



MÁSTER UNIVERSITARIO EN BIG DATA

TRABAJO FIN DE MÁSTER

SISTEMA DE VISIÓN ARTIFICIAL PARA EL PROCESADO DE IMÁGENES Y CLASIFICACIÓN DE ACTIVOS EN REDES DE DISTRIBUCIÓN

Autor: Guillermo Valle Gutiérrez

Director: Guillermo González-Santander de la Cruz

Madrid

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
Sistema de visión artificial para el procesado de imágenes y clasificación de activos en
redes de distribución

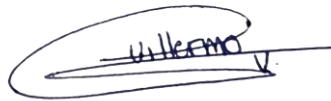
en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2024/25 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.



Fdo.: Guillermo Valle Gutiérrez

Fecha: 09/05/2025

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Firmado por GONZALEZ-
SANTANDER DE LA CRUZ,
GUILLERMO (AUTENTICACIÓN) el
día 13/05/2025 con un
certificado emitido por AC

Fdo.: Guillermo González-Santander de la Cruz

Fecha: 13/05/2025



MÁSTER UNIVERSITARIO EN BIG DATA

TRABAJO FIN DE MÁSTER

SISTEMA DE VISIÓN ARTIFICIAL PARA EL PROCESADO DE IMÁGENES Y CLASIFICACIÓN DE ACTIVOS EN REDES DE DISTRIBUCIÓN

Autor: Guillermo Valle Gutiérrez

Director: Guillermo González-Santander de la Cruz

Madrid

Agradecimientos

Quiero expresar mi agradecimiento sobre todo a mis padres, Santos y Juana, y a mi hermana María, por la educación recibida a lo largo de mi vida. También quiero agradecer a mis amigos de Toledo, cuya amistad y apoyo han sido fundamentales a lo largo de los años. Agradecer también a todas las personas que he conocido en Madrid, en el C.M.U Barberán y Collar; mis compañeros en la UPM y en ICAI; las que he conocido como jugador de fútbol en el Chamberí FS; y las que he conocido en mi vida laboral en KPMG, The Cocktail y baobab, por haber hecho que estos años vividos en Madrid hayan sido de los mejores de mi vida.

Gracias,

Guillermo

SISTEMA DE VISIÓN ARTIFICIAL PARA EL PROCESADO DE IMÁGENES Y CLASIFICACIÓN DE ACTIVOS EN REDES DE DISTRIBUCIÓN

Autor: Valle Gutiérrez, Guillermo.

Director: González-Santander de la Cruz, Guillermo.

Entidad Colaboradora: baobab soluciones SL

RESUMEN DEL PROYECTO

El presente trabajo presenta el desarrollo de un sistema basado en técnicas de visión artificial y *Deep Learning* para gestión del inventario de activos en redes de distribución eléctrica. Para ello, se ha utilizado un modelo de detección *open-set* (Grounding DINO) y clasificadores específicos realizando un *fine-tuning* de los modelos (YOLOv8). Obteniendo una herramienta capaz de identificar visualmente postes eléctricos y clasificarlos en función de características técnicas como material y función. Los resultados obtenidos muestran un alto rendimiento en las tareas de detección y clasificación, validando la viabilidad de aplicar este enfoque en el sector energético.

Palabras clave: poste eléctrico, YOLOv8, Grounding DINO, visión artificial, Deep Learning, inventario de activos, redes de distribución

1. Introducción

La gestión de inventarios de activos es un aspecto relevante en las empresas que cuenten con grandes cantidades de activos. Este es el caso de la compañía energética Iberdrola, que cuenta con redes extensas compuestas por millones de elementos físicos cuya identificación y mantenimiento requiere de un alto conocimiento técnico y una elevada experiencia. Es por eso, por lo que se ha evaluado la posibilidad de realizar la gestión de inventarios apoyándose en ciertas técnicas de visión artificial para poder automatizar algunos procesos manuales. Este trabajo se enmarca en el proyecto IA4TES, una iniciativa pública orientada al desarrollo de soluciones de inteligencia artificial para la transición energética y su digitalización. El caso de uso abordado se centra en la mejora de la gestión del inventario de postes eléctricos mediante técnicas de procesamiento de imágenes.

2. Definición del Proyecto

El objetivo principal del trabajo ha sido desarrollar una solución basada en técnicas de visión artificial capaz de procesar imágenes de activos físicos para extraer información relevante y clasificar el inventario. Para ello, se ha diseñado un pipeline dividido en dos módulos principales:

- Un módulo de **detección *open-set*** basado en el modelo **Grounding DINO**, capaz de localizar postes en imágenes sin necesidad de reentrenamiento.
- Un módulo de **clasificación** construido sobre la arquitectura **YOLOv8**, que ha sido ajustado mediante *fine-tuning* para clasificar los postes eléctricos (tipo de material, subtipo, función de apoyo y función especial de apoyo).

3. Solución técnica

El sistema implementado sigue un flujo automatizado dividido en dos etapas principales, que permiten transformar imágenes generales de campo en información estructurada y útil para la gestión de activos eléctricos.

1. **Detección del activo con Grounding DINO:** Como se representa en la parte superior izquierda de la Ilustración 1, la imagen original es procesada inicialmente por el modelo **Grounding DINO**, encargado de localizar el activo relevante (*electric pylon*) dentro de la imagen. Este modelo, basado en una arquitectura Transformer con integración de lenguaje natural, permite realizar detecciones en escenarios de conjunto abierto (*open-set*), utilizando como entrada descripciones textuales del objeto a localizar. El resultado de esta etapa es un recorte centrado en el activo detectado, que aísla visualmente el poste y descarta información de fondo no relevante. La palabra clave utilizada para detectar los activos ha sido *electric pylon*.



Ilustración 1. Detección del activo con Grounding DINO

2. **Clasificación mediante modelos YOLO especializados:** A continuación, el recorte generado se transmite a un conjunto de modelos **YOLOv8**, entrenados mediante *fine-tuning* para abordar tareas específicas de clasificación. Tal y como se ilustra en la parte inferior derecha de la Ilustración 2, cada modelo opera sobre una dimensión concreta de clasificación del poste:

- **Tipo de material**
- **Subtipo de material**
- **Función de apoyo**
- **Función especial de apoyo**

La salida de este bloque es una clasificación multi etiqueta que resume las propiedades clave del activo, generando información estructurada directamente integrable en los sistemas de inventario.



