



Trabajo Fin De Grado

Automatización y digitalización de una máquina de
escaleras

Automation and digitalization of a stair machine

Javier Alquézar Alquézar

Escuela de Ingeniería y Arquitectura
Grado en Ingeniería Electrónica y Automática

Daniel Latorre Trébol

Jorge Júlvez Bueno

2024

AGRADECIMIENTOS

A Daniel Latorre, Germán Sacramento y Jorge Júlvez, por su ayuda, paciencia y disposición para la elaboración del proyecto. Gracias por compartir sus conocimientos.

A mis padres y mi hermana por el apoyo incondicional desde el inicio de la carrera. Gracias por creer en mí y motivarme.

A Irene por apoyarme día a día y ser una fuente de alegría en mi vida.

A mis abuelas por todas las velas que han puesto por mí.

A mis compañeros de clase, gracias por darme las fuerzas y la actitud necesaria para afrontar cualquier proyecto.

RESUMEN

La automatización industrial ha escalado en importancia dentro de las empresas contemporáneas. La globalización y la necesidad de competir en mercados internacionales impulsan a las empresas a aprovechar los avances en áreas como la informática y la robótica. Esta revolución facilita el trabajo de los empleados, aumentando la eficiencia y la productividad.

El principal objetivo del proyecto es la automatización de una máquina de fabricación de escaleras real, realizando un estudio de calidad sobre el producto final. En este proyecto se analizan los resultados de las operaciones de conformado de la máquina de fabricación de escaleras, clasificándolos en válidos o pendientes de revisión.

El trabajo fin de grado se desarrolla dentro de la empresa Avanti Wind Systems y consiste en la implementación de las soluciones a los problemas reales identificados en la máquina de fabricación de escaleras. Se plantea una comparativa de una solución real propuesta por una empresa asociada a Avanti Wind Systems y una solución desarrollada por medios propios, diseñada utilizando una Raspberry PI 5 y unos módulos de captura de imágenes.

Inicialmente se detalla el funcionamiento de la máquina de fabricación de escaleras, analizando los posibles problemas y explorando en profundidad como mejorarlos. Luego, se detalla la solución planteada por la empresa asociada y la solución implementada por medios propios, orientada en la captura y análisis de imágenes. Se implementa el control de calidad desarrollado al funcionamiento real de la máquina de escaleras, realizando la conexión entre la Raspberry y el autómata programable de la máquina de escaleras.

Se incluyen anexos complementarios al proyecto con información adicional específica y técnica.

ABSTRACT

Industrial automation has increased in importance within contemporary companies. Globalization and the need to compete in international markets drive companies to take advantage of advances in areas such as computing and robotics. This revolution makes the work of employees easier, increasing efficiency and productivity.

The main objective of the project is the automation of a real staircase manufacturing machine, carrying out a quality study on the final product. In this project, the results of the forming operations of the staircase manufacturing machine are analysed, classifying them as valid or pending review.

The final degree project is developed within the company Avanti Wind Systems and consists of the implementation of solutions to the real problems identified in the staircase manufacturing machine. A comparison is made between a real solution proposed by a company associated with Avanti Wind Systems and a solution developed by its own means, designed using a Raspberry PI 5 and some image capture modules.

Initially, the operation of the staircase manufacturing machine is detailed, analysing possible problems and exploring in depth how to improve them. Then, the solution proposed by the associated company and the solution implemented by its own means, focused on image capture and analysis, are detailed. The quality control developed is implemented to the actual operation of the stairs machine, making the connection between the Raspberry and the programmable controller of the stairs machine.

Complementary annexes to the project are included with additional specific and technical information.

ÍNDICE

| | |
|--|-----------|
| CAPÍTULO 1 INTRODUCCIÓN | 8 |
| 1.1 Motivación..... | 8 |
| 1.2 Objetivos..... | 8 |
| CAPÍTULO 2 DESCRIPCIÓN DE LA MÁQUINA DE ESCALERAS | 9 |
| 2.1 Introducción a la máquina de escaleras | 9 |
| 2.2 Funcionamiento de la máquina de escaleras | 9 |
| 2.3 Descripción de defectos de fabricación | 10 |
| 2.3.1 Análisis módulo de expandido | 10 |
| 2.3.2 Análisis módulo de remachado | 11 |
| CAPÍTULO 3 DESARROLLO DE LA VERSIÓN COMERCIAL..... | 13 |
| 3.1 Implementación de mejoras..... | 13 |
| 3.2 Desarrollo de la solución comercial | 13 |
| 3.2.1 Prueba y programación | 13 |
| 3.2.2 Conclusión | 15 |
| CAPÍTULO 4 DESARROLLO DE LA VERSIÓN NO COMERCIAL..... | 16 |
| 4.1 Descripción general | 16 |
| 4.2 Elaboración de la versión ad hoc..... | 17 |
| 4.2.1 Captura de imágenes..... | 17 |
| 4.2.1.1 Hardware captura de imágenes | 17 |
| 4.2.1.2 Criterio de control del módulo de expandido | 18 |
| 4.2.1.3 Criterio de control del módulo de remachado | 19 |
| 4.2.2 Detección de objetos en imágenes | 20 |
| 4.2.2.1 Elaboración del conjunto de datos | 21 |
| 4.2.2.2 Etiquetado de imágenes | 21 |
| 4.2.2.3 Entrenamiento del modelo de detección | 23 |
| 4.2.3 Integración a nivel industrial | 24 |
| 4.2.3.1 Factores externos al desarrollo de la solución | 24 |
| 4.2.3.2 Integración de la solución a la máquina de escaleras | 26 |
| 4.2.3.3 Conexión Automata – Raspberry Pi 5..... | 28 |
| 4.2.3.4 Desarrollo del Script | 30 |
| 4.2.3.5 Estudio del tiempo de ciclo..... | 31 |
| CAPÍTULO 5 CONCLUSIONES | 33 |
| ANEXOS | 34 |

| | | |
|---------------------|--|-----------|
| ANEXO 1 | Descripción más detallada del funcionamiento de la máquina..... | 34 |
| A.1.1 | Módulo de alimentación de largueros..... | 34 |
| A.1.2 | Módulo de punzonado..... | 34 |
| A.1.3 | Módulo de inserción de peldaños..... | 35 |
| A.1.4 | Módulo de expandido..... | 36 |
| A.1.5 | Módulo de remachado..... | 37 |
| ANEXO 2 | Hardware del modelo..... | 38 |
| A.2.1 | Raspberry Pi 5..... | 38 |
| A.2.2 | Raspberry Pi Camera Module 3..... | 38 |
| A.2.3 | Soporte módulo de captura..... | 38 |
| A.2.4 | Autómata siemens S7-1200..... | 45 |
| ANEXO 3 | Software utilizado..... | 46 |
| A.3.1 | Yolo v8..... | 46 |
| A.3.2 | Python..... | 46 |
| A.3.3 | Librería Snap 7..... | 46 |
| ANEXO 4 | Base de datos para la realización del entrenamiento del modelo YOLOv8..... | 47 |
| ANEXO 5 | Script utilizado para la programación de la Raspberry Pi 5..... | 48 |
| A.5.1 | Script captura e inferencia peldaños..... | 48 |
| A.5.2 | Script visualización de resultados..... | 53 |
| ANEXO 6 | Solución desarrollada por medios propios..... | 54 |
| A.6.1 | Prototipo del modelo..... | 54 |
| A.6.2 | Costes de la versión no comercial..... | 54 |
| ANEXO 7 | Diagrama de Gantt..... | 57 |
| BIBLIOGRAFÍA | | 58 |

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1 Vista de la máquina de fabricación de escaleras | 9 |
| Figura 2 Dedos expansores..... | 10 |
| Figura 3 Marcas expandido | 11 |
| Figura 4 Marcas incompletas expandido | 11 |
| Figura 5 Remache..... | 12 |
| Figura 6 Remachado incorrecto..... | 12 |
| Figura 7 Prototipo de BCNVision | 14 |
| Figura 8 Captura y etiquetado | 14 |
| Figura 9 Programación modelo expandido incorrecto | 15 |
| Figura 10 Diagrama de bloques de la solución no comercial..... | 16 |
| Figura 11 2 series de 3 marcas y remache en el peldaño..... | 17 |
| Figura 12 Soporte captura de imágenes | 18 |
| Figura 13 Marcas expandido completas | 18 |
| Figura 14 Marcas expandido incompletas | 19 |
| Figura 15 Resultado fallo total módulo de expandido | 19 |
| Figura 16 Grieta remache | 20 |
| Figura 17 Remachado incorrecto | 20 |
| Figura 18 Remachado correcto..... | 20 |
| Figura 19 Etiquetado remache y expandido correcto | 22 |
| Figura 20 Etiquetado remache incorrecto..... | 22 |
| Figura 21 Matriz de confusión | 24 |
| Figura 22 Captura de imagen tras auto enfoque..... | 25 |
| Figura 23 Foco de luz de la cámara superior..... | 25 |
| Figura 24 Luz incidente en el peldaño | 26 |
| Figura 25 Reflejo producido en el larguero..... | 26 |
| Figura 26 Zona implementación del modelo..... | 27 |
| Figura 27 Soporte control de calidad..... | 27 |
| Figura 28 Control de dispositivos autómeta programable..... | 28 |
| Figura 29 Fin de carrera remachadora..... | 29 |
| Figura 30 Diagrama de flujo del script..... | 30 |
| Figura 31.A Sistema de alimentación de largueros | 34 |
| Figura 32.A Módulo de punzonado | 35 |
| Figura 33.A Sistema elevador y cargador de peldaños..... | 35 |
| Figura 34.A Unidad de inserción de peldaños..... | 36 |
| Figura 35.A Unidad de expansión | 36 |
| Figura 36.A Unidad principal de expandido..... | 37 |
| Figura 37.A Unidad de remachado..... | 37 |
| Figura 38.A Prototipo de la solución..... | 54 |
| Figura 39.A Evolución de la calidad en la captura de imágenes | 54 |
| Figura 40.A Diagrama de Gantt | 57 |

CAPÍTULO 1 INTRODUCCIÓN

1.1 Motivación

En la actualidad, la automatización se ha convertido en una parte crucial de las empresas. Se aumenta la inversión en las tecnologías de automatización y en la digitalización de sus procesos, uno de los objetivos para poder lograr así un aumento de eficiencia y productividad. Este proyecto pretende implementar soluciones innovadoras para afrontar la digitalización de la empresa. Además, se implementa la solución en un entorno real, con la influencia de factores externos que conllevan en la elaboración de la solución.

Por ello, con el departamento de I+D+I, se explora tecnología que se desarrolla en interno para su posible uso en otras fábricas y aplicaciones similares. Además, esta motivación aumenta por el contexto de realizar un proyecto en Zaragoza, ciudad que atrae a multitud de empresas debido al aumento de relevancia nacional en el sector industrial. Todo esto ha influido en la elección de la temática del trabajo fin de grado.

1.2 Objetivos

El objetivo principal consiste en diseñar un sistema para una máquina de fabricación de escaleras basado en tecnologías digitales que permita la mejora de la calidad y la productividad.

Para cumplir este objetivo, se alcanzarán diversos subobjetivos siendo los principales:

- Análisis de una máquina funcional: detección de defectos de fabricación y posibles mejoras para su implementación en una máquina de fabricación de escaleras trabajando en entorno real.
- Descripción y desarrollo de dos soluciones, comercial y no comercial: programación y desarrollo de una versión elaborada por medios propios, cuyo objetivo es alcanzar la fiabilidad y flexibilidad de la solución comercial, reduciendo costes de desarrollo e implementación.
- Diseño de un sistema de visión capaz de capturar imágenes de calidad, adaptando el desarrollo a un entorno real.
- Desarrollo de un modelo de análisis y detección de objetos en imágenes mediante el modelo Yolo v8.
- Integración del sistema en el entorno industrial, realizando la conexión entre el modelo y el autómatas programable, además de la implementación del modelo como un módulo complementario a la máquina de fabricación de escaleras para la visualización de los resultados.

CAPÍTULO 2 DESCRIPCIÓN DE LA MÁQUINA DE ESCALERAS

2.1 Introducción a la máquina de escaleras

Avanti Wind Systems fabrica y suministra sistemas internos completos para torres de aerogeneradores, como son escaleras, puntos de anclaje, líneas de vida, elevadores de servicio, equipos de protección individual, además de componentes eléctricos.

La máquina de fabricación de escaleras es la encargada de producir escaleras de aluminio. La unión de tramos de escalera conforma una escalera de mayor tamaño, que se sitúa en el interior de un molino de viento para su uso.

2.2 Funcionamiento de la máquina de escaleras

La máquina de escaleras está dividida en diferentes módulos, cada uno con una función específica: alimentación de largueros, punzonado, inserción de peldaños, expandido y remachado. La figura 1 muestra la máquina de fabricación de escaleras.

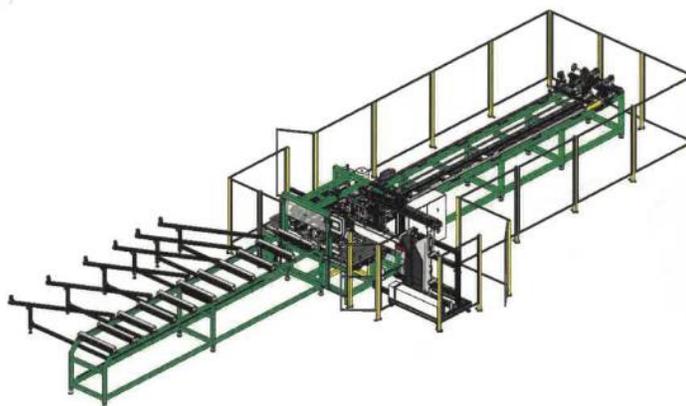


Figura 1 Vista de la máquina de fabricación de escaleras

El proceso de fabricación de escaleras comienza con la alimentación de la máquina de fabricación de escaleras con largueros. A continuación, se realizan cíclicamente cuatro operaciones de conformado:

1. Punzonado de largueros.
2. Inserción de peldaño en los largueros.
3. Expandido de la parte interna del peldaño en el larguero.
4. Remachado exterior del peldaño.

Finalizadas todas las operaciones de conformado en cada peldaño de la escalera, el operario procede a la extracción de la escalera para su posterior almacenaje.

Se incluyen anexos complementarios con una descripción detallada del funcionamiento de la máquina de escaleras, con información adicional y técnica, añadiendo datos más específicos.

2.3 Descripción de defectos de fabricación

En esta subsección se va a explicar en detalle el análisis del funcionamiento de la máquina de fabricación de escaleras, orientado a la descripción de los errores con su frecuencia de aparición.

Se analizaron las operaciones de conformado del producto, divididas por módulos, destacando en ellos los errores más comunes que sucedieron a lo largo de 48 meses.

2.3.1 Análisis módulo de expandido

Se destaca la operación de expandido de la máquina de fabricación de escaleras. Durante la operación se proporciona una ganancia de resistencia entre los peldaños y el larguero que conforman el producto final. El proceso de expandido se realiza mediante unos dedos expansores que, al introducirse en el interior del peldaño, son empujados por un cilindro hidráulico.

El resultado de la operación provoca en el interior del peldaño una deformación mecánica permanente mediante unos dedos expansores, que fija la posición del peldaño respecto a los largueros. La figura 2 muestra los dedos expansores encargados de la operación de expandido.



Figura 2 Dedos expansores

Estas marcas delimitan los bordes exterior e interior del larguero, asegurando la fijación entre el peldaño y el larguero, evitando el movimiento. Por tanto, es de vital importancia que las marcas estén realizadas en la misma posición y tengan una extensión similar. La figura 3 muestra las marcas realizadas por el expandido en el peldaño.



Figura 3 Marcas expandido

La aparición de las marcas tras el expandido indica que la operación se ha realizado correctamente, sin errores. Durante el estudio del funcionamiento de la máquina y del producto, se observó que dependiendo del tipo de aluminio con el que se conforman las escaleras, las marcas resultantes no son siempre las mismas.

Para un mismo proveedor de aluminio, las propiedades del material de cada lote pueden ser diferentes. Para aluminio AW6063, caracterizado por una resistencia a la tracción de 205 MPa, de características más duras y difíciles de tratar, la operación resulta en unas marcas en el interior del peldaño de menor longitud. En cambio, con aluminio AW6060, caracterizado por una resistencia a la tracción de 260 MPa, menos duro, las marcas producidas por el expandido son las adecuadas y las esperadas. La figura 4 muestra el resultado incompleto de las marcas realizadas por el expandido.



Figura 4 Marcas incompletas expandido

La frecuencia de aparición de este defecto es de 1 peldaño por cada 10 escaleras, lo que puede provocar el movimiento del peldaño durante el uso. Por ello, la identificación del defecto es prioritaria. Cabe destacar que el ritmo de producción no permite al operario revisar el resultado de las operaciones de conformado de la escalera, siendo necesario implementar un control de calidad automatizado.

2.3.2 Análisis módulo de remachado

El proceso de remachado, junto con el expandido, otorgan a la escalera un aumento de resistencia ante el movimiento del peldaño con el larguero, asegurando la fijación entre ambos. La figura 5 muestra el remache del peldaño al larguero.

