Universidad Zaragoza
1542

Master Final Project

EFFICIENT ANOMALY DETECTION IN CCTV VIDEOS

DETECCIÓN EFICIENTE DE ANOMALÍAS EN VIDEOS CCTV

Author

Sonia Rubio Llamas

Director

Eduardo Montijano Muñoz

ESCUELA DE INGENIERÍA Y ARQUITECTURA
2024/2025

Acknowledgments

I would like to thank all the people who have supported me and have been with me actively or passively throughout this year. To the professors of the master's program for passing on their knowledge and helping us to think outside the box. And above all, to the friends I take with me, from Zaragoza and abroad. Without their help and support I would still be in the first semester. There have been tears but also laughts.

Thanks to Eduardo Montijano for guiding me through all the project, and to Ana Cristina Murillo and Sara Casao for accompanying me and helping me while I was doing the internship, as well as to the people of the Robotics group for welcoming me in their meetings. And to Kiko and Oscar from Buavi for giving me the opportunity to work with them and continue in the world of computer science and artificial intelligence.

Special thanks to my parents and my sister for always believing in me. To Ana and Mario, for being my great and unconditional supporters. And last but not least, to my nephew Dani, who came into my life a year and a half ago to brighten it up.

Resumen

En los últimos años, el crecimiento exponencial de los sistemas de videovigilancia CCTV ha evidenciado la necesidad de métodos automáticos de detección de anomalías para garantizar la seguridad en espacios públicos y privados. Esta tesis busca mejorar la precisión y eficiencia de dichos métodos, haciéndolos prácticos para escenarios reales. El desequilibrio entre eventos normales y anómalos, la diversidad de contextos y las altas demandas computacionales de los modelos de aprendizaje profundo son retos clave, especialmente en entornos con recursos computacionales limitados, como CPUs.

Hemos evaluado tres algoritmos de detección de anomalías débilmente supervisados — PEL4VAD, UR-DMU y BN-WVAD— utilizando la base de datos UCF-Crime. Estos métodos, que solo necesitan etiquetas a nivel de video durante el entrenamiento, fueron seleccionados por sus buenos resultados. Sin embargo, nuestros experimentos han revelado que la métrica estándar ROC AUC utilizada puede no reflejar adecuadamente su buen funcionamiento en conjuntos desbalanceados. Hemos propuesto en su lugar el uso de la métrica Precision-Recall AUC (PR AUC), que equilibra mejor la precisión y la sensibilidad.

Para mejorar la eficiencia computacional, hemos implementado técnicas de reducción de la tasa de muestreo y optimizado la extracción de características con I3D, reduciendo tiempos de procesamiento sin comprometer la precisión. Además, introdujimos estrategias como el uso de contexto y funciones de pérdida balanceadas, como la entropía cruzada binaria balanceada y la pérdida focal Tversky.

Estas mejoras han permitido obtener métricas más equilibradas e interpretables. Los resultados demuestran la viabilidad de implementar estos modelos en hardware con recursos limitados, haciéndolos adecuados para aplicaciones de vigilancia en tiempo real.

Abstract

In recent years, the exponential rise in CCTV video surveillance systems has highlighted the need for automated anomaly detection to ensure safety in public and private spaces. This thesis focuses on enhancing the efficiency and accuracy of video anomaly detection frameworks, making them suitable for real-world scenarios. Challenges such as the imbalance between normal and anomalous events, diverse contexts, and the computational demands of deep learning models are particularly critical in resource-constrained environments.

We evaluated three state-of-the-art weakly supervised anomaly detection algorithms — PEL4VAD, UR-DMU, and BN-WVAD — on the UCF-Crime dataset, which contains real-world surveillance videos. These methods, relying on video-level labels during training, reduce the need for detailed annotations. Our experiments identified limitations in the standard ROC AUC metric, which can misrepresent performance in imbalanced datasets. To address this, we proposed Precision-Recall AUC (PR AUC) as a more reliable alternative, balancing precision and recall.

To improve computational efficiency, we implemented downsampling techniques and optimized feature extraction with the I3D model, significantly reducing processing times without compromising accuracy. Additionally, we introduced strategies such as contextual integration and balanced loss functions like Weighted Binary Cross-Entropy and Focal Tversky Loss.

These enhancements produced more balanced and interpretable metrics. The results validate the feasibility of deploying these models on resource-limited hardware, such as CPUs, making them viable for real-time applications. This work marks a significant advance toward practical, scalable, and efficient video anomaly detection systems.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

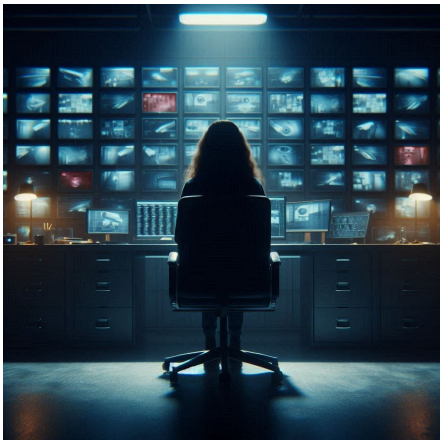
1.1 Motivation and Context

In recent years, video surveillance systems have become an essential tool for ensuring safety in public spaces, generating vast amounts of data daily, exceeding human capabilities to monitor effectively. Consequently, there is an urgent need for automated systems capable of detecting anomalies in real time, reducing the strain on human operators, and improving response times to critical situations. As illustrated in Figure 1.1, the increase of cameras and the immense amount of video footage emphasize the importance of the task.

Anomaly detection in video surveillance involves identifying patterns or behaviors that deviate from what is considered normal in a given context. Examples include violent incidents, unauthorized access, unattended objects, or suspicious behaviors. To address this challenge, deep learning methods have emerged as a powerful solution, providing reliable and accurate detection. However, the practical deployment of these systems in real-world scenarios presents several high-level challenges that need to be addressed to make such solutions feasible and effective.

One of the main challenges lies in the limited availability of examples of anomalies. Because of their nature, anomalous events represent only a small fraction of the total video data. The vast majority of recorded footage corresponds to normal behaviors, making it difficult for models to learn and generalize patterns that represent anomalies. This imbalance in the data poses a significant challenge for achieving reliable performance.

(a) Source: `https://shorturl.at/9k9QE`



(b) Source: Image created with the assistance of DALL·E 2

Figure 1.1: Representation of the amount of cameras and video CCTV footage.

Another critical challenge is the diversity of real-world contexts where these systems are implemented. The same behavior can be normal in one context but anomalous in another. For example, the presence of a car is typical on a highway but would be unusual in a park. Developing a system that can generalize well across diverse environments and contexts remains a significant issue.

Additionally, achieving computational efficiency is a major challenge, particularly for real-time processing. State-of-the-art deep learning methods, while highly accurate, are often computationally expensive. Surveillance systems deployed in practical settings frequently operate under limited hardware resources, where access to high-performance servers or GPUs cannot be guaranteed. In such scenarios, finding a balance between accuracy and efficiency becomes a top priority to ensure the feasibility of real-time detection.

This project aims to improve both the computational efficiency and the detection quality of current algorithms developed to detect anomalies in CCTV footage. By leveraging state-of-the-art techniques in artificial intelligence and machine learning, the goal is to build solutions that are not only accurate but also cost-effective and suitable for real-world deployment.

This work has been carried out in collaboration with the Robotics, Computer Vision and Artificial Intelligence Group at the University of Zaragoza and the Zaragoza-based

company Buavi, under the framework of the project "Reconocimiento de comportamientos peligrosos mediante inteligencia artificial" from *Programa de Apoyo a los Digital Innovation Hubs (PADIH)*. The project aims to address practical challenges such as detecting workplace accidents or elderly falls in nursing homes in an efficient, fast and low-computational way. This collaboration has served as a longer-term collaboration to apply these solutions to real-world scenarios.

1.2 Goals

The primary objective of this thesis is to develop an efficient and reliable framework for video anomaly detection tailored for CCTV footage, leveraging current advances in artificial intelligence and computer vision. The specific goals are as follows:

1. **Evaluation of existing state-of-the-art frameworks:** Perform a detailed review and local deployment of state-of-the-art anomaly detection algorithms. This includes evaluating their performance and limitations within the context of CCTV footage to identify potential areas for improvement.
2. **Enhancement of computational efficiency:** Optimize the computational efficiency of selected algorithms to enable real-time anomaly detection on resource-constrained hardware, such as CPUs, without significantly compromising accuracy.
3. **Improvement of Detection Accuracy:** Explore novel strategies, such as leveraging contextual information and balanced loss functions, to improve metrics like Area Under the ROC Curve (ROC AUC), Area Under the Precision-Recall Curve (PR AUC) and False Alarm Rate (FAR).

1.3 Scope

To achieve the defined goals, we have carried out several tasks and addressed multiple challenges:

– **Review of scientific literature:** Numerous scientific articles describing state-

of-the-art algorithms for anomaly detection have been studied. Complementary materials were also reviewed to deepen the understanding of fundamental concepts and methods used in these algorithms.

- **Deployment and configuration:** The project required the setup and configuration of three anomaly detection algorithms (PEL4VAD, UR-DMU, and BN-WVAD) on local hardware. This process included preparing datasets, running experiments under different settings, and ensuring the reproducibility of results to analyze their performance.

- **Algorithm modification and experimentation:** Improvements to the selected algorithms were implemented to enhance performance. These modifications included preprocessing techniques, such as adjustments to the I3D feature extractor and the inclusion of contextual information, as well as changes in the training phase (e.g., adjustments to the loss function). Additionally, optimizations for execution on different hardware (CPU/GPU) were explored, along with alternative evaluation metrics like PR AUC and FAR to better assess the results.

- **Workship collaboration:** Regular meetings were held with Buavi, a Zaragoza-based company, to share intermediate results and discuss how the developed methods could address practical challenges in applications such as workplace safety and elderly care monitoring.

- **Active participation in a research group:** Participation with the Robotics Group at the University of Zaragoza was a key aspect of this project. Weekly meetings provided opportunities to present results, discuss scientific challenges, and receive valuable feedback in a collaborative environment.

1.4 Planning

The work presented in this thesis was carried out in several stages, ensuring a structured and comprehensive approach to achieving the defined goals. Figure 1.2 illustrates the Gannt chart of the plannning.

1. **Literature Review:** An in-depth review of existing video anomaly detection methods, focusing on their advantages, limitations, and applicability to CCTV data.

2. **Dataset Preparation:** Selection and preprocessing of the UCF-Crime dataset, including feature extraction using Inflated 3D ConvNets (I3D) and downsampling to improve computational efficiency.

3. **Algorithm Development:** Implementation of weakly supervised methods, such as PEL4VAD, with modifications to enhance their computational and detection performance.

4. **Experimental Evaluation:** Extensive testing on the UCF-Crime dataset to compare the proposed methods against existing baselines, using metrics such as ROC AUC, PR AUC and FAR.

5. **Optimization and Fine-Tuning:** Iterative improvements to the algorithms based on experimental results, including adjustments to the loss functions and hyperparameters.

6. **Documentation and Presentation:** Compilation of results and insights into a coherent thesis, highlighting contributions to the field and directions for future work.

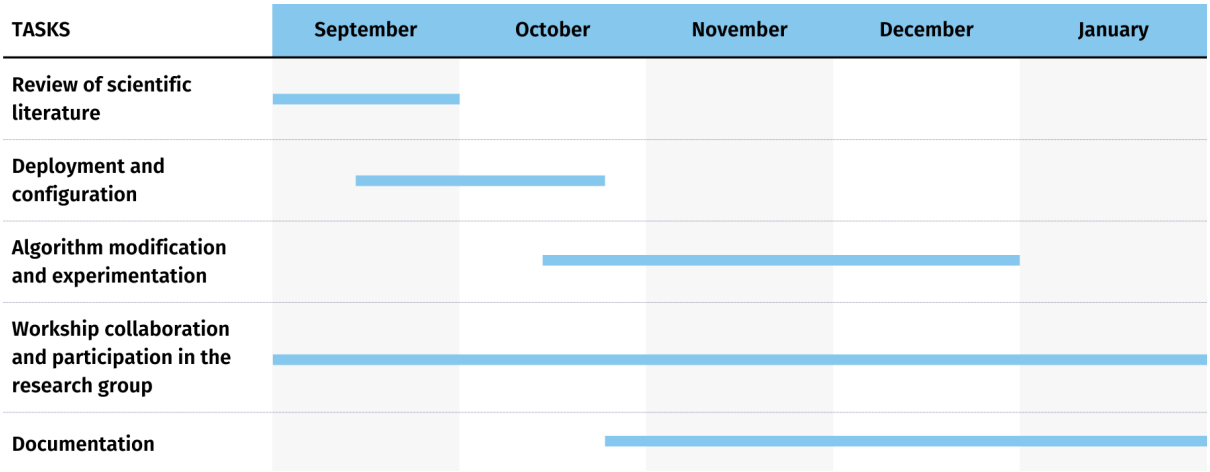


Figure 1.2: Gantt Chart for the Project Plan

1.5 Organization of the Thesis

This thesis is organized into seven main chapters, covering everything from the introduction and theoretical background to the results and final conclusions. The first chapter introduces the context of the project, the goals we have set, its scope, and the planning we have followed.