Ilustración 2 – Clasificación del activo con YOLO

Ambos componentes han sido integrados en una aplicación web desarrollada con Streamlit, permitiendo a los usuarios cargar imágenes, lanzar el análisis y visualizar los resultados desde una interfaz sencilla. El despliegue se ha realizado en Google Cloud, utilizando Docker, Terraform y Jenkins para una infraestructura reproducible y escalable.

4. Resultados

- Grounding DINO ha permitido realizar detecciones precisas sin necesidad de etiquetas previas, validando su idoneidad para tareas de inventario open-set en redes de distribución.
- Las matrices de confusión y curvas Precision-Recall han confirmado la capacidad del sistema para diferenciar correctamente entre los tipos de postes. En la Tabla 1. mAP de los modelos de clasificación se muestra un resumen de la métrica mAP (mean Average Precision).

<i>Modelo de clasificación</i>	<i>mAP</i>
Tipo de material	0.961
Subtipo de material	0.92
Función de apoyo	0.8
Función especial de apoyo	0.71

Tabla 1. mAP de los modelos de clasificación

5. Conclusiones

El sistema desarrollado ha demostrado su viabilidad como herramienta de apoyo en la supervisión automatizada de activos eléctricos. La combinación de un detector *open-set* y clasificadores especializados ha permitido abordar el problema de forma modular y eficiente. Además, el despliegue en la nube y la interfaz accesible facilitan su uso por personal no técnico. Este trabajo sienta las bases para futuras ampliaciones orientadas a la integración con sistemas corporativos y la extensión a otros tipos de activos.

Referencias

- [1] J. D. T. D. J. M. Ross Girshick, "Rich feature hierarchies for accurate object detection and semantic segmentation," *UC Berkeley*, p. 21, 2014.
- [2] D. M. D. F. E. T. Guillaume Maître, «Autonomous inspections of power towers with an UAV,» *Institutional Repository - Research Portal Dépôt Institutionnel - Portail de la Recherche*, 2021.
- [3] S. M. D. F. a. J. V. Anicetus Odo, «Towards the Automatic Visual Monitoring of Electricity Pylons from Aerial Images,» *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2020.
- [4] S. Fang, C. Haiyang, L. Sheng y W. Xiaoyu, «A Framework of Power Pylon Detection for UAV-based Power Line Inspection,» *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, 2020.
- [5] Z. Z. T. R. F. L. H. Z. J. Y. C. L. J. Y. H. S. J. Z. L. Z. Shilong Liu, "Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection," 2023.
- [6] S. D. R. G. A. F. Joseph Redmon, *You Only Look Once: Unified, Real-Time Object Detection*, Washington: Allen Institute for AI, 2016.

COMPUTER VISION SYSTEM FOR IMAGE PROCESSING AND ASSET CLASSIFICATION IN DISTRIBUTION NETWORKS

Author: Valle Gutiérrez, Guillermo.

Supervisor: González-Santander de la Cruz, Guillermo.

Collaborating Entity: baobab soluciones SL

ABSTRACT

This paper presents the development of a system based on artificial vision and Deep Learning techniques for asset inventory management in electricity distribution networks. For this purpose, an open-set detection model (Grounding DINO) and specific classifiers have been used, performing a fine-tuning of the models (YOLOv8). A tool capable of visually identifying electric pylons and classifying them according to technical characteristics such as material and function has been obtained. The results obtained show a high performance in the detection and classification tasks, validating the feasibility of applying this approach in the energy sector.

Keywords: electric pylon, YOLOv8, Grounding DINO, artificial vision, Deep Learning, asset inventory, distribution networks.

1. Introduction

Asset inventory management is a relevant aspect in companies with large amounts of assets. This is the case of the energy company Iberdrola, which has extensive networks made up of millions of physical elements whose identification and maintenance requires a high level of technical knowledge and experience. For this reason, the possibility of inventory management has been evaluated using certain artificial vision techniques to automate some manual processes. This work is part of the IA4TES project, a public initiative aimed at developing artificial intelligence solutions for the energy transition and its digitalization. The use case addressed focuses on improving the management of the inventory of electricity pylons using image processing techniques.

2. Project definition

The main objective of the work has been to develop a solution based on artificial vision techniques capable of processing images of physical assets to extract relevant information and classify the inventory. To this end, a pipeline divided into two main modules has been designed:

- An open-set detection module based on the Grounding DINO model, capable of locating electric pylons in images without the need for retraining.
- A classification module built on the YOLOv8 architecture, which has been fine-tuned by fine-tuning to classify the detected electric pylons (material type, subtype, support function and special support function).

3. Technical solution

The implemented system follows an automated flow divided into two main stages, which allow transforming general field images into structured and useful information for electrical asset management.

1. **Asset detection with Grounding DINO:** As depicted in the upper left part of Figure 1, the original image is initially processed by the Grounding DINO model, which is in charge of locating the relevant asset (electric pylon) within the image. This model, based on a Transformer architecture with natural language integration, allows detections to be performed in open-set scenarios, using textual descriptions of the object to be located as input. The result of this stage is a clipping centered on the detected asset, which visually isolates the pole and discards non-relevant background information. The keyword used to detect the assets has been electric pylon.



Figure 1. Asset detection with Grounding DINO

2. **Classification using specialized YOLO models:** The generated snippet is then passed to a set of YOLOv8 models, trained by fine-tuning to address specific classification tasks. As illustrated in the bottom right of Figure 2, each model operates on a specific pole characterization dimension:
 - **Material type**
 - **Material subtype**
 - **Supporting function**
 - **Special supporting function**

The output of this block is a multi-label classification summarizing the key properties of the asset, generating structured information directly integrable in inventory systems.



Figure 2. Asset classification using YOLO models

Both components have been integrated into a web application developed with Streamlit, allowing users to upload images, launch the analysis and visualize the results from a simple interface. The deployment has been done on Google Cloud, using Docker, Terraform and Jenkins for a reproducible and scalable infrastructure.

4. Results

- Grounding DINO has enabled accurate detections without the need for pre-labelling, validating its suitability for open-set inventory tasks in distribution networks.
- Confusion matrices and Precision-Recall curves confirmed the system's ability to correctly differentiate between pole types. Table 1. mAP of the classification models shows a summary of the mAP (mean Average Precision) metric.