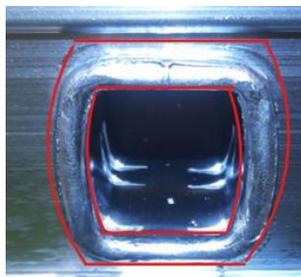


Figura 5 Remache

El remachado es fácilmente observable de forma visual en el borde exterior del peldaño. Cabe destacar que la mayoría de los defectos que la máquina ha tenido con remaches realizados incorrectamente, han sido precedidos por una operación de expandido incorrecta.

Como se observa en figura 6, el espacio entre el peldaño y el larguero es visible cuando se ha realizado un expandido incorrecto, lo que a su vez provoca que el remachado sea incorrecto. En definitiva, el control de la correcta realización de la operación de expandido es de vital importancia para garantizar un producto ideal y apto para su uso. La figura 6 muestra un remache realizado incorrectamente.



Figura 6 Remachado incorrecto

Cabe destacar que, durante los 48 meses analizados, el único error aislado realizado por la remachadora fue un giro de la matriz de remachado en 90°, provocando que el remachado se realizase girado 90° respecto a la orientación definida. Este error fue fácilmente detectado y solucionado.

CAPÍTULO 3 DESARROLLO DE LA VERSIÓN COMERCIAL

3.1 Implementación de mejoras

Como se ha comentado en el capítulo anterior, se realiza un estudio del funcionamiento de la máquina de fabricación de escaleras. Los dos módulos de mayor relevancia son el módulo de remachado y el módulo de expandido, encargados de otorgar y garantizar la seguridad durante el uso del producto final.

Actualmente, la máquina de escaleras carece de métodos para comprobar el resultado de las operaciones de conformado de la escalera. Para solucionar esto, se va a desarrollar un módulo encargado del control de calidad del producto, capaz de detectar los defectos de fabricación más notables de forma automatizada.

3.2 Desarrollo de la solución comercial

Se buscó una solución comercial con el objetivo de que el módulo detecte las operaciones de conformado de la escalera y compruebe los resultados.

Por tanto, se plantea una solución basada a la visión por computador, mediante el análisis de imágenes. Se captura una imagen del interior del peldaño, realizando la detección de las marcas realizadas por el módulo encargado del expandido y el remache producido por el módulo encargado de remachado.

Mediante el análisis y la detección de objetos, se logrará un control de calidad del producto, comprobando el estado de las operaciones de conformado de la escalera.

3.2.1 Prueba y programación

En esta subsección se explica la realización de pruebas con una cámara de visión de BCNvision. Cabe destacar que esta cámara de visión es un prototipo que no está disponible en el mercado.

Se realiza la conexión vía Ethernet a nuestro ordenador, donde se realiza la programación y el etiquetado de imágenes. La figura 7 muestra la solución prototipada de BCNvision.



Figura 7 Prototipo de BCNVision

La programación consiste en crear un modelo válido para que la cámara, en tiempo real, sea capaz de captar y determinar si el peldaño es válido o no. Para lograr esto, se utiliza el programa In-Sigth Vision.

El objetivo es desarrollar un modelo con una fiabilidad superior al 70% para que sea apto. Se desarrollaron 2 modelos independientes, uno encargado de la detección de las marcas del expandido y otro encargado de la detección de errores tras el remachado.

Para el desarrollo de ambos modelos no se necesita ninguna luz exterior, ya que la cámara puede aislar la luz exterior, enfocando y captando únicamente lo que se encuentra en el centro de la lente, colocada a una distancia entre 10 y 15cm.

El proceso de desarrollo del modelo consiste en las siguientes tareas:

- Captura de imágenes.
- Creación de clases y etiquetado de las imágenes.
- Clasificación de las imágenes.

La figura 8 muestra el etiquetado y captura de imágenes mediante In-Sigth Vision.

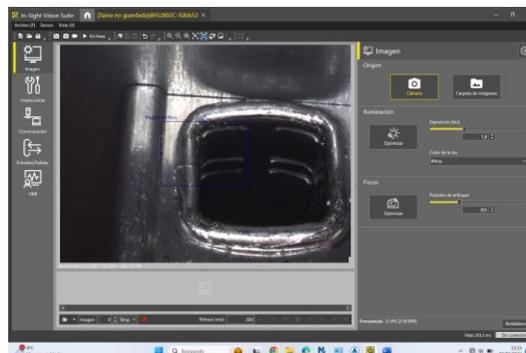


Figura 8 Captura y etiquetado

Para la clasificación de marcas de expandido válidas se siguió un criterio general: que en la imagen aparecieran las seis marcas realizadas por el proceso de expandido. En caso de no aparecer, se clasifican como marcas no válidas, entrenando así nuestro modelo. La figura 9 muestra la programación del modelo en In-Sigth Vision.

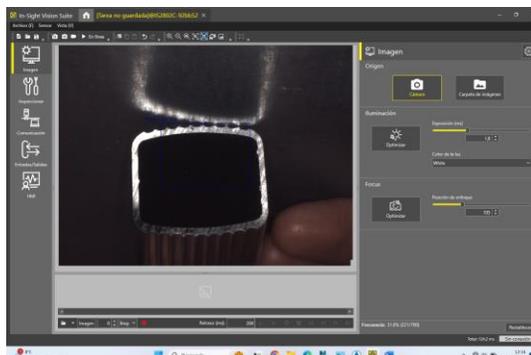


Figura 9 Programación modelo expandido incorrecto

Se realiza un proceso similar con el modelo de detección del remachado, clasificando mediante el etiquetado en remachado válido o no válido, siguiendo el criterio de aparición de grietas o fallo completo del remachado que conlleva la destrucción del peldaño.

Ajustamos el modelo con exactitud, y conforme se añaden diferentes imágenes de multitud de peldaños, el porcentaje de seguridad y la capacidad de análisis de la cámara aumenta, aumentando nuestra capacidad de éxito.

3.2.2 Conclusión

Tras la programación de la cámara proporcionada por BCNVision, se obtiene una solución con un porcentaje de fiabilidad elevado, superior al 80%. Se diseña una solución comercial que consiste en cuatro cámaras, situadas dos en cada lateral de la máquina de fabricación de escaleras. Cada cámara realiza el análisis del resultado del remachado y de las marcas superior e inferior realizadas por el expandido, dependiendo de la orientación de la cámara.

El desarrollo de la solución real, conlleva un coste de varios miles de euros. Por tanto, el retorno de inversión es demasiado bajo para ser viable económicamente debido a la baja incidencia de los defectos y considerando el coste del actual sistema de inspección visual. Por esta razón, se propone el desarrollo de una versión ad hoc, encargada del control de calidad de los resultados de las operaciones realizadas por los módulos de expandido y remachado.

CAPÍTULO 4 DESARROLLO DE LA VERSIÓN NO COMERCIAL

4.1 Descripción general

Como se comenta en la motivación del proyecto, el objetivo principal consiste en implementar y digitalizar las mejoras planteadas en una máquina de fabricación de escaleras. Por tanto, se desarrolla un modelo encargado del control de calidad del producto, capaz de detectar los defectos de fabricación de las operaciones realizadas por los módulos de expandido y remachado del peldaño en ambos laterales de la escalera.

El modelo desarrollado está conformado por dos módulos de captura de imágenes, una tablet para visualizar los resultados del modelo, y dos controladores para la adquisición de imágenes y la inferencia del modelo de detección de clases de objetos en imágenes. El controlador es una Raspberry Pi 5. La figura 10 muestra el diagrama de bloques diseñado para la solución no comercial.

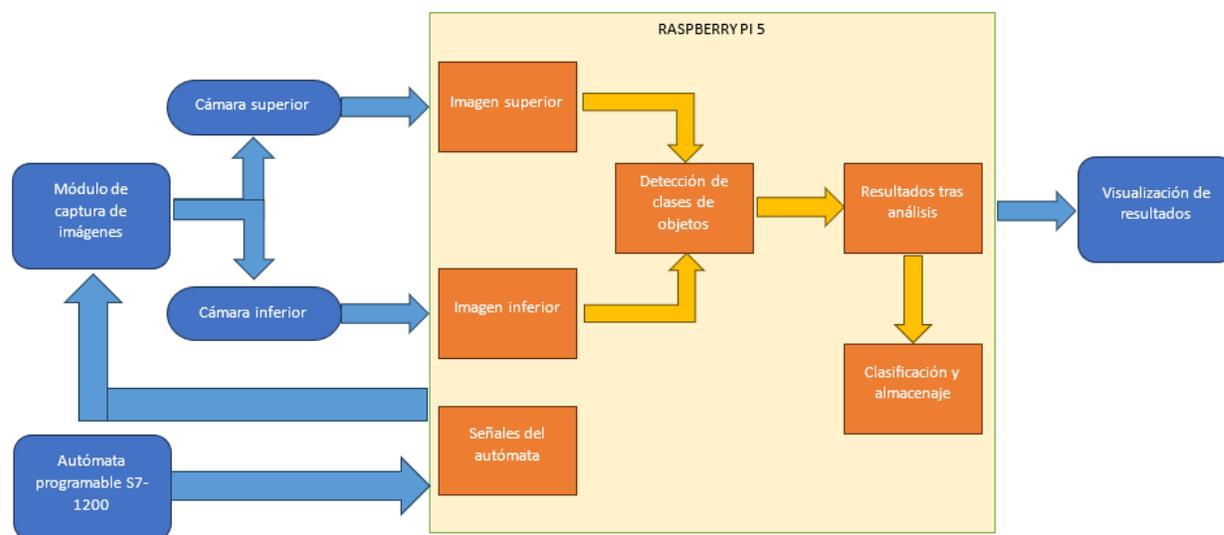


Figura 10 Diagrama de bloques de la solución no comercial

Se ha desarrollado un modelo basado en la captura y análisis de imágenes de cada peldaño por ambos largueros. El módulo de captura de imágenes consta de dos cámaras y dos focos de luz. Cada cámara captura la parte superior o inferior del peldaño, registrando claramente las marcas realizadas por el módulo de expandido y el remache realizado por el módulo de remachado.

Además, mediante un modelo previamente entrenado, se detectan las clases de objetos definidas en las imágenes capturadas y se clasifican dependiendo del estado de los peldaños, informando finalmente al operario de los resultados.

El objetivo de desarrollar una solución no comercial frente a la solución comercial es lograr un proceso eficiente y fiable, además de reducir notablemente los costes de la versión comercial.

4.2 Elaboración de la versión ad hoc

El desarrollo de la versión ad hoc se divide en varias secciones, en las que se explican la captura de imágenes, el desarrollo del modelo de detección de objetos y la implementación de la solución a nivel industrial.

4.2.1 Captura de imágenes

En esta subsección se explica la captura de imágenes en un entorno real y su implementación en la máquina de escaleras, utilizando una Raspberry Pi 5 y módulos de captura de imágenes.

El objetivo del módulo de captura de imágenes es captar la parte más exterior del peldaño, enfocando y capturando con claridad el resultado de las operaciones de conformado de los módulos de expandido y remachado, obteniendo imágenes de alta calidad.

El resultado del módulo de expandido queda reflejado dentro del peldaño como series de tres marcas por cada una de las cuatro esquinas del peldaño. Por tanto, dos cámaras captarán cada una, las esquinas inferiores y superiores del peldaño, captando dos series de tres marcas y el remache realizado. La figura 11 muestra las 2 series de marcas y el remache realizado en el peldaño.



Figura 11 2 series de 3 marcas y remache en el peldaño

4.2.1.1 Hardware captura de imágenes

Es necesario capturar cuatro imágenes por peldaño, dos módulos de captura de imágenes por cada extremo del peldaño. Para lograr la mayor calidad de imágenes, el módulo de captura de imágenes debe situarse en un punto fijo, evitando que el movimiento del módulo provoque un desenfoque de las cámaras. Esto hizo necesario diseñar un soporte para las dos cámaras, con capacidad de movimiento en tres ejes, para poder modificar la posición de las cámaras y obtener unas imágenes más claras. La figura 12 muestra el soporte realizado para la captura de imágenes.

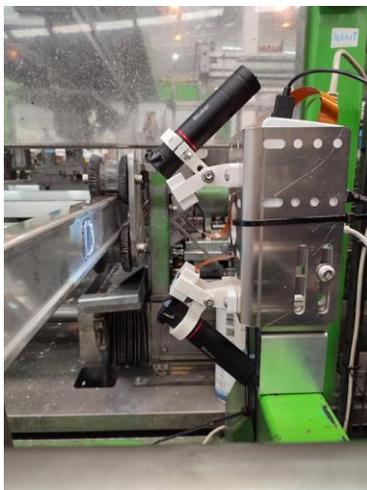


Figura 12 Soporte captura de imágenes

Los soportes contienen las dos cámaras encargadas de la fotografía, conectadas a una Raspberry Pi 5, que ejecuta el software para la captura de imágenes. Las cámaras están encapsuladas para evitar riesgos de rotura y asegurar una mayor capacidad de movimiento. Además, se han incluido focos de luz que ayudan a aumentar la calidad de la captura de imagen.

4.2.1.2 Criterio de control del módulo de expandido

Se establece un criterio de control de calidad para el resultado de las operaciones de conformado del módulo de expandido. Como se explicó en el capítulo anterior, dependiendo del tipo de aluminio, las marcas realizadas por el expandido pueden variar. Para ello, se establecen tres categorías de marcas.

1. **Marcas completas:** Los dedos expansores encargados de realizar estas marcas completan al 100% su apertura, dejando en el interior del peldaño marcas en forma de 'L' de igual forma en cada lateral que conforma la esquina del peldaño. La figura 13 muestra las marcas completas tras la operación de expandido.



Figura 13 Marcas expandido completas

Este tipo de marcas es el resultado ideal de la operación de expandido de la máquina de escaleras.

2. Marcas incompletas: Como se menciona anteriormente, las marcas pueden realizarse incorrectamente dependiendo de las características del tipo de aluminio. La figura 14 muestra las marcas realizadas incorrectamente tras la operación de expandido.



Figura 14 Marcas expandido incompletas

Se observa en la figura 14 que las marcas no están realizadas en su totalidad. Durante el estudio del criterio a asignar, se comprobó que la frecuencia de aparición de este tipo de marcas era muy habitual. En el 95% de los casos, el peldaño no producía movimiento, siendo la mayoría de las escaleras completamente funcionales y aptas para su uso.

Por tanto, se asigna un criterio de revisión a este tipo de serie de marcas. Se avisará al operario del estado del peldaño para una posterior revisión, ya que no se asegura con una fiabilidad del 100% que este tipo de marcas no afecten al movimiento del peldaño.

3. Ausencia de marcas: Esta categoría se produce por un fallo total del módulo de expandido, como la rotura de los dedos expansores o defectos puntuales del cilindro encargado de su apertura. La figura 15 muestra la ausencia de marcas tras un fallo en el módulo de expandido.



Figura 15 Resultado fallo total módulo de expandido

4.2.1.3 Criterio de control del módulo de remachado

Se establece a su vez un criterio de control de calidad para el resultado de las operaciones de conformado del módulo de remachado. Los resultados de la operación del módulo de remachado son fácilmente visibles, clasificándose en tres categorías:

1. Defectos menores: Un defecto en la operación de remachado puede provocar una grieta o un pequeño desperfecto en el remache. Este tipo de defecto, fácilmente detectable, se produce normalmente al cambiar el tipo de aluminio que alimenta la máquina. La figura 16 muestra el resultado de un remache realizado incorrectamente.



Figura 16 Grieta remache

2. Defectos mayores: Un fallo en la operación de remachado puede resultar en la destrucción del peldaño o en grietas que lo doblan, dejándolo inservible. Estos defectos son fácilmente detectables por el operario. La figura 17 muestra un remache realizado incorrectamente.



Figura 17 Remachado incorrecto

3. Operación correcta: La operación correcta del remachado asegura que el peldaño esté firmemente fijado al larguero, incrementando la seguridad y evitando el movimiento del peldaño. La figura 18 muestra un remache realizado correctamente.



Figura 18 Remachado correcto

4.2.2 Detección de objetos en imágenes

En esta subsección se expone el desarrollo de un modelo utilizando la herramienta YOLO v8, para la detección de objetos en las imágenes capturadas anteriormente. Se utiliza como criterio de detección el criterio de control detallado en el apartado anterior, con la creación de clases de objetos que detectará el modelo desarrollado.

4.2.2.1 Elaboración del conjunto de datos

Para desarrollar el modelo de detección de objetos mediante YOLO v8, es imperativo obtener un *dataset* fiable. Para lograr esto, se capturan series de imágenes de multitud de peldaños diferentes, realizando una clasificación conforme el criterio de control establecido en la sección de captura de imágenes.

Se capturan imágenes de peldaños durante varios turnos de trabajo de 8 horas, utilizando aluminios tipo AW6060 y AW6063. A lo largo de 12 meses, se recopilaron todas las imágenes de peldaños con defectos producidos por la máquina de escaleras, aumentando así la fiabilidad y precisión del modelo. El *dataset* final consta de 533 imágenes, detalladas en el Anexo 4.