Chapter 2 reviews the state of the art in anomaly detection for CCTV videos. It explains the most relevant methods and approaches, the datasets commonly used, and the key challenges in this field.

Chapter 3 focuses on the algorithms we analyzed in this project. It provides a detailed description of the selected models—PEL4VAD, UR-DMU, and BN-WVAD—and their local performance based on experimental evaluations.

Chapter 4 discusses the strategies used to improve computational efficiency. This includes frame rate evaluation, and comparisons of performance under different hardware configurations.

Chapter 5 explores the limitations of traditional evaluation metrics in highly imbalanced datasets. It introduces a a more appropriate alternative and evaluates its impact through experiments and case studies. Additionally, a window-based evaluation method is proposed to better align with real-world anomaly detection scenarios.

Chapter 6 looks into improvements in detection quality. Here, we haved analyzed the metrics used, proposed changes to the loss functions, and explored the impact of adding contextual information on the accuracy of the models.

Finally, Chapter 7 presents a summary of the main contributions of this work and proposes future research directions.

Chapter 2

State of the Art in Video Anomaly

Video surveillance systems are increasingly used in public spaces such as marketplaces, transportation hubs, and other crowded areas to ensure safety and security. These systems generate an enormous amount of video data every day, far beyond the capacity of human operators to monitor effectively. As a result, automated systems for detecting anomalies, such as unusual or suspicious activities in real-time have become essential.

Video anomalies refer to patterns or behaviors that deviate from what is considered normal, based on the specific context. These deviations can manifest as anomalous activities, such as fights, riots, traffic violations, or stampedes, or as anomalous objects, like weapons in restricted areas or unattended luggage in public spaces. Detecting these anomalies in an automated way is crucial for improving the efficiency and effectiveness of video surveillance systems.

Timely and accurate detection of anomalies is vital for security systems to enable quick and appropriate responses.

2.1 Methods and Approaches

In the context of video anomaly detection, the goal is not only to define what constitutes an anomaly but also to create systems capable of recognizing these events automatically. Machine learning and deep learning approaches have emerged as effective tools for this task. These methods use large datasets to learn patterns of normal behavior, allowing

them to identify deviations as potential anomalies. Deep learning techniques, in particular, leverage Convolutional Neural Networks (CNNs) to extract spatial features and other architectures to analyze temporal dynamics, ensuring a comprehensive understanding of video data over time.

In video anomaly detection, it is challenging to develop a universal solution that works across all datasets and scenarios. Instead, most approaches rely on tailored methods that address the specific characteristics and requirements of individual datasets or applications.

Zaheer et al. [1] classified existing approaches for video anomaly detection (VAD) into four categories, based on the level of human intervention and the amount of prior knowledge (i.e., labeled data) required during the training process. (Fig. 2.1)

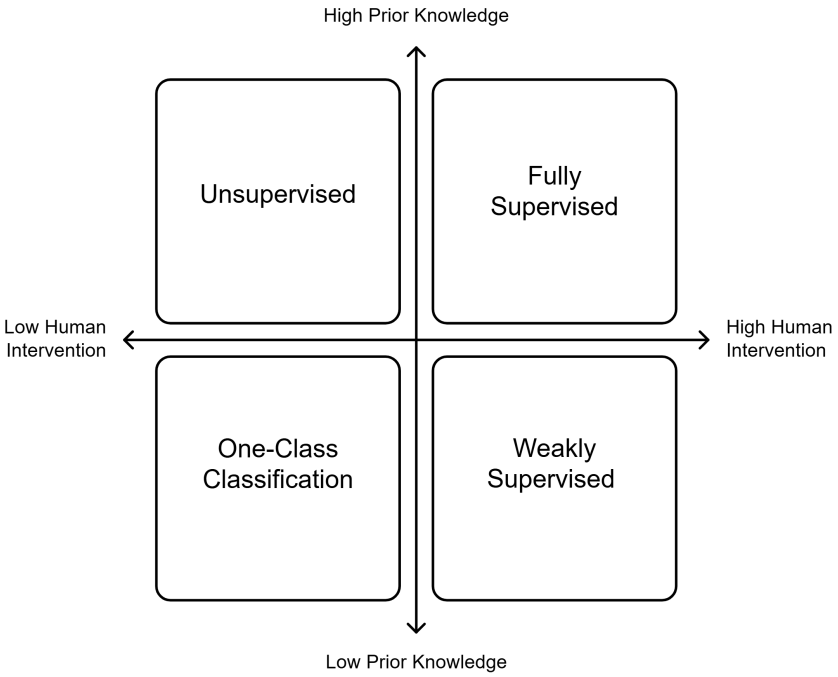


Figure 2.1: Categories of video anomaly detection methods based on supervision: unsupervised, fully supervised, one-class classification, and weakly supervised.

1. **Fully supervised approaches:** These methods rely on detailed annotations for both normal and abnormal behaviors, typically at the frame level, across the entire training dataset.

2. **One-class classification:** In this approach, the system is trained only on data labeled as normal, without the need for abnormal examples.

3. **Weakly supervised approaches:** These methods require video-level annotations indicating whether a video is normal or anomalous, but without specific frame-level details.

4. **Unsupervised methods:** These do not require any labeled data, and the system learns to identify anomalies solely based on patterns in the data itself.

Fully supervised methods are often impractical for VAD due to the difficulty in obtaining balanced datasets with a similar number of both normal and abnormal samples. Labeling each frame is time-consuming, and abnormal events are rare, making it hard to gather comprehensive training data. One-class classifiers address this by focusing only on normal data, using models like autoencoders [2] and GANs [3] to detect deviations. However, these models may misclassify anomalous instances or unseen normal events, as they cannot capture the full range of normal variations.

Weakly supervised methods reduce the need for detailed labeling by using video-level annotations, but this comes with less precision in identifying specific anomalous frames. Finally, unsupervised methods, which require no labels, offer scalability but can struggle to distinguish between rare normal events and true anomalies due to the absence of supervision.

Advances in unsupervised and weakly supervised learning, which do not require extensive labeled data, have shown promise in addressing these challenges. However, achieving robust and reliable anomaly detection in real-world settings remains an open research problem.

2.2 Datasets

To develop an effective prediction algorithm, selecting the right dataset for training is crucial. In this case, the ideal dataset would consist of numerous normal and abnormal video recordings from CCTV cameras. Three of the most widely used datasets in this field are UCF-Crime [4], XD-Violence [5], and ShanghaiTech [6].

2.2.1 XD-Violence

P. Wu et al. [5] introduced XD-Violence, the largest multimodal dataset for detecting violent events. It consists of 4,754 untrimmed videos, with a total duration of 217 hours. The dataset includes both visual and audio signals, as well as weak video-level labels. It contains 2,405 violent videos and 2,349 non-violent videos. These videos come from diverse sources, including surveillance footage, movies, car cameras, and video games. The dataset covers six types of violence: abuse, car accidents, explosions, fighting, riots, and shootings. A notable challenge in this dataset is the presence of artistic effects, such as camera movements and scene changes, which complicate the anomaly detection process.

2.2.2 ShanghaiTech

The ShanghaiTech dataset, proposed by Zhong et al. [6], features 13 scenes with varied lighting conditions and camera angles. It contains over 270,000 frames and includes 130 abnormal events. A key feature of this dataset is the pixel-level ground truth annotations for the abnormal events, which allows for precise evaluation of detection methods. For weakly supervised learning, the dataset is reorganized into 238 training videos and 199 test videos.

2.2.3 UCF-Crime Dataset

Sultani et al. [4] introduced UCF-Crime, a large-scale dataset designed specifically for anomaly detection in real-world surveillance environments. This dataset contains 128 hours of video, comprising 1,900 long, untrimmed surveillance videos. These videos include 13 types of realistic anomalies: abuse, arrest, arson, burglary, robbery, stealing, shooting, shoplifting, assault, fighting, explosion, road accidents, vandalism and normal videos. For weakly supervised learning, UCF-Crime provides 1,610 videos for training and 290 videos for testing. During training, only video-level annotations are available, meaning each video is labeled as either normal or containing an anomaly, but the exact location of the anomaly within the video is unknown. In contrast, frame-level annotations are provided for testing purposes. Along with the dataset, the authors introduced an anomaly detection method, which will be reviewed in Section 3.1.

The UCF-Crime dataset was selected for its focus on real-world surveillance, offering a broader range of scenarios compared to the more specific ShanghaiTech dataset. With 1,900 untrimmed videos covering various types of anomalies, UCF-Crime provides the diversity needed to develop a robust anomaly detection model.

While XD-Violence offers diversity, its inclusion of audio and artistic effects makes it less suitable for this project, which focuses solely on visual data. UCF-Crime ensures practical relevance while maintaining a manageable scope.

2.3 Metric Evaluation

For evaluating anomaly detection models, the Area Under the Curve (AUC) is the primary metric used in the literature. The ROC AUC is used to assess each model's ability to distinguish between normal and anomalous frames. The ROC curve plots the true positive rate (also known as sensitivity or recall) against the false positive rate (1 - specificity) at various threshold levels. These metrics are mathematically defined in equations 2.1 and 2.2, and a visual representation can be seen in Figure 2.2. A perfect classifier achieves a true positive rate (TPR) of 1 and a false positive rate (FPR) of 0, indicating it can correctly identify all anomalies without any false alarms. The closer the ROC curve approaches this ideal point in the top-left corner of the plot, the better the model's performance. Conversely, if the curve appears as a diagonal line (linear), it indicates the model performs no better than random guessing.

$$\textit{True Positive Rate (TPR)} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \tag{2.1}$$

$$\textit{False Positive Rate (FPR)} = \frac{\text{False Positives (FP)}}{\text{False Positives (FP)} + \text{True Negatives (TN)}} \tag{2.2}$$

The AUC, or area under this curve, quantifies the overall performance. A higher AUC indicates that the model is better at identifying anomalies (high true positive rate) while minimizing false alarms (low false positive rate). In other words, it measures the model's capacity to correctly classify frames as normal or abnormal.

In addition, the False Alarm Rate (FAR) is also used individually, representing the proportion of normal frames mistakenly classified as anomalous. To calculate FAR, a

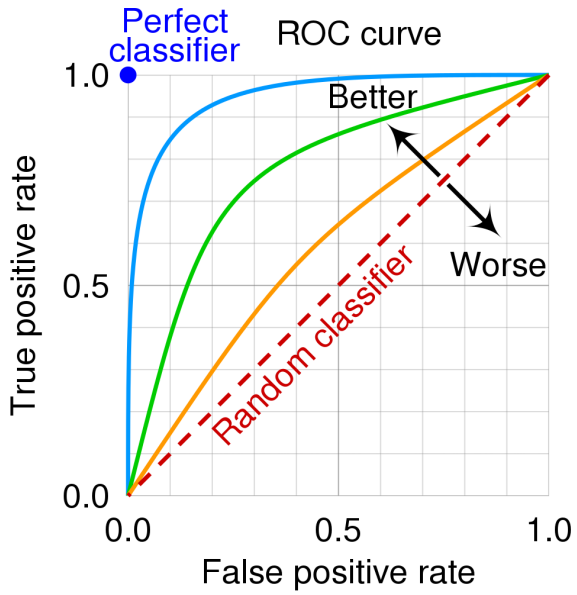


Figure 2.2: ROC AUC. Source: https://shorturl.at/ljUNt

threshold value of 0.5 is typically applied, where any prediction score above this threshold is considered an anomaly. A lower FAR means that the model is better at avoiding false positives, making it more reliable in real-world applications where normal events should not be frequently misclassified as anomalies.

2.4 Key Insights and Challenges

Detecting video anomalies is a complex task influenced by several factors. Different challenges make it difficult to find a general and unique solution that works across all scenarios. Anomalies are highly context-dependent, meaning that an event considered normal in one situation could be anomalous in another. For example, running may be expected in a park but could be suspicious in a crowded subway station. Moreover, anomalies are rare and diverse, creating a significant imbalance between anomalous and normal data. This imbalance complicates the use of supervised learning approaches, since acquiring labeled data for anomalies remains challenging. Additionally, anomalies are often undefined—they can take many forms and are influenced by the specific environment, context, and timing. This is in contrast to tasks like action recognition, which typically involve well-defined categories.

Environmental factors, such as changes in lighting, occlusions, or crowd density, further complicate anomaly detection. Additionally, the settings of the surveillance system, such as camera angles and resolutions, can also impact the detection process. On top of these challenges, the large amount of data and computational effort required to train and deploy neural networks adds another layer of complexity, especially when the goal is to detect anomalies in real time, before they evolve into significant public health issues.

Morover, while analyzing the results and the metrics, we have identified inconsistencies in both the choice of metrics and their application. The imbalance between anomalous and normal data misrepresents the ROC AUC metric.