<i>Classification models</i>	<i>mAP</i>
Material type	0.961
Material subtype	0.92
Supporting function	0.8
Special supporting function	0.71

Table 1. mAP of the classification models

5. Conclusions

The developed system has demonstrated its viability as a support tool for automated monitoring of electrical assets. The combination of an open-set detector and specialized classifiers has allowed addressing the problem in a modular and efficient way. In addition, the cloud deployment and accessible interface facilitate its use by non-technical personnel. This work lays the groundwork for future extensions aimed at integration with corporate systems and extension to other asset types.

References

- [1] J. D. T. D. J. M. Ross Girshick, "Rich feature hierarchies for accurate object detection and semantic segmentation," *UC Berkeley*, p. 21, 2014.
- [2] D. M. D. F. E. T. Guillaume Maître, «Autonomous inspections of power towers with an UAV,» *Institutional Repository - Research Portal Dépôt Institutionnel - Portail de la Recherche*, 2021.
- [3] S. M. D. F. a. J. V. Anicetus Odo, «Towards the Automatic Visual Monitoring of Electricity Pylons from Aerial Images,» *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2020.
- [4] S. Fang, C. Haiyang, L. Sheng y W. Xiaoyu, «A Framework of Power Pylon Detection for UAV-based Power Line Inspection,» *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, 2020.
- [5] Z. Z. T. R. F. L. H. Z. J. Y. C. L. J. Y. H. S. J. Z. L. Z. Shilong Liu, "Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection," 2023.
- [6] S. D. R. G. A. F. Joseph Redmon, *You Only Look Once: Unified, Real-Time Object Detection*, Washington: Allen Institute for AI, 2016.

Índice de la memoria

Capítulo 1. Introducción	5
1.1 Motivación del proyecto.....	5
Capítulo 2. Descripción de las tecnologías	7
2.1 Detección de objetos mediante CV	7
2.2 Grounding DINO.....	7
2.2.1 Contexto.....	7
2.2.2 Arquitectura de Grounding DINO.....	8
2.2.3 Preentrenamiento y datos.....	8
2.2.4 Ventajas.....	9
2.3 YOLO (You Only Look Once)	9
2.3.1 Contexto.....	9
2.3.2 Funcionamiento de YOLO.....	10
2.3.3 Flujo de inferencia	11
2.3.4 Ventajas.....	11
2.4 Streamlit	12
2.5 Docker	13
2.6 Jenkins y Terraform para CI/CD	14
2.6.1 Jenkins.....	14
2.6.2 Terraform	15
Capítulo 3. Estado de la cuestión.....	16
Capítulo 4. Definición del trabajo.....	19
4.1 Justificación.....	19
4.2 Objetivos	20
4.3 Metodología.....	20
4.4 Planificación.....	21
Capítulo 5. Solución técnica implementada	23
5.1 Dato de entrada.....	24
5.1.1 Cantidad de datos de entrada.....	24

5.1.2	Diversidad del dato de entrada	25
5.2	Diseño del flujo de ejecución	27
5.3	Detección de activos.....	28
5.4	Clasificación de activos.....	33
5.4.1	Fine-tuning de YOLO	33
5.4.2	Particionamiento de datos para el entrenamiento	34
5.4.3	Hiperparámetros del modelo.....	34
5.4.4	Modelos entrenados.....	36
5.5	Interfaz y despliegue	37
Capítulo 6. Análisis de resultados		40
6.1	Resultados tipo de material	41
6.1.1	Curva Precision-Recall	41
6.1.2	Matriz de Confusión	42
6.2	Resultados subtipo de material.....	43
6.2.1	Curva Precision-Recall	43
6.2.2	Matriz de Confusión	44
6.3	Resultados función de apoyo.....	45
6.3.1	Curva Precision-Recall	45
6.3.2	Matriz de Confusión	46
6.4	Resultados función especial de apoyo.....	47
6.4.1	Curva Precision-Recall	47
6.4.2	Matriz de Confusión	48
Capítulo 7. Conclusiones y trabajos futuros.....		50
7.1	Conclusiones	50
7.2	Trabajos futuros.....	51
Capítulo 8. Bibliografía.....		52
ANEXO I - Stack tecnológico utilizado		53

Índice de figuras

Figura 1. Proceso de inferencia de YOLO	11
Figura 2. Cronograma del proyecto	22
Figura 3. Flujo de ejecución completo para la clasificación de activos	28
Figura 4: Detección de activos haciendo uso de Grounding Dino	33
Figura 5. Clasificación de activos haciendo uso de YOLO.....	37
Figura 6. Streamlit App subida de una imagen y detección automática del poste eléctrico	37
Figura 7. Resultados mostrados en la interfaz de Streamlit	38
Figura 8. Curva de Precision-Recall para tipo de material.....	42
Figura 9. Matriz de confusión en tipo de material.....	43
Figura 10. Curva de Precision-Recall para subtipo de material	44
Figura 11. Matriz de confusión en subtipo de material	45
Figura 12. Curva de Precision-Recall para función de apoyo	46
Figura 13. Matriz de confusión en función de apoyo.....	47
Figura 14. Curva de Precision-Recall para función especial de apoyo	48
Figura 15. Matriz de confusión en función especial de apoyo	49

Índice de tablas

Tabla 1. mAP de los modelos de clasificación	9
Tabla 2. Bloques y tareas del proyecto.....	21
Tabla 3. Formato del fichero de metadatos	24
Tabla 4. Resumen metadato para el tipo de material.....	25
Tabla 5. Resumen metadato para el subtipo de material	26
Tabla 6. Resumen metadato para la función del activo	26
Tabla 7. Resumen metadato para la función especial del activo	27
Tabla 8. Tabla resultante del módulo de detección de activos	29
Tabla 9. Resumen metadato para el tipo de material tras filtrar duplicados	30
Tabla 10. Resumen metadato para el subtipo de material tras filtrar duplicados	31
Tabla 11. Resumen metadato para la función de apoyo del activo tras filtrar duplicados ..	31
Tabla 12. Resumen metadato para la función especial del activo tras filtrar duplicados....	32
Tabla 13. Configuración de hiperparámetros de Ultralytics para YOLO.....	35
Tabla 14: ejemplo de una matriz de confusión.....	41

Capítulo 1. INTRODUCCIÓN

La gestión eficiente de inventarios de activos constituye un elemento fundamental en numerosas organizaciones, especialmente aquellas que operan con infraestructuras complejas y distribuidas geográficamente. En el caso del sector energético, por ejemplo, compañías como Iberdrola gestionan una vasta red compuesta por millones de componentes físicos —como postes eléctricos, transformadores, subestaciones y líneas de distribución— que forman parte crítica del sistema eléctrico. La identificación, clasificación y mantenimiento de estos activos requiere una precisión elevada, tanto para asegurar la integridad de los datos como para optimizar los procesos operativos y de mantenimiento. En este contexto, las tecnologías basadas en inteligencia artificial —y en particular la visión artificial— han emergido como herramientas clave para automatizar procesos tradicionalmente manuales y dependientes del conocimiento experto.