El objetivo del modelo es detectar con alta precisión la calidad del producto conformado por la máquina de escaleras. Se desarrolla un único modelo de detección, capaz de identificar tanto las marcas realizadas por el módulo de expandido como el remache realizado por el módulo de remachado. El modelo clasifica los remaches y marcas como adecuados o inadecuados e informará al operario sobre los resultados de la inferencia.

4.2.2.2 Etiquetado de imágenes

Se utiliza la herramienta *Label Studio* para etiquetar las imágenes y definir las clases de objetos que el modelo detectará.

Se establecen cuatro clases de objetos, basadas en el criterio de control de calidad definido previamente. Las clases de objetos que definen el criterio de control de calidad de las operaciones del módulo encargado del expandido son:

- Serie de 3 marcas detectadas de forma completa: Se asigna a aquellas imágenes donde aparecen las tres series de marcas completas mediante un cuadro de etiquetado. Esta clase se define en el modelo como *Expandido_OK*.
- Serie de 3 marcas detectadas de forma incompleta: Se etiquetan en las imágenes mediante un cuadro de etiquetado, asignando el tipo de clase incompleta, que necesitan revisión por parte del operario. Esta clase de objeto se define en el modelo como *Expandido_CHECK*.

La clase de *Expandido_CHECK*, engloba todas aquellas marcas realizadas incorrectamente, incluyendo el resultado de las marcas del expandido prácticamente inexistente. Se decide crear únicamente esta clase, sin distinguir entre la ruptura del peldaño por un expandido incorrecto y la realización incompleta del expandido. Esto se debe a que, en caso de producirse algún problema con los dedos expansores, produciendo la ruptura de parte del peldaño, será necesaria la revisión por parte del operario. Asimismo, si las marcas no se han realizado de forma idónea, podría derivar en movimiento del peldaño respecto al larguero, requiriendo también la revisión por parte del operario. Por tanto, para aumentar la eficacia de detección del modelo, no se distingue entre estas dos situaciones. Se crean y etiquetan todas las imágenes, clasificando las series de marcas en los dos tipos de clases de objetos mencionados.

Además, se decide realizar el etiquetado de las tres marcas como un único objeto, ya que, en caso de etiquetarlas marca por marca, se perdía efectividad en la detección. Las dos marcas situadas

en la parte más interna del peldaño tienen características muy similares, mientras que la marca más externa, es diferente en tamaño y profundidad. Esto genera una colisión de marcas en el modelo de prueba, donde no se detectan las marcas más externas.

El modelo también detecta de forma simultánea el estado del remache realizado por el módulo de remachado. La detección del estado del remache consta de dos tipos de clases, siguiendo un criterio de control de calidad similar al de las marcas del expandido:

- Remachado realizado de forma correcta: El etiquetado se realiza de forma general, definiendo como objeto a detectar todo el remache realizado adecuadamente. Esta clase de objeto se define en el modelo como *Rivet_OK*. La figura 19 muestra el etiquetado del remache y expandido realizado correctamente en Label Studio.

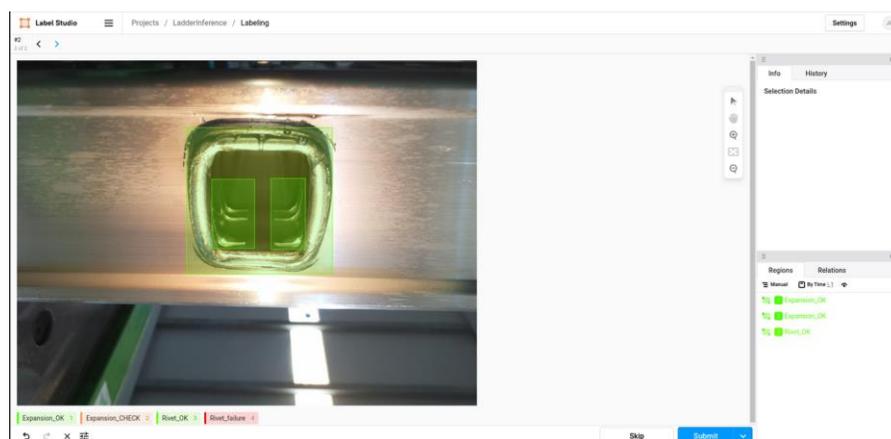


Figura 19 Etiquetado remache y expandido correcto

- Remachado realizado de forma incorrecta: Según el criterio de control, el defecto producido por un remachado incorrecto resulta en una grieta o defecto en el peldaño, provocando su ruptura parcial. Se define como objeto el defecto producido en el remache. Esta clase de objeto se define en el modelo como *Rivet_failure*. La figura 20 muestra el etiquetado del remache y expandido realizado incorrectamente en Label Studio.

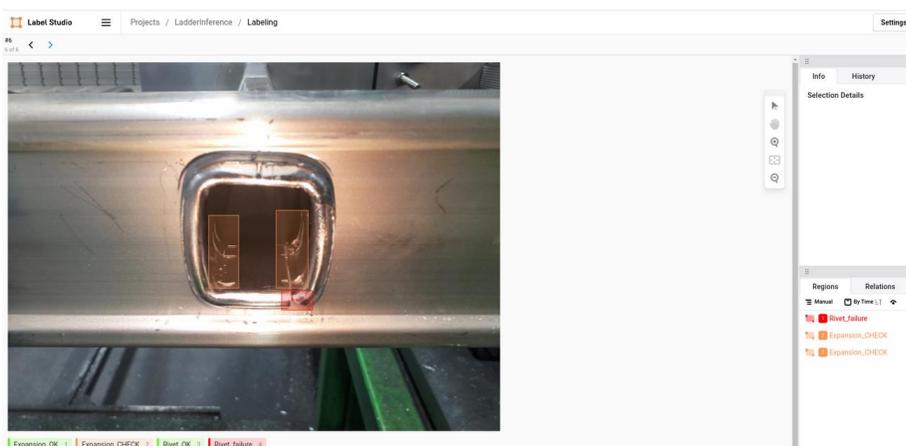


Figura 20 Etiquetado remache incorrecto

Se establecen las clases de objetos a detectar por el modelo, realizando el etiquetado en cada imagen. Las imágenes etiquetadas se exportan en formato YOLO v8, obteniendo las imágenes etiquetadas y unos ficheros en formato .txt, que contienen el tipo de clase de cada objeto etiquetado en la imagen junto con las coordenadas del cuadro rectangular de la etiqueta en la imagen.

4.2.2.3 Entrenamiento del modelo de detección

Para la creación del modelo entrenado de detección de objetos en imágenes, es necesario realizar un entrenamiento mediante la herramienta YOLO v8.

Se realiza un único modelo, encargado de la detección de marcas producidas por el módulo de expandido y de la detección del estado del remache producido por el módulo de remachado, utilizando el conjunto de datos previamente etiquetado.

Se utiliza para realizar el modelo YOLO v8 Nano, el más pequeño de los modelos disponibles. Este modelo presenta los menores requisitos y sus características proporcionan un aumento de velocidad de procesamiento e inferencia. El modelo YOLO v8 Nano consta de 3.2 millones de parámetros y se ejecuta en una CPU virtual. (Ultralytics, sf)

Además, se emplea el modelo basado en la predicción de objetos que, durante entrenamiento, aumenta la fiabilidad en la detección de objetos. Se establecen los requisitos hardware necesarios para realizar el entrenamiento, con el objetivo de reducir el tiempo de inferencia. Los hiperparámetros del modelo YOLO v8 Nano se fijan de forma predeterminada. El tamaño de las imágenes de entrada se reduce a 640x640 frente a una captura original de 2839x2128, para mejorar la velocidad de procesamiento y la detección de objetos en imágenes.

Se asigna el número de *epochs* a 20 para reducir el tiempo de entrenamiento. Durante cada *epoch*, el modelo se entrena con todos los datos disponibles, actualizando sus parámetros internos para mejorar su capacidad de hacer predicciones precisas.

Se configura el tipo de modelo sin definir parámetros específicos que afecten a la posterior detección del modelo, por lo que no se especifica un umbral de confianza en la configuración. Esto se debe a la necesidad de una configuración segura, ya que las operaciones de conformado de la escalera tienen un rango de operación muy amplio, dependiendo de las características del tipo de aluminio. En consecuencia, se decide aumentar la sensibilidad del modelo, evitando ajustar el umbral de confianza y reduciendo el riesgo de perder detecciones más sutiles.

Finalmente, tras la realización del entrenamiento, se obtiene una matriz de confusión que indica los porcentajes de acierto y confianza en la detección de objetos. La figura 21 muestra los resultados tras el entrenamiento del modelo mediante una matriz de confusión.

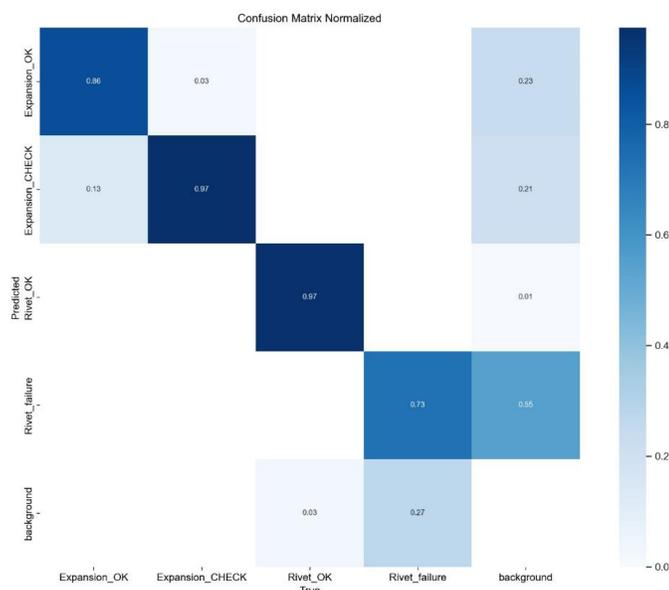


Figura 21 Matriz de confusión

Tras el entrenamiento, se obtuvo una matriz de confusión que muestra un nivel de precisión del modelo superior al 70% en las cuatro clases de objetos. Los resultados específicos son los siguientes:

- El modelo detecta con un 86% de acierto los peldaños correctos (*Expansion_OK*), confundiendo únicamente un 14% con un resultado incompleto del expandido que requiere revisión, debido a que el límite en muchas ocasiones es mínimo. Por otro lado, el modelo detecta con un 97% de fiabilidad las marcas que deben ser revisadas por el operario, siendo esta la clase más prioritaria a detectar.
- El modelo detecta con un 97% de acierto el remache realizado correctamente. Finalmente, se detectan con un 73% los errores producidos en el remache, que provocan grietas o la destrucción del peldaño. El porcentaje disminuye debido el tipo de error producido abarca multitud de roturas de diferentes tamaños y formas.

4.2.3 Integración a nivel industrial

En esta subsección se describe la integración del sistema de visión y análisis tanto mecánicamente como a nivel de control. Además, se explican los factores externos que afectaron al desarrollo de la solución y como fueron solucionados.

4.2.3.1 Factores externos al desarrollo de la solución

El factor más determinante que afecta a la captura de imagen es la vibración. Las operaciones de conformado de la máquina de fabricación de escaleras producen una vibración considerable en toda la máquina. El módulo encargado del remachado es la operación de conformado que mayor

vibración produce. Este inconveniente afecta la realización de las fotografías, ya que el enfoque se ve alterado debido al movimiento de todo el soporte del modelo.

Para solucionar estos problemas de vibración y movimiento en la lente de la cámara se utilizó el autoenfoco. Este se fuerza como parámetro en la cámara, asignando la zona de la imagen que contiene las marcas y el remachado. El auto enfoque se realiza previamente al instante de captura de la imagen, para evitar todo tipo de vibraciones o un enfoque incorrecto. La figura 22 muestra la evolución del enfoque en la captura de imágenes.



Figura 22 Captura de imagen tras auto enfoque

El objetivo del módulo de captura de imágenes es captar los resultados de las operaciones de conformado de la máquina de escaleras de forma clara, obteniendo una imagen de alta calidad. Para lograr esto, se añade en la parte superior e inferior del soporte de los módulos de captura un espacio para unos focos de luz que ayudan a destacar la parte que necesitamos captar de forma clara. El objetivo de los focos de luz es resaltar las marcas realizadas en el peldaño por el módulo de expandido y el remache realizado por el módulo de remachado, destacando la solución que se desea capturar sobre el resto del peldaño. La figura 23 muestra el foco de luz de la cámara superior.



Figura 23 Foco de luz de la cámara superior

Otro factor determinante para la captura de imagen es el reflejo producido por el aluminio. El aluminio es un material que refleja el 80% de la luz que incide sobre su superficie. (Weerg, 2020)

Este reflejo es captado por las cámaras, provocando una pérdida de enfoque y nitidez en las imágenes capturadas. Para mejorar la calidad de la imagen, captando de forma clara las marcas y el remachado y evitando el reflejo del material, se ajustan los parámetros de la cámara que afectan a la cantidad de luz capturada. La figura 24 muestra la luz incidente por el foco de luz en el peldaño.



Figura 24 Luz incidente en el peldaño

Se reduce la cantidad de luz capturada por la cámara asignando una ganancia ISO de valor 150. Además, se reduce el tiempo de exposición de la cámara mediante un valor de apertura f20. La luz incidente en el peldaño es muy elevada, destacando las marcas realizadas por el expandido y el remachado. Al reducir la cantidad de luz capturada por la cámara alrededor del peldaño, este se oscurece en la imagen capturada, manteniendo la zona central clara y óptima para la captura de imágenes, destacando sobre el resto del peldaño.

Sin embargo, en la parte exterior del peldaño y junto al borde del larguero, se produce un reflejo considerable de la luz incidente, distorsionando la parte superior e inferior del peldaño y afectando su captura óptima. La figura 25 muestra la evolución de la disminución del reflejo producido en el larguero.



Figura 25 Reflejo producido en el larguero

Como solución al reflejo producido en el borde del larguero, se colocan unas tapas en las lentes de los focos de luz, lo que reduce notablemente el reflejo entre el borde del larguero y el exterior del peldaño. Las tapas ocupan un 40% de la lente sin afectar a la calidad de la imagen del interior del peldaño.

4.2.3.2 Integración de la solución a la máquina de escaleras

Tras el estudio del funcionamiento de la máquina, se observa que, una vez finalizadas las operaciones de conformado de la escalera, la producción concluye dejando la escalera en el interior de la máquina. El operario debe desplazar la escalera al exterior de la máquina para almacenarla, lo que hace que los dos últimos peldaños sean inaccesibles para el módulo de captura.

Por tanto, la implementación del módulo de control de calidad debe situarse a continuación del módulo de remachado de la máquina de escaleras, permitiendo así la captura de la todos los peldaños excepto los dos últimos que permanecen en el interior de la máquina. La figura 26 muestra la zona de implementación del modelo en la máquina de escaleras.



Figura 26 Zona implementación del modelo

El objetivo del modelo es capturar los resultados de las operaciones de conformado de la escalera mediante dos cámaras, cada una encargada de capturar la parte superior y la parte inferior de cada peldaño. Para ello, se diseña un soporte que permite el movimiento en los tres ejes, adaptándose a la máquina de escaleras. Las características, detalles y medidas del soporte se especifican en el Anexo. La figura 27 muestra el soporte de captura de imágenes.

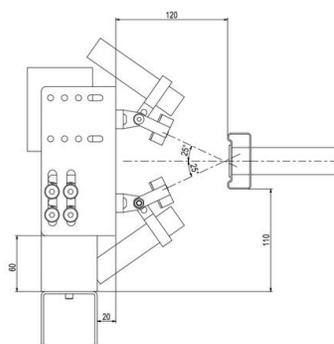


Figura 27 Soporte control de calidad

El soporte contiene la Raspberry Pi 5, conectada a las dos cámaras que realizan las fotografías, situadas con el mismo grado de inclinación respecto al centro del peldaño. Esto permite captar adecuadamente el interior del peldaño, logrando capturar las seis marcas inferiores y el remachado con una de las cámaras, y las seis marcas superiores y el remachado con la otra cámara.

Los módulos de captura están atornillados al soporte, respetando el grado de inclinación respecto al centro del peldaño. El departamento de I+D+I de Avanti Wind Systems fabricó mediante impresora 3D unas capsulas que contienen los módulos de captura, proporcionando protección y facilitando un ajuste correcto del grado de inclinación.

4.2.3.3 Conexión Automata – Raspberry Pi 5

En esta subsección se explica cómo se realiza la conexión entre la Raspberry Pi 5 y el autómata, así como la comunicación entre ambos módulos, recibiendo señales internas del autómata programable desde la Raspberry Pi 5.

La conexión entre la Raspberry Pi 5 y el autómata de Siemens S7-1200, encargado del funcionamiento de la máquina, se establece mediante una conexión Ethernet. Por tanto, se configura la dirección IP de la Raspberry Pi 5 para que se encuentre en el mismo rango de subred que el autómata programable, es decir, compartan la misma parte de red y difieran el último octeto.

