la precisión de la identificación de defectos superficiales basada en CNN. El modelo obtuvo resultados sobresalientes en términos de exactitud, puntuación de dados y precisión media en las tareas de clasificación, segmentación y detección de defectos.

Además, el estudio ha explorado la combinación de métodos de destilación de conocimiento y postcuantización para reducir el tamaño y mejorar la velocidad de inferencia en tiempo real de modelos de aprendizaje automático en dispositivos de recursos computacionales limitados (“edge”). Los resultados han mostrado que este método puede lograr una reducción de 4 veces en el tamaño del modelo y los requisitos de ancho de banda de memoria. El entrenamiento iterativo del modelo Defect-Aux-Net utilizando enfoques centrados tanto en los datos como en el modelo, así como la implementación de Operaciones de aprendizaje automático (MLops) para supervisar el rendimiento del modelo, ha sido validado obteniéndose resultados de la máxima calidad.

Por último, se ha implementado un sistema completo e innovador para la inspección y corrección de elementos de fijación que utiliza robótica avanzada y el modelo Defect-Aux-net. El sistema utiliza un brazo robótico KUKA KR 60 HA que proporciona la capacidad de realizar tareas de manipulación de las piezas a inspeccionar con gran precisión y eficiencia. El equipo Magnaflux realiza el proceso de aplicación de partículas magnéticas, mientras que el motor de inferencia de IA integrado en los procesadores Intel Edge AI analiza las imágenes captadas por las cámaras y detecta los defectos de los elementos de fijación.

Este sistema proporciona una solución completa para la inspección de elementos de fijación y tiene el potencial de revolucionar el proceso MPI. El uso de tecnologías avanzadas como la robótica y la visión por ordenador no sólo aumenta la eficacia y precisión del proceso de inspección, sino que también reduce la intervención humana, minimizando así las posibilidades de error humano y mejorando la ergonomía, dado que las piezas pueden alcanzar un peso de hasta 20 kg. Este sistema de vanguardia se ha desarrollado en Tekniker y se ha puesto en marcha en Erreka Fastening Solutions. El sistema es un testimonio de las posibilidades de combinar tecnologías avanzadas para aplicaciones prácticas y de gran impacto industrial.

6. DISSEMINATION OF RESULTS

Dissemination of results

6

Published articles:

- **Journal of Big Data 2021:** V. Sampath, I. Maurtua, J. J. Aguilar Martín, and A. Gutierrez, “A survey on generative adversarial networks for imbalance problems in computer vision tasks,” J. Big Data, vol. 8, no. 1, pp. 1–59, Dec. 2021.
- **IEEE Transactions on Industrial Informatics 2023:** V. Sampath, I. Maurtua, J. J. A. Martín, A. Rivera, J. Molina and A. Gutierrez, "Attention Guided Multi-Task Learning for Surface defect identification," in IEEE Transactions on Industrial Informatics, doi: 10.1109/TII.2023.3234030.
- **Sensors 2023:** V. Sampath, I. Maurtua, J. J. Aguilar Martín, A. Iriondo, I. Lluvia, and G. Aizpurua, “Intraclass Image Augmentation for Defect Detection Using Generative Adversarial Neural Networks,” Sensors, vol. 23, no. 4, p. 1861, Feb. 2023, doi: 10.3390/s23041861.
- **Applied Sciences 2023:** T. Chen, V. Sampath, M.C. May, S. Shan, O.J. Jorg, J.J. Aguilar Martin, F. Stamer, G. Fantoni, G. Tosello, M.Calaon, “Machine Learning in Manufacturing towards Industry 4.0: From ‘For Now’ to ‘Four-Know,’”Applied Sciences, vol. 13, no. 3, p. 1903, Feb. 2023, doi: 10.3390/app13031903.

Conference contributions:

- **International Conference on Electrical, Computer and Energy Technologies (ICECET) 2022:** V. Sampath, I. Maurtua, J. J. Aguilar Martín, A. Iriondo, I. Lluvia and A. Rivera, "Vision Transformer based knowledge distillation for fasteners defect detection," 2022 International Conference on Electrical, Computer and Energy Technologies (ICECET), 2022, pp. 1-6, doi: 10.1109/ICECET55527.2022.9872566.

Research Training at DTU:

Training 1: DIGIMAN4.0 Innovation Management and Digital Transformation Workshop.

Position: Visiting Scholar in the Department of Mechanical Engineering at the Technical University of Denmark (DTU).

Period: From 25th April 2022 to 06th May 2022.

About Workshop: The training course was oriented to provide a technical and economic framework to explain the rationale behind Industry 4.0 and more in general digital transformation. The course started decomposing industry 4.0 enabling technologies into its constituents finding for commonalities and showing possible path to transform a traditional enterprise into a modern data-driven organization.

Training 2: DIGIMAN 4.0 Workshop ’Teaching and Learning’.

Position: Visiting Scholar in the Department of Mechanical Engineering at the Technical University of Denmark (DTU).

Period: From 25th April 2022 to 06th May 2022.

About Workshop: The training course was oriented to provide a range of innovative teaching and learning methodologies that can be applied in various educational settings. The course explores how different learning theories can inform instructional design, and how technology can be used to enhance student engagement and active learning. The course aimed at teaching about various pedagogical approaches, such as project-based learning, inquiry-based learning, and game-based learning, and gain practical experience in designing and implementing such approaches.

7. APPENDIX

290

7 Appendix

7.1 Impact factor of publications

- **Journal of Big Data:** The JCR journal impact factor of Journal of Big Data in the year 2021 was 10.835. It is in the position 6 of 110 (Q1) of the category "Computer Science, Theory and Methods".
- **IEEE Transactions on Industrial Informatics:** The JCR journal impact factor of IEEE Transactions on Industrial Informatics in the year 2021 was 11.648. It is in the position 3 of 65 (Q1) of the category "Automation and Control Systems".
- **Sensors:** The JCR journal impact factor of Sensors in the year 2021 was 3.847. It is in the position 16 of 76 (Q2) of the category "Instruments and Instrumentation".
- **Applied Sciences:** The JCR journal impact factor of Sensors in the year 2021 was 2.838. It is in the position 39 of 92 (Q2) of the category "Engineering, Multidisplinary".

7.2 Co-authorship justification

This subsection describes the main contributions of the author in each of the publications of this thesis compendium:

IEEE Transactions on Industrial Informatics and Sensors:

- Study of the state of the art: Perform a comprehensive review of the existing literature, including the latest research and advancements in the field.
- Data collection and curation: Identify relevant data sources, collects and preprocesses the data, and applies appropriate quality control measures to ensure the accuracy and reliability of the data.
- Conceptualization of the deep learning architectural design and development of the model: Design a deep learning architecture tailored to the specific

problem at hand, which includes defining the network structure, selecting the appropriate activation functions, loss functions, and optimization algorithms.