El presente Trabajo Fin de Máster se enmarca en el proyecto IA4TES, una iniciativa financiada por el Ministerio de Asuntos Económicos y Transformación Digital a través de la Secretaría de Estado de Digitalización e Inteligencia Artificial. El proyecto IA4TES promueve el desarrollo de soluciones avanzadas de inteligencia artificial orientadas a la transición energética sostenible, incluyendo casos de uso que impulsen la eficiencia operativa en sectores estratégicos. Dentro de este marco, el trabajo se centra en la aplicación de técnicas de procesamiento de imágenes para mejorar la gestión del inventario de activos, con un enfoque específico en los postes eléctricos.

1.1 MOTIVACIÓN DEL PROYECTO

Tradicionalmente, el proceso de inventariado de activos ha requerido personal altamente especializado, familiarizado con tipologías técnicas y catálogos complejos. Este enfoque, si bien efectivo, implica una alta inversión de tiempo y recursos, además de ser propenso a errores humanos y a inconsistencias en la calidad de los datos. Con el creciente volumen de

activos y la necesidad de mantener actualizadas sus características, surge la necesidad de adoptar soluciones innovadoras que garanticen eficiencia, escalabilidad y precisión.

La motivación de este proyecto es doble: por un lado, aportar una solución tecnológica innovadora a una necesidad operativa concreta, y por otro, contribuir a la transformación digital de sectores estratégicos mediante el uso responsable y eficiente de la inteligencia artificial. La automatización del inventario de activos tiene un impacto directo en la reducción de costes, la mejora en la toma de decisiones y la sostenibilidad a largo plazo, al facilitar una gestión más precisa y menos intensiva en recursos. Asimismo, al disminuir la dependencia del conocimiento experto, esta tecnología democratiza el acceso a procesos técnicos y abre nuevas oportunidades para la digitalización de territorios y empresas con menor grado de madurez tecnológica.

Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

En este apartado se profundiza en las principales tecnologías de detección y segmentación de objetos que se han empleado en el proyecto, junto con las tecnologías empleadas para desplegar la solución desarrollada.

2.1 DETECCIÓN DE OBJETOS MEDIANTE CV

La detección de objetos es una técnica propia de la visión por computador (*computer vision*¹) que permite identificar y ubicar distintos elementos presentes en una imagen o secuencia de vídeo. No solo reconoce que hay en la imagen (como lo haría un clasificador), sino también dónde está, dibujando por ejemplo una caja alrededor del objeto detectado.

2.2 GROUNDING DINO

2.2.1 CONTEXTO

Mientras que la mayoría de detectores funcionan en un *closed set* (conjunto cerrado de clases que el modelo es capaz de identificar), muchos escenarios requieren identificar objetos arbitrarios definidos en tiempo de ejecución (por nombre o descripción). Para hacer énfasis, los modelos que están entrenados en un conjunto cerrado de clasificaciones, no son capaces de clasificar una nueva clase.

Grounding DINO es capaz de abordar esta necesidad introduciendo el lenguaje natural como interfaz para la detección, abriendo el sistema a conceptos no vistos durante el entrenamiento. En el contexto del presente proyecto, esta capacidad ha permitido detectar **postes eléctricos** directamente mediante descripciones textuales, sin necesidad de reentrenar

¹ La visión por ordenador (*computer vision* - CV) es un campo de la inteligencia artificial (IA) que permite a los ordenadores «ver» e interpretar imágenes, vídeos y otros datos visuales, de forma similar a como lo hacen los humanos.

el modelo para esta clase específica, agilizando la implementación sin necesidad de entrenar un modelo de identificación.

2.2.2 ARQUITECTURA DE GROUNDING DINO

El modelo se construye sobre una arquitectura Transformer² (DINO), y añade una fusión estrecha entre la información visual y el texto de entrada. Se compone de los siguientes bloques clave:

- *Feature Enhancer*: Extrae y mejora las características visuales y textuales utilizando mecanismos de atención cruzada entre imagen y el texto. Esto permite al modelo relacionar regiones visuales con descripciones semánticas.
- *Language-Guided Query Selection*: A partir de la descripción textual (por ejemplo: *electric pylon*), selecciona regiones de la imagen que podrían contener el objeto descrito. Estas regiones actúan como puntos de partida para el proceso de detección.
- *Cross-Modality Decoder*: Refina las predicciones combinando información visual y textual para generar las cajas de detección y asociarlas con las expresiones correspondientes.

2.2.3 PREENTRENAMIENTO Y DATOS

Grounding DINO ha sido preentrenado en tres tipos de datos:

- Datos de detección tradicionales (como COCO³).
- Datasets de *grounding* con cajas asociadas a descripciones.
- Datos de *captioning*, que mejoran su comprensión semántica del lenguaje.

² Un *transformer* es un modelo de aprendizaje profundo (Deep Learning) que usa el mecanismo de autoatención, en el que se le da un peso diferente a cada parte de la entrada.

³ El conjunto de datos COCO (*Common Objects in Context*) es un conjunto de datos a gran escala sobre detección, segmentación y subtítulos de objetos.

Gracias a esta combinación, el modelo aprende tanto a localizar objetos como a alinear conceptos visuales y lingüísticos, lo que le permite realizar detecciones en escenarios abiertos, sin necesidad de reentrenamiento.

2.2.4 VENTAJAS

- *Zero-shot detection*: detecta objetos nuevos con solo proporcionar su nombre o descripción.
- Alta flexibilidad: puede trabajar con expresiones complejas (“poste de madera junto a una caja de distribución”).
- Precisión: logra resultados competitivos frente a modelos entrenados específicamente para tareas cerradas.