These challenges highlight the limitations of current methods in video anomaly detection. The diversity of anomalies, dataset imbalance, and environmental variability demand improvements in detection accuracy. At the same time, the computational cost of deploying such systems calls for more efficient solutions. The following chapters propose methods to address these challenges, focusing on improving both the efficiency and accuracy of video anomaly detection systems, offering more practical and robust solutions for real-world applications.

Chapter 3

Overview of Anomaly Detection Algorithms

Numerous authors have contributed to the field of Video Anomaly Detection (VAD). For this project, we have chosen to analyze three of the most advanced methods in the literature. These methods were selected based on their high performance metrics and the availability of open-source code, ensuring both relevance and reproducibility. In addition to Sultani et al.'s baseline approach, called Deep Multiple Instance (MIL), the selected algorithms are PEL4VAD [7], UR-DMU[8], and BN-WVAD [9].

3.1 Deep Multiple Instance (MIL) framework

Alongside introducing the UCF-Crime dataset, Sultani et al. proposed an approach for learning video anomaly detection using a deep multiple instance (MIL) framework based on weakly labeled training videos. In their method, entire videos (classified as either normal or anomalous) are treated as "bags" and individual video segments as "instances" within a Multiple Instance Learning (MIL) framework. C3D features [10] are extracted from these segments to capture spatiotemporal information. The model is trained using a fully connected neural network and a ranking loss function, which computes the score difference between the highest-scoring segment in the positive bag (anomalous video) and segments in the negative bag (normal video). Sparsity and temporal smoothness constraints in the loss function further refine the model's ability to localize anomalies. As

the proposers of the UCF-Crime dataset, their algorithm represents a strong baseline for both the anomaly detection task and the dataset itself.

3.2 PEL4VAD

Among the selected algorithms, PEL4VAD is perhaps the most significant, as it serves as the primary baseline for this project's experiments and contributions.

PEL4VAD (Prompt-Enhanced Learning for Video Anomaly Detection) is a state-of-the-art framework for weakly-supervised video anomaly detection, particularly designed to address the challenge of missing frame-level annotations in surveillance videos. The method effectively combines three key components: the Temporal Context Aggregation (TCA) module, the Prompt-Enhanced Learning (PEL) module, and a Score Smoothing (SS) strategy, each contributing to its high performance in detecting anomalies.

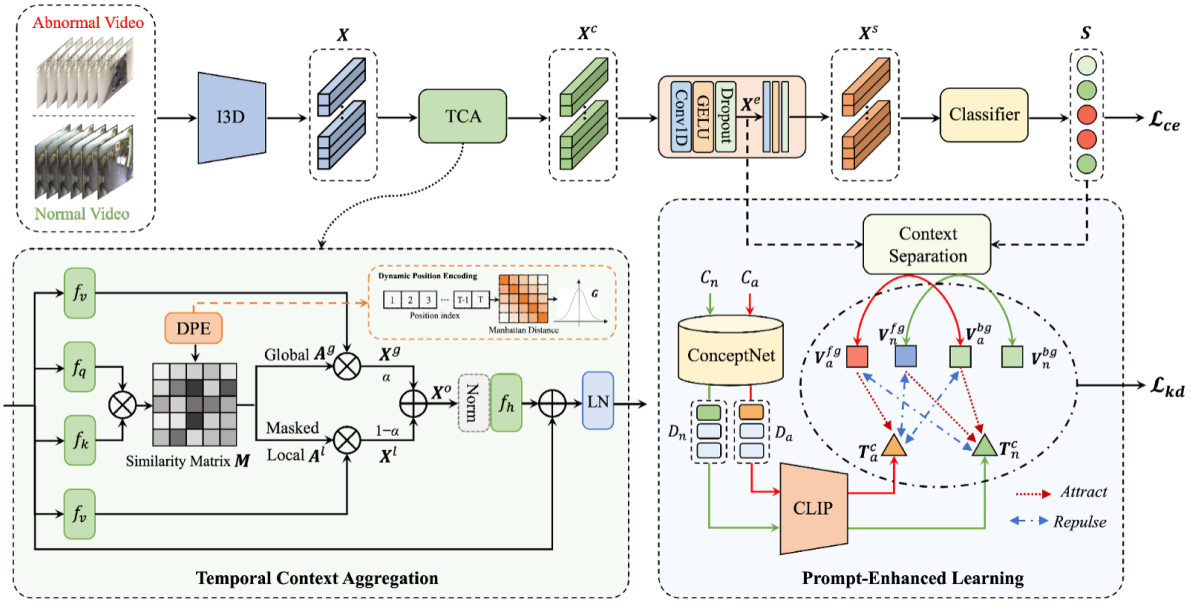


Figure 3.1: Overview of the PEL4VAD framework

The TCA module plays a crucial role in capturing both local and global temporal dependencies within video snippets. Instead of using separate branches for different contexts, it reuses a similarity matrix, applying efficient attention mechanisms to reduce computational complexity while maintaining the ability to model fine-grained temporal

patterns. This design is augmented by a Dynamic Position Encoding (DPE) that encodes the relative positions of video snippets, further enhancing the model's ability to distinguish temporal order.

The PEL module introduces external semantic knowledge into the model by utilizing ConceptNet [11], a large knowledge graph, to create semantic prompts. These prompts, derived from a pre-trained CLIP model [12], guide the model in identifying and differentiating anomalous from normal events. This cross-modal alignment between visual features and semantic prompts is achieved using a Kullback-Leibler (KL) divergence loss, which ensures the consistency of the feature representations with their corresponding semantic knowledge.

To refine the anomaly scores, the Score Smoothing (SS) strategy is implemented. By applying a moving average, it mitigates the impact of transient noise and false positives, ensuring smoother and more reliable predictions. This step is crucial for improving the temporal consistency of anomaly detection, especially in videos with fluctuating or ambiguous anomaly patterns.

PEL4VAD demonstrates strong performance, achieving an ROC AUC of 86.76 on the UCF-Crime dataset. Its ability to model temporal dependencies and integrate semantic knowledge makes it highly effective for weakly-supervised anomaly detection.

3.3 UR-DMU

The main goal of UR-DMU (Uncertainty Regulated Dual Memory Units) is to enhance weakly supervised video anomaly detection by addressing two key challenges: learning robust feature representations for normal and anomalous events, and managing the uncertainty caused by noisy surveillance data, such as camera shifts or scene variations. Additionally, UR-DMU leverages I3D (Inflated 3D ConvNet) as a feature extractor to improve its performance. Its approach focuses on explicitly separating normal and abnormal patterns to reduce false alarms and improve detection reliability. The idea is encapsulated by the framework illustrated in Figure 3.2. UR-DMU achieves this through three core innovations. First, it utilizes a Global and Local Multi-Head Self-Attention (GL-MHSA) module to capture both long-term and short-term temporal dependencies

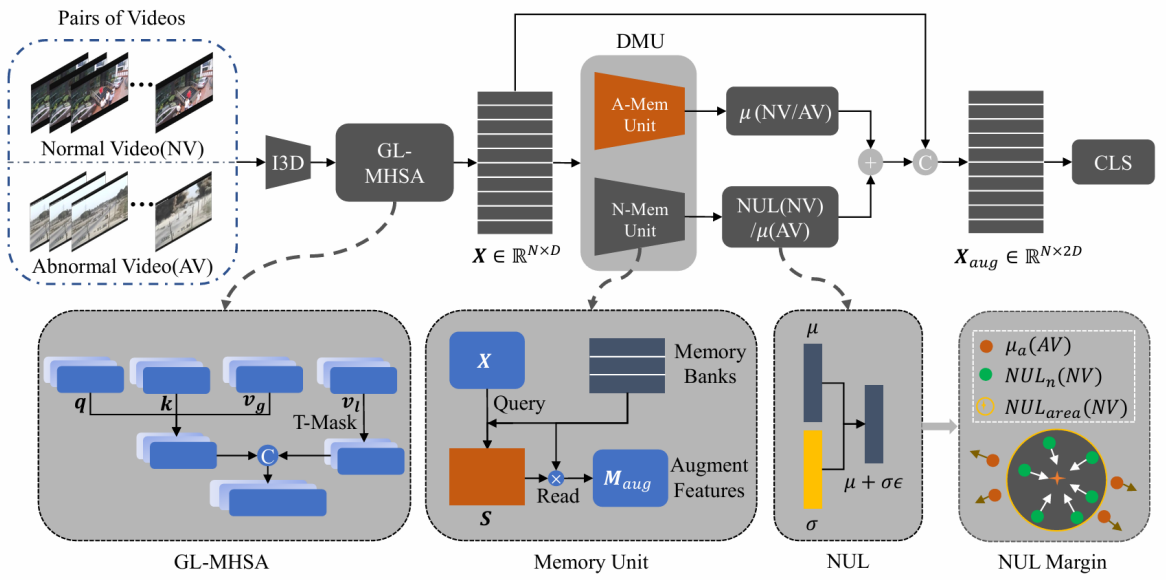


Figure 3.2: Overview of the UR-DMU framework

in videos. This enhances the model's ability to understand the evolving dynamics of anomalies and normal patterns over time.

Second, it introduces Dual Memory Units (DMU), consisting of separate memory banks for normal and anomalous patterns. This explicit separation increases the margin between normal and abnormal feature spaces, enabling the model to handle ambiguous or borderline cases more effectively. A specially designed loss function ensures that each memory bank learns its respective patterns with high precision.

Third, UR-DMU incorporates a Normal Data Uncertainty Learning (NUL) mechanism to address noise in normal data. By constraining normal feature representations to follow a Gaussian distribution, this module increases the model's resilience to small variations in normal patterns while clearly isolating anomalies as out-of-distribution (OOD) events.

On the UCF-Crime dataset, UR-DMU achieves outstanding performance, with a ROC AUC of 86.97% on UCF-Crime, demonstrating its effectiveness in detecting anomalies with high accuracy.

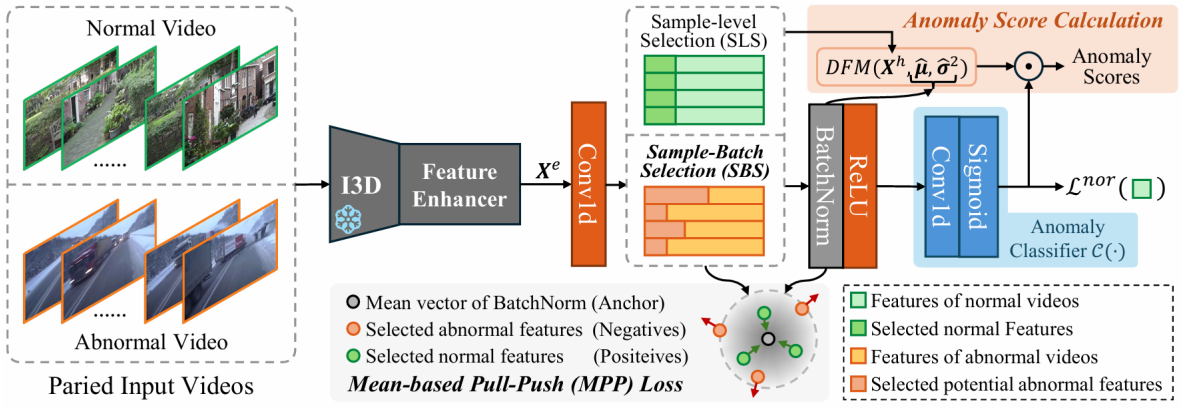


Figure 3.3: Overview of the BN-WVAD framework

3.4 BN-WVAD

The primary objective of BN-WVAD is to enhance the robustness and accuracy of weakly supervised video anomaly detection (WVAD) by leveraging the statistical properties of Batch Normalization (BatchNorm). This method addresses critical challenges in WVAD, such as unreliable abnormality criteria, sensitivity to noisy labels, and limitations in identifying abnormal snippets when the proportion of anomalies varies across videos. The method also incorporates I3D as a feature extractor.

As illustrated in Figure 3.3, the core of BN-WVAD is the implementation of BatchNorm as a statistical instrument to model normal patterns within video data. During training, BatchNorm calculates a mean vector that effectively represents the "normal" feature distribution, as the majority of snippets in most mini-batches are normal. This insight allows the model to treat abnormal snippets as statistical outliers, providing a robust criterion for distinguishing normal from abnormal patterns. Unlike previous methods that rely on simple assumptions, such as feature magnitude, BN-WVAD introduces a more resilient approach that improves anomaly detection accuracy.

To further enhance its effectiveness, BN-WVAD incorporates a Mean-based Pull-Push (MPP) loss that ensures normal features are tightly clustered around the mean vector while maximizing the separation between normal and abnormal features. This loss function strengthens the model's ability to differentiate between normal and anomalous snippets, even in scenarios with high variability or noise.