- Development of methodology: Propose a methodology that outlines the steps to be followed to apply the deep learning model to the data and obtain meaningful results.
- Validation of results: Evaluate the performance of the deep learning model, compares it to existing benchmarks, and reports the results using appropriate statistical measures.
- Formal Analysis: Perform a rigorous analysis of the results obtained, identifies the limitations and challenges of the proposed approach, and discusses the implications of the findings.
- Development of the article's framework and focus: Outline the structure of the article and defines its main focus, including the research questions, hypotheses, and objectives.
- Drafting, reviewing, editing, and analysis of the original manuscript: Draft the manuscript, incorporates feedback from reviewers and collaborators, and ensures the clarity and coherence of the text, making sure it conforms to the standards of the relevant publication.

Journal of Big Data:

- Study of the state of the art: Conduct an extensive and thorough review of the existing literature to identify the current state of the field of GAN.
- Conceptualization: Contribute to conceptualizing the research questions and objectives, as well as developing a framework for analyzing and interpreting the data.
- Methodology: Contribute to the development of methodology by identifying and describing the data sources and collection methods, as well as the analytical tools and techniques used to analyze the data.
- Collecting resources: Contribute by collecting and organizing the relevant resources, including published literature, data sets, and other sources of information.
- Validation: Validate by ensuring the accuracy and validity of the data and analysis, and by addressing potential biases and limitations.
- Formal analysis: Perform the formal analysis by applying appropriate statistical and analytical techniques to the data.
- Investigation: Contribute to the investigation by exploring and interpreting the findings, and by identifying areas for future research and development.
- Drafting, reviewing, editing, and analysis of the original manuscript: Contribute by drafting and revising the manuscript, incorporating feedback

from peer reviewers and editors, and ensuring that the final product is of high quality and meets the standards of the field.

Applied Sciences:

- Conceptualization: Contribute to conceptualizing the research questions and objectives, as well as developing a framework for analyzing and interpreting the data. Identify the scope of the survey and the research gaps, and develop a clear research question and objectives.
- Original article draft preparation: Contribute to the writing process by preparing the first draft of the article. Organize the article with an introduction, literature review, methodology, results, and discussion sections, and ensure that the draft is clear, concise, and effectively communicates the survey findings.
- Review and editing: Contribute to the writing process by reviewing and editing the draft manuscript. Revise and improve the clarity, structure, and flow of the manuscript, ensure consistency in formatting and citation style, and proofread for errors in grammar, spelling, and punctuation. Incorporate feedback from peer reviewers and editors to strengthen the manuscript and ensure that it is of high quality and meets the standards of the target journal or publication.

Bibliography

[1] Qinglin Qi and Fei Tao. “Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison”. In: *Ieee Access* 6 (2018), pp. 3585–3593 (cit. on p. 2).

[2] Ethirajan Manavalan and Kandasamy Jayakrishna. “A review of Internet of Things (IoT) embedded sustainable supply chain for industry 4.0 requirements”. In: *Computers & Industrial Engineering* 127 (2019), pp. 925–953 (cit. on p. 2).

[3] Jay Lee, Hung-An Kao, and Shanhu Yang. “Service innovation and smart analytics for industry 4.0 and big data environment”. In: *Procedia cirp* 16 (2014), pp. 3–8 (cit. on p. 2).

[4] Marina Paolanti, Luca Romeo, Andrea Felicetti, et al. “Machine learning approach for predictive maintenance in industry 4.0”. In: *2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*. IEEE. 2018, pp. 1–6 (cit. on p. 2).

[5] Tomasz Żabiński, Tomasz Mączka, Jacek Kluska, et al. “Failures prediction in the cold forging process using machine learning methods”. In: *Artificial Intelligence and Soft Computing: 13th International Conference, ICAISC 2014, Zakopane, Poland, June 1-5, 2014, Proceedings, Part I 13*. Springer. 2014, pp. 622–633 (cit. on p. 2).

[6] Mostafa Moussa and Hoda ElMaraghy. “Master assembly network for alternative assembly sequences”. In: *Journal of Manufacturing Systems* 51 (2019), pp. 17–28 (cit. on p. 4).

[7] Hoda ElMaraghy, Mostafa Moussa, Waguih ElMaraghy, and Mohamed Abbas. “Integrated product/system design and planning for new product family in a changeable learning factory”. In: *Procedia Manufacturing* 9 (2017), pp. 65–72 (cit. on p. 4).

[8] Andreas Schumacher, Selim Erol, and Wilfried Sihn. “A maturity model for assessing Industry 4.0 readiness and maturity of manufacturing enterprises”. In: *Procedia Cirp* 52 (2016), pp. 161–166 (cit. on p. 4).

[9] Grand View Research. *Industrial Fasteners Market Size, Share Trends Analysis Report*. 2022. URL: https://www.grandviewresearch.com/industry-analysis/industrial-fasteners-market (visited on Sept. 30, 2022) (cit. on p. 4).

[10] Hoejin Kim, Yirong Lin, and Tzu-Liang Bill Tseng. “A review on quality control in additive manufacturing”. In: *Rapid Prototyping Journal* (2018) (cit. on p. 4).

[11] F Gonzalez and R Pous. “Quality control in manufacturing process by near infrared spectroscopy”. In: *Journal of pharmaceutical and biomedical analysis* 13.4-5 (1995), pp. 419–423 (cit. on p. 4).

[12] Walter Andrew Shewhart. “Economic quality control of manufactured product 1”. In: *Bell System Technical Journal* 9.2 (1930), pp. 364–389 (cit. on p. 5).

[13] Senyong Chen, Yi Qin, Chee-Mun Choy, and JG Chen. “Testing an injection forging process for the production of automotive fasteners”. In: *Procedia engineering* 207 (2017), pp. 508–513 (cit. on p. 6).

[14] Shao-Yi Hsia and Po-Yueh Shih. “Wear improvement of tools in the cold forging process for long hex flange nuts”. In: *Materials* 8.10 (2015), pp. 6640–6657 (cit. on p. 6).

[15] Jeoung Han Kim, Chae Hoon Lee, Jae Keun Hong, Jae Ho Kim, and Jong Taek Yeom. “Effect of surface treatment on the hot forming of the high strength Ti-6Al-4V fastener”. In: *Materials transactions* 50.8 (2009), pp. 2050–2056 (cit. on pp. 7, 8).