Este enfoque resulta especialmente útil en el contexto del proyecto, donde se desea detectar postes eléctricos sin necesidad de entrenar el modelo desde cero. Grounding DINO permite abordar esta necesidad de forma directa y eficaz, reduciendo significativamente el coste de anotación y entrenamiento.

2.3 *YOLO (YOU ONLY LOOK ONCE)*

2.3.1 CONTEXTO

La detección clásica de objetos solía apoyarse en pipelines de dos etapas (propuesta de regiones + clasificación), como R-CNN [1], que extraían primero un conjunto de regiones candidatas y luego evaluaban cada una con un clasificador. YOLO (*You Only Look Once*) introduce un enfoque de una sola etapa, donde una única red neuronal⁴ procesa la imagen completa y predice a la vez las cajas delimitadoras y sus probabilidades de clase, transformando la tarea en un problema de regresión espacial.

⁴ Una red neuronal es una técnica de Deep Learning que utiliza un conjunto de neuronas artificiales interconectadas, organizadas en capas, y cuya estructura se inspira en el funcionamiento del cerebro humano.

2.3.2 FUNCIONAMIENTO DE YOLO

- División de la imagen en rejilla (*grid*): YOLO divide la imagen de entrada en una cuadrícula de celdas (por ejemplo, una cuadrícula de 13x13 o 19x19 dependiendo de la resolución de la red). Cada celda es responsable de predecir los objetos cuya parte central se encuentra dentro de esa celda. Si un objeto está centrado en una celda específica, esa celda es la encargada de predecir el objeto y su clase.
- Predicción de cajas delimitadoras (*bounding boxes*): Para cada celda, el modelo predice varias *bounding boxes* que indican las posibles ubicaciones de objetos en la imagen. Cada caja delimitadora se describe por:
 - Coordenadas de la caja: (x, y) que representan el centro de la caja con respecto a la celda, y el ancho (w) y la altura (h) de la caja, en porcentaje respecto al alto y ancho de la imagen total.
 - Confianza: Mide cuán segura está la celda de que la caja contiene un objeto. Esto se calcula como la probabilidad de que haya un objeto en la caja multiplicado por la precisión de la caja.
- Clasificación de objetos: cada celda predice, además, la categoría a la que pertenece el objeto contenido en la caja delimitadora. Estas categorías dependen del conjunto de datos utilizado para el entrenamiento y pueden incluir, por ejemplo, personas, vehículos, animales, entre otros.
- Filtrado de predicciones: Para reducir las predicciones incorrectas, YOLO utiliza un umbral de confianza para filtrar aquellas predicciones cuya probabilidad es baja. Cuando varias cajas predicen el mismo objeto, se aplica un proceso conocido como Non-Maximum Suppression (NMS)⁵, que elimina las detecciones redundantes y conserva únicamente aquella con mayor confianza.

En la Figura 2 se muestra un ejemplo del proceso de inferencia de YOLO:

⁵ NMS es una técnica de postprocesamiento utilizada en la detección de objetos para eliminar las detecciones duplicadas y seleccionar los objetos detectados más relevantes.

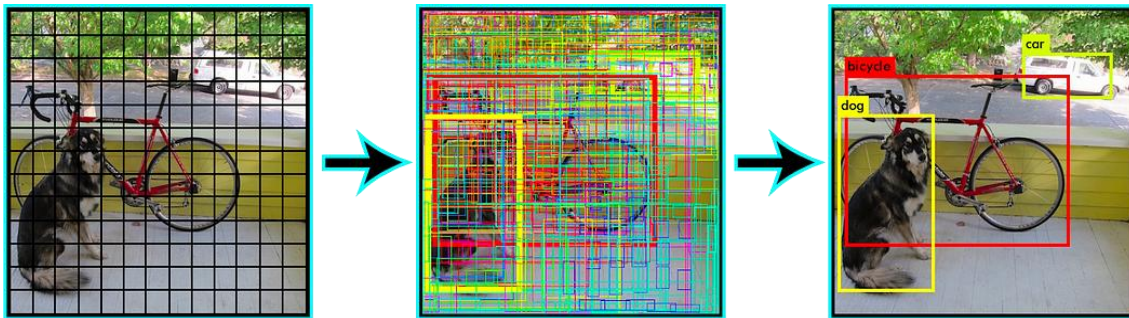


Figura 1. Proceso de inferencia de YOLO

2.3.3 FLUJO DE INFERENCIA

- Para cada celda, la red genera coordenadas de caja y la confianza (*objectness score*).
- Se filtran las predicciones con umbral de confianza y se aplica *Non-Max Suppression* para eliminar solapamientos excesivos.

2.3.4 VENTAJAS

- Rápido y preciso: Detecta objetos en una sola pasada por la imagen, ideal para aplicaciones en tiempo real.
- Reutilización de conocimiento: El *fine-tuning*⁶ parte de modelos preentrenados (como COCO), lo que acelera el entrenamiento y mejora la generalización.
- Adaptación a nuevos objetos: Permite entrenar el modelo para detectar clases específicas del proyecto, como distintos tipos de postes eléctricos.
- Eficiencia en datos y recursos: Necesita menos datos etiquetados y menos tiempo de cómputo que entrenar desde cero.
- Fácil de integrar: Compatible con entornos Python⁷ y librerías, lo que facilita su uso en aplicaciones web o sistemas productivos.

⁶ El *fine-tuning* es el proceso de tomar un modelo de aprendizaje automático preentrenado y seguir entrenándolo en un conjunto de datos más pequeño y específico.

⁷ Python es un lenguaje de programación de alto nivel y propósito general.