BN-WVAD also introduces a Sample-Batch Selection (SBS) strategy, which combines sample-level and batch-level selection techniques. This hybrid approach helps identify abnormal snippets more comprehensively, especially in videos with varying levels of anomaly prevalence. By addressing the limitations of purely sample-level or batch-level strategies, SBS ensures that significant anomalous events are not overlooked.

The framework's effectiveness is ALSO demonstrated on the UCF-Crime dataset, achieving a ROC AUC of 87.24%, surpassing previous methods.

3.5 Baseline Evaluation

In this section, we have described the experiments conducted to evaluate the baseline performance of the selected algorithms as reported by their original implementations. Although each method applies particular strategies, the overall framework remains the same: first, features are extracted from the video frames, resulting in a feature vector of size $(X, 1024)$, where X corresponds to the number of extracted features. These features are then fed into a prediction neural network, which outputs a single value for every 16 frames. This output value, which lies in the range [0, 1], represents the probability of that frame to be an anomaly (1) or a normal frame (0).

The first major step in this project has been to replicate the three selected algorithms: PEL4VAD, UR-DMU, and BN-WVAD, to ensure their usability and perform further experiments.

Thanks to the availability of open-source code implementations, we have been able to verify and reproduce the results provided by the authors. The source code can be found on GitHub for each method: PEL4VAD [1], UR-DMU [2], and BN-WVAD [3]. We have made minor adjustments to the folder structure for storing all the data, creating separate versions optimized for GPU and CPU usage, given that the original versions were exclusively designed for GPU.

We have performed the experiments using two computational environments with the

[1] `https://github.com/yujiangpu20/PEL4VAD`
[2] `https://github.com/henrryzh1/UR-DMU`
[3] `https://github.com/cool-xuan/BN-WVAD`

following specifications:

- **PC1:** Intel Core i5-7200U processor and an NVIDIA GeForce 940MX GPU with 4GB of memory. In this setup, we have only used the CPU, referred to as CPU1.
- **PC2:** Intel Core i5-14400F processor and an NVIDIA GeForce RTX 4060 GPU with 8GB of memory. In this case, both the CPU (CPU2) and GPU (GPU2) have been used.

Unless specified otherwise, we have executed all the experiments using the GPU (GPU2) for faster computation.

We have evaluated the time required for training and prediction, as well as the main performance metrics. These results are summarized in Tables 3.1 and 3.2, respectively.

Table 3.1 shows the time required for training and prediction across the different algorithms and environments. Due to hardware limitations, training has only been executed on GPU2. Prediction times have been also measured on both CPU1 and CPU2 for comparison. The prediction times vary depending on the hardware used. As expected, predictions are significantly faster when using GPU2 compared to CPU1 or CPU2.

	Training Time			**Prediction Time**		
	PEL4VAD	**UR-DMU**	**BN-WVAD**	**PEL4VAD**	**UR-DMU**	**BN-WVAD**
CPU1	-	-	-	4'48"	8'19"	6'53"
CPU2	-	-	-	3'10"	2'8"	1'59"
GPU2	3º43'18"	4º27'51"	3º50'45"	0'43"	0'46"	0'59"

Table 3.1: Training and prediction times for each algorithm

As mentioned in Section 2.2.3, the UCF-Crime dataset consists of 290 videos in the test set, which corresponds to a total duration of 10 hours, 19 minutes, and 47 seconds. Given that the dataset is recorded at 30 frames per second (fps), the total number of frames to process is 1,111,131. To measure the prediction time, we considered the full process, including loading the checkpoint (the trained model weights), loading the dataset, and generating predictions. It is important to note that the model, the checkpoint and the dataset are loaded only once at the start.

We have made a deeper analysis for the PEL4VAD method, observing that the time required to load the checkpoint is minimal, at approximately 0.07 seconds. Once the model is initialized, the prediction process itself is extremely fast, with processing times on the order of 10^{-6} seconds per frame. However, the main bottleneck in the pipeline occurs during the dataset processing stage, where it takes around 23.53 seconds to process the entire dataset. This step involves handling the input frames, preparing the data for prediction, and ensuring that the network can efficiently process and output the results.

To evaluate the performance of the algorithms, we have measured three main metrics: ROC AUC (Receiver Operating Characteristic - Area Under the Curve), PR AUC (Precision-Recall Area Under the Curve), and FAR (False Alarm Rate). The results are presented in Table 3.2, where we compare the original results reported by the authors (paper) with our results reproduced locally (ours).

It is important to highlight two things; firslty, none of the papers gives importance or explain the effect of the PR AUC with the UCF-Crime dataset. Secondly, the FAR calculated in the original PEL4VAD test only considers predictions on normal videos. Thus, we have also measured the FAR across the entire test dataset, which includes both normal and abnormal videos. This adjustment makes the FAR metric more representative of the overall performance. We provide a detailed explanation of this decision in Chapter 5. For the UR-DMU algorithm, the authors did not provide the code to compute the FAR; instead, they only reported the result in their paper. As a result, we were unable to reproduce the FAR metric based solely on predictions from normal videos. This cases where the FAR could not be calculated, the values are marked with a dash (-). The same situation applies to the BN-WVAD algorithm. Additionally, the FAR values marked with an asterisk (*) indicate that these results were not provided by the authors. These FAR values were computed using the authors' checkpoint, but with our own FAR calculation method, which includes both normal and abnormal videos in the evaluation.

The metrics for PEL4VAD and UR-DMU show consistent results when compared to the original values reported by the authors. However, we have encountered some issues with the BN-WVAD implementation. First, although the authors report performance metrics for the UCF-Crime dataset, the code was specifically written and adjusted for the XD-Violence dataset. As a result, we have modified the code to ensure compatibility with the UCF-Crime dataset used in this project.

	PEL4VAD		UR-DMU		BN-WVAD	
	Paper	**Ours**	**Paper**	**Ours**	**Paper**	**Ours**
ROC AUC	86.76	86.60	86.97	86.89	87.24	83.85
PR AUC	33.99	34.87	35.58	35.20	36.26	25.79
FAR (normal)	0.43	0.23	1.05	-	-	-
FAR (abnormal)	7.32*	7.31	11.60*	8.42	-	21.41

Table 3.2: Performance evaluation metrics

Second, we observed that the final output values of the prediction network in BN-WVAD were not normalized to the expected range of [0, 1]. Instead, the predictions reached values as high as 24. To address this, we have also modified the code and applied a min-max normalization to scale the output values to the range [0, 1]. It is important to note that this normalization only affects the visualization of the results, as the evaluation metrics remain unaffected.

Chapter 4

Improvement of Computational Efficiency

In the context of video anomaly detection, computational efficiency plays a vital role in ensuring that the systems can operate effectively in real-world scenarios, where hardware resources are often limited. This chapter explores strategies to optimize the performance of detection frameworks by reducing computational demands without sacrificing accuracy. By focusing on efficient feature extraction and evaluating the trade-offs between processing speed and detection performance, we aim to make these methods more practical for large-scale deployment in surveillance systems.

4.1 Feature Extraction with I3D

In deep learning, raw data, especially high-dimensional data like video, must be processed to make it suitable for analysis by a neural network. This preprocessing step, known as feature extraction, involves identifying and encoding essential patterns while reducing redundancy, transforming complex visual content into a numerical representation that facilitates learning. For video data, feature extraction presents unique challenges due to the dual need to capture both spatial information (appearance) and temporal dynamics (motion).

Traditional 2D convolutional networks are effective at extracting spatial features but struggle to represent temporal patterns. As a result, specialized methods like I3D [13]

and C3D [10] have been developed, enabling effective spatiotemporal feature extraction.

This project employs I3D as the feature extraction method, chosen for its ability to model both the appearance and motion patterns in videos. This makes it particularly effective for video anomaly detection tasks, where understanding the interplay between spatial and temporal features is crucial.

The I3D (Two-Stream Inflated 3D ConvNet) is a deep learning model designed for video action recognition. It extends 2D convolutional networks by inflating 2D filters into 3D, enabling the model to capture both spatial and temporal information across video frames. Based on the Inception architecture from GoogleNet [14], I3D uses a two-stream approach: the RGB stream processes appearance features, while the optical flow stream captures motion by analyzing frame-to-frame differences.

An advantage of I3D is its ability to capture temporal patterns effectively, even within short snippets of 64 frames, making it suitable for video analysis tasks. Furthermore, I3D is pretrained on large-scale video datasets, such as Kinetics [15], which helps it learn robust features. This pretraining enables the model to achieve state-of-the-art performance in tasks like action recognition, video classification, and anomaly detection.

4.1.1 I3D for Video Anomaly Detection

In our study, we have employed the I3D network as a feature extractor for all three algorithms under evaluation, as it was originally the methodology used by the original authors. The network provides three different extraction methods: one based solely on RGB data, another based on optical flow, and a third combining both. After performing experiments to evaluate the metrics when incorporating optical flow, we observed only minimal changes in results. Consequently, we have decided to follow the original approach and proceed exclusively with RGB data, aligning with the methodology followed by the authors of the three algorithms.

To save time, we have downloaded[1] the pre-extracted features instead of running the

[1] https://stuxidianeducn-my.sharepoint.com/personal/pengwu_stu_xidian_edu_cn/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fpengwu%5Fstu%5Fxidian%5Fedu%5Fcn%2FDocuments%2FUCF%2DCrime%2FI3D&ga=1

feature extraction process ourselves. the extraction of features for a single 16-frame segment takes approximately 0.53 seconds. Based on this time, the full extraction process for the entire dataset would have taken approximately 10 hours under ideal conditions. Morover, at the beginning of this project, we only had access to CPU1 for computation, which significantly limited our processing capabilities. According to the time comparison between CPU1 and GPU1 in Table 3.1 from Section 3.5, feature extraction on CPU1 takes approximately 18 times longer than on GPU1. This means that the total extraction time would have increased to approximately 7 days, a duration that we considered unfeasible given the constraints of our research timeline.

On the other side, for the local experiments that required feature extraction, we used the code implementation provided by the UR-DMU authors [2] following the I3D methodology.

An additional aspect to note is the use of 10-crop data augmentation in all three algorithms by the authors. Data augmentation is crucial in deep learning, particularly when training data is limited. It prevents overfitting, which could otherwise compromise the generalization capability of the model. The 10-crop technique generates 10 versions of each resized 224x224 frame: five spatial crops taken from each corner and the center of the frame, along with their horizontal flips. This augmentation effectively increases the dataset size, improving the robustness of the trained models.

The resulting feature vector for each video is of the shape [10, X, 1024], saved as a `.npy` file. Here, 10 corresponds to the augmentation, where each row represents a specific crop. The dimension 1024 is fixed by the I3D neural network, and X depends on the total number of frames in the video and the chunk size. This chunk size represents the number of frames processed at a time, and it's fixed to 16. For instance, the video "Abuse028," consisting of 1412 frames, produces a feature vector of size [10, 89, 1024]. This vector is then split into 10 individual `.npy` files of size [89, 1024], each corresponding to a crop, to ensure compatibility with the prediction network. Consequently, each vector of size [1, 1024] represents 16 frames. This framework can be seen in Figure 4.1

The prediction network is designed to process feature vectors as input data. While the prediction network performs inference almost instantaneously ($5 \cdot 10^{-5}$s), the bottleneck

[2] `https://github.com/henrryzh1/UR-DMU/blob/master/feature_extract/README.md`

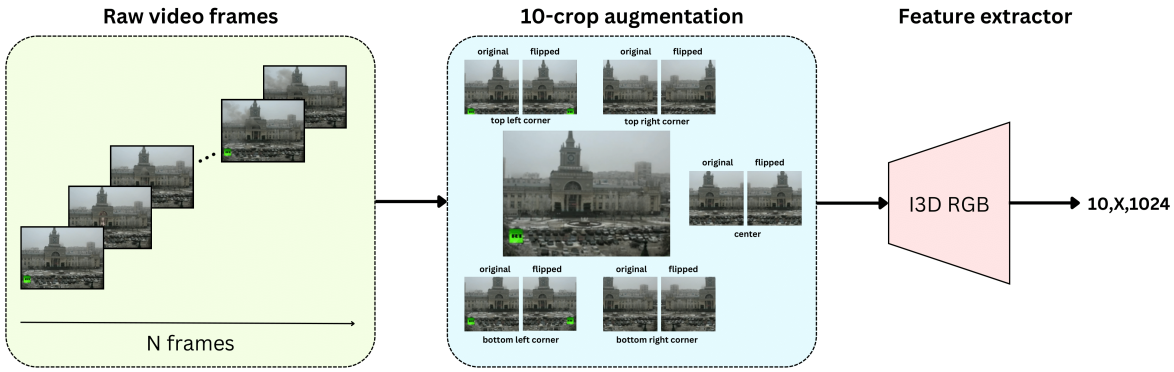


Figure 4.1: Framework of the feature extraction process.

lies in the feature extraction process. Extracting a feature vector for 16 frames using the I3D model takes approximately 0.5 seconds. Consequently, processing the entire test dataset requires around 40 hours on a GPU, making real-time analysis infeasible. This limitation underscores the need for optimizing the feature extraction process.