Además, se utiliza la librería Snap 7 para la comunicación entre el autómata y la Raspberry Pi 5, permitiendo la comunicación mediante lenguaje Python con el autómata programable. Esto conlleva la necesidad de conectarnos desde la Raspberry a la misma subred que el autómata y, es necesario por el protocolo de conexión de la librería Snap 7, el rack y el slot donde se encuentran la CPU de nuestro autómata programable. La figura 28 muestra el control de dispositivos de autómata programable. Más detalles específicos sobre la librería Snap 7 se encuentran en el Anexo.



Figura 28 Control de dispositivos autómata programable

Se decide realizar la comunicación mediante cable Ethernet y la librería Snap 7 para no modificar el programa cargado en el autómata. Se evaluaron otras opciones, como la comunicación programando clientes Modbus TCP en la Raspberry Pi 5 y un servidor Modbus TCP en el autómata, estableciendo la comunicación mediante peticiones Modbus. Otra opción era utilizar los módulos de entradas digitales de la Raspberry Pi 5, cableando directamente las salidas del autómata como entradas al módulo de entradas/salidas digitales de la Raspberry Pi 5. Sin embargo, estas opciones resultaban más complicadas, por lo que se optó por la conexión mediante cable Ethernet y la librería Snap 7.

Como se explica en secciones anteriores, es crucial evitar las vibraciones para lograr captar las imágenes con la mayor calidad posible. Esto se consigue mediante la comunicación entre la Raspberry Pi 5 y el autómata, recibiendo la señal adecuada del autómata para que el modelo realice las capturas y el análisis con fiabilidad.

Por tanto, se estudió el funcionamiento de la máquina mediante el programa cargado en el autómata con la herramienta TIAPortal. Se determinó el tiempo de ciclo de los procesos de remachado, la operación que genera la mayor cantidad de vibraciones, y el momento más idóneo para realizar las fotografías para evitar dichas vibraciones.

El proceso del módulo encargado del remachado consiste en:

- 1.- El peldaño alcanza la posición para la operación de remachado.
- 2.- 0.5 segundos de espera.
- 3.- El cilindro de la remachadora avanza hasta entrar en contacto con un fin de carrera.
- 4.- Se realiza la operación de remachado, produciendo mucha vibración (0.5 segundos).
- 5.- El cilindro retrocede hasta entrar en contacto con el fin de carrera de reposo (0.5 segundos).
- 6.- Tiempo de espera hasta la finalización de otras operaciones (2.5 segundos).

Por tanto, el momento para realizar las fotografías es tras el proceso de remachado, cuando el cilindro ha retrocedido y se encuentra en la posición de reposo. La señal utilizada para realizar la fotografía es la del sensor fin de carrera. La figura 29 muestra el fin de carrera de la remachadora.

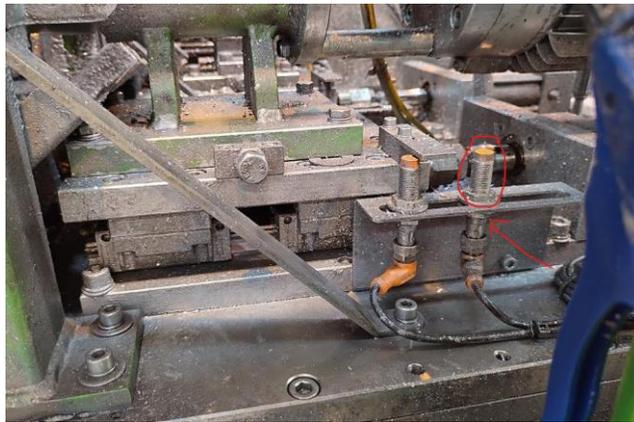


Figura 29 Fin de carrera remachadora

La distancia entre el módulo de control de calidad y el módulo encargado de la operación de remachado del peldaño es considerable, equivalente a la distancia entre dos peldaños. Por lo tanto, cuando el módulo de remachado está realizando la operación sobre el tercer peldaño, el módulo de control de calidad realiza las fotografías del primer peldaño.

Se leen como señales de entrada del autómeta el valor del contador de peldaños remachados, que se incrementa cuando se realiza la operación de conformado del módulo de remachado, y el valor del número de peldaños a realizar por la máquina. El proceso de lectura mediante la librería Snap 7 se realiza leyendo la dirección de memoria donde se almacenan los valores del estado de las señales.

- Sensor fin de carrera (Reposo remachadora), asignado a la dirección %M105.7, de tipo booleano.
- Contador de peldaños remachados, asignado a la dirección %MW324, de tipo Word.
- Valor asignado de número de peldaños a conformar, asignado la dirección %MW324, de tipo Word.

4.2.3.4 Desarrollo del Script

Se programa en la Raspberry Pi 5 un archivo en lenguaje Python encargado del control para la adquisición de las imágenes y la inferencia del modelo, clasificando los resultados según el criterio de control establecido. La figura 30 muestra el diagrama de flujo del desarrollo del script de captura y análisis de imágenes.

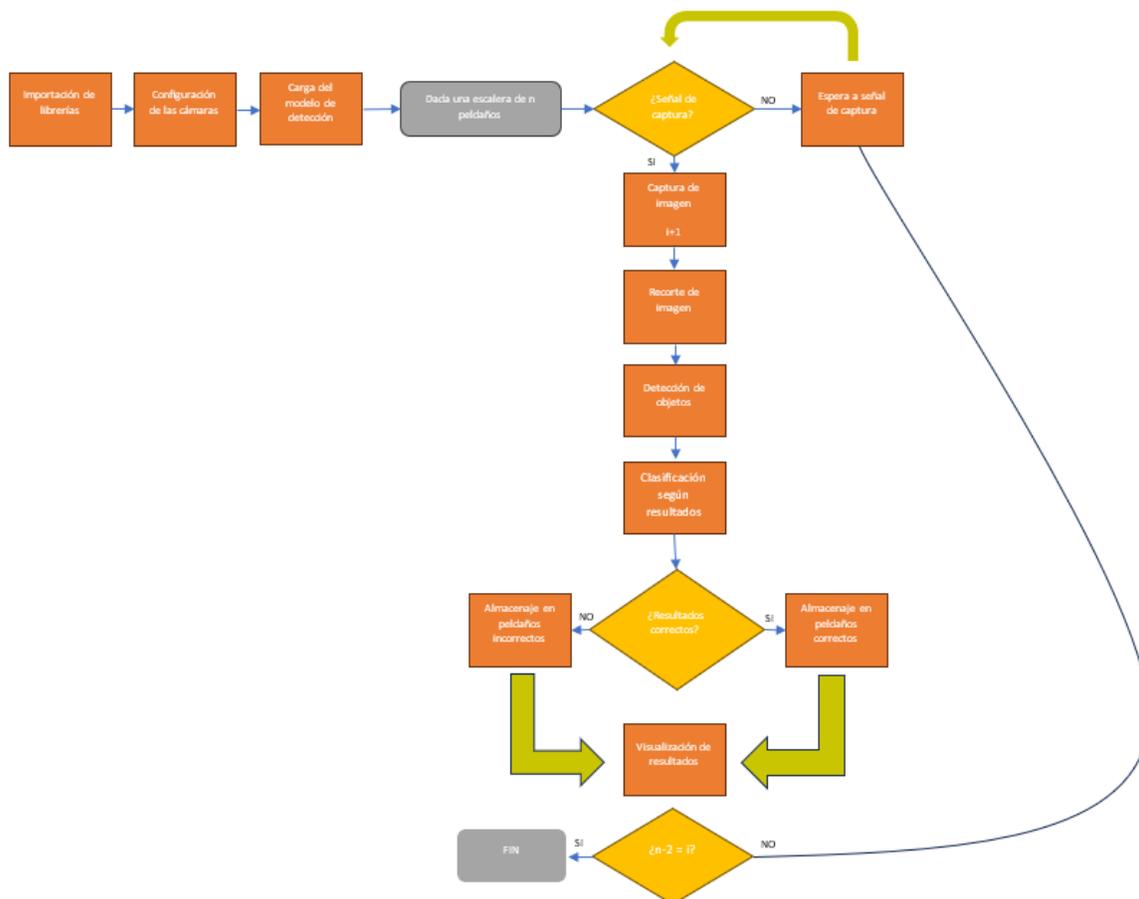


Figura 30 Diagrama de flujo del script

El modelo opera de forma cíclica. Con el flanco de subida de la señal recibida del autómata, se realiza la captura de imágenes de las partes superior e inferior de cada peldaño, seguido de su recorte, análisis y clasificación. El proceso se repite para cada peldaño hasta alcanzar el número de peldaños a producir, según lo indicado por el autómata, exceptuando los dos últimos. Esto se debe a que el funcionamiento y la programación de la máquina de escaleras dejan los dos últimos peldaños en posiciones no accesibles para el modelo desarrollado.

Uno de los objetivos principales de la solución es informar al operario de los resultados de las operaciones de conformado de la escalera. Para facilitar esta visualización, se implementa una tablet que muestra los resultados. Se establece una conexión WiFi entre la Raspberry Pi 5 y la tablet. Utilizando la librería Flask, se desarrolla una aplicación web sencilla en lenguaje Python

donde la Raspberry envía el contenido del archivo results.txt, que almacena los resultados de la inferencia de cada peldaño de la escalera. Para establecer la conexión, la tablet y la Raspberry deben estar conectadas a la misma red. La aplicación web permite al operario observar los resultados de forma rápida y sencilla. La interfaz de la tablet consta de dos aplicaciones web, cada una mostrando el estado de cada peldaño de la escalera en cada lateral del larguero.

4.2.3.5 Estudio del tiempo de ciclo

Finalmente, se realizó un estudio del tiempo que necesita la Raspberry Pi 5 para realizar todos los procesos de captura y análisis de la imagen. El tiempo máximo entre peldaños es de 6.8 segundos. Considerando un margen de seguridad, la Raspberry Pi 5 debe realizar todos los procesos en un tiempo inferior a 6 segundos. Este hecho hizo necesario realizar un estudio del tiempo de ciclo, dividido en varias subsecciones:

- Tiempo máximo en lograr el enfoque de las dos cámaras: El valor máximo es de 3 segundos.
- Tiempo máximo de captura de imagen de ambas cámaras: El valor máximo es de 0.2 segundos.
- Tiempo de recorte de las imágenes capturadas: El valor máximo es de 0.2 segundos.
- Tiempo de inferencia en la detección de objetos en la imagen recortada mediante el modelo previamente cargado: El valor máximo es de 1.4 segundos.
- Tiempo de anotación y etiquetado de los objetos en la imagen analizada: El valor máximo es de 0.2 segundos.
- Tiempo de clasificación y almacenaje de las imágenes anotadas y los resultados de la inferencia: El valor máximo es de 0.2 segundos.

Se toman varias decisiones para garantizar el cumplimiento del tiempo de ciclo:

1. Realizar el autoenfoco una única vez.
2. Reducir el tamaño de las imágenes para la detección de objetos.
3. Utilizar un único modelo de detección.
4. Realizar las capturas de imagen en formato array.

Realizar el autoenfoco de ambas cámaras para cada peldaño, previo a la captura de imagen, mejora la calidad de la imagen, pero el tiempo de autoenfoco de ambas cámaras alcanzó en algunas ocasiones los 10 segundos, interrumpiendo el ritmo de producción. Se determina que el autoenfoco debe realizarse una única vez al comienzo de ciclo, manteniendo la zona de píxeles donde debe enfocar la cámara durante toda la producción.

Además, tras el estudio del funcionamiento del script, se identifica el tiempo de inferencia en la detección de objetos en las imágenes como la operación de mayor coste computacional. Para reducir este tiempo, se decide recortar previamente las imágenes, reduciendo así la carga computacional de 1.2 segundos a 0.7 segundos. Cabe destacar la utilización de un único modelo de detección, ya que la inferencia de dos modelos en dos imágenes aumenta el tiempo de ciclo de la operación de inferencia de 1.4 a 3 segundos.

Finalmente, se realiza la captura de imagen en formato array para reducir el coste computacional. Se comparan los tiempos de ciclo: capturar una imagen en formato jpeg requiere

un tiempo de ciclo de 0.8 segundos, mientras que la captura en formato array requiere solo 0.2 segundos.

La suma total de los tiempos de ciclo máximos de cada operación es de 5.2 segundos. Por tanto, tras los cambios realizados, se observa que la suma de todos los procesos realizados por la Raspberry Pi 5 es inferior al tiempo de ciclo permitido, logrando que el modelo opere con alta fiabilidad y flexibilidad.

Cabe destacar que los tiempos se registraron durante el desarrollo del script en un turno de trabajo de 8 horas, donde la Raspberry estaba trabajando a temperaturas elevadas tras 8 horas de captura e inferencia de imágenes. Además, para el estudio del tiempo de ciclo máximo, se registraron todos los tiempos máximos.

CAPÍTULO 5 CONCLUSIONES

El trabajo fin de grado plantea varias soluciones al problema de la automatización de una máquina de fabricación de escaleras. Se desarrolla una solución de tipo comercial, mediante prototipos implementados por empresas externas a Avanti Wind Systems, y una solución no comercial desarrollada por medios propios. Se logra implementar un control de calidad del producto conformado por la máquina de fabricación de escaleras.

El desarrollo de la solución ad hoc permite, mediante la captura de imágenes y análisis de objetos, la detección y clasificación automatizada del resultado de las operaciones de conformado, garantizando la seguridad para el uso del producto final.

Dado que la seguridad es de vital importancia en el sector eólico, la programación de un modelo capaz de detectar los defectos de fabricación en los peldaños de la escalera proporciona una mejora en el control del producto y un aumento en la fiabilidad durante el uso.

En el presente trabajo se han expuesto dos soluciones, una desarrollada por medios propios y otra comercial. La solución comercial ofrece mayor facilidad de implementación y adaptabilidad al entorno, además de una fiabilidad elevada, pero con unos costes muy elevados. En cambio, la solución desarrollada por medios propios alcanza altos porcentajes de fiabilidad, obteniendo imágenes de alta calidad e implementando un modelo único y fiable, aunque adaptado a un entorno y material de trabajo específico.

La solución desarrollada por medios propios destaca por un rendimiento adecuado a los requisitos de Avanti Wind Systems, con una fiabilidad aceptable, reduciendo considerablemente los costes de desarrollo de 20.938 € a 453,18€. El desarrollo de la solución junto con el prototipo ha supuesto un coste total de 708,30 €.

En conclusión, la solución desarrollada por medios propios proporciona un modelo acorde a los requisitos e incidencias de Avanti Wind Systems, desarrollada con un presupuesto considerablemente menor, alcanzando una eficiencia similar respecto a la solución comercial, aunque con mayor limitación, adaptada a un entorno específico y cerrado.

ANEXOS

ANEXO 1 Descripción más detallada del funcionamiento de la máquina

A.1.1 Módulo de alimentación de largueros

La producción de escaleras comienza con la alimentación de largueros, que conformarán el producto final. La operación se inicia con la unidad de tracción colocándose a 3.5 metros del soporte móvil. Una vez introducidos los largueros, estos son detectados por los sensores de proximidad y la unidad de tracción comienza el proceso. La figura 31.A muestra el sistema de alimentación de largueros.

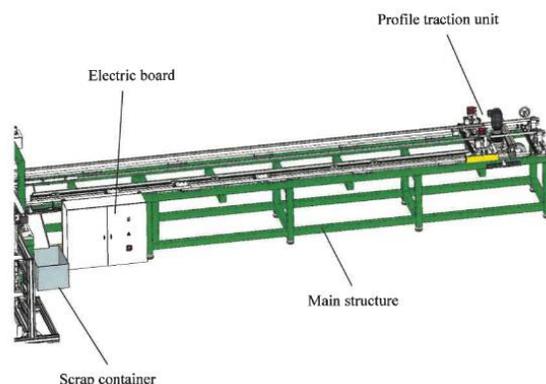


Figura 31.A Sistema de alimentación de largueros

La unidad de tracción se desplaza hasta la parte final de la estructura, encargándose del movimiento longitudinal de los largueros. En esta etapa se realizan los procesos de punzonado, inserción, expandido y remachado. Cuando la unidad de tracción se sitúa a 3 metros del soporte móvil interno, este se encarga del movimiento longitudinal de la máquina de escaleras hasta que se completen las operaciones de conformado de la escalera.

A.1.2 Módulo de punzonado

El proceso de conformado de la escalera comienza con el punzonado del larguero. La operación consiste en perforar el larguero y empujar la parte desechable a un contenedor situado en la parte inferior del sistema. La figura 32.A muestra el módulo de punzonado.

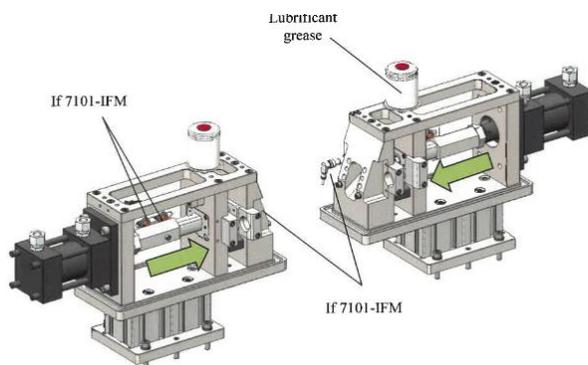


Figura 32.A Módulo de punzonado

La unidad está compuesta por un cilindro hidráulico regulado con dos sensores inductivos. A través del mandril interno y de las matrices externas, se realizan los agujeros en los largueros. Un sensor de proximidad detecta la pieza a punzonar. Para evitar la adhesión del material, es necesario lubricar en cada ciclo de punzonado.