2.4 STREAMLIT

Streamlit es un *framework*⁸ de Python de código abierto específicamente diseñado para crear aplicaciones web interactivas de forma rápida y con muy pocas líneas de código, ideal para proyectos de ciencia de datos y Machine Learning:

- **Ejecución basada en scripts:** Al ejecutar `streamlit run app.py`, Streamlit reconstruye la interfaz cada vez que detecta un cambio en el código o una interacción del usuario. Esta mecánica *hot-reload* acelera el ciclo de desarrollo.
- **Componentes nativos:** Dispone de una biblioteca completa de controles UI (`st.button`, `st.selectbox`, `st.slider`, `st.file_uploader`, etc.), así como funciones para mostrar gráficos (Matplotlib, Altair, Plotly), tablas y mapas con apenas una línea de código. Permite además insertar HTML/CSS/JavaScript personalizado si se requiere un grado mayor de personalización.
- **Gestión de estado y sesiones:** Con `st.session_state` es posible guardar variables entre ejecuciones del script, lo que habilita flujos más complejos (formularios, pasos secuenciales, procesos en segundo plano) sin tener que recurrir a bases de datos externas.
- **Integración de modelos ML:** Facilita la carga de modelos entrenados en `scikit-learn`, TensorFlow, PyTorch o cualquier otra librería.
- **Despliegue simplificado:** La propia App es un único archivo Python (o un pequeño conjunto de módulos), por lo que se puede empaquetar y ejecutar en cualquier entorno que tenga Python instalado. En producción, se suele ejecutar tras un servidor (por ejemplo, con gunicorn) o dentro de un contenedor Docker para garantizar aislamiento y reproducibilidad.

⁸ Un *framework* es una estructura real o conceptual destinada a servir de soporte o guía para la construcción de algo que amplíe la estructura hasta convertirla en algo útil.

2.5 DOCKER

Docker es la tecnología de contenedorización de referencia que permite empaquetar aplicaciones y todas sus dependencias en unidades ligeras y portables llamadas contenedores:

- Imágenes inmutables:
 - Una imagen es el esquema principal: contiene el sistema de archivos (OS minimalista), librerías, dependencias y el código de la aplicación.
 - Se define mediante un Dockerfile, donde cada instrucción (FROM, COPY, RUN, ENTRYPOINT, etc.) crea una nueva capa, favoreciendo el cacheo y la construcción incremental.
- Contenedores aislados:
 - Un contenedor es la instancia en ejecución de una imagen. Dispone de su propio espacio de procesos, sistema de archivos, red y configuración de puertos, pero comparte el kernel⁹ con el host, lo que lo hace mucho más ligero que una máquina virtual.
 - Permiten montar volúmenes para persistir datos y configurar redes virtuales para comunicar contenedores entre sí o con el exterior.
- Beneficios clave:
 - Portabilidad: Alta fiabilidad del funcionamiento de la implementación en producción al replicar el entorno.
 - Reproducibilidad: versiones de librerías y sistema fijas.
 - Aislamiento: evita conflictos entre aplicaciones y entornos.
 - Escalabilidad: despliegue rápido de nuevas instancias ante picos de carga.

⁹ El kernel es el *software* que controla el *hardware* y coordina el funcionamiento de un sistema operativo. Se le conoce también como núcleo del sistema operativo.

En el proyecto, se encapsula tanto la aplicación Streamlit como el modelo (y sus dependencias) en un contenedor Docker, de modo que cada versión sea consistente y fácil de distribuir.

2.6 JENKINS Y TERRAFORM PARA CI/CD

Para poder garantizar una herramienta que garantice la integración continua (*Continuous Integration*) CI y la entrega continua (*Continuous Delivery*) CD se ha optado por el uso de Jenkins y Terraform. De esta forma se orquesta el despliegue de forma automática de la aplicación desarrollada en Streamlit. El CI/CD garantiza que cada cambio validado en el repositorio de código ejecuta el pipeline de despliegue de la nueva versión en el entorno de producción.

2.6.1 JENKINS

Todo el flujo de trabajo se define en un archivo llamado Jenkinsfile, que normalmente se guarda dentro del mismo repositorio del proyecto. Este archivo describe paso a paso (por medio de *stages*) qué se debe hacer cada vez que se ejecuta el pipeline (es decir, el conjunto de tareas automatizadas). Las tareas principales que suele hacer son las siguientes:

- Actualización del código desde el repositorio.
- Ejecución de pruebas.
- Construcción de la imagen Docker.
- Despliegue en el entorno de Google Cloud mediante Terraform (para el caso de uso implementado).

Jenkins permite configurar la ejecución automática de los pipelines en respuesta a distintos eventos, como un *push* al repositorio, la creación de una *pull request* o según una programación definida. Además, puede integrarse con herramientas de mensajería como Slack o el correo electrónico para notificar el estado de cada proceso de compilación y despliegue.

2.6.2 TERRAFORM

Terraform es una herramienta de *Infrastructure as Code* (IaC)¹⁰ que permite definir, aprovisionar y gestionar recursos en la nube de manera declarativa. En este proyecto, se ha utilizado para desplegar de forma automatizada los servicios necesarios en Google Cloud, garantizando:

- Automatización del entorno: se definen en código todos los recursos necesarios para la App (como instancias de Compute Engine, reglas de firewall o configuración de red), garantizando consistencia y trazabilidad.
- Control de versiones: cualquier cambio en la infraestructura queda registrado, permitiendo volver fácilmente a versiones anteriores en caso de error.
- Despliegue reproducible: al ejecutar los scripts Terraform desde Jenkins, se asegura que el entorno de producción está siempre alineado con la infraestructura definida en el repositorio.

¹⁰ IaC es la práctica de utilizar código para gestionar y aprovisionar la infraestructura de TI en lugar de procesos manuales. IaC es una práctica DevOps clave que ayuda a las organizaciones a automatizar las tareas de infraestructura.

Capítulo 3. ESTADO DE LA CUESTIÓN

En este apartado se lleva a cabo una revisión de la bibliografía existente sobre resoluciones de problemas similares. La revisión se ha realizado para los modelos de clasificación con visión por computador y sobre modelos de detección.

En [2] se describe una metodología para inspeccionar torres de redes eléctricas a partir de las imágenes de cámaras de UAVs autónomos, segmentadas por la red neuronal convolucional U-Net. De esta manera, se establece un entorno de simulación basado en la física para producir imágenes etiquetadas para entrenar el modelo U-Net, lo cual ahorra tiempo y costos en la recolección de datos físicos. Para encontrar la mejor configuración de un modelo U-Net en términos de características de entrenamiento para la segmentación en el mundo real, se evalúan múltiples condiciones de entrenamiento y prueba, mostrando los hallazgos del estudio sobre especificar características que maximizan el rendimiento del modelo incluso con un pequeño número de imágenes en el escenario del mundo real. También se realiza un análisis de posprocesamiento para extraer estadísticas relevantes para orientar adecuadamente al UAV.