[16] Roman Chumakov. “Optimal control of screwing speed in assembly with thread-forming screws”. In: *The International Journal of Advanced Manufacturing Technology* 36.3 (2008), pp. 395–400 (cit. on p. 7).

[17] Ali Abolmaali, Jason Treadway, Pranesh Aswath, Frank K Lu, and Emily McCarthy. “Hysteresis behavior of t-stub connections with superelastic shape memory fasteners”. In: *Journal of Constructional Steel Research* 62.8 (2006), pp. 831–838 (cit. on p. 7).

[18] Pablo Martinez, Mohamed Al-Hussein, and Rafiq Ahmad. “Intelligent vision-based online inspection system of screw-fastening operations in light-gauge steel frame manufacturing”. In: *The International Journal of Advanced Manufacturing Technology* 109.3 (2020), pp. 645–657 (cit. on p. 8).

[19] GV Pachurin, SM Shevchenko, MV Mukhina, LI Kutepova, and JV Smirnova. “The factor of structure and mechanical properties in the production of critical fixing hardware 38XA”. In: *Tribology in Industry* 38.3 (2016), p. 385 (cit. on p. 8).

[20] AA Filippov, GV Pachurin, VI Naumov, and NA Kuz’min. “Low-cost treatment of rolled products used to make long high-strength bolts”. In: *Metallurgist* 59.9 (2016), pp. 810–817 (cit. on p. 8).

[21] Shashi S Pathak, Michael D Blanton, Sharathkumar K Mendon, and James W Rawlins. “Investigation on dual corrosion performance of magnesium-rich primer for aluminum alloys under salt spray test (ASTM B117) and natural exposure”. In: *Corrosion Science* 52.4 (2010), pp. 1453–1463 (cit. on p. 9).

[22] Piotr Łukasz Kłosek. “Controlling variability in assembly production of aircraft structures”. PhD thesis. Instytut Organizacji Systemów Produkcyjnych, 2022 (cit. on p. 9).

[23] Radu Godina and João CO Matias. “Quality control in the context of industry 4.0”. In: *International Joint conference on Industrial Engineering and Operations Management*. Springer. 2018, pp. 177–187 (cit. on p. 9).

[24] Sang M Lee, DonHee Lee, and Youn Sung Kim. “The quality management ecosystem for predictive maintenance in the Industry 4.0 era”. In: *International Journal of Quality Innovation* 5.1 (2019), pp. 1–11 (cit. on p. 9).

[25] Foivos Psarommatis, Gökan May, Paul-Arthur Dreyfus, and Dimitris Kiritsis. “Zero defect manufacturing: state-of-the-art review, shortcomings and future directions in research”. In: *International journal of production research* 58.1 (2020), pp. 1–17 (cit. on p. 9).

[26] João DIAS. “Normas NP, ISO e EN, Relacionadas com o Desenho Técnico”. In: *Lisboa: IST-Departamento de Engenharia Mecânica* (2000) (cit. on p. 9).

[27] BSEN ISO and BRITISH STANDARD. “Mechanical properties of fasteners made of carbon steel and alloy steel”. In: (2009) (cit. on p. 9).

[28] KP Shah. “Fundamentals of threaded fasteners”. In: *Retrieved April* 6 (2019), p. 2021 (cit. on p. 10).

[29] Sandeep Kumar Dwivedi, Manish Vishwakarma, and Akhilesh Soni. “Advances and researches on non destructive testing: A review”. In: *Materials Today: Proceedings* 5.2 (2018), pp. 3690–3698 (cit. on pp. 10, 11).

[30] Joshua Pelleg. *Mechanical properties of materials*. Vol. 190. Springer, 2013 (cit. on p. 11).

[31] Sam Zhang, Lin Li, and Ashok Kumar. *Materials characterization techniques*. CRC press, 2008 (cit. on p. 11).

[32] Bengisu Yilmaz, Aadhik Asokkumar, Elena Jasiūnienė, and Rymantas Jonas Kažys. “Air-coupled, contact, and immersion ultrasonic non-destructive testing: Comparison for bonding quality evaluation”. In: *Applied Sciences* 10.19 (2020), p. 6757 (cit. on p. 12).

[33] Joko Siswantoro. “Application of color and size measurement in food products inspection”. In: *Indonesian Journal of Information Systems (IJIS)* 1.2 (2019), pp. 90–107 (cit. on pp. 12, 13).

[34] Tito Endramawan, Emin Haris, Felix Dionisius, and Yuliana Prinka. “Aplikasi Non Destructive Test Penetrant Testing (Ndt-Pt) Untuk Analisis Hasil Pengelasan Smaw 3g Butt Joint”. In: *JTT (Jurnal Teknologi Terapan)* 3.2 (2017) (cit. on pp. 12, 14).

[35] Javier Garcıa-Martın, Jaime Gómez-Gil, and Ernesto Vázquez-Sánchez. “Non-destructive techniques based on eddy current testing”. In: *Sensors* 11.3 (2011), pp. 2525–2565 (cit. on pp. 13, 14).

[36] Stuart Wilkinson, Steven M Duke, et al. *Comparative testing of radiographic testing, ultrasonic testing and phased array advanced ultrasonic testing non destructive testing techniques in accordance with the AWS D1. 5 bridge welding code.* Tech. rep. Florida. Dept. of Transportation, 2014 (cit. on pp. 14, 15).

[37] Suwei Li, Kezhong Shi, Kun Yang, and Jianzhong Xu. “Research on the defect types judgment in wind turbine blades using ultrasonic NDT”. In: *IOP Conference Series: Materials Science and Engineering*. Vol. 87. 1. IOP Publishing. 2015, p. 012056 (cit. on pp. 15, 16).

[38] Samira Gholizadeh, Z Leman, and BT Hang Tuah Baharudin. “A review of the application of acoustic emission technique in engineering”. In: *Structural Engineering and Mechanics* 54.6 (2015), pp. 1075–1095 (cit. on p. 16).

[39] MJ Lovejoy. *Magnetic particle inspection: a practical guide*. Springer Science & Business Media, 1993 (cit. on p. 16).

[40] Abolfazl Zolfaghari, Amin Zolfaghari, and Farhad Kolahan. “Reliability and sensitivity of magnetic particle nondestructive testing in detecting the surface cracks of welded components”. In: *Nondestructive Testing and Evaluation* 33.3 (2018), pp. 290–300 (cit. on p. 16).