A.1.3 Módulo de inserción de peldaños

La unidad de inserción de peldaños se compone de varias subunidades:

El elevador de peldaños que, de forma automática, inserta peldaños en el cargador de peldaños, de manera independiente al funcionamiento de la máquina de escaleras. El cargador cuenta con dos sensores de carga máxima y mínima, mientras los peldaños no alcancen el sensor de carga máxima, el elevador continúa en funcionamiento. La figura 33.A muestra el sistema elevador y cargador de peldaños.

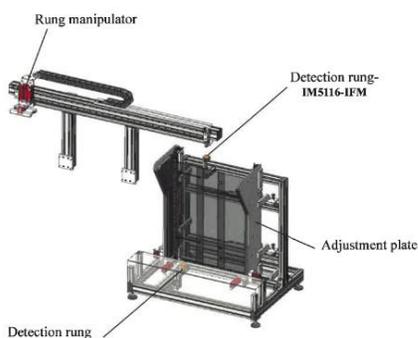


Figura 33.A Sistema elevador y cargador de peldaños

La segunda subunidad se encarga de la inserción de los peldaños en los agujeros realizados por el módulo de punzonado. Los peldaños se insertan siguiendo una guía hasta alcanzar el larguero posterior. La figura 34.A muestra la unidad de inserción de peldaños.

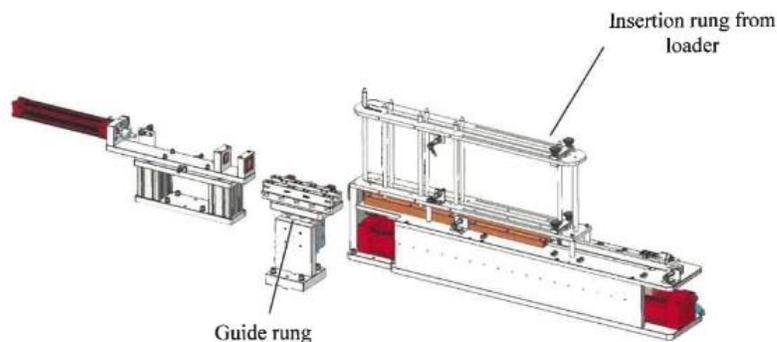


Figura 34.A Unidad de inserción de peldaños

A.1.4 Módulo de expandido

La operación de expandido se realiza mediante una unidad compuesta por dos cilindros hidráulicos: uno de ellos encargado de la expansión y otro del movimiento de las herramientas de expandido en el interior del peldaño. La figura 35.A muestra la unidad de expansión.

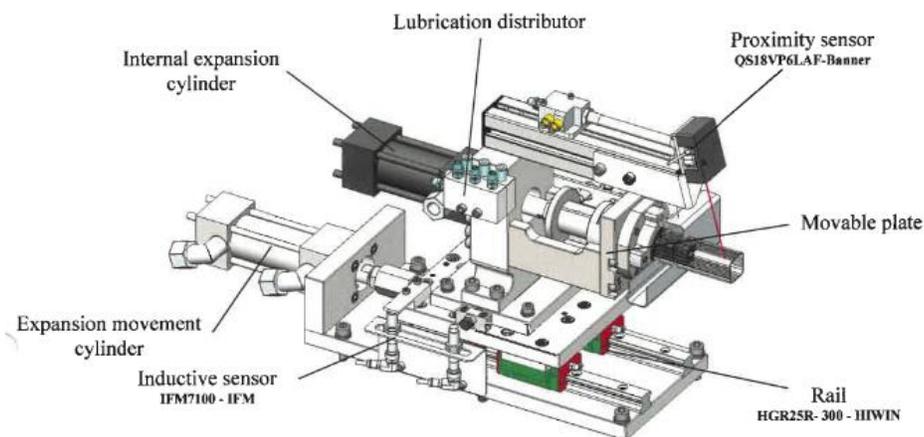


Figura 35.A Unidad de expansión

El proceso de expandido aumenta la resistencia del peldaño con los largueros, evitando su movimiento. Esta operación es la más relevante de la máquina de escaleras, siendo de vital importancia para asegurar la fijación del peldaño en la escalera. La figura 36.A muestra la unidad principal de expandido.

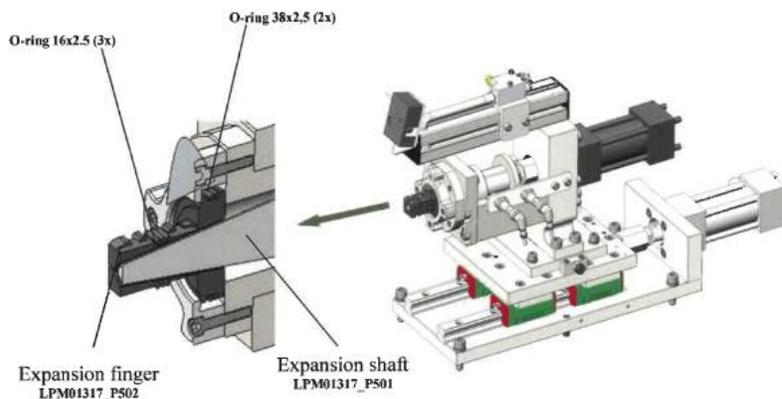


Figura 36.A Unidad principal de expandido

La unidad principal de expandido consta de dedos expansores que, mediante un cilindro, se abren para expandir el peldaño y asegurar su fijación.

A.1.5 Módulo de remachado

Esta operación es la última del proceso de conformado de la escalera. Se realiza después del proceso de expandido, aumentando la resistencia entre el peldaño y el larguero. La figura 37.A muestra la unidad de remachado.

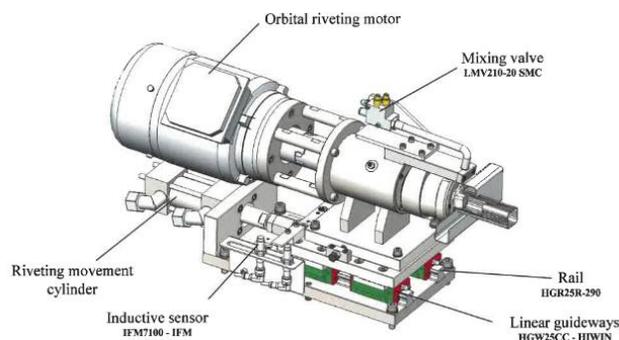


Figura 37.A Unidad de remachado

El proceso consiste en realizar un remache en la parte más exterior del larguero, mediante un cilindro hidráulico y un motor. Esta operación logra un remachado visualmente aceptable y añade resistencia para evitar el movimiento del peldaño en la escalera.

ANEXO 2 Hardware del modelo

A.2.1 Raspberry Pi 5

La Raspberry Pi 5 se utiliza como controlador para la adquisición de las imágenes y la inferencia del modelo.

Raspberry Pi 5 es la última versión desarrollada por Raspberry Pi Foundation. La Raspberry Pi es una serie de ordenadores monoplaca de bajo costo, enfocada para uso educativo y aplicaciones relacionadas con la robótica. Permite la conexión de cámaras capaces de capturar imágenes.

A.2.2 Raspberry Pi Camera Module 3

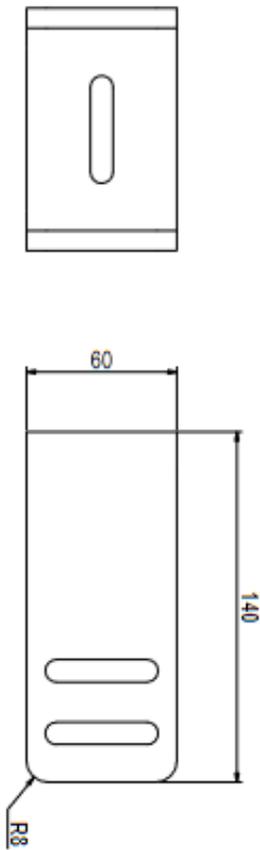
Raspberry Pi Camera Module 3 es una cámara compacta de Raspberry Pi. El módulo puede capturar imágenes de alta calidad, hasta 3 megapíxeles, y su funcionamiento es compatible con la librería libcamera, incluyendo funciones de enfoque automático.

Se decide utilizar este tipo de cámara para el desarrollo del modelo debido a sus funciones de enfoque automático, necesarias para mitigar el efecto de las vibraciones de la máquina de escalera, además de contar con una óptica mejorada respecto a versiones anteriores.

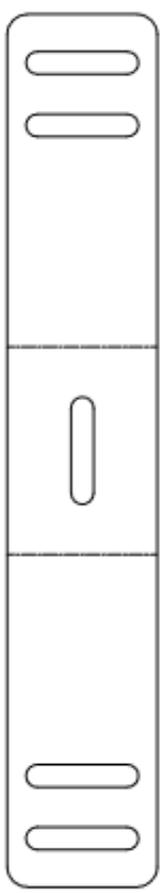
A.2.3 Soporte módulo de captura

Se exponen los planos del soporte realizado para el módulo de control de calidad, compuesto por dos cámaras que capturan imágenes del modelo, junto con sus focos de luz que mejoran la calidad de la imagen.

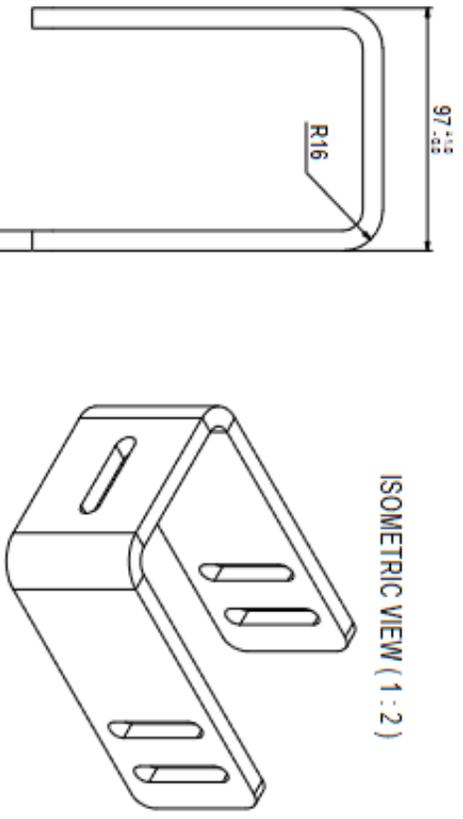
El soporte permite movimiento en 3 ejes, facilitando el posicionamiento de la solución.



UNFOLDED VIEW (1:2)



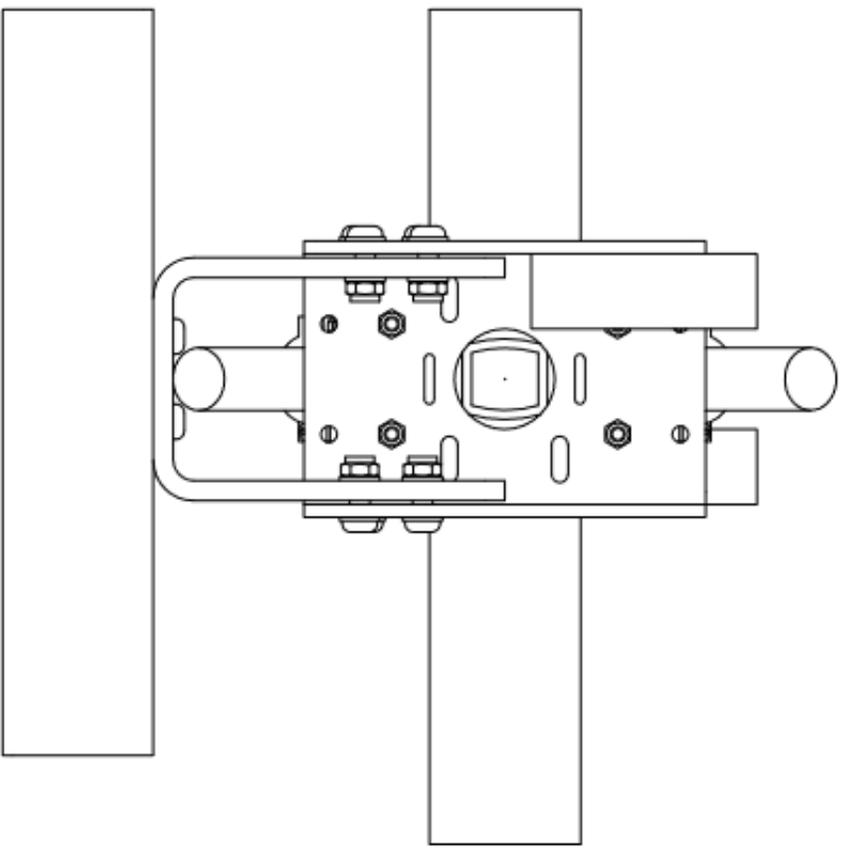
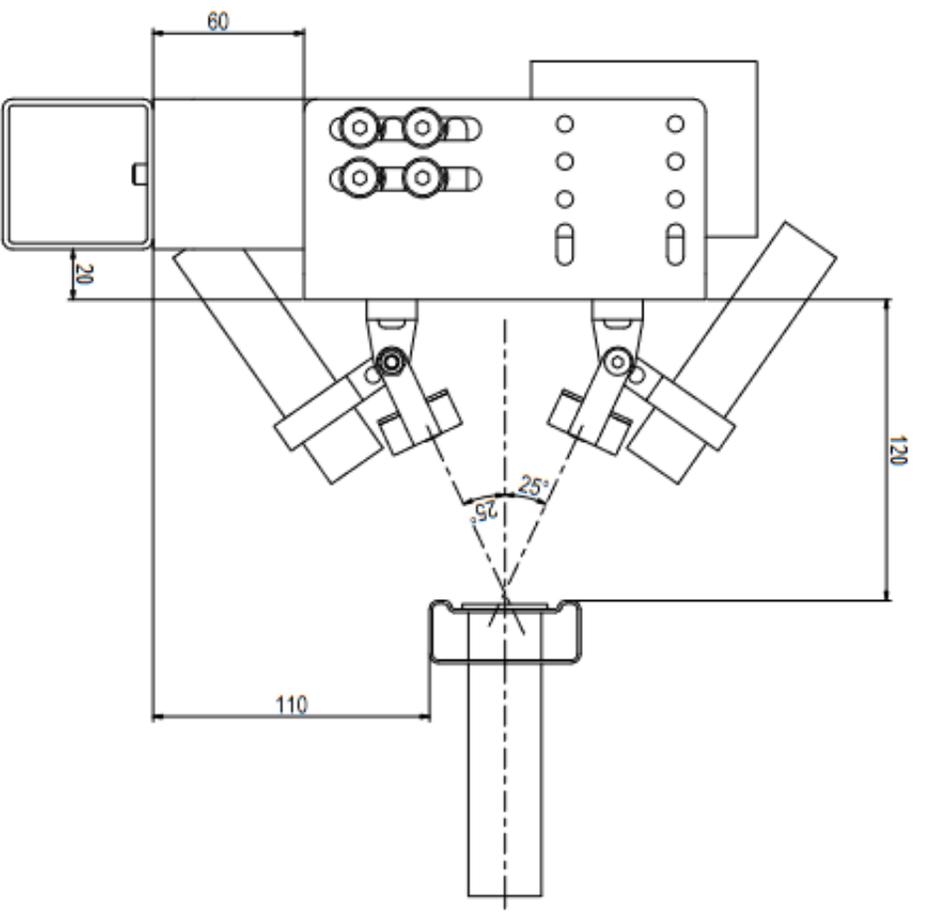
ISOMETRIC VIEW (1:2)



NOTE: Drawing is partially annotated, main dimensions and dimensions of holes and their relative position on each face shall be assured based on the unfolded view.

| | | | | | |
|---|--|---------------------------------|------------|-----------------------|-----------------|
| | | ISO 2768-MK NO TREATMENT | | A3 AW 5754H111 | |
| | | Scale: 1/1 | Sheet: 1/1 | Scale: Deformed | Weight: 0.41 kg |
| AVANTI | | Rev: 01 | Date: | Appr: | Release: |
| Increased height | | Change Description | | | |
| New document | | Ladder camera support 02 | | | |
| Ladder machine vision system | | Ladder machine vision system | | | |
| This document and its contents are reserved to AVANTI and its customers. Any reproduction, distribution, modification, copying, or alteration in any form is strictly prohibited without the express written consent of AVANTI. | | Part No: AD00008067 | | | |