4.2 Downsampling and Computational Cost

To address the computational inefficiency of the feature extraction process, we have conducted an experiment focused on downsampling, which involves reducing the video frame rate. Instead of analyzing every frame at the standard 30 frames per second (fps) of the original videos, we have opted to maintain only 1 out of every X frames. This approach effectively decreases the total number of frames processed, reducing the time required for feature extraction.

Downsampling not only reduces computational demands but also aligns with the nature of video anomaly detection tasks. In many cases, anomalies are characterized by events taking place over several seconds, making it unnecessary to analyze every single frame. By carefully selecting the downsampling rate, we aim to determine whether the algorithms are robust to reduced temporal resolution while still preserving sufficient information for anomaly detection.

To explore this question regarding the impact of downsampling on detection performance, we have evaluated the performance of the algorithms using features extracted at

Frame Rate (fps)	PEL4VAD			URDMU			BN-WVAD		
	ROC AUC	PR AUC	FAR	ROC AUC	PR AUC	FAR	ROC AUC	PR AUC	FAR
30	86.76	33.74	**6.89**	86.97	34.68	**11.60**	**83.86**	25.79	21.41
15	87.66	36.42	12.64	**87.43**	37.45	12.30	83.37	27.58	20.21
10	**87.75**	**37.16**	16.07	87.30	23.70	13.66	83.58	28.28	22.88
6	87.41	36.83	19.21	86.56	40.65	15.72	83.51	**29.80**	22.83
3	85.39	34.30	20.57	84.74	38.73	17.33	82.03	29.28	20.01
2	84.07	31.62	19.52	84.69	**41.20**	17.22	82.40	28.65	19.07
1	82.33	23.70	13.73	81.73	37.46	14.95	81.33	25.10	**16.73**

Table 4.1: Evaluation metrics at different frame rates.

different frame rates. Specifically, we analyzed how the metrics such as ROC AUC, PR AUC, and FAR varied when reducing the frame rate. The results of this experiment, shown in Table 4.1, provide insights into the robustness of the algorithms.

Interestingly, we observed that at slightly reduced frame rates, performance metrics showed improvement in some cases. For example, the highest ROC AUC for PEL4VAD (87.75) was achieved at 10 fps, and the PR AUC for URDMU peaked at 41.20 at 2 fps. However, this trend was not consistent when the frame rate was reduced further. At very low frame rates, the performance began to decline, indicating that there is a limit to how much temporal information can be removed without impacting detection accuracy.

Additionally, while some metrics improved at lower frame rates, this was followed by an increase in the False Alarm Rate. A higher FAR indicates a greater number of false positives, which could lead to unnecessary alerts. This trade-off highlights the importance of selecting an appropriate downsampling rate that balances computational efficiency and anomaly detection reliability.

It is also worth noting that despite the reduced temporal resolution, the overall detection performance has remained relatively high. We believe this is due to the specific nature of the videos used in the experiments. The original I3D models were trained on video snippets of 64 frames at 25 fps. We believe that this training videos were very diverse. The dataset likely includes both videos with slow, static content (resembling high-frame-rate scenarios) and videos with rapid, dynamic actions (similar to scenarios with downsampling). This variety could explain why reducing the frame rate in our experiments does not significantly degrade the performance of the model.

In Figure 4.2, we have presented a clear example of how the three algorithms perform when comparing their original predictions with the downsampled ones, as well as the

ground truth, specifically at a 2fps rate.

Overall, our conclusions are positive. The experiments confirm that predicting fewer frames does not significantly degrade detection performance and, in some cases, even enhances it. This validates that downsampling is a practical and efficient strategy to improve the computational feasibility of anomaly detection without sacrificing accuracy.

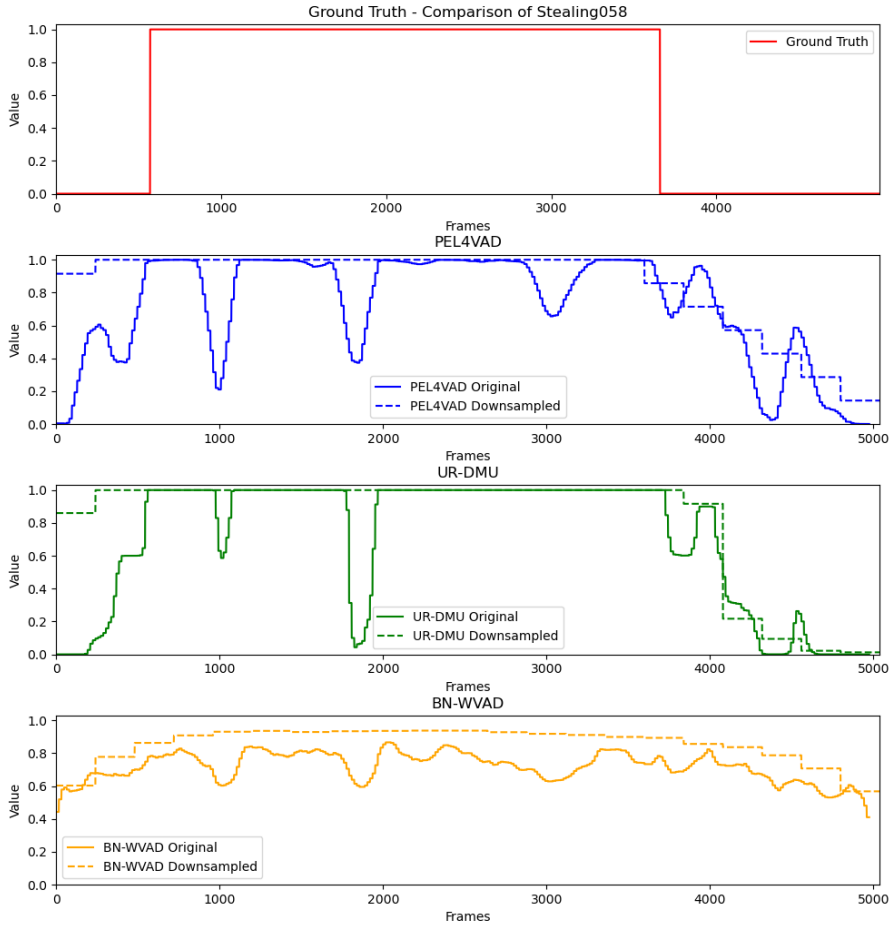


Figure 4.2: Comparison of original predictions, downsampled predictions at 2fps, and ground truth for the three algorithms.

In addition to evaluating detection performance across different frame rates, we have also re-trained and fine-tuned the PEL4VAD algorithm specifically for a frame rate of 2 fps. For this purpose, we extracted features from videos by keeping only 2 out of every 30 frames. The goal of this retraining was to explore whether the performance metrics could be further improved. The results, shown in Table 4.2, demostrates that, although there were no significant improvements, the fine-tuned version of PEL4VAD achieved the best metrics among the tested models, with a ROC AUC of 85.83, a PR AUC of 34.83,

Frame Rate (fps)	PEL4VAD			UR-DMU			BN-WVAD		
	ROC AUC	PR AUC	FAR	ROC AUC	PR AUC	FAR	ROC AUC	PR AUC	FAR
2	84.07	31.62	19.52	**84.69**	**41.20**	**17.22**	**82.40**	**28.65**	19.07
2 fine-tuning	**85.83**	**34.83**	**19.71**	81.69	20.31	24.01	81.17	25.80	**20.21**

Table 4.2: Evaluation metrics at 2fps, original and fine-tuning

and a FAR of 19.715.

Moreover, we have analyzed the computational costs associated with different frame rates. Table 4.3 summarizes the real-time processing time and resource usage for each frame rate. The reported time corresponds to the actual time required to process 16 frames. For example, at the original frame rate of 30 frames per second, processing 16 frames takes approximately 0.53 seconds. Similarly, for lower frame rates, the processing time is scaled accordingly—for instance, at 1 frame per second, the time increases to 16 seconds due to the reduced number of frames analyzed per second.

We have considered both the prediction time and the feature extraction time for 16 consecutive frames, as previously discussed in Section 3.5, to provide a comprehensive view of the computational demands. This computational cost, as shown in Table 4.3, reveals the real-time processing times and resource usage at different frame rates. As expected, reducing the frame rate results in a significant decrease in computational requirements. For instance, processing videos at 1 fps reduces GPU usage to just 3.72%, compared to the 111.53% required at 30 fps. This reduction reflects the substantial savings in computational resources that downsampling can offer.

In addition to this, Table 4.4 shows the time required for each descriptor calculation, including prediction and extraction times. The prediction time is relatively small, whereas feature extraction time is more significant.

We reiterate that downsampling is especially beneficial in the context of CCTV camera systems, which often operate on low-computational CPU-based setups while still delivering reliable results.

Our findings confirm that downsampling is an effective strategy for video anomaly detection. By lowering the frame rate, we have improved computational efficiency without sacrificing—and sometimes even improving—detection performance. This approach is particularly advantageous in large-scale scenarios with limited computational resources.

Frame Rate (fps)	Real Time Processed (s)	GPU2 (%)	CPU2 (%)
30	0.53	25.98	111.53
15	1.07	12.99	55.76
10	1.60	8.66	37.18
6	2.67	5.20	22.31
3	5.33	2.60	11.15
2	8.00	1.73	7.44
1	16.00	0.87	3.72

Table 4.3: Computational cost.

Time/Descriptor (s)	GPU2	CPU2
Prediction	$5.59 \cdot 10^{-5}$	$4.05 \cdot 10^{-4}$
Extraction	0.1385	0.5944

Table 4.4: Computational time.

Chapter 5

Improvement of Evaluation Metrics

Evaluating the performance of anomaly detection systems is essential for understanding their reliability in real-world applications. This chapter addresses the challenges posed by the limitations of traditional evaluation metrics and considers alternative approaches that better align with the imbalanced nature of video datasets. Additionally, we have explored the use of window-based evaluation, which shifts the focus from frame-by-frame analysis to detecting anomalies over broader intervals.

5.1 Evaluation Metrics and Dataset Limitations

While analyzing the results and the metrics, we have identified inconsistencies in both the choice of metrics and their application. Firstly, the FAR is traditionally calculated only using normal videos. Following the approach proposed by Sultani et al., the assumption is that in real-world surveillance settings, most video frames are normal. Consequently, a robust anomaly detection model should have a low FAR on normal videos. However, this approach overlooks a critical point: in abnormal videos, the vast majority of frames are also normal. By prioritizing the reduction of false alarms, the evaluation places less emphasis on correctly identifying the relatively rare but crucial anomalous frames. We believe that detecting these anomalies is far more critical than merely reducing false alarms, with the FAR remaining within an acceptable range.

This leads to the second issue, which lies in the inherent imbalance of the dataset,

which has been largely unaddressed in this prior studies. While the number of normal and abnormal videos in the dataset appears balanced (290 videos in total, with 150 normal and 140 abnormal, representing 51.72% and 48.27%, respectively), this balance does not extend to the frame level. Out of 1,111,131 total frames, only 84,148 are anomalous, representing a mere 7.58% of the dataset. This extreme imbalance misrepresents the ROC AUC metric, which performs well in balanced datasets but becomes less informative in this scenario. Out of all the anomalous frames, just 37.96% were accurately predicted, whereas the normal frames represented the 76.57% of all normal frames.

For instance, the ROC AUC for PEL4VAD is 86.76, suggesting good overall performance. However, this high value is due to correctly predicting the majority of normal frames rather than the relatively rare anomalous frames. To illustrate, consider an extreme case where a dataset contains 10 anomalous frames and 100,000 normal frames. A model with a true positive rate (TPR) of 100% and a false positive rate (FPR) of 1% would achieve a perfect classifier score in ROC AUC terms, predicting well the 10 positives, but it would also produce 1,000 false alarms—a problematic outcome in real-world applications.

Given these limitations, we have concluded that the Precision-Recall AUC (PR AUC) is a more appropriate metric for our task. Unlike ROC AUC, PR AUC focuses on the trade-off between precision and recall, emphasizing the model's ability to detect anomalies while keeping false alarms under control. Recall, or sensitivity, measures how many actual anomalies were correctly predicted, and it is defined as 2.1. Similarly, precision quantifies the proportion of predicted anomalies that are actually correct, and it is defined as 5.1.