[41] Hongwei Hao, Luming Li, and Yuanhui Deng. "Vision system using linear CCD cameras in fluorescent magnetic particle inspection of axles of railway wheelsets". In: *Health Monitoring and Smart Nondestructive Evaluation of Structural and Biological Systems IV*. Vol. 5768. SPIE. 2005, pp. 442–449 (cit. on p. 17).

[42] Jiwen Lu, Gang Wang, and Pierre Moulin. "Image set classification using holistic multiple order statistics features and localized multi-kernel metric learning". In: *Proceedings of the IEEE international conference on computer vision*. 2013, pp. 329–336 (cit. on p. 18).

[43] Moe Win, AR Bushroa, MA Hassan, NM Hilman, and Ari Ide-Ektessabi. "A contrast adjustment thresholding method for surface defect detection based on mesoscopy". In: *IEEE Transactions on Industrial Informatics* 11.3 (2015), pp. 642–649 (cit. on p. 18).

[44] Xuewu Zhang, Wei Li, Ji Xi, Zhuo Zhang, and Xinnan Fan. "Surface defect target identification on copper strip based on adaptive genetic algorithm and feature saliency". In: *Mathematical Problems in Engineering* 2013 (2013) (cit. on p. 18).

[45] Maoxiang Chu, Rongfen Gong, Song Gao, and Jie Zhao. "Steel surface defects recognition based on multi-type statistical features and enhanced twin support vector machine". In: *Chemometrics and Intelligent Laboratory Systems* 171 (2017), pp. 140–150 (cit. on p. 18).

[46] Huijun Hu, Ya Liu, Maofu Liu, and Liqiang Nie. "Surface defect classification in large-scale strip steel image collection via hybrid chromosome genetic algorithm". In: *Neurocomputing* 181 (2016), pp. 86–95 (cit. on p. 19).

[47] Ma Ricci, Aa Ficola, MLa Fravolini, et al. "Magnetic imaging and machine vision NDT for the on-line inspection of stainless steel strips". In: *Measurement Science and Technology* 24.2 (2012), p. 025401 (cit. on p. 19).

[48] Yong Jie Zhao, Yun Hui Yan, and Ke Chen Song. "Vision-based automatic detection of steel surface defects in the cold rolling process: considering the influence of industrial liquids and surface textures". In: *The International Journal of Advanced Manufacturing Technology* 90.5 (2017), pp. 1665–1678 (cit. on p. 19).

[49] Doo-chul Choi, Yong-Ju Jeon, Seung Hun Kim, et al. "Detection of pinholes in steel slabs using Gabor filter combination and morphological features". In: *Isij International* 57.6 (2017), pp. 1045–1053 (cit. on p. 19).

[50] Şaban Öztürk and Bayram Akdemir. “Real-time product quality control system using optimized Gabor filter bank”. In: *The International Journal of Advanced Manufacturing Technology* 96.1 (2018), pp. 11–19 (cit. on p. 19).

[51] Doo-Chul Choi, Yong-Ju Jeon, Sang Jun Lee, Jong Pil Yun, and Sang Woo Kim. “Algorithm for detecting seam cracks in steel plates using a Gabor filter combination method”. In: *Applied optics* 53.22 (2014), pp. 4865–4872 (cit. on p. 19).

[52] Xiu-yong Wu, Ke Xu, and Jin-wu Xu. “Application of undecimated wavelet transform to surface defect detection of hot rolled steel plates”. In: *2008 Congress on Image and Signal Processing*. Vol. 4. IEEE. 2008, pp. 528–532 (cit. on p. 19).

[53] Xiaoming Liu, Ke Xu, Peng Zhou, Dongdong Zhou, and Yujie Zhou. “Surface defect identification of aluminium strips with non-subsampled shearlet transform”. In: *Optics and Lasers in Engineering* 127 (2020), p. 105986 (cit. on p. 19).

[54] Bayram Akdemir and S Öztürk. “Glass surface defects detection with wavelet transforms”. In: *International Journal of Materials, Mechanics and Manufacturing* 3.3 (2015), pp. 170–173 (cit. on p. 19).

[55] Seyyed Hadi Seifi, Wenmeng Tian, Haley Doude, Mark A Tschopp, and Linkan Bian. “Layer-wise modeling and anomaly detection for laser-based additive manufacturing”. In: *Journal of Manufacturing Science and Engineering* 141.8 (2019), p. 081013 (cit. on p. 19).

[56] Jinjiang Wang, Peilun Fu, and Robert X Gao. “Machine vision intelligence for product defect inspection based on deep learning and Hough transform”. In: *Journal of Manufacturing Systems* 51 (2019), pp. 52–60 (cit. on p. 20).

[57] Stephen Marsland. *Machine learning: an algorithmic perspective*. Chapman and Hall/CRC, 2011 (cit. on p. 20).

[58] Moritz Hardt, Eric Price, and Nati Srebro. “Equality of opportunity in supervised learning”. In: *Advances in neural information processing systems* 29 (2016) (cit. on pp. 20, 40).

[59] Horace B Barlow. “Unsupervised learning”. In: *Neural computation* 1.3 (1989), pp. 295–311 (cit. on pp. 20, 40).

[60] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. “Reinforcement learning: A survey”. In: *Journal of artificial intelligence research* 4 (1996), pp. 237–285 (cit. on pp. 21, 40).

[61] Harith Al-Sahaf, Ying Bi, Qi Chen, et al. “A survey on evolutionary machine learning”. In: *Journal of the Royal Society of New Zealand* 49.2 (2019), pp. 205–228 (cit. on p. 21).

[62] Burr Settles. “Active learning literature survey”. In: (2009) (cit. on p. 21).

[63] Yonglong Tian, Chen Sun, Ben Poole, et al. “What makes for good views for contrastive learning?” In: *Advances in neural information processing systems* 33 (2020), pp. 6827–6839 (cit. on pp. 21, 22).

[64] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. “Curriculum learning”. In: *Proceedings of the 26th annual international conference on machine learning*. 2009, pp. 41–48 (cit. on pp. 21, 22).

[65] Sebastian Ruder. “An overview of multi-task learning in deep neural networks”. In: *arXiv preprint arXiv:1706.05098* (2017) (cit. on pp. 21, 22, 37, 39).

[66] Ricardo Vilalta and Youssef Drissi. “A perspective view and survey of meta-learning”. In: *Artificial intelligence review* 18 (2002), pp. 77–95 (cit. on pp. 21, 23).

[67] Oded Maron and Tomás Lozano-Pérez. “A framework for multiple-instance learning”. In: *Advances in neural information processing systems* 10 (1997) (cit. on pp. 21, 24).