La inspección visual de redes eléctricas de transmisión y distribución se describe en [3], donde helicópteros vuelan en las cercanías de torres de alta tensión con el objetivo de recopilar imágenes, que tienden a capturar imágenes. Pero este enfoque lleva consigo costos, riesgos e ineficiencia. Este documento presenta una revisión de la investigación dirigida a reconocer los elementos de líneas de transmisión y distribución, especialmente cuando se utilizan plataformas de vehículos aéreos no tripulados (UAV). Utilizando un conjunto de datos reciente de imágenes aéreas de alta resolución de torres de energía de media y baja tensión, se explora el aprendizaje profundo como base para el análisis de datos de imágenes. Los métodos de aprendizaje profundo permiten la clasificación confiable de torres.

Las características de los UAVs con cámara en comparación con la inspección manual tradicional son la visualización 3D de la torre de transmisión, alta eficiencia en el trabajo,

costos laborales relativamente bajos, etc. En [4], se menciona que la inspección regular de las líneas eléctricas es esencial para la transmisión segura y confiable de energía eléctrica. Describimos un marco para detectar pilonas eléctricas que combina información de diferentes fuentes: calibración y proyección de cámara, agrupación de modelos de pilona, extracción y emparejamiento de características. Este marco se propone y se aplica a imágenes reales de pilonas eléctricas, y se verifica su rendimiento. Los resultados demuestran que la metodología puede detectar de manera automática y precisa las pilonas eléctricas y distinguir situaciones anómalas, como la ausencia de postes, validando así la propuesta.

En [5] se presenta Grounding DINO, un detector de objetos de “open-set” que fusiona el modelo *Transformer* DINO con un pre-entrenamiento basado en *grounding* de lenguaje, permitiendo detectar objetos arbitrarios a partir de entradas humanas (nombres de categoría o expresiones referenciales). Para lograr la generalización a conceptos abiertos, el trabajo divide el detector cerrado en tres fases y propone una fusión estrecha entre visión y lenguaje mediante: un *feature enhancer* para enriquecer las representaciones visuales, una selección de consultas guiada por lenguaje y un decodificador *cross-modal* que integra ambas modalidades. Grounding DINO se preentrena en gran escala con datos de detección de objetos, *grounding* y *captioning*, y se evalúa tanto en detección *zero-shot* (COCO, LVIS, ODinW) como en benchmarks de detección referencial, alcanzando un 52.5 AP en COCO *zero-shot* y un récord de 26.1 AP en ODinW *zero-shot*.

En [6] se presenta YOLO (*You Only Look Once*), un enfoque unificado de detección de objetos que reformula la tarea como un problema de regresión de cajas delimitadoras y probabilidades de clase, en lugar de reutilizar clasificadores preexistentes. Una única red neuronal procesa la imagen completa en una sola evaluación, permitiendo optimizar el sistema de extremo a extremo directamente sobre el rendimiento de detección. El modelo base de YOLO procesa imágenes en tiempo real a 45 fps, mientras que su versión reducida, Fast YOLO, alcanza 155 fps y dobla el mAP de otros detectores en tiempo real. Aunque YOLO comete más errores de localización, presenta menos falsos positivos en el fondo y

aprende representaciones muy generales de los objetos, superando métodos como DPM y R-CNN al generalizar desde imágenes naturales a dominios como obras de arte.

Capítulo 4. DEFINICIÓN DEL TRABAJO

4.1 JUSTIFICACIÓN

El análisis crítico de los enfoques revisados evidencia que las soluciones existentes se focalizan bien en la segmentación semántica de infraestructuras (p. ej. U-Net), bien en la detección rápida de componentes concretos (YOLO) o en la generalización *open-set* a través de Transformers (Grounding DINO), pero ninguna aborda de forma integrada la extracción simultánea de múltiples atributos de cada activo (tipo y subtipo de material, función de apoyo y función especial), junto con la capacidad de incorporar nuevas clases sin reentrenamiento y de validar o completar automáticamente la información del inventario.

Esta carencia técnica repercute directamente en la operativa de gestión de activos, obligando aún a ciclos manuales de inspección o a soluciones fragmentadas que generan datos inconexos y costosos de mantener. Por ello, el proyecto IA4TES se justifica plenamente, pues propone:

- Un flujo unificado que combina detección rápida y clasificación detallada en un único sistema de visión artificial.
- Mecanismos de validación redundante y enriquecimiento de metadatos que corrigen errores e integran información faltante.

De este modo, el proyecto realizado para IA4TES consigue resolver el problema de la gestión de inventarios de activos, ofreciendo una solución escalable, ágil y alineada con las demandas de digitalización y mantenimiento predictivo en entornos industriales y de infraestructuras.

4.2 OBJETIVOS

El objetivo principal del proyecto es **desarrollar una solución basada en técnicas de visión artificial capaz de procesar imágenes de activos físicos para extraer información relevante de inventario**. A partir de una simple fotografía, la herramienta debe identificar automáticamente las principales características del activo como su tipo, material o configuración y, en los casos más complejos, reconocer modelos específicos mediante su correspondencia con un catálogo predefinido. Esta automatización busca mejorar la precisión, eficiencia y escalabilidad en la gestión del inventario de activos.

A partir del objetivo principal, se han definido los siguientes objetivos específicos:

- **Integrar y adaptar un modelo preentrenado de detección *open-set*** para identificar automáticamente activos en imágenes, sin necesidad de reentrenamiento al incorporar nuevas clases.
- **Implementar un sistema de clasificación automática** que permita distinguir entre diferentes atributos de los activos (tipo y subtipo de material, función de apoyo y función especial de apoyo) a partir de los resultados de la identificación.

4.3 METODOLOGÍA

Para el desarrollo del proyecto se han seguido los principios de desarrollo ágil en su modo de trabajar. Esto implica, entre otros beneficios:

- Desarrollo del proyecto enfocado a resultados rápidos.
- Entregas frecuentes planificadas de las funcionalidades que sean utilizables lo antes posible.
- Mantenimiento de documentación ligera y flexible sobre las herramientas y el código fuente.
- Inclusión del cliente en las pruebas, adaptando el desarrollo a posibles cambios en especificaciones a través de reuniones.