The PR AUC metric is particularly well-suited for imbalanced datasets, as it focuses on the performance of the minority class (anomalies) rather than being skewed by the majority class (normal frames). As shown in Section 4.2, PR AUC values tend to be significantly lower than ROC AUC values. Although PR AUC has been analyzed in some of these studies, it is often given secondary importance or evaluated using different datasets, which limits its adoption.

$$Precision = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}} \tag{5.1}$$

5.1.1 Experiments

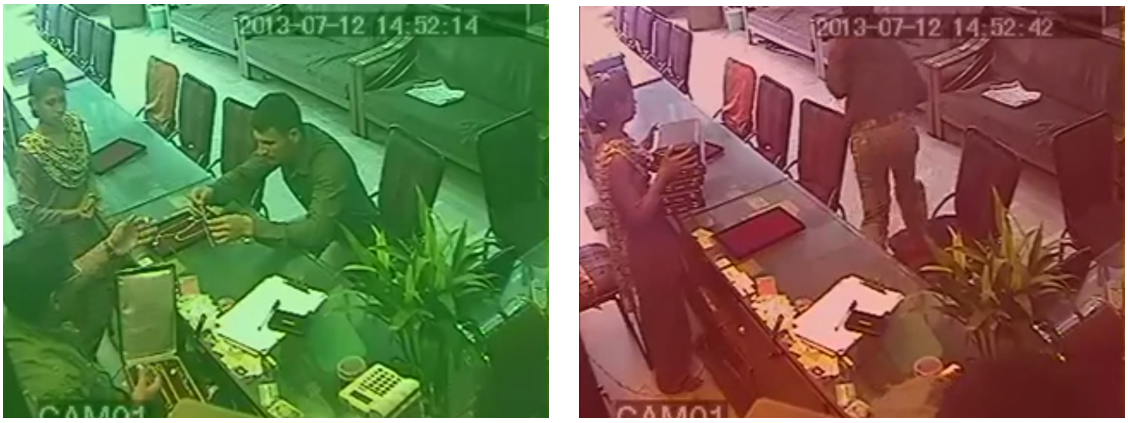
To further support our argument, we have performed a detailed numerical analysis to demonstrate how high ROC AUC values can mask poor anomaly detection performance at the individual video level. While the ROC AUC metric has been widely used to evaluate models for video anomaly detection, we have identified specific cases where this metric fails to provide an accurate representation of the model's practical performance. This analysis aims to highlight the limitations of relying solely on ROC AUC and the importance of complementary metrics, such as PR AUC and FAR, to provide a more balanced evaluation of a model's performance. All these results has been evaluated using the PEL4VAD algorithm.

Next, we have summarized some case studies, and we have illustrated them in Figure 5.5.

- **Misleading High ROC AUC:** Consider the video *"Shoplifting037"*, which has a ROC AUC of 80.68 using the PEL4VAD algorithm, a value close to the average ROC AUC performance across all videos. The video contains 1,386 frames, of which only 60 are anomalous. Despite this, at a threshold of 0.5, the model has predicted 1,326 frames correctly but has failed to detect any of the anomalous frames. Additionally, there have been no false alarms. The high ROC AUC score, in this case, is misleading, as it suggests that the model performs well overall, but in reality, it has completely failed to detect the anomalies in this video. In contrast, the PR AUC for this video is only 9.93, which provides a more realistic assessment of the model's inability to detect anomalies under these conditions. This highlights the need to consider metrics like PR AUC, which better capture the model's performance in imbalanced scenarios where anomalies are rare.

  Figure 5.1 presents two frames from the video: the first frame represents a normal situation within the context of the video, while the second frame represents the abnormal event of the shoplifting incident that was not detected by the prediction network.

- **High PR AUC Despite Poor Recall:** In the case of *"Shooting047"*, the model has correctly identified only 12.31% of the anomalous frames (256 out of 3,481



(a) *Normal frame* **(b)** *Abnormal frame*

Figure 5.1: Visualization of video *Shoplifting037*.

abnormal frames). Despite this low recall, the PR AUC is relatively high (81%). This high PR AUC is primarily due to the low number of false alarms, with only 59 false positives out of the total 4,460 frames in the video, giving a FAR of 6.02. This example illustrates how PR AUC emphasizes the trade-off between precision and recall, rewarding models that maintain high precision by minimizing false alarms. However, such high PR AUC values may still mask poor recall, which could be critical in real-life anomaly detection scenarios. However, of all the videos, this is not the most common case; normally, if the number of false alarms is 0, the number of predicted anomalous frames is also 0.

Figure 5.2 presents two frames from the video.

(a) *Normal frame* **(b)** *Abnormal frame*

Figure 5.2: Visualization of video *Shooting047*.

– **High False Alarms Impacting PR AUC:** In contrast, the limitations of ROC AUC are further evident in videos where a model achieves a high true positive rate but at the cost of a significant number of false alarms, as mentioned before. For instance, in the case of *"Abuse030"*, the model has predicted all 85 anomalous frames correctly, achieving a perfect recall. However, it has also produced 507 false alarms across the video, with a FAR of 34.74, significantly affecting the model's practical utility. Despite this, the ROC AUC remains relatively high at 84.88, which suggests good performance. The PR AUC, however, drops to a more representative 15.19.

  Figure 5.3 presents two frames from the video.

(a) *Normal frame* **(b)** *Abnormal frame*

Figure 5.3: Visualization of video *Abuse030*.

– **Balanced High ROC AUC and PR AUC:** On the other hand, when both ROC AUC and PR AUC are similar and high, we have concluded that the model has performed well in terms of both precision and recall, with minimal false alarms. For example, in the video *"Burglary037"*, the ROC AUC is 97.28, and the PR AUC is 98.94. This video contains 1,411 abnormal frames, all of which have been correctly predicted, with only 109 false alarms, and 1,922 total frames. In such cases, the high values of both metrics align, demonstrating that the model has achieved a balanced and reliable performance.

  Figure 5.4 presents two frames from the video.

From these examples, we have concluded that while ROC AUC remains a valuable metric for evaluating anomaly detection models, it is insufficient on its own, especially

(a) *Normal frame* **(b)** *Abnormal frame*

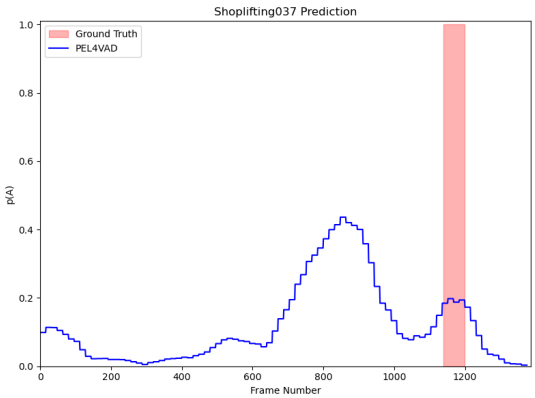
Figure 5.4: Visualization of video *Burglary037*.

in the context of highly imbalanced datasets like ours. PR AUC provides a complementary perspective, as it captures the trade-off between precision and recall, focusing on the model's ability to detect anomalies while controlling false alarms. To achieve reliable and practical performance, we argue that metrics such as PR AUC should be considered alongside FAR, as these provide a more balanced assessment of the model's behavior. Balancing these metrics ensures that the model not only identifies anomalies effectively but also minimizes the false alarms that could infer its performance in real-world scenarios. This approach is essential for robust evaluation and ensures that anomaly detection systems are both accurate and practical.

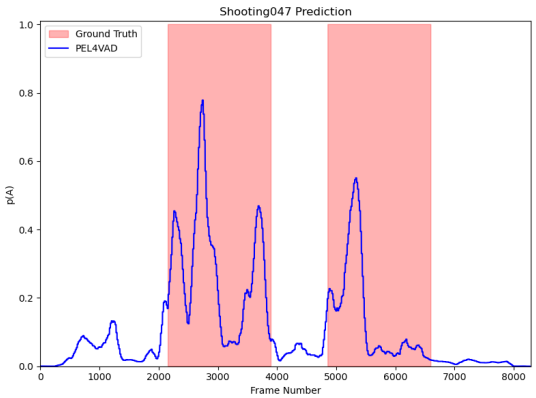
Following this individual metrics, we haved made another analaysis to see which type of videos are more suscepcible to the model. The *Assault* videos has the highest metrics, with 97.57 ROC AUC and 95.63 PR AUC, and 2.74 FAR. the worst type of anomaly is *Explosion*, with 59.53 ROC AUC, 16.97 PR AUC and 35.23 FAR. The most difference with both metric is the *Abuse* anomaly, with 79.19 ROC AUC, 11.03 PR and 53.44 FAR.

5.1.2 Dataset Limitations

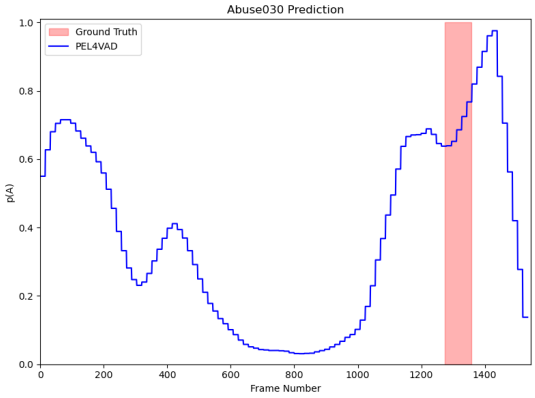
We have also identified a significant issue in the dataset related to the labeling of anomalies. For instance, in examples such as the video *"Explosion016"*, visualized in Figure 5.6, the predicted anomalous region extends until frame 700, whereas the ground truth only spans from frame 180 to frame 450. Analyzing the video, we have observed that



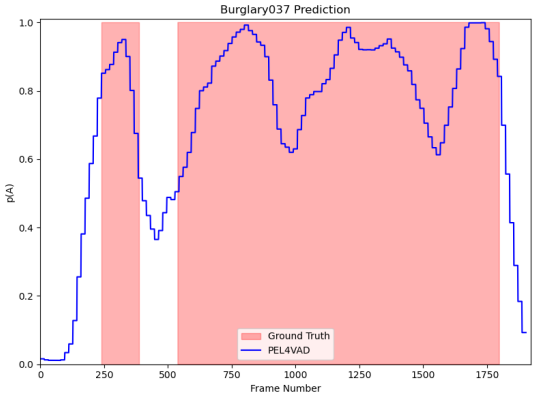
(a) *Shoplifting037*. High ROC AUC but no anomalous frames detected. PR AUC provides a realistic assessment of 9.93%.



(b) *Shooting047*. High PR AUC (81%) due to low false alarms, but poor recall (12.31%).



(c) *Abuse030*. Perfect recall but high false alarms (507), leading to low PR AUC of 15.19.

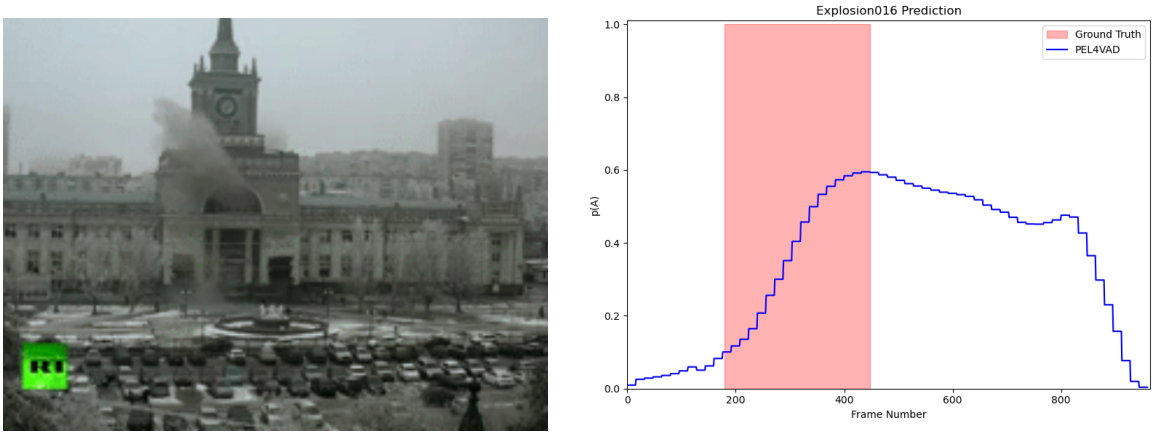


(d) *Burglary037*. High ROC AUC (97.28) and PR AUC (98.94), with balanced precision and recall.

Figure 5.5: Illustration of case studies showcasing the relationship between ROC AUC, PR AUC, and the practical performance of the model.

although the explosion itself occurs in a small number of frames, the subsequent presence of smoke—a direct consequence of the explosion—persists in the scene for a much longer duration. However, the labeling marks these frames as "normal."

This inconsistency suggests that the labeling might have been carried out with a narrow focus, strictly targeting the core anomaly type (e.g., the explosion) and ignoring other related events that occur as part of the same anomalous scene. This approach introduces a level of subjectivity into the labeling process and does not account for the wide spectrum of unusual actions that can characterize an anomalous scenario.



(a) *Explosion Frame.* **(b)** *Prediction with Ground Truth.*

Figure 5.6: Visualization of video *Explosion016*.

Such labeling inaccuracies may significantly impact the evaluation metrics, potentially leading to an underestimation of the models' performance. For example, predictions identifying smoke as part of the anomalous event could be penalized, even though they align with the scene's context. This highlights the critical importance of adopting a more comprehensive and precise labeling strategy to better reflect the complexities of real-world anomalies.