[68] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. “Generalizing from a few examples: A survey on few-shot learning”. In: *ACM computing surveys (csur)* 53.3 (2020), pp. 1–34 (cit. on pp. 21, 24).

[69] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. “A survey of transfer learning”. In: *Journal of Big data* 3.1 (2016), pp. 1–40 (cit. on pp. 21, 25).

[70] Yue Wu, Yinpeng Chen, Lijuan Wang, et al. “Large scale incremental learning”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 374–382 (cit. on pp. 21, 25).

[71] Daniel Lowd and Christopher Meek. “Adversarial learning”. In: *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. 2005, pp. 641–647 (cit. on pp. 21, 26).

[72] Chelsea Finn, Pieter Abbeel, and Sergey Levine. “Model-agnostic meta-learning for fast adaptation of deep networks”. In: *International conference on machine learning*. PMLR. 2017, pp. 1126–1135 (cit. on p. 23).

[73] Yoonho Lee and Seungjin Choi. “Gradient-based meta-learning with learned layerwise metric and subspace”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 2927–2936 (cit. on p. 23).

[74] Jiaxin Chen, Li-Ming Zhan, Xiao-Ming Wu, and Fu-lai Chung. “Variational metric scaling for metric-based meta-learning”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 04. 2020, pp. 3478–3485 (cit. on p. 23).

[75] Zhi-Hua Zhou. “Multi-instance learning: A survey”. In: *Department of Computer Science & Technology, Nanjing University, Tech. Rep* 1 (2004) (cit. on p. 24).

[76] Flood Sung, Yongxin Yang, Li Zhang, et al. “Learning to compare: Relation network for few-shot learning”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 1199–1208 (cit. on p. 24).

[77] Sachin Ravi and Hugo Larochelle. “Optimization as a model for few-shot learning”. In: *International conference on learning representations*. 2017 (cit. on p. 24).

[78] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. “Understanding of a convolutional neural network”. In: *2017 international conference on engineering and technology (ICET)*. Ieee. 2017, pp. 1–6 (cit. on p. 26).

[79] Keiron O’Shea and Ryan Nash. “An introduction to convolutional neural networks”. In: *arXiv preprint arXiv:1511.08458* (2015) (cit. on p. 27).

[80] Naila Murray and Florent Perronnin. “Generalized max pooling”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 2473–2480 (cit. on p. 27).

[81] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. “Learning deep features for discriminative localization”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2921–2929 (cit. on p. 27).

[82] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324 (cit. on pp. 28, 41).

[83] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Communications of the ACM* 60.6 (2017), pp. 84–90 (cit. on p. 28).

[84] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014) (cit. on p. 28).

[85] Christian Szegedy, Wei Liu, Yangqing Jia, et al. “Going deeper with convolutions”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9 (cit. on p. 28).

[86] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778 (cit. on pp. 28, 77).

[87] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. “Rethinking the inception architecture for computer vision”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2818–2826 (cit. on p. 28).

[88] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. “Densely connected convolutional networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708 (cit. on p. 28).

[89] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587 (cit. on pp. 28, 41).

[90] Ross Girshick. “Fast r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1440–1448 (cit. on p. 28).

[91] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. “Faster r-cnn: Towards real-time object detection with region proposal networks”. In: *Advances in neural information processing systems* 28 (2015) (cit. on p. 28).

[92] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. “Mask r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2961–2969 (cit. on p. 28).

[93] Wei Liu, Dragomir Anguelov, Dumitru Erhan, et al. “Ssd: Single shot multibox detector”. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer. 2016, pp. 21–37 (cit. on pp. 28, 89).

[94] Mohammad Javad Shafiee, Brendan Chywl, Francis Li, and Alexander Wong. “Fast YOLO: A fast you only look once system for real-time embedded object detection in video”. In: *arXiv preprint arXiv:1709.05943* (2017) (cit. on p. 28).

[95] Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully convolutional networks for semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440 (cit. on pp. 28, 41).

[96] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. “Segnet: A deep convolutional encoder-decoder architecture for image segmentation”. In: *IEEE transactions on pattern analysis and machine intelligence* 39.12 (2017), pp. 2481–2495 (cit. on p. 28).

[97] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer. 2015, pp. 234–241 (cit. on p. 28).

[98] Foivos I Diakogiannis, François Waldner, Peter Caccetta, and Chen Wu. “ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 162 (2020), pp. 94–114 (cit. on p. 28).

[99] Abdul Mueed Hafiz and Ghulam Mohiuddin Bhat. “A survey on instance segmentation: state of the art”. In: *International journal of multimedia information retrieval* 9.3 (2020), pp. 171–189 (cit. on p. 28).

[100] Yanming Guo, Yu Liu, Theodoros Georgiou, and Michael S Lew. “A review of semantic segmentation using deep neural networks”. In: *International journal of multimedia information retrieval* 7 (2018), pp. 87–93 (cit. on pp. 28, 29).

[101] Yu He, Kechen Song, Hongwen Dong, and Yunhui Yan. “Semi-supervised defect classification of steel surface based on multi-training and generative adversarial network”. In: *Optics and Lasers in Engineering* 122 (2019), pp. 294–302 (cit. on p. 29).

[102] Vidhya Natarajan, Tzu-Yi Hung, Sriram Vaikundam, and Liang-Tien Chia. “Convolutional networks for voting-based anomaly classification in metal surface inspection”. In: *2017 IEEE International Conference on Industrial Technology (ICIT)*. IEEE. 2017, pp. 986–991 (cit. on p. 29).

[103] Jonathan Masci, Ueli Meier, Gabriel Fricout, and Jürgen Schmidhuber. “Multi-scale pyramidal pooling network for generic steel defect classification”. In: *The 2013 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2013, pp. 1–8 (cit. on p. 29).

[104] Di He, Ke Xu, and Peng Zhou. “Defect detection of hot rolled steels with a new object detection framework called classification priority network”. In: *Computers & Industrial Engineering* 128 (2019), pp. 290–297 (cit. on p. 29).

[105] Yu He, Kechen Song, Qinggang Meng, and Yunhui Yan. “An end-to-end steel surface defect detection approach via fusing multiple hierarchical features”. In: *IEEE Transactions on Instrumentation and Measurement* 69.4 (2019), pp. 1493–1504 (cit. on p. 29).

[106] Xupeng Kou, Shuaijun Liu, Kaiqiang Cheng, and Ye Qian. “Development of a YOLO-V3-based model for detecting defects on steel strip surface”. In: *Measurement* 182 (2021), p. 109454 (cit. on p. 29).