Ladder machine control side - Setup 1



Position representation of both sides of ladder

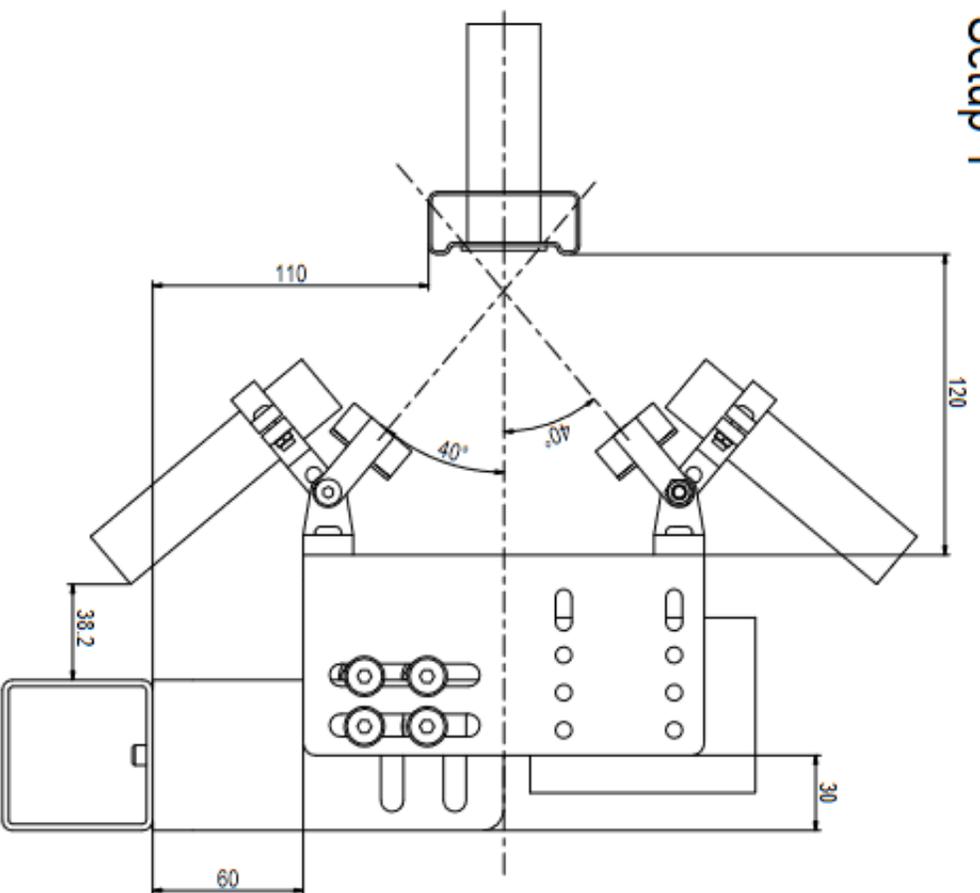
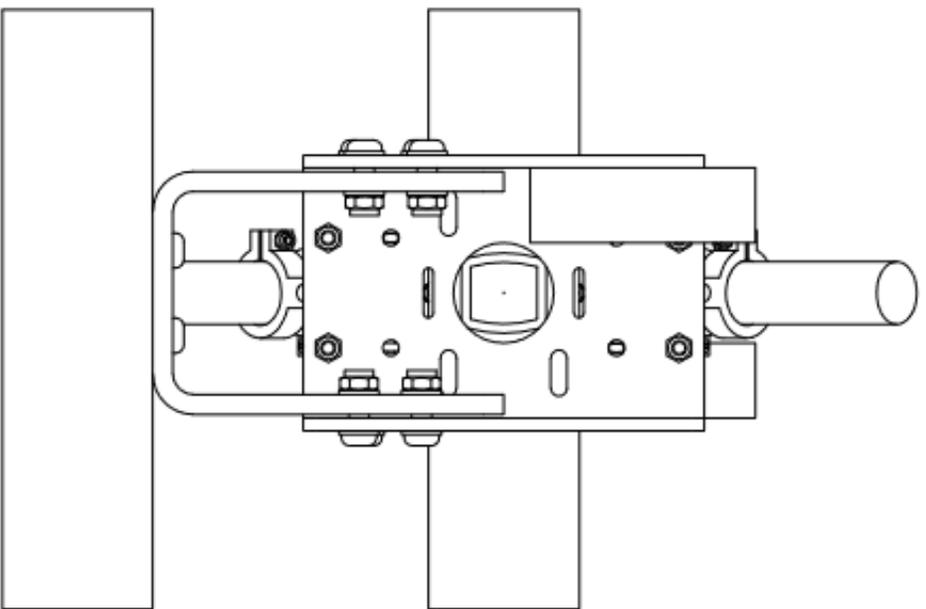
Change Description

| Rev. | Date | Appr. | Released | Change Description |
|------|------|-------|----------|--------------------|
| 02 | | | | |

| | | |
|--|--|--|
| Technical ISO 2768-MK Scale to 1/5 Scale A3 | Technical NO TREATMENT Scale Not applicable | Title Ladder machine assy with vision sys Project Ladder machine assy vision system Part no. N/A Part no. AD000008075 Part 02 |
|--|--|--|

This document and its contents are proprietary to AVANTI and are provided by AVANTI under a license agreement. AVANTI shall retain all rights in this document and shall not be held responsible for any damage or loss of data resulting from the use of this document. AVANTI shall not be held responsible for any damage or loss of data resulting from the use of this document. AVANTI shall not be held responsible for any damage or loss of data resulting from the use of this document.

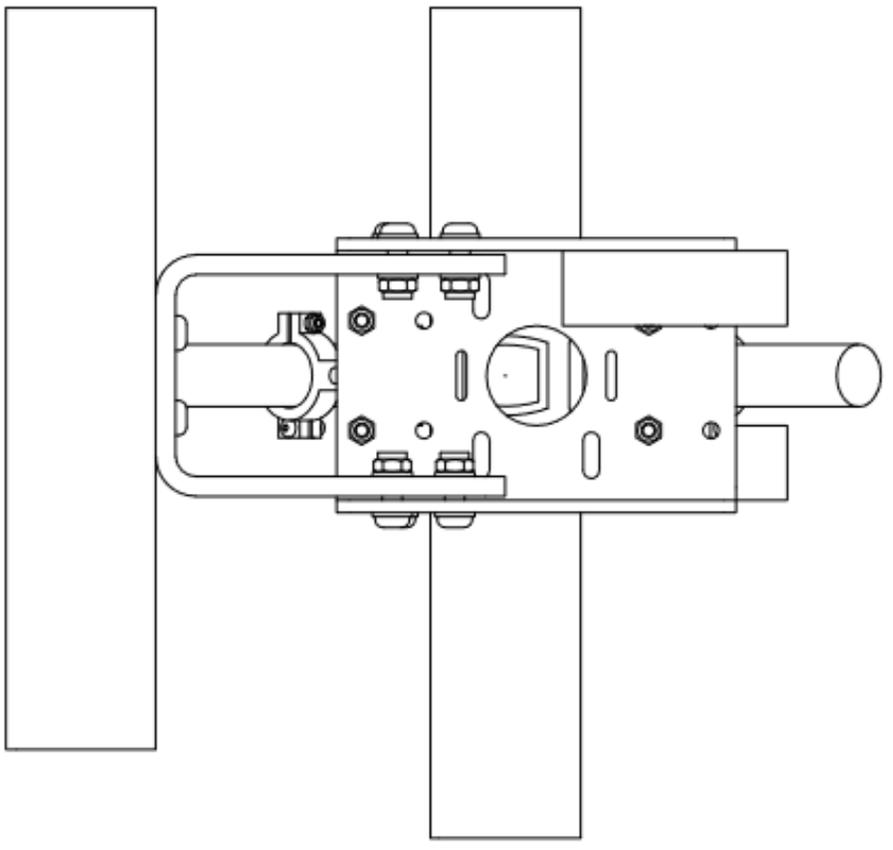
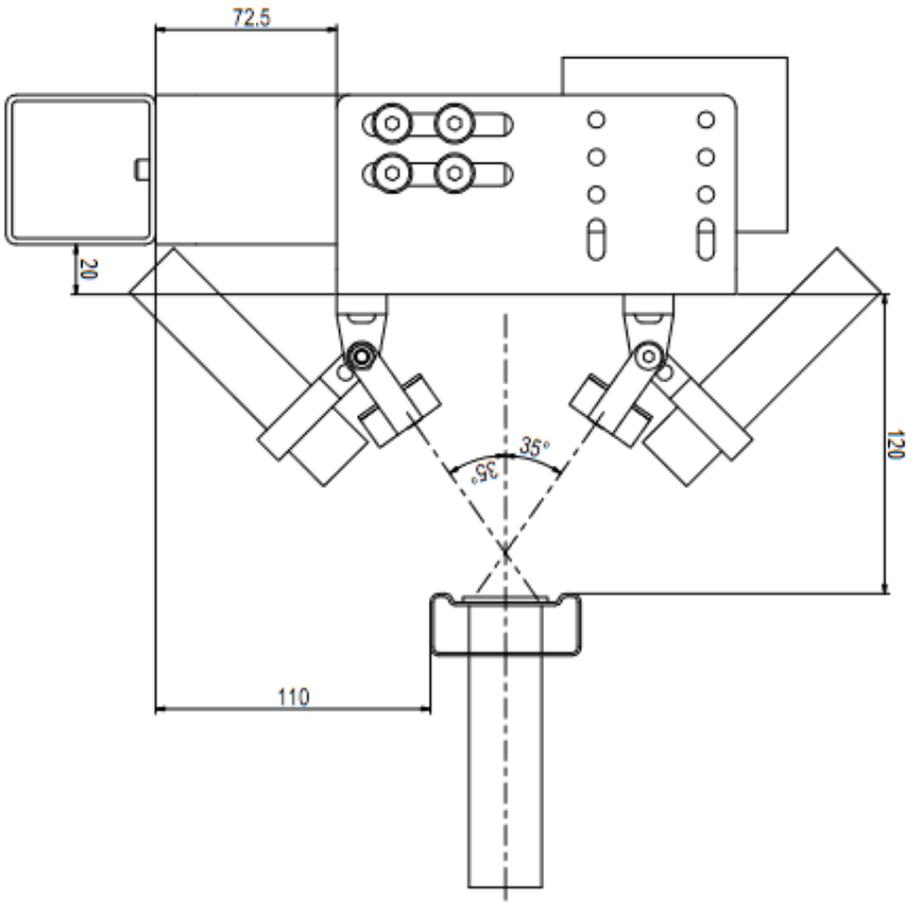
Ladder machine unloading side - Setup 1



| Standard | | Material | | Revision | | Change Description | |
|--|--------------|--|--|--|-------|--------------------|--|
| ISO 2768-mK | NO TREATMENT | Q2 | | Date | Appr. | Released | |
| Scale: 2/5 | Scale: A3 | Position representation of both sides of laser | | | | | |
| Title: Ladder machine assy with vision sys | | Scale: Not applicable | | Project: Ladder machine assy vision system | | | |
| Part No: AD000008075 | | Rev: 02 | | Ctg. No.: N/A | | | |

This document and its contents are proprietary to AVANTI and its affiliates. Any use, reproduction, or distribution of this document without the express written consent of AVANTI is strictly prohibited. AVANTI reserves the right to modify this document without notice. AVANTI shall not be held liable for any damages or losses resulting from the use of this document. AVANTI shall not be held liable for any damages or losses resulting from the use of this document.

Ladder machine control side - Setup 2

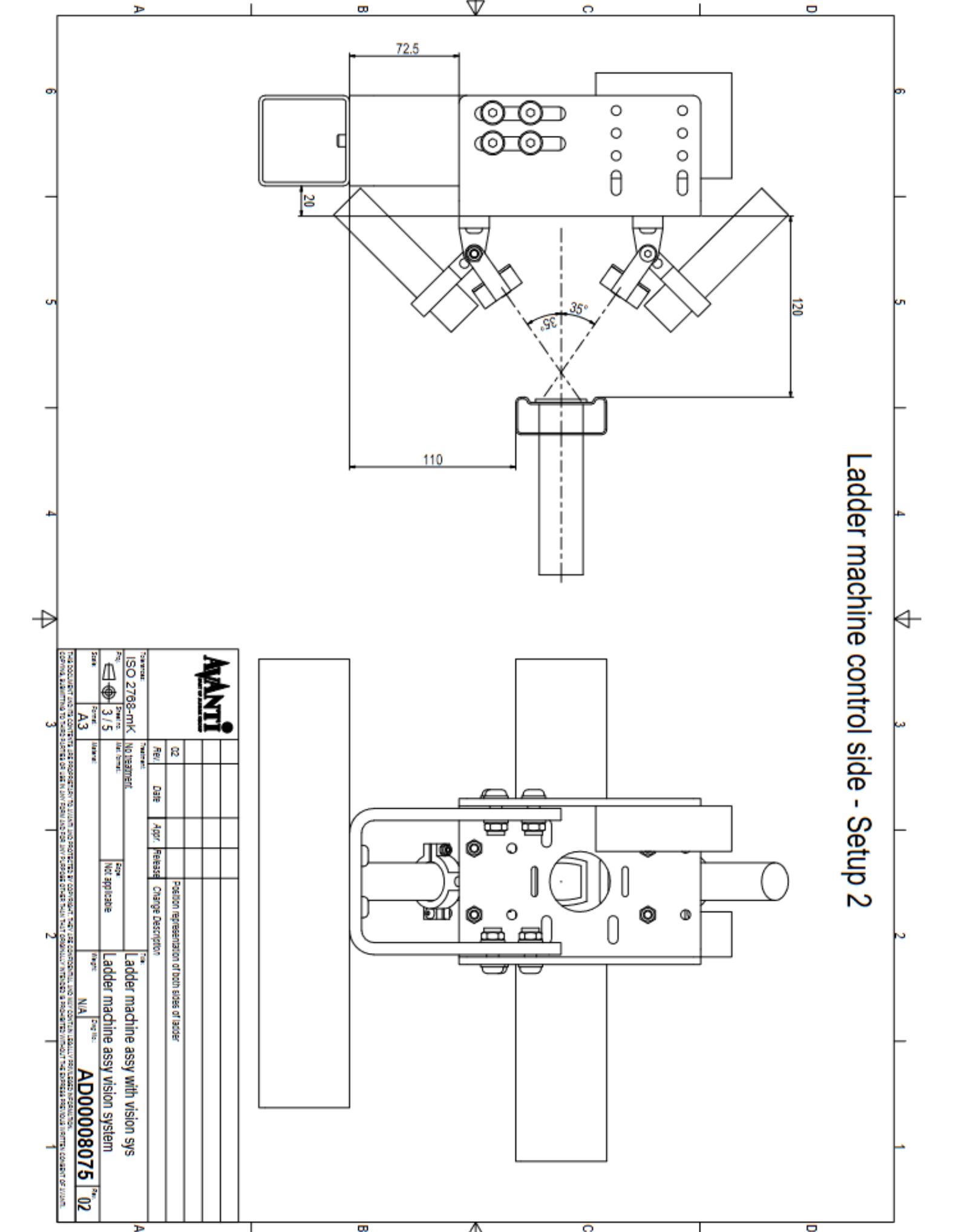


| Rev. | Date | Appr. | Released | Change Description |
|------|------|-------|----------|---|
| 02 | | | | Position representation of both sides of ladder |