5.2 Evaluation Metrics at Window Level

The main goal of this work is not to fully automate anomaly detection or place complete responsibility on the machine. Instead, the aim is to assist human operators who may need to monitor dozens of CCTV cameras in real time, which can be overwhelming and lead to errors. The system is intended to serve as a tool to guide the operator's attention by flagging potential anomalies, reducing their workload. Therefore, it is not essential to detect the exact frames where the anomaly occurs but rather to signal that an anomaly might be happening within a segment of time. This redirects the focus toward a simpler but effective approach, which involves evaluating predictions over a window of frames rather than frame by frame.

For this purpose, we have evaluated the performance of the models using a window size of 30 frames, matching the frame rate of 30fps, representing an interval of 1 second. A

window is considered anomalous if more than 50% of its frames (at least 15) are predicted as anomalous, meaning they exceed the threshold of 0.5. Similarly, for the ground truth, a window is labeled as anomalous if more than 50% of its frames are marked as anomalies. This method reduces the granularity of the evaluation but aligns better with the intended use case of guiding human intervention.

The evaluation results using this window-based approach are summarized in Table 5.1. From the analysis, we have observed that the ROC AUC values decrease for all three algorithms when evaluated at the window level compared to the original frame-by-frame evaluation. For instance, PEL4VAD's ROC AUC drops from 86.76 to 69.57, while UR-DMU decreases from 86.97 to 73.18, and BN-WVAD falls from 83.85 to 74.34 (in comparison to our local stated metric, not with the original one).

In contrast, the PR AUC improves for all models in the window-based evaluation. For example, PEL4VAD's PR AUC increases from 33.99 to 42.44, while UR-DMU rises from 35.58 to 44.97, and BN-WVAD improves from 36.26 to 46.80. This increase occurs because PR AUC focuses on the trade-off between precision and recall. By grouping frames into windows, false positives are reduced, and the model's predictions become more consistent, which positively impacts precision and recall.

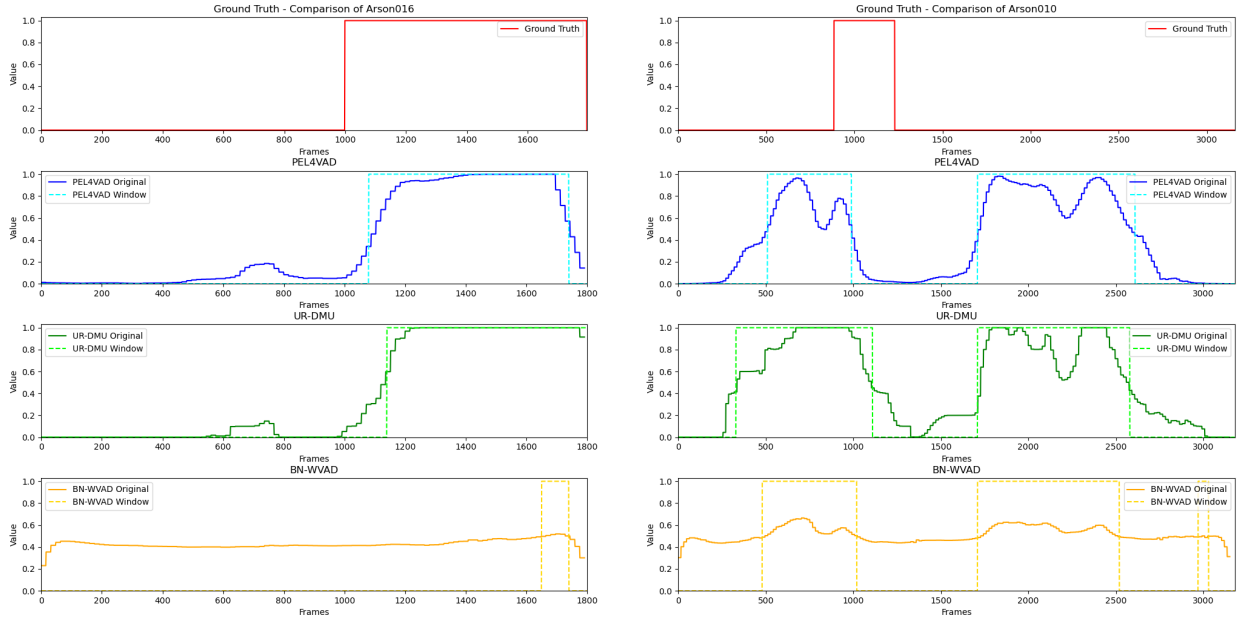
The False Alarm Rate remains relatively stable for most models, with minor changes observed between frame-level and window-level evaluations. For example, PEL4VAD's FAR slightly decreases from 7.32% to 7.21%, while UR-DMU experiences a small increase from 11.60% to 12.10%. The BN-WVAD algorithm, however, shows a more pronounced increase in FAR under the window-based evaluation, reaching 21.67%.

Taking into account our new focus on balancing PR AUC and FAR, this analysis represents a better implementation for real-time scenarios.

	PEL4VAD		**UR-DMU**		**BN-WVAD**	
	Original	**Window**	**Original**	**Window**	**Original**	**Window**
ROC AUC	**86.76**	69.57	**86.97**	73.18	**83.85**	74.34
PR AUC	33.99	**42.44**	35.58	**44.97**	25.79	**46.80**
FAR (abnormal)	7.32	**7.21**	**11.60**	12.10	**21.41**	21.67

Table 5.1: Evaluation metrics using window-level evaluation.

Figure 5.7 illustrates two examples of this approach: (a) a good example where the window-level predictions successfully align with the ground truth and provide clear guidance, and (b) a bad example where the frame-by-frame predictions fail to accurately localize anomalies, so does the window-level approach.



(a) *Arson016* **(b)** *Arson010*

Figure 5.7: Visualization of two videos using window-level prediction.

Chapter 6

Improvement of Detection Quality

Enhancing the quality of anomaly detection is crucial for building reliable systems capable of handling real-world complexities. This chapter investigates methods to improve detection performance by leveraging contextual information and optimizing training strategies through advanced loss functions. By addressing challenges such as dataset imbalance and the distinction between normal and anomalous events, we aim to refine the models' ability to provide more accurate and consistent predictions.

6.1 Contextual Information for Anomaly Detection

In line with the framework of PEL4VAD, which incorporates both global and local contexts, we have explored an additional strategy to enhance anomaly detection. Our approach involves adding prior and subsequent normal context to video sequences. The goal is to improve the network's ability to differentiate between normal and anomalous frames, thereby decreasing the predicted anomaly scores for normal frames while retaining high scores for truly anomalous ones.

To evaluate this idea, we used I3D feature vectors and selected the 20% of frames with the lowest variance from each video. These frames, which represent the most static and consistent parts of the video, were added at the beginning and end of the original sequence, repeating this process five times. The augmented video features were then processed by the anomaly detection network.

Figure 6.1(a) and Figure 6.1(b) illustrate the effects of adding this context. In the original predictions, anomaly scores before the anomalous events were relatively high, around 0.5. After adding context, these scores dropped to approximately 0.15, as shown in the figures. However, when analyzing the evaluation metrics across the dataset, we observed that the improvements were minimal and, in some cases, the performance even decreased.

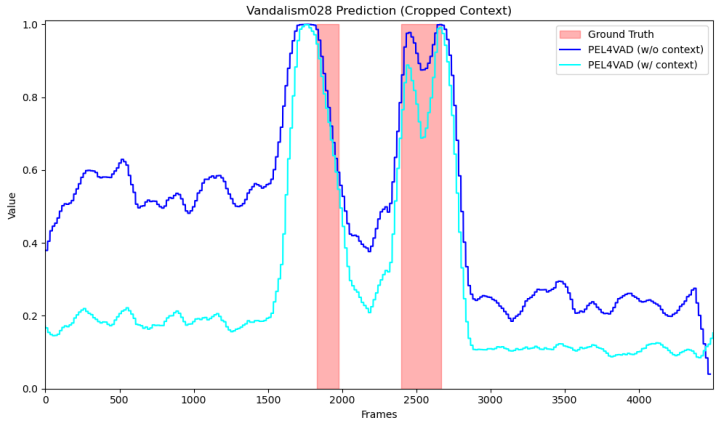
Table 6.1 compares the performance metrics for predictions with and without context across the three algorithms: PEL4VAD, UR-DMU, and BN-WVAD. While PEL4VAD showed a small decrease in the FAR metric, the overall ROC AUC and PR AUC scores remained almost unchanged. For UR-DMU and BN-WVAD, the results were less favorable, with notable drops in both metrics. For instance, in BN-WVAD, the ROC AUC decreased from 87.24% to 81.12%, and the PR AUC dropped significantly from 36.26% to 23.54%.

Metric	**PEL4VAD**		**UR-DMU**		**BN-WVAD**	
	Original	**Context**	**Original**	**Context**	**Original**	**Context**
ROC AUC	**86.76**	86.34	**86.97**	86.78	**83.85**	81.12
PR AUC	33.99	**34.39**	**35.58**	34.48	**25.79**	23.54
FAR (abnormal)	7.32	**3.36**	**11.60**	12.13	21.41	**17.91**

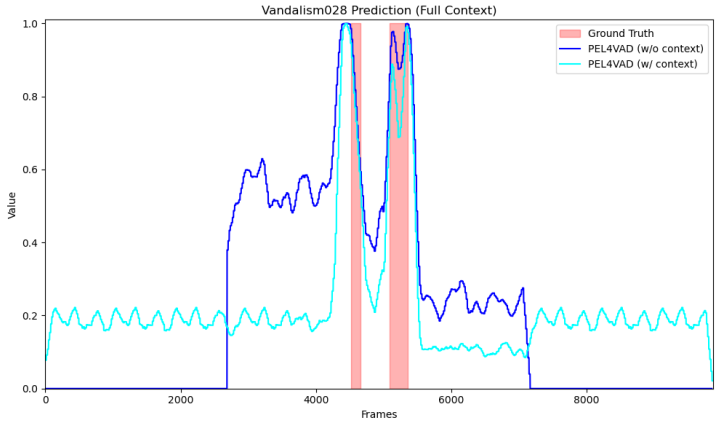
Table 6.1: Evaluation metrics for different methods, with and without context.

To gain a deeper understanding of these results, we examined individual videos in more detail. In general, the expected improvement from *Vandalism028* was not the predominant behavior, as most videos showed only slight reductions in the anomaly scores, as illustrated in Figure 6.2(a). In more significant cases, such as *"Fighting003"* and *Explosion028*, shown in Figure 6.2(b) and Figure 6.2(d), respectively, the model performed well without context, correctly classifying most anomalous frames with a threshold of 0.5. However, when context was added, many of these frames fell below the threshold, negatively impacting the metrics. On the other hand, in *"RoadAccidents020"*, shown in Figure 6.2(c), the additional context significantly reduced false positives, resulting in a better balance between true positives and false positives.

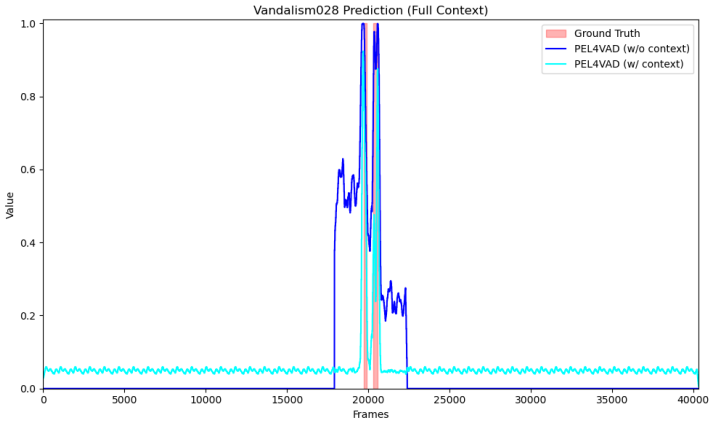
We have also tested a more exaggerated approach by repeating the normal context 20 times. As shown in Figure 6.1(c), this extended context caused both normal and



(a)



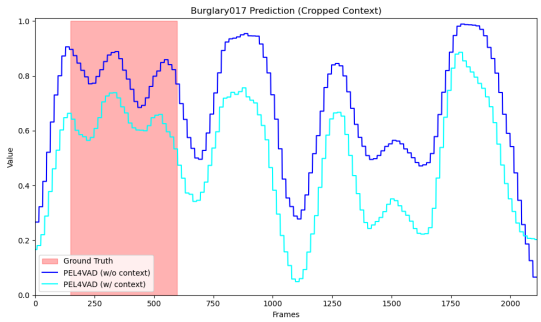
(b)



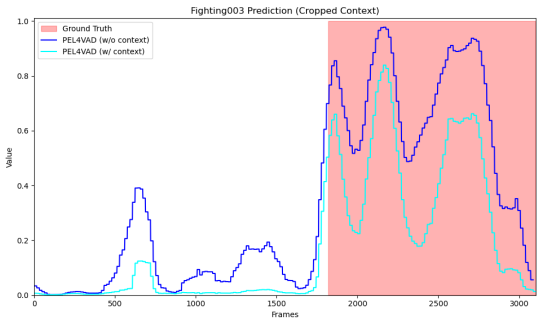
(c)

Figure 6.1: Illustration of predictions for *Vandalism028* showcasing different configurations: (a) Only the original frames range with the context added 5 times. (b) Frame range with the context added 5 times. (c) Frame range with the context added 20 times.