[107] Ruoxu Ren, Terence Hung, and Kay Chen Tan. “A generic deep-learning-based approach for automated surface inspection”. In: *IEEE transactions on cybernetics* 48.3 (2017), pp. 929–940 (cit. on p. 29).

[108] Hua Yang, Yifan Chen, Kaiyou Song, and Zhouping Yin. “Multiscale feature-clustering-based fully convolutional autoencoder for fast accurate visual inspection of texture surface defects”. In: *IEEE Transactions on Automation Science and Engineering* 16.3 (2019), pp. 1450–1467 (cit. on p. 29).

[109] Robby Neven and Toon Goedemé. “A multi-branch U-Net for steel surface defect type and severity segmentation”. In: *Metals* 11.6 (2021), p. 870 (cit. on p. 29).

[110] Xiaofei Zhou, Hao Fang, Xiaobo Fei, Ran Shi, and Jiyong Zhang. “Edge-Aware Multi-Level Interactive Network for Salient Object Detection of Strip Steel Surface Defects”. In: *IEEE Access* 9 (2021), pp. 149465–149476 (cit. on p. 29).

[111] Guorong Song, Kechen Song, and Yunhui Yan. “EDRNet: Encoder–decoder residual network for salient object detection of strip steel surface defects”. In: *IEEE Transactions on Instrumentation and Measurement* 69.12 (2020), pp. 9709–9719 (cit. on p. 29).

[112] Hongwen Dong, Kechen Song, Yu He, et al. “PGA-Net: Pyramid feature fusion and global context attention network for automated surface defect detection”. In: *IEEE Transactions on Industrial Informatics* 16.12 (2019), pp. 7448–7458 (cit. on p. 29).

[113] Xiaoxue Ren, Zhenchang Xing, Xin Xia, et al. “Neural network-based detection of self-admitted technical debt: From performance to explainability”. In: *ACM transactions on software engineering and methodology (TOSEM)* 28.3 (2019), pp. 1–45 (cit. on pp. 36, 37, 40).

[114] Guizhong Fu, Peize Sun, Wenbin Zhu, et al. “A deep-learning-based approach for fast and robust steel surface defects classification”. In: *Optics and Lasers in Engineering* 121 (2019), pp. 397–405 (cit. on pp. 36, 40).

[115] Donggyun Im, Sangkyu Lee, Homin Lee, et al. “A data-centric approach to design and analysis of a surface-inspection system based on deep learning in the plastic injection molding industry”. In: *Processes* 9.11 (2021), p. 1895 (cit. on p. 36).

[116] Lidan Shang, Qiushi Yang, Jianing Wang, Shubin Li, and Weimin Lei. “Detection of rail surface defects based on CNN image recognition and classification”. In: *2018 20th International Conference on Advanced Communication Technology (ICACT)*. IEEE. 2018, pp. 45–51 (cit. on pp. 36, 72).

[117] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, et al. “Grad-cam: Visual explanations from deep networks via gradient-based localization”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 618–626 (cit. on pp. 37, 40, 97, 264).

[118] Joffrey L Leevy, Taghi M Khoshgoftaar, Richard A Bauder, and Naeem Seliya. “A survey on addressing high-class imbalance in big data”. In: *Journal of Big Data* 5.1 (2018), pp. 1–30 (cit. on pp. 38, 41).

[119] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, et al. “Generative adversarial networks”. In: *Communications of the ACM* 63.11 (2020), pp. 139–144 (cit. on pp. 38, 41, 52).

[120] Nour Eldeen Khalifa, Mohamed Loey, and Seyedali Mirjalili. “A comprehensive survey of recent trends in deep learning for digital images augmentation”. In: *Artificial Intelligence Review* (2022), pp. 1–27 (cit. on p. 39).

[121] Shuanlong Niu, Bin Li, Xinggang Wang, and Yaru Peng. “Region-and strength-controllable GAN for defect generation and segmentation in industrial images”. In: *IEEE Transactions on Industrial Informatics* 18.7 (2021), pp. 4531–4541 (cit. on p. 39).

[122] Benyi Yang, Zhenyu Liu, Guifang Duan, and Jianrong Tan. “Mask2Defect: A Prior Knowledge-Based Data Augmentation Method for Metal Surface Defect Inspection”. In: *IEEE Transactions on Industrial Informatics* 18.10 (2021), pp. 6743–6755 (cit. on p. 39).

[123] Gongjie Zhang, Kaiwen Cui, Tzu-Yi Hung, and Shijian Lu. “Defect-GAN: High-fidelity defect synthesis for automated defect inspection”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2021, pp. 2524–2534 (cit. on p. 39).

[124] Heliang Zheng, Jianlong Fu, Tao Mei, and Jiebo Luo. “Learning multi-attention convolutional neural network for fine-grained image recognition”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 5209–5217 (cit. on p. 39).

[125] Xiaojin Jerry Zhu. “Semi-supervised learning literature survey”. In: (2005) (cit. on p. 40).

[126] Connor Shorten and Taghi M Khoshgoftaar. “A survey on image data augmentation for deep learning”. In: *Journal of big data* 6.1 (2019), pp. 1–48 (cit. on pp. 41, 43, 45).

[127] Connor Shorten, Taghi M Khoshgoftaar, and Borko Furht. “Text data augmentation for deep learning”. In: *Journal of big Data* 8 (2021), pp. 1–34 (cit. on p. 41).

[128] Qingsong Wen, Liang Sun, Fan Yang, et al. “Time series data augmentation for deep learning: A survey”. In: *arXiv preprint arXiv:2002.12478* (2020) (cit. on p. 41).

[129] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. “Autoaugment: Learning augmentation strategies from data”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 113–123 (cit. on pp. 41, 43).

[130] Francisco J Moreno-Barea, Fiammetta Strazzera, José M Jerez, Daniel Urda, and Leonardo Franco. “Forward noise adjustment scheme for data augmentation”. In: *2018 IEEE symposium series on computational intelligence (SSCI)*. IEEE. 2018, pp. 728–734 (cit. on pp. 41, 45).

[131] Terrance DeVries and Graham W Taylor. “Dataset augmentation in feature space”. In: *arXiv preprint arXiv:1702.05538* (2017) (cit. on p. 42).

[132] Agnieszka Mikołajczyk and Michał Grochowski. “Data augmentation for improving deep learning in image classification problem”. In: *2018 international interdisciplinary PhD workshop (IIPhDW)*. IEEE. 2018, pp. 117–122 (cit. on p. 42).

[133] Ren Wu, Shengen Yan, Yi Shan, Qingqing Dang, and Gang Sun. “Deep image: Scaling up image recognition”. In: *arXiv preprint arXiv:1501.02876* 7.8 (2015), p. 4 (cit. on p. 47).