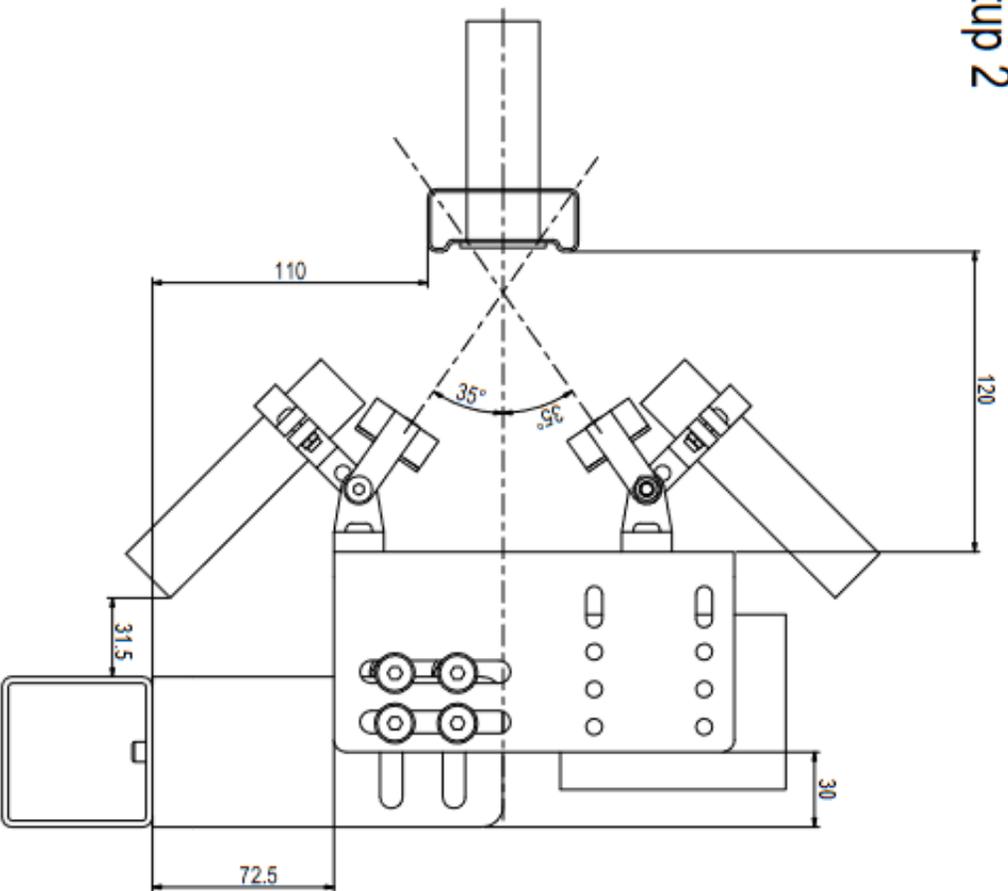
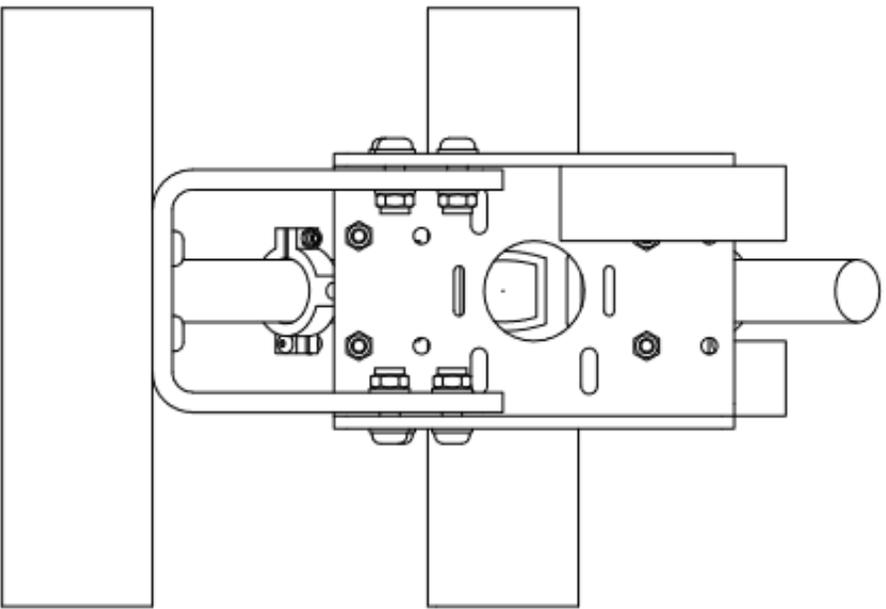
| | | |
|--|---|--|
| Character: ISO 2768-mK Scale: 3/5 Scale: A3 | Treatment: NO TREATMENT Size: Not applicable | Title: Ladder machine assy with vision sys Ladder machine assy vision system |
|--|---|--|

THIS DOCUMENT IS THE PROPERTY OF AVANTI AND IS LOANED TO YOU BY COMPANY. YOU WILL BE RESPONSIBLE FOR RETURNING IT TO AVANTI. AVANTI WILL NOT BE RESPONSIBLE FOR ANY DAMAGE TO THIS DOCUMENT. AVANTI WILL NOT BE RESPONSIBLE FOR ANY DAMAGE TO THIS DOCUMENT. AVANTI WILL NOT BE RESPONSIBLE FOR ANY DAMAGE TO THIS DOCUMENT. AVANTI WILL NOT BE RESPONSIBLE FOR ANY DAMAGE TO THIS DOCUMENT.

Scale: A3 Date: Appr.: Released: Change Description: Title: Ladder machine assy with vision sys Ladder machine assy vision system Eng. No.: AD00008075 Rev.: 02



Ladder machine unloading side - Setup 2

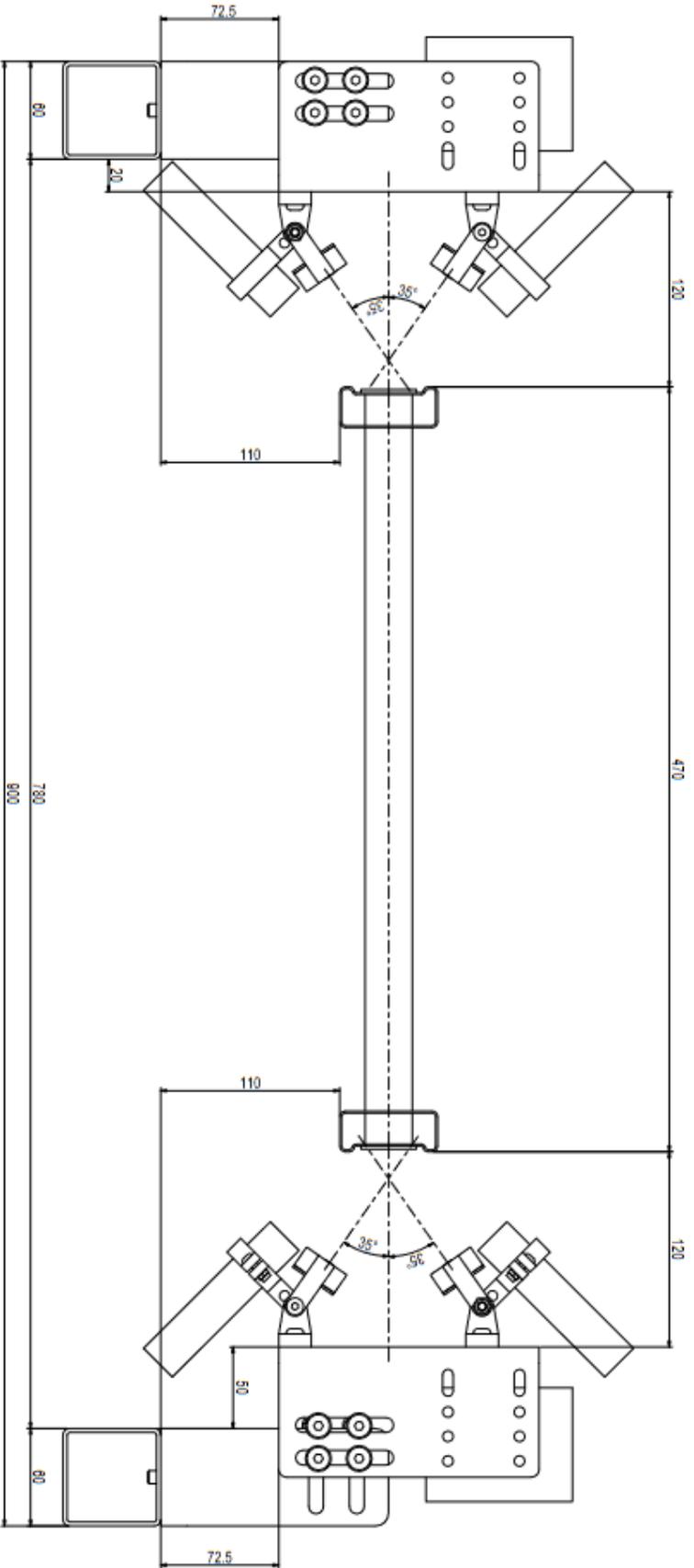


Position representation of both sides of ladder

| | | | |
|---------|-------------|-----------|-------------------------------------|
| FIGURE | ISO 2768-MK | TOLERANCE | NO TOLERANCE |
| FIG. TO | 4/5 | REF. | DATE |
| SCALE | A3 | APPR. | RELEASE |
| SCALE | A3 | SCOPE | Change Description |
| SCALE | A3 | SCOPE | Not applicable |
| SCALE | A3 | SCOPE | Ladder machine Assy with vision sys |
| SCALE | A3 | SCOPE | Ladder machine Assy vision system |
| SCALE | A3 | SCOPE | AD00008075 |
| SCALE | A3 | SCOPE | 02 |

THIS DOCUMENT AND ITS CONTENTS ARE PROVIDED TO YOU BY AVANTI AND ARE SUBJECT TO AVANTI'S CONFIDENTIALITY AND PROPRIETARY RIGHTS POLICY. AVANTI ASSUMES NO LIABILITY FOR ANY DAMAGE OR LOSS OF PROFITS OR BUSINESS OPERATIONS, INCLUDING REVENUE, ARISING FROM THE USE OF THIS DOCUMENT OR THE INFORMATION CONTAINED THEREIN. AVANTI DISCLAIMS ANY AND ALL WARRANTIES, INCLUDING MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. AVANTI'S LIABILITY IS LIMITED TO THE EXTENT PERMITTED BY APPLICABLE LAW. AVANTI'S LIABILITY IS LIMITED TO THE EXTENT PERMITTED BY APPLICABLE LAW.

Complete assembly - Both sides



| | | | |
|-------------------------------------|-------------|---|----|
| | | Position representation of both sides of ladder | |
| | | Change Description | |
| Version | ISO 2768-MK | Part | 1 |
| Material | 316 | Material | |
| Scale | 1:5 | Scale | A2 |
| Ladder machine assy with vision sys | | Ladder machine assy vision system | |
| Part No. | AD00008075 | Part No. | 02 |

A.2.4 Autómata siemens S7-1200

Se utiliza para la implementación del módulo de control de calidad en la máquina de fabricación de escaleras, previamente programado, un autómata programable S7-1200. Este controlador lógico programable (PLC), fabricado por Siemens, una empresa alemana líder en tecnología, forma parte de la familia de productos SIMATIC S7 y se utiliza en aplicaciones de automatización industrial para controlar y supervisar procesos.

El PLC S7-1200 permite establecer una comunicación mediante la librería Python Snap7, facilitando así la integración con el sistema de control de calidad desarrollado.

ANEXO 3 Software utilizado

A.3.1 Yolo v8

Yolo v8 es un popular modelo de detección de objetos y segmentación de imágenes, fue desarrollado por Joseph Redmon y Ali Farhadi en la Universidad de Washington. Lanzado en 2015.

Se utiliza la versión YOLO v8, siendo la última versión de YOLO de Ultralytics, aumentando el rendimiento, la flexibilidad y la eficacia de versiones anteriores. Admite una gama completa de tareas de visión como la detección, la segmentación, la estimación de la pose, el seguimiento y la clasificación.

A.3.2 Python

Python es un lenguaje de programación de alto nivel que permite trabajar rápidamente e integrar sistemas de manera más eficaz. Su principal objetivo es proporcionar legibilidad de código y productividad para los desarrolladores, además de su capacidad para manejar grandes conjuntos de datos.

Se decide utilizar Python para el desarrollo del modelo debido a su muy popularidad en la industria, su sencillez, legibilidad y facilidad de comprensión. Además, Python es de código abierto y cuenta con una comunidad innovadora y extensa, que ha desarrollado multitud de librerías útiles para el desarrollo de este proyecto.

A.3.3 Librería Snap 7

Snap7 es un software multi plataforma para comunicarse de igual manera nativa con los PLCs del fabricante SIEMENS, utilizando tecnología Ethernet y el protocolo S7. Es un software de código abierto, orientado para ser utilizado en comunicaciones Ethernet y S7, sin necesidad de un adaptador especial.

Se decide utilizar Snap7 para lograr la comunicación entre la Raspberry Pi 5 y el autómata programable debido a la sencillez de uso. Dado que el modelo solo requiere la lectura de tres señales del autómata programable, no se necesita una comunicación compleja. Se garantiza el proceso y la lectura correcta de los datos.

ANEXO 4 Base de datos para la realización del entrenamiento del modelo YOLOv8

Se utiliza para la realización de un modelo de detección de objetos en imágenes, un conjunto de datos que contiene imágenes de peldaños tomadas con el mayor rango de sensibilidad posible. Por tanto, se recogen imágenes que contienen:

- Peldaños conformados con aluminios de características duras.
- Peldaños conformados con aluminios de características blandas.
- Peldaños capturados en condiciones climáticas soleadas, donde la luz es elevada.
- Peldaños capturados en condiciones climáticas nubosas, donde la luz incidente se reduce.
- Peldaños capturados con alguna breve vibración, que desenfoca levemente la imagen.
- Peldaños conformados de manera idónea según el criterio de calidad.
- Peldaños conformados para su posterior revisión según el criterio de calidad.
- Peldaños con roturas, que fueron almacenados durante 12 meses.

Se conforma un *dataset* fiable, con bastante rango de imágenes, para crear un modelo fiable, con alta sensibilidad de detección, acorde a las características necesarias de Avanti Wind Systems.

Finalmente, cabe destacar, en caso de producirse cambios de proveedores, que provocan el cambio del tipo de aluminio, podrían incluirse en este conjunto de datos imágenes de peldaños nuevos, mejorando así el modelo, aumentando la sensibilidad y flexibilidad del mismo.

Enlace del *dataset*:

https://drive.google.com/drive/folders/1QddJB0iY0KanhW_TV3CMXVmqe-5tx5iA?usp=drive_link

ANEXO 5 Script utilizado para la programación de la Raspberry Pi 5

A.5.1 Script captura e inferencia peldaños

Se adjunta el script encargado de la captura de imágenes, con su posterior detección de clases de objetos en imágenes y su clasificación según el criterio de calidad definido en el capítulo 4.

Se utiliza para la inicialización de los módulos de captura de imagen, la librería PiCamera2 declarando como objetos las 2 cámaras encargadas de la captura de imágenes, cada una de ellas encargadas de la parte superior e inferior del peldaño.

Se cargan los modelos encargados de la detección de objetos, previamente etiquetados y definidos haciendo uso de la librería Ultralytics.

Se ajustan los parámetros de captura de las cámaras, asignando un valor de Ganancia ISO de 1.5 y un valor de tiempo de exposición de 20000.

Se realiza la conexión entre el autómata programable y la Raspberry Pi 5 mediante la librería snap7. La Raspberry Pi 5 lee y guarda como variables el número de peldaños a realizar por la máquina, el número de peldaños remachados y la señal booleana del sensor fin de carrera de la remachadora.

Dada una escalera de n peldaños:

1.- Al activarse la señal de captura y siendo 2 el valor de contador de peldaños remachados, se realiza un autoenfoco de las cámaras en la zona central de la imagen, para evitar vibraciones y aumentar la calidad de la imagen. Se realiza únicamente el autoenfoco al comienzo de la producción de la nueva escalera.

2.- Al activarse la señal de captura, para cada peldaño i entre $[1, n-2]$, comienza la captura de imagen. Por tanto, cuando el peldaño se encuentra en la posición del módulo de control de calidad, realiza:

- Captura de imagen del peldaño i , en formato array.

- Recorta imagen mediante la función “crop_image”, obteniendo la imagen en formato array recortada a tamaño 2839×2128 . La función “Crop_image” se define al comienzo del script, recibe como parámetro de entrada una imagen y devuelve la imagen recortada en formato array.

3.- Se realiza la detección de objetos en la imagen, utilizando el modelo en tipo predictivo, previamente cargado.

4.- Con el resultado de la inferencia, mediante cuadros de etiquetados realizados con la librería Annotator, que indican la clase detectada, obtenemos las imágenes con las clases de objetos anotadas.

5.- Se clasifican las imágenes tras la detección:

- Si cumplen los criterios de control establecidos tanto en la imagen de la cámara encargada de la parte superior, como de la cámara encargada de la parte inferior del peldaño, se guardan en un directorio de peldaño correcto.

- Si no cumplen los criterios de control establecidos en ambas imágenes, se almacenan en la carpeta de imágenes pendientes de revisión.

Los criterios de control establecidos para una imagen correcta contienen 2 objetos de clase *Expansion_OK* y 1 objeto de la clase *Rivet_OK*.

Los criterios de control establecidos para una imagen incorrecta contienen al menos 1 objeto de clase *Expansion_CHECK* o 1 objeto de la clase *Rivet_failure*.

6.- Se almacenan los resultados en un archivo results.txt, anotando el peldaño analizado, con el resultado y valoración del estado del peldaño.

El modelo se realiza de forma cíclica y continúa, una vez finaliza la escalera, cuando se han alcanzado el número de peldaños pendientes de analizar, comienza un nuevo ciclo con una nueva escalera.