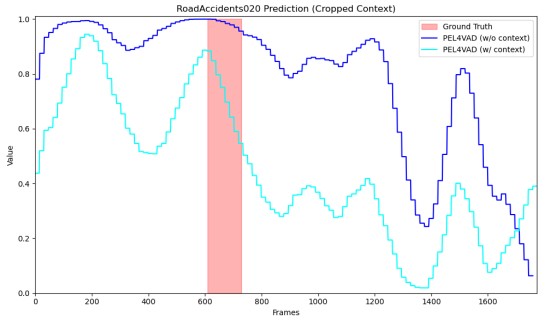
anomalous frame scores to decrease slightly, as before. However, the overall metrics were worse, indicating that adding too much context can introduce noise or abrupt transitions that confuse the network.



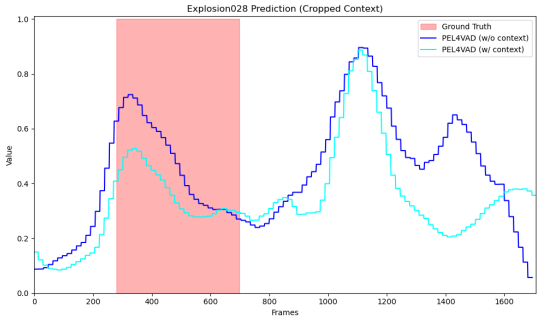
(a) *Burglary017*



(b) *Fighting003*



(c) *RoadAccidents020*



(d) *Explosion028*

Figure 6.2: Illustration of predictions for different scenarios.

Although we have selected frames with the lowest variance, we observed that in some cases, the "normal" context still contained elements of abnormal activity. To address this, we created a smaller test subset, referred to as the "minitest" set, including the 20% of the dataset with a balanced selection of 19 anomalous videos and 19 normal videos, and a representative amount of normal and abnormal frames. For this subset, we manually selected the most static and representative normal frames from each video to ensure the chosen context was truly normal. Consistent with earlier observations, repeating the context five times produced better results than repeating it 20 times. However, even with these adjustments, the metrics showed no significant improvement, as summarized in Table 6.2.

Our analysis of the minitest has revealed several challenges in using normal context to

Metric	Original	Automatic Context	Manual Context
ROC AUC	**89.56**	86.41	88.17
PR AUC	23.23	**24.38**	21.80
FAR	4.12	**2.84**	3.35

Table 6.2: Evaluation metrics with manual and automatic context for the PEL4VAD algorithm.

improve anomaly detection. Defining what constitutes "normal" is highly dependent on the scene. An empty park might be normal in one context, while in another, the presence of people could represent the norm. Adding static frames as context does not always reflect the true context of the video. Additionally, repeating context in loops introduced sharp transitions between the original video and the added frames, which the network sometimes misclassified as anomalies. Using smoother transitions or gradual blending could help mitigate this issue. While adding context reduced false positives in some cases, the overall impact on metrics like PR AUC and ROC AUC was minimal. This suggests that contextual information alone, added witouth prior smoothing, is insufficient to significantly enhance performance.

Future work could focus on dynamic context selection strategies, incorporating a deeper semantic understanding of scenes to identify truly representative normal frames. Techniques like blending or temporal interpolation could create smoother transitions, helping the network better process contextual information without introducing noise or false anomalies.

6.2 Balanced Loss Functions and Training Strategies

One critical aspect that affects the prediction quality of anomaly detection models is the choice of the loss function. Loss functions define the optimization goal during training and directly influence the network's ability to distinguish between normal and anomalous patterns.

For the PEL4VAD algorithm, the original loss function proposed by the authors is the binary cross-entropy (BCE), which employs a Multiple Instance Learning (MIL) framework, where each video is divided into multiple segments, or "bags," for training. They

determine the video-level prediction p_i by taking the mean value of the top-k anomaly scores. For positive bags, they set k = [T/16 +1], and for negative bags, they set k = 1. Given a mini batch containing B samples (128 in this case) with video-level ground-truth y_i, the binary cross entropy is formulated as:

$$L_{ce} = \prod_{n=1}^{B} -y_i log(p_i) \tag{6.1}$$

This is known as BCE Loss, which computes the binary cross-entropy loss. The behavior of this loss function can be intuitively understood as follows:

- **For anomalous frames ($y_i = 1$):** When the prediction is accurate (p = 0.95), the loss is small (0.05), which is desirable. However, if the model incorrectly predicts a lower anomaly score (p = 0.33), the loss increases significantly, penalizing the model appropriately for its error.
- **For normal frames ($y_i = 0$):** When the prediction is accurate (p=0.1), the loss remains low (0.1). Conversely, if the model predicts a higher anomaly score for normal frames, the loss will also increase, reflecting the model's misclassification.

Although BCE effectively minimizes the difference between predictions and ground truth, its standard implementation presents challenges in datasets with significant class imbalance, such as UCF-Crime.

To address this issue, we have explored two alternative implementations of the loss function: Weighted Binary Cross-Entropy (WBCE) and Focal Tversky Loss (FTL), aiming to improve anomaly detection performance under imbalanced conditions.

6.2.1 Weighted Binary Cross-Entropy (WBCE)

To address the imbalance problem, we have implemented a weighted binary cross-entropy loss function (WBCE). WBCE modifies the standard BCE loss by assigning a higher weight to the minority class (anomalies), ensuring that misclassifications of anomalies are penalized more heavily than those of normal frames. The weight for the anomalous class,

referred to as β, is calculated as

$$\beta = \frac{number\ of\ normal\ frames}{number\ of\ anomalies}. \tag{6.2}$$

This weight essentially increases the influence of the anomalous frames during training, giving them equal importance to the normal frames. For the normal class, the weight is set to 1. This guarantees that the loss penalizes anomalies more than normal frames. The standard BCE loss and the weighted BCE loss are defined as follows

$$L_{BCE} = -(y_i \cdot log(p_i) + (1 - y_i) \cdot log(1 - p_i)), \tag{6.3}$$

$$L_{WBCE} = -(\beta \cdot y_i \cdot log(p_i) + (1 - y_i) \cdot log(1 - p_i)), \tag{6.4}$$

being β the weight for the anomalies, y_i the ground truth label, and p_i the predicted probability.

By controlling β, we aim to reduce the bias of the model toward predicting normal frames, allowing it to focus more on detecting anomalies. As observed in Table 6.3, the metrics achieved with WBCE are similar to those obtained with the original BCE. While there is a slight improvement in the PR AUC, other metrics remain unchanged.

6.2.2 Focal Tversky Loss (FTL)

To further address the imbalance, we have implemented the Focal Tversky Loss (FTL), which is an extension of the Tversky Index. The Tversky Index (TI) measures the similarity between predicted and ground truth labels, taking into account both false positives (FP) and false negatives (FN). It is defined as

$$\text{TI} = \frac{\text{TP}}{TP + \alpha FN + \beta FP}, \tag{6.5}$$

where α and β are parameters that control the relative weight of FN and FP, and $\alpha + \beta = 1$. Setting $\alpha > \beta$ penalizes false negatives more heavily, which is particularly useful for imbalanced datasets like UCF-Crime.

FTL modifies the Tversky loss by introducing a focusing parameter γ that adjusts the loss non-linearly,

$$\text{FTL} = (1 - TI)^{\gamma}. \tag{6.6}$$

The parameter γ increases the gradient for harder examples, forcing the model to focus on regions where the Tversky Index is low. This is particularly effective for datasets with extreme imbalance, as it emphasizes harder-to-classify examples, such as anomalies in our case.

However, despite its theoretical advantages, FTL did not perform as expected. While it reduced the False Alarm Rate, this improvement came at the cost of lower ROC AUC and PR AUC values, as seen in Table 6.3.

Table 6.3 summarizes the results of the three loss functions: BCE, WBCE, and FTL. The original BCE function provides the highest ROC AUC, indicating its robustness across the dataset. WBCE shows a slight improvement in PR AUC, suggesting better performance in detecting anomalies. FTL, while reducing FAR, led to a decrease in other metrics, underlining the trade-offs involved.

Metric	**Original**	**WBCE**	**FTL**
ROC AUC	**86.76**	86.60	84.88
PR AUC	33.74	**34.87**	30.43
FAR	6.89	6.65	**3.26**

Table 6.3: Evaluation metrics with different loss functions for the PEL4VAD algorithm.

Chapter 7

Conclusions and Future Work

7.1 Summary of Contributions

In this thesis, we have addressed critical challenges in the domain of anomaly detection for CCTV video surveillance. The project aimed to balance computational efficiency and detection quality, providing scalable and reliable solutions for real-world deployment. Throughout our research, we have focused on evaluating state-of-the-art algorithms, optimizing their computational requirements, and proposing improvements to detection quality.

Our contributions can be summarized as follows:

1. **Comprehensive Algorithm Analysis:** We have set up and evaluated three advanced anomaly detection frameworks: PEL4VAD, UR-DMU, and BN-WVAD. By deploying these algorithms locally, we have ensured their reproducibility and analyzed their strengths and limitations in terms of performance and computational cost.

2. **Improvement of Computational Efficiency:** By introducing downsampling strategies and feature extraction improvements, we have been able to reduce the computational demands of the algorithms. Our experiments showed that downsampling frame rates can maintain, and in some cases improve, anomaly detection performance while reducing processing time. This makes real-time applications feasible on resource-constrained hardware, such as CPUs.

3. **Improvement of Metric Evaluation:** We have identified the limitations of commonly used metrics, such as ROC AUC, in imbalanced datasets, and proposed the use of PR AUC and FAR as more appropriate alternatives.

4. **Improvement of Detection Quality:** We have explored novel strategies, including modifying loss functions and incorporating contextual information, to enhance detection accuracy.

7.2 Future Directions

While this thesis has provided meaningful insights and advancements, several aspects remain open for future exploration and refinement. We outline the following directions:

1. **Improved Dataset Representation:** Despite the progress made, the datasets used in this study have limitations, such as imbalanced frame-level annotations and limited diversity in real-world scenarios. Future work could focus on creating or using more representative datasets that include a broader range of anomalies and environmental conditions.

2. **Weakly Supervised to Fully-Supervised Transition:** One of the main challenges in anomaly detection is the lack of frame-level labels in training datasets. Enhancing detection precision may be possible by integrating predictions from the weakly-supervised technique into a fully-supervised method.

3. **Enhanced Loss Functions:** Although we have experimented with weighted loss functions, more advanced techniques could be explored.

4. **Dynamic Contextual Information:** As observed during our experiments, adding static frames to provide context for anomaly detection yielded limited improvements and introduced challenges such as sharp transitions between original and repeated frames. Future work could focus on dynamic context selection strategies that identify truly representative normal frames, ensuring the added context is relevant to the specific video. Techniques like blending or temporal interpolation could also be explored to smooth transitions and reduce noise introduced by static context.

Taking advantage of artificial intelligence to overcome this strategies could be a potential area for future work.

Bibliography

[1] M. Z. Zaheer, A. Mahmood, M. H. Khan, M. Segu, F. Yu, and S.-I. Lee, "Generative cooperative learning for unsupervised video anomaly detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 14 744–14 754.

[2] Y. Le Cun and F. Fogelman-Soulié, "Modèles connexionnistes de l'apprentissage," *Intellectica*, vol. 2, no. 1, pp. 114–143, 1987.

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[4] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6479–6488.

[5] P. Wu, J. Liu, Y. Shi, Y. Sun, F. Shao, Z. Wu, and Z. Yang, "Not only look, but also listen: Learning multimodal violence detection under weak supervision," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*. Springer, 2020, pp. 322–339.

[6] W. Luo, W. Liu, and S. Gao, "A revisit of sparse coding based anomaly detection in stacked rnn framework," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 341–349.

[7] Y. Pu, X. Wu, L. Yang, and S. Wang, "Learning prompt-enhanced context features for weakly-supervised video anomaly detection," *IEEE Transactions on Image Processing*, 2024.

[8] H. Zhou, J. Yu, and W. Yang, "Dual memory units with uncertainty regulation for weakly supervised video anomaly detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 3, 2023, pp. 3769–3777.

[9] Y. Zhou, Y. Qu, X. Xu, F. Shen, J. Song, and H. T. Shen, "Batchnorm-based weakly supervised video anomaly detection," *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.

[10] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.

[11] R. Speer, J. Chin, and C. Havasi, "Conceptnet 5.5: An open multilingual graph of general knowledge," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017.

[12] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[13] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[15] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev *et al.*, "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017.