[134] Humza Naveed. “Survey: Image mixing and deleting for data augmentation”. In: *arXiv preprint arXiv:2106.07085* (2021) (cit. on p. 47).

[135] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. “Random erasing data augmentation”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 07. 2020, pp. 13001–13008 (cit. on p. 47).

[136] Luke Taylor and Geoff Nitschke. “Improving deep learning with generic data augmentation”. In: *2018 IEEE symposium series on computational intelligence (SSCI)*. IEEE. 2018, pp. 1542–1547 (cit. on p. 47).

[137] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. “Image-to-image translation with conditional adversarial networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1125–1134 (cit. on pp. 56, 65).

[138] Alec Radford, Luke Metz, and Soumith Chintala. “Unsupervised representation learning with deep convolutional generative adversarial networks”. In: *arXiv preprint arXiv:1511.06434* (2015) (cit. on p. 56).

[139] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. “Progressive growing of gans for improved quality, stability, and variation”. In: *arXiv preprint arXiv:1710.10196* (2017) (cit. on p. 56).

[140] Emily L Denton, Soumith Chintala, Rob Fergus, et al. “Deep generative image models using a laplacian pyramid of adversarial networks”. In: *Advances in neural information processing systems* 28 (2015) (cit. on p. 56).

[141] Daniel Jiwoong Im, Chris Dongjoo Kim, Hui Jiang, and Roland Memisevic. “Generating images with recurrent adversarial networks”. In: *arXiv preprint arXiv:1602.05110* (2016) (cit. on p. 56).

[142] Tu Nguyen, Trung Le, Hung Vu, and Dinh Phung. “Dual discriminator generative adversarial nets”. In: *Advances in neural information processing systems* 30 (2017) (cit. on p. 57).

[143] Milind Shah, Vinay Vakharia, Rakesh Chaudhari, et al. “Tool wear prediction in face milling of stainless steel using singular generative adversarial network and LSTM deep learning models”. In: *The International Journal of Advanced Manufacturing Technology* 121.1-2 (2022), pp. 723–736 (cit. on p. 57).

[144] Arnab Ghosh, Viveka Kulharia, Vinay P Namboodiri, Philip HS Torr, and Puneet K Dokania. “Multi-agent diverse generative adversarial networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8513–8521 (cit. on p. 57).

[145] Mehdi Mirza and Simon Osindero. “Conditional generative adversarial nets”. In: *arXiv preprint arXiv:1411.1784* (2014) (cit. on p. 57).

[146] Augustus Odena, Christopher Olah, and Jonathon Shlens. “Conditional image synthesis with auxiliary classifier gans”. In: *International conference on machine learning*. PMLR. 2017, pp. 2642–2651 (cit. on p. 57).

[147] Shabab Bazrafkan and Peter Corcoran. “Versatile auxiliary classifier with generative adversarial network (vac+ gan), multi class scenarios”. In: *arXiv preprint arXiv:1806.07751* (2018) (cit. on p. 57).

[148] Xi Chen, Yan Duan, Rein Houthooft, et al. “Infogan: Interpretable representation learning by information maximizing generative adversarial nets”. In: *Advances in neural information processing systems* 29 (2016) (cit. on p. 57).

[149] Xiaoqiang Li, Liangbo Chen, Lu Wang, Pin Wu, and Weiqin Tong. “Scgan: Disentangled representation learning by adding similarity constraint on generative adversarial nets”. In: *IEEE Access* 7 (2018), pp. 147928–147938 (cit. on p. 57).

[150] Xiaolong Wang and Abhinav Gupta. “Generative image modeling using style and structure adversarial networks”. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*. Springer. 2016, pp. 318–335 (cit. on p. 59).

[151] Han Zhang, Tao Xu, Hongsheng Li, et al. “Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 5907–5915 (cit. on p. 60).

[152] Tero Karras, Samuli Laine, and Timo Aila. “A style-based generator architecture for generative adversarial networks”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 4401–4410 (cit. on p. 61).

[153] Zifeng Wu, Chunhua Shen, and Anton Van Den Hengel. “Wider or deeper: Revisiting the resnet model for visual recognition”. In: *Pattern Recognition* 90 (2019), pp. 119–133 (cit. on pp. 63, 65).

[154] Mingxing Tan and Quoc Le. “Efficientnet: Rethinking model scaling for convolutional neural networks”. In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114 (cit. on pp. 63, 65).

[155] Olga Russakovsky, Jia Deng, Hao Su, et al. “Imagenet large scale visual recognition challenge”. In: *International journal of computer vision* 115 (2015), pp. 211–252 (cit. on p. 64).

[156] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. "Unpaired image-to-image translation using cycle-consistent adversarial networks". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2223–2232 (cit. on p. 65).

[157] Pádraig Cunningham, Matthieu Cord, and Sarah Jane Delany. "Supervised learning". In: *Machine learning techniques for multimedia: case studies on organization and retrieval* (2008), pp. 21–49 (cit. on p. 70).

[158] Yoav Freund, Robert E Schapire, et al. "Experiments with a new boosting algorithm". In: *icml*. Vol. 96. Citeseer. 1996, pp. 148–156 (cit. on p. 70).

[159] Leonard E Baum and Ted Petrie. "Statistical inference for probabilistic functions of finite state Markov chains". In: *The annals of mathematical statistics* 37.6 (1966), pp. 1554–1563 (cit. on p. 70).

[160] Geoffrey E Hinton. "Deep belief networks". In: *Scholarpedia* 4.5 (2009), p. 5947 (cit. on p. 70).

[161] Joseph Cohen, Baoyang Jiang, and Jun Ni. "Machine learning for diagnosis of event synchronization faults in discrete manufacturing systems". In: *Journal of Manufacturing Science and Engineering* 144.7 (2022) (cit. on p. 71).

[162] Abdul Mujeeb, Wenting Dai, Marius Erdt, and Alexei Sourin. "One class based feature learning approach for defect detection using deep autoencoders". In: *Advanced Engineering Informatics* 42 (2019), p. 100933 (cit. on p. 71).

[163] NA Kasim, MZ Nuawi, JA Ghani, et al. "Enhancing Clustering Algorithm with Initial Centroids in Tool Wear Region Recognition". In: *International Journal of Precision Engineering and Manufacturing* 22 (2021), pp. 843–863 (cit. on p. 71).

[164] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018 (cit. on p. 72).