Finalmente cabe destacar como implementación del modelo para su posterior funcionamiento, la implementación del archivo para realizarse de forma automática cuando el operario alimente la Raspberry Pi 5. Se define el archivo Ladder_inference.py como un servicio que debe ejecutarse al iniciar la Raspberry Pi 5.

```

1 ▾ """
2     This file captures images from two cameras of the steps of a ladder.
3     Captured images are cropped to decrease inference time.
4     A trained YOLO model is loaded and annotation is done on the cropped image
5     results with defined classes. The results in images are stored in directories
6     based on the established control criteria.
7     The results indicating whether the step is correct or incorrect are
8     stored in a list, which will be displayed to the operator on a screen
9
10    Ladder_inference.py
11    """
12
13    import snap7
14    from snap7.util import*
15    from picamera2 import Picamera2, Preview
16    from libcamera import controls
17    import time
18    import ultralytics
19    import os
20    from ultralytics import YOLO
21    import cv2
22    from ultralytics.utils.plotting import Annotator
23    import datetime
24    import numpy as np
25
26    # Load the trained model in YOLO
27    model = YOLO("/home/avanti/ladder_inference/best.pt")
28
29    # Create 2 objects
30    picam0 = Picamera2(1)
31    picam1 = Picamera2(0)
32
33    # Configuration of the camera 0
34    camera_config_0 = picam0.create_still_configuration(main={"size": (4056, 3040)}, lores={"size": (640, 480)}, display="lores")
35    picam0.configure(camera_config_0)
36    picam0.set_controls({"AnalogueGain":1.5})
37    picam0.set_controls({"ExposureTime":10000})
38 ▾ picam0.set_controls({"AfMode":controls.AfModeEnum.Auto,
39                      "AfSpeed":controls.AfSpeedEnum.Fast,
40                      "AfMetering":controls.AfMeteringEnum.Windows,
41                      "AfWindows":[(1480,980,930,1010)]})
42
43    # Configuration of the camera 1
44    camera_config_1 = picam1.create_still_configuration(main={"size": (4056, 3040)}, lores={"size": (640, 480)}, display="lores")
45    picam1.configure(camera_config_1)
46    picam1.set_controls({"AnalogueGain":1.5})
47    picam1.set_controls({"ExposureTime":10000})
48 ▾ picam1.set_controls({"AfMode":controls.AfModeEnum.Auto,
49                      "AfSpeed":controls.AfSpeedEnum.Fast,
50                      "AfMetering":controls.AfMeteringEnum.Windows,
51                      "AfWindows":[(1480,980,930,1010)]})
52
53    # Read the current ladder value
54    cont_file = "/home/avanti/Desktop/ladder_cont.txt"
55 ▾ with open(cont_file, "r") as f:
56     cont_ladder = int(f.read())
57
58    # Connection to PLC
59    plc = snap7.client.Client()
60    plc.connect("150.150.150.10",0,1)
61    print('conectado')
62
63    # Function for image cropping
64 ▾ def crop_image(image_to_crop, x=0.5, y=0.5, width_ratio=0.5, height_ratio=0.5):
65     """
66     This function crops an image by creating a bounding box on normalized coordinates X, Y, where the bounding
67     box size dimensions are:
68     - new width is the width of the original image multiplied by the width ratio
69     - new height is the height of the original image multiplied by the height ratio
70     :param image_to_crop: This is the image to be cropped
71     :param x: This is the normalized coordinate X of the center of the new cropped image
72     :param y: This is the normalized coordinate Y of the center of the new cropped image
73     :param width_ratio: This is the ratio of the new width to the original image width
74     :param height_ratio: This is the ratio of the new height to the original image height
75     return: processed_image
76     """
77     # Get the image size
78     height, width, _ = image_to_crop.shape
79
80     # Calculate the coordinates for cropping
81     # New image width and height
82     a = width * width_ratio
83     b = height * height_ratio
84
85     # New image center
86     x = width * x
87     y = height * y
88
89     # Calculate the coordinates for cropping
90     x1 = int(x - a / 2)
91     x2 = int(x + a / 2)
92     y1 = int(y - b / 2)
93     y2 = int(y + b / 2)
94
95     # Crop the image
96     processed_image = image_to_crop[y1:y2, x1:x2]
97     return processed_image
98
99 ▾ while True:
100     # Start the cameras
101     picam0.start()
102     picam1.start()

```

```

102 picam1.start()
103
104 # Initialization of variables
105 i = 0
106
107 while True:
108     while True:
109         # Read the status of the limit switch of the riveted in the initial position
110         capture_signal = plc.mb_read(105,1) # Adress of the sensor
111         capture_signal = get_bool(capture_signal, 0, 7)
112
113         # Read the number of steps #MW80
114         num_steps = plc.mb_read(80,2) # Adress of the steps cont
115         num_steps = int.from_bytes(num_steps, byteorder = 'big')
116
117         # Read the number of riveted steps
118         riveted_steps = plc.mb_read(324,2) # Adress of the riveted steps cont
119         riveted_steps = int.from_bytes(riveted_steps, byteorder = 'big')
120
121         # Cycle start condition
122         # Autofocus is performed on both cameras
123         # Results of the previous inference are cleared
124         if capture_signal == True and riveted_steps > 1:
125             if(riveted_steps == 2):
126                 print('NEW LADDER')
127                 time.sleep(3)
128                 print("Focusing")
129                 success=picam1.autofocus_cycle()
130                 print(success)
131                 success=picam0.autofocus_cycle()
132                 print(success)
133                 with open("results_left_now.txt", "w")as file:
134                     file.write("")
135             break
136
137         print('NEW STEP')
138
139         # Increment the number of photographed steps
140         i += 1
141
142         # Image filenames
143         filename_up = f"_ladder_{cont_ladder}_left_side_step_{i}_up_{datetime.datetime.now().strftime('%d_%m_%Y')}.jpg"
144         filename_down = f"_ladder_{cont_ladder}_left_side_step_{i}_down_{datetime.datetime.now().strftime('%d_%m_%Y')}.jpg"
145         filename_step = f"left_side_step_{i}_{datetime.datetime.now().strftime('%d_%m_%Y')}.jpg"
146
147         # Capture image
148         time_capture_start = time.time()
149         image_step_up = picam0.capture_array("main")
150         image_step_down = picam1.capture_array("main")
151         print('Image captured')
152
153         # Crop the image captured
154         image_cropped_up = crop_image(image_step_up, x=0.5, y=0.5, width_ratio=0.7, height_ratio=0.7)
155         image_cropped_down = crop_image(image_step_down, x=0.5, y=0.5, width_ratio=0.7, height_ratio=0.7)
156         image_cropped_up = np.ascontiguousarray(image_cropped_up)
157         image_cropped_down = np.ascontiguousarray(image_cropped_down)
158         print('Image cropped')
159         time_capture_end = time.time()
160         print('Time to capture:',(time_capture_end-time_capture_start))
161
162         # Detection of objects in the images
163         time_inference_start = time.time()
164         results_up = model.predict(image_cropped_up,agnostic_nms=True)[0]
165         results_down = model.predict(image_cropped_down,agnostic_nms=True)[0]
166
167         annotator_up = Annotator(image_cropped_up)
168         annotator_down = Annotator(image_cropped_down)
169
170         # Add boxes in the images
171         for box in results_up.bboxes:
172             b = box.xyxy[0]
173             c = int(box.cls)
174             annotator_up.box_label(b, model.names[c])
175
176         for box in results_down.bboxes:
177             b = box.xyxy[0]
178             c = int(box.cls)
179             annotator_down.box_label(b, model.names[c])
180
181         # Images with inference results
182         image_with_detections_up = annotator_up.result()
183         image_with_detections_down = annotator_down.result()
184         time_inference_end = time.time()
185         print('Time inference:',(time_inference_end-time_inference_start))
186
187         # Number of classes detected
188         detected_classes_up = [int(box.cls) for box in results_up.bboxes]
189         detected_classes_down = [int(box.cls) for box in results_down.bboxes]
190
191         # Save annotated image to output folder
192         # Classes
193         #0: Expansion_OK
194         #1: Expansion_CHECK
195         #2: Rivet_OK
196         #3: Rivet_failure
197         if(detected_classes_up.count(1) == 2 and detected_classes_up.count(2) == 1):
198             output_path_up = os.path.join('/home/avanti/ladder_inference/results/corrects/up', filename_up)
199             cv2.imwrite(output_path_up, image_with_detections_up)
200             final_results_up = True
201             print('Up Correct')
202         elif detected_classes_up.count(0) >= 1 or detected_classes_up.count(3) >= 1:
203             output_path_up_1 = os.path.join('/home/avanti/ladder_inference/results/revision/up', filename_up)
204             cv2.imwrite(output_path_up_1, image_with_detections_up)

```

```

204     cv2.imwrite(output_path_up_1, image_with_detections_up)
205     final_results_up = False
206     print('Up Check')
207     else:
208         output_path_up_2 = os.path.join('/home/avanti/ladder_inference/results/unknown/up', filename_up)
209         cv2.imwrite(output_path_up_2, image_with_detections_up)
210         final_results_up = False
211
212     if(detected_classes_down.count(1) == 2 and detected_classes_down.count(2) == 1):
213         output_path_down = os.path.join('/home/avanti/ladder_inference/results/corrects/down', filename_down)
214         cv2.imwrite(output_path_down, image_with_detections_down)
215         final_results_down = True
216         print('Down Correct')
217     elif detected_classes_down.count(0) >= 1 or detected_classes_down.count(3) >= 1:
218         output_path_down_1 = os.path.join('/home/avanti/ladder_inference/results/revision/down', filename_down)
219         cv2.imwrite(output_path_down_1, image_with_detections_down)
220         final_results_down = False
221         print('Down Check')
222     else:
223         output_path_down_2 = os.path.join('/home/avanti/ladder_inference/results/unknown/down', filename_down)
224         cv2.imwrite(output_path_down_2, image_with_detections_down)
225         final_results_down = False
226
227     # Results that will store in the list
228     if (final_results_up == True and final_results_down == True):
229         step_result = "CORRECT"
230     else:
231         step_result = "INCORRECT"
232
233     # Store in the list step_result
234     with open("resultsleft.txt", "a") as file:
235         if(i == 1):
236             file.write(f"New_Ladder_{cont_ladder}\n")
237             file.write(f"Results_{step_result}_{filename_step}\n")
238
239     with open("results_left_now.txt", "w") as file:
240         if(i == 1):
241             file.write(f"New_Ladder_{cont_ladder}\n")
242             file.write(f"Results_{step_result}_{filename_step}\n")
243
244     print('STEP ANALIZED')
245     time.sleep(2)
246
247     # Production end condition
248     if i == num_steps-3 :
249         time.sleep(3)
250         break
251
252     # Stop the cameras and increment the counter of ladders
253     picam0.stop()
254     picam1.stop()
255     cont_ladder += 1
256
257     # Store the new value
258     with open(cont_file, "w") as f:
259         f.write(str(cont_ladder))

```

A.5.2 Script visualización de resultados

Se adjunta el script encargado de la creación de una aplicación web, donde se muestran los resultados tras la inferencia del estado del peldaño.

Para lograr esto, se realiza la conexión entre la Raspberry Pi 5 y una Tablet mediante Wifi, se crea en lenguaje Python mediante la librería *Flask*, una aplicación web sencilla donde la Raspberry envía como texto plano el contenido del archivo `results.txt`, donde se almacenan los resultados de la inferencia de cada peldaño de la escalera.

```

1  ▾ """
2      This file creates a web application that shows the results
3      of the inference performed by the ladder_inference.py file
4      This app can be accessed at http://192.168.23.30:5000/get_results
5      server_ladder.py
6  """
7  from flask import Flask, send_file
8
9  # Create the application
10 app = Flask(__name__)
11
12 # Define the route
13 @app.route('/get_results')
14
15 # Define the .txt file to display
16 ▾ def get_results():
17     results_file = "results.txt"
18     return send_file(results_file, mimetype='text/plain')
19
20 # Start the web application
21 if __name__ == '__main__':
22     app.run(host='0.0.0.0', port=5000)

```

Se crea un servidor web, donde se muestran los resultados de la inferencia realizada. Esto permite la visualización desde cualquier dispositivo de los resultados a tiempo real, ingresando la dirección definida en el archivo.

ANEXO 6 Solución desarrollada por medios propios

A.6.1 Prototipo del modelo

Se realizaron durante el desarrollo de la solución diversas pruebas, relacionadas el módulo de captura de imágenes, donde comenzamos con diversas cámaras, mencionadas en el archivo de listas de materiales. La figura 38.A muestra el prototipo del modelo.



Figura 38.A Prototipo de la solución

Se realizaron pruebas para obtener la mejor calidad posible, comenzando por diversos soportes y lentes, capturando imágenes muy precarias que ayudaron a conformar el ángulo, la distancia y todos los efectos producidos por el entorno que debían de ser solucionados por el módulo de captura.

Se puede observar en la Figura 39.A la evolución de las imágenes capturadas por los prototipos hasta las imágenes capturadas por el modelo final.



Figura 39.A Evolución de la calidad en la captura de imágenes

A.6.2 Costes de la versión no comercial

Se detallan el coste de la versión no comercial en las siguientes tablas:

| Materiales sistema de visión | | | | | |
|---|-----|------------------------|--------------|----------|---|
| DESCRIPTION | QTY | SUPPLIER | SUPPLIER P/N | PRECIO | Link |
| Ordenadores de placa única (SBC) Raspberry Pi 5 Board 8GB | 2 | Mouser Electronics INC | 358-SC1112 | 148,80 € | https://www.mouser.es/ProductDetail/Raspberry-Pi/SC1112?qs=HoCaDK9Nz5c86n0i5EQ%2FPA%3D%3D |
| Accesorios de Raspberry Pi Raspberry Pi 5 Case Red/White | 2 | Mouser Electronics INC | 358-SC1159 | 18,60 € | https://www.mouser.es/ProductDetail/Raspberry-Pi/SC1159?qs=HoCaDK9Nz5d9gZHizvivRA%3D%3D |
| Adaptadores CA de montaje en la pared Raspberry Pi 5 27W USB-C PD PSU EU, Black | 2 | Mouser Electronics INC | 358-SC1157 | 22,32 € | https://www.mouser.es/ProductDetail/Raspberry-Pi/SC1157?qs=HoCaDK9Nz5fNeKL9V81LRA%3D%3D |
| Cameras y módulos de cámara Camera Module 3 Wide | 4 | Mouser Electronics INC | 358-SC0874 | 130,20 € | https://www.mouser.es/ProductDetail/Raspberry-Pi/SC0874?qs=rQFj71Wb1eXBS1leyCexZA%3D%3D |
| Cables de acoplamiento FFC / FPC Raspberry Pi 5 FPC Cables - Camera, 300mm | 5 | Mouser Electronics INC | 358-SC1129 | 9,30 € | https://www.mouser.es/ProductDetail/Raspberry-Pi/SC1129?qs=HoCaDK9Nz5ccJ4%252BJpQhyXw%3D%3D |
| Tarjeta Micro SD | 2 | Amazon | | 31,98 € | LINK |
| Linterna LED recargable | 4 | Amazon | | 40,00 € | LINK |
| Cable microHDMI - HDMI (10m) | 2 | Amazon | | 51,98 € | LINK |

| Otros Materiales Utilizados en las pruebas | | | | | |
|--|-----|------------------------|--------------|-----------------|---|
| DESCRIPTION | QTY | SUPPLIER | SUPPLIER P/N | PRECIO | Link |
| Cameras y módulos de cámara Raspberry Pi HQ Camera | 2 | Mouser Electronics INC | 358-SC0818 | 93,00 € | link |
| Lentes para cámaras PT3611614M10MP 16mm, 10MP lens (C-mount) | 2 | Mouser Electronics INC | 358-SC0123 | 93,00 € | link |
| Cameras y módulos de cámara Camera Module 3 | 1 | Mouser Electronics INC | 358-SC0872 | 23,25 € | https://www.mouser.es/ProductDetail/Raspberry-Pi/SC0872?qs=rQFj71Wb1eUxGr948C6EEw%3D%3D |
| Lente 6-10 mm | 1 | Amazon | | | |
| Soporte pinza flexible cámara | 1 | Amazon | | 21,99 € | LINK |
| Cable HDMI 1.5 m | 2 | Amazon | | 23,88 € | LINK |
| TOTAL | | | | 708,30 € | |
| TOTAL, SIN PROTOTIPO | | | | 453,18€ | |

ANEXO 7 Diagrama de Gantt

El trabajo fin de grado se desarrolló en varias etapas, comenzando por un análisis detallado del funcionamiento de la máquina, incluyendo todas las mejoras a realizar en la máquina de fabricación de escaleras. Finalizando con las 2 soluciones planteadas para posibles mejoras, destacando las diversas subtareas que ocuparon la mayor parte del tiempo para lograr el desarrollo final del modelo desarrollado por medios propios.

Se visualizan en la figura 40.A los componentes básicos del proyecto, las tareas que se programan y el tiempo que requieren para su elaboración.



Figura 40.A Diagrama de Gantt

Además, se destaca la elaboración de la memoria del trabajo fin de grado como otra subtarea añadida a la elaboración del proyecto fin de grado.

BIBLIOGRAFÍA

- [1] *The Python Standard Library*. (s.f.). Obtenido de <https://docs.python.org/3/library/index.html>
- [2] *Raspberry Camera Software*. (s.f.). Obtenido de https://www.raspberrypi.com/documentation/computers/camera_software.html
- [3] Silvia Illescas. (s.f.). *Curso de fotografía entendiendo la apertura*. Obtenido de <https://www.dzoom.org.es/curso-de-fotografia-entendiendo-la-apertura/>
- [4] *Dinâmica Virtual*. (2017). *Manual de funcionamiento*. Obtenido de Avanti Wind Systems.
- [5] *Snap 7 library*. (s.f.). Obtenido de <https://snap7.sourceforge.net/>
- [6] *Ultralytics*. (s.f.). Obtenido de <https://github.com/ultralytics/ultralytics>
- [7] *PLC S7-1200 de Siemens*. (s.f.). Obtenido de <https://programacion Siemens.com/s7-1200/>
- [8] *BCNVision*. (s.f.). Obtenido de <https://bcnvision.es/es>
- [9] *In-Sight Vision*. (s.f.). Obtenido de <https://www.cognex.com/es-es/training/training-descriptions/in-sight-vision-suite>
- [10] *Label Studio*. (s.f.). Obtenido de <https://labelstud.io/>
- [11] *Weerg Staff*. (2023). *Aluminio características y propiedades*. Obtenido de <https://www.weerg.com/es/guias/aluminio-que-es-propiedades-caracteristicas>