Además, se ha utilizado el desarrollo guiado por pruebas para garantizar que todas las funcionalidades cumplen con los requisitos establecidos.

Durante el desarrollo del proyecto se han programado reuniones de seguimiento semanales para coordinar los esfuerzos y se haya podido informar sobre el progreso del proyecto y de las tareas pendientes.

4.4 PLANIFICACIÓN

Las distintas tareas contempladas en el proyecto, y descritas con un mayor detalle en el apartado de desarrollo realizado se muestran agrupadas por bloques en la Tabla 2:

<i>Bloque</i>	<i>Tarea</i>	<i>Descripción de la tarea</i>
Preparativos	T01	Preparación imágenes
	T02	Preprocesado de imágenes
	T03	Montaje flujo de entrenamiento
	T04	Preparación infra
Modelos	T01	Modelo material cuerpo
	T02	Modelo subtipo material
	T03	Modelo función normal
	T04	Modelo función especial
Documentación	T01	Estado del arte
	T02	Documentación técnica
	T03	Otra documentación

Tabla 2. Bloques y tareas del proyecto

En la Figura 2 se muestra el cronograma del proyecto, la duración total del cuál será de 20 semanas.

Bloque	W01-W04	W05	W06-W19												W20					
Preparativos	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Modelos					■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
Documentacion	■	■				■		■		■		■		■		■		■	■	

Figura 2. Cronograma del proyecto

Capítulo 5. SOLUCIÓN TÉCNICA IMPLEMENTADA

Este capítulo describe la solución técnica para automatizar el inventario de activos haciendo uso de técnicas de *computer vision*. Se ha estructurado el desarrollo del trabajo en bloques que se centran en un aspecto específico de todo el proceso: desde el tipo de datos que se tienen disponibles, hasta la construcción de módulos funcionales que permiten la detección y clasificación automática de los activos en las imágenes.

Se comienza introduciendo el **conjunto de datos** que será la base del proyecto, junto con sus características destacadas, formato y problemas de evolución. Luego, se muestra el proceso de diseño del **módulo de detección de postes eléctricos** basado en un modelo preentrenado de detección de conjunto abierto (*open-set*). Este módulo está diseñado para encontrar la ubicación de los activos en las imágenes de una manera robusta y eficiente.

En segundo lugar, se presentan detalles sobre el **módulo de clasificación**, que es responsable de crear el conjunto de etiquetas para cada poste detectado, describiendo sus principales características físicas y funcionales. La solución propuesta consiste en un módulo responsable de convertir las detecciones visuales en información estructurada que pueda ser utilizada directamente para la gestión del inventario.

Finalmente, la solución **se despliega en una infraestructura en la nube orquestada con Terraform**, donde la aplicación se ejecuta en Google Cloud mediante contenedores Docker y se accede a través de una interfaz web desarrollada en Streamlit. El sistema es sofisticado y puede integrarse perfectamente con otras tecnologías para mantener su rendimiento y eficiencia, además de garantizar su disponibilidad y facilidad de uso, lo que incluye integración continua y servicios complementarios.

5.1 DATO DE ENTRADA

La calidad y la cantidad de datos son elementos fundamentales en el entrenamiento exitoso de modelos de visión artificial para la clasificación avanzada de imágenes en el contexto de la gestión de inventarios de activos. En este apartado, se proporciona una visión general de la **cantidad** de datos de origen, así como de los **metadatos asociados**, provenientes de Iberdrola Data Space Innovation, para entrenar los cuatro modelos requeridos para clasificar el tipo de material, el subtipo de material, la función de apoyo y la función especial de apoyo de los activos.

5.1.1 CANTIDAD DE DATOS DE ENTRADA

La cantidad de datos fuente que se necesitan para entrenar modelos de *computer vision* depende de varios factores, como la complejidad de las características a clasificar, la variabilidad en los activos y las condiciones bajo las cuales se capturan las imágenes. Por ejemplo, cuando se trata de diferenciar activos según su tipo de material, subtipo de material, función de apoyo y función especial de apoyo, se necesitan una gran cantidad de imágenes representativas, así como el rango de escenarios y circunstancias que se encuentran en el entorno operativo.

Al inicio del proyecto, ese número era de 10,942 imágenes. Cómo metadato de entrada se contaba con un fichero que contenía la clase de cada imagen asociada a cada clasificación, en la Tabla 3 se muestra cómo era el fichero de metadatos:

<i>nombre</i>	<i>tipo_material</i>	<i>subtipo_material</i>	<i>funcion_material</i>	<i>funcion_especial</i>
0001.jpg	40	210	98	9
0002.jpg	20	80	4	2
0003.jpg	1	30	1	98

Tabla 3. Formato del fichero de metadatos

5.1.2 DIVERSIDAD DEL DATO DE ENTRADA

Tras analizar el fichero de metadatos, se ha observado que hay 1.021 imágenes que no contienen etiqueta asociada para ninguna de las clasificaciones. Lo que reduce el número de imágenes origen a 9.921. A continuación, se muestra una tabla resumen del fichero de metadatos para cada clasificación.

La Tabla 4 muestra el resumen de metadatos del cliente para la clasificación del tipo de material:

<i>Nombre de la clase</i>	<i>Nº de imágenes “tipo_material”</i>
1 – HORMIGÓN	2.430
20 - METÁLICO CELOSIA	5.041
30 - METÁLICO CHAPA (TUBULAR)	886
40 - METÁLICO PRESILLA	1.564

Tabla 4. Resumen metadato para el tipo de material

La Tabla 5 muestra el resumen de metadatos del cliente para la clasificación del subtipo de material:

<i>Nombre de la clase</i>	<i>Nº de imágenes “subtipo_material”</i>
1 - HORMIGON ESTANDAR	1.254
10 - HORMIGON DOBLE	587
30 - HORMIGON PORTICO	589
80 - CELOSIA C	1.778
90 - CELOSIA S1	591
100 - CELOSIA S2	625

120 - CELOSIA 45T	605
140 - CELOSIA MAT	886
150 - OTRAS CELOSIAS	556
200 - CHAPA GALVANIZADA (TUBULAR)	886
210 - METALICO PRESILLA	1.564

Tabla 5. Resumen metadato para el subtipo de material

La Tabla 6 muestra el resumen de metadatos del cliente para la clasificación de la función de apoyo del activo:

<i>Nombre de la clase</i