[165] Xuefeng Ni, Ziji Ma, Jianwei Liu, Bo Shi, and Hongli Liu. "Attention network for rail surface defect detection via consistency of intersection-over-union (IoU)-guided center-point estimation". In: *IEEE Transactions on Industrial Informatics* 18.3 (2021), pp. 1694–1705 (cit. on p. 72).

[166] Qirui Ren, Jiahui Geng, and Jiangyun Li. "Slighter Faster R-CNN for real-time detection of steel strip surface defects". In: *2018 Chinese Automation Congress (CAC)*. IEEE. 2018, pp. 2173–2178 (cit. on p. 73).

[167] Laith Alzubaidi, Jinglan Zhang, Amjad J Humaidi, et al. "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions". In: *Journal of big Data* 8 (2021), pp. 1–74 (cit. on p. 73).

[168] Xinyi Le, Junhui Mei, Haodong Zhang, Boyu Zhou, and Juntong Xi. "A learning-based approach for surface defect detection using small image datasets". In: *Neurocomputing* 408 (2020), pp. 112–120 (cit. on p. 73).

[169] Selim Seferbekov, Vladimir Iglovikov, Alexander Buslaev, and Alexey Shvets. "Feature pyramid network for multi-class land segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2018, pp. 272–275 (cit. on pp. 76, 77).

[170] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. "Focal loss for dense object detection". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988 (cit. on pp. 83, 89).

[171] Kaggle Severstal. "Steel Defect Detection". In: *Can You Detect and Classify Defects in Steel* (2019) (cit. on p. 84).

[172] Zhaowei Cai and Nuno Vasconcelos. "Cascade r-cnn: Delving into high quality object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 6154–6162 (cit. on p. 89).

[173] Babak Hassibi and David Stork. "Second order derivatives for network pruning: Optimal brain surgeon". In: *Advances in neural information processing systems* 5 (1992) (cit. on p. 94).

[174] Aojun Zhou, Anbang Yao, Yiwen Guo, Lin Xu, and Yurong Chen. "Incremental network quantization: Towards lossless cnns with low-precision weights". In: *arXiv preprint arXiv:1702.03044* (2017) (cit. on pp. 94, 95).

[175] Cheng Tai, Tong Xiao, Yi Zhang, Xiaogang Wang, et al. "Convolutional neural networks with low-rank regularization". In: *arXiv preprint arXiv:1511.06067* (2015) (cit. on pp. 94, 96).

[176] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network". In: *arXiv preprint arXiv:1503.02531* (2015) (cit. on pp. 94, 96).

[177] Jiayi Liu, Samarth Tripathi, Unmesh Kurup, and Mohak Shah. "Pruning algorithms to accelerate convolutional neural networks for edge applications: A survey". In: *arXiv preprint arXiv:2005.04275* (2020) (cit. on p. 94).

[178] Ruichi Yu, Ang Li, Chun-Fu Chen, et al. "Nisp: Pruning networks using neuron importance score propagation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 9194–9203 (cit. on p. 94).

[179] Shaohui Lin, Rongrong Ji, Chenqian Yan, et al. “Towards optimal structured cnn pruning via generative adversarial learning”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 2790–2799 (cit. on p. 94).

[180] Pius Kwao Gadosey, Yujian Li, and Peter T Yamak. “On pruned, quantized and compact CNN architectures for vision applications: an empirical study”. In: *Proceedings of the International Conference on Artificial Intelligence, Information Processing and Cloud Computing*. 2019, pp. 1–8 (cit. on p. 95).

[181] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, et al. “Pact: Parameterized clipping activation for quantized neural networks”. In: *arXiv preprint arXiv:1805.06085* (2018) (cit. on p. 95).

[182] Haibao Yu, Qi Han, Jianbo Li, et al. “Search what you want: Barrier panelty nas for mixed precision quantization”. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*. Springer. 2020, pp. 1–16 (cit. on p. 95).

[183] Daniel Povey, Gaofeng Cheng, Yiming Wang, et al. “Semi-orthogonal low-rank matrix factorization for deep neural networks.” In: *Interspeech*. 2018, pp. 3743–3747 (cit. on p. 96).

[184] Yong Chen, Wei He, Naoto Yokoya, and Ting-Zhu Huang. “Hyperspectral image restoration using weighted group sparsity-regularized low-rank tensor decomposition”. In: *IEEE transactions on cybernetics* 50.8 (2019), pp. 3556–3570 (cit. on p. 96).

[185] Jungchan Cho and Minsik Lee. “Building a compact convolutional neural network for embedded intelligent sensor systems using group sparsity and knowledge distillation”. In: *Sensors* 19.19 (2019), p. 4307 (cit. on p. 97).

[186] From Wikipedia, the free encyclopedia. *Tensor Processing Unit*. [Online; accessed 29-September-2012]. 2010 (cit. on p. 261).

[187] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. “Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks”. In: *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE. 2018, pp. 839–847 (cit. on p. 264).

[188] Haofan Wang, Zifan Wang, Mengnan Du, et al. “Score-CAM: Score-weighted visual explanations for convolutional neural networks”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 2020, pp. 24–25 (cit. on p. 265).

[189] Linda Guiga and AW Roscoe. “Neural Network Security: Hiding CNN Parameters with Guided Grad-CAM.” In: *ICISSP*. 2020, pp. 611–618 (cit. on p. 266).

This doctoral thesis aims to improve the efficiency and accuracy of defect detection in ferromagnetic parts by developing an automated method based on deep learning. Specifically, the thesis focuses on fasteners, a type of ferromagnetic part widely used in industries such as automotive, aerospace, and machinery. The production of fasteners involves several processes, including casting, forging, and machining, which can introduce defects in the surface or near-surface of the parts. The detection and identification of these defects are critical to ensuring the integrity and reliability of the fasteners in their intended applications. The proposed system analyzes raw images of fasteners to detect and locate defects by being trained on labeled images of fasteners with and without defects.

To address the challenges of limited data, imbalanced classes, and overfitting, the proposed methodology employs a data-centric approach using data augmentation and GAN-based synthetic images to expand the size and diversity of the training dataset. Additionally, the use of multi-task learning improves the performance of the defect detection model by allowing it to learn from multiple sources of information and to generalize better to new tasks. Finally, explainable AI techniques are employed to make the defect detection model more interpretable and explainable, with GradCAM generating the most interpretable and explainable heatmaps. The proposed methodology represents a significant advancement in the field of fastener inspection and quality control, with potential applications in various industries. The automated defect detection system developed at Tekniker has been successfully tested at Erreka Fastening solutions, demonstrating its potential to improve the efficiency and accuracy of defect detection in the manufacturing process of ferromagnetic parts.

Universidad
Zaragoza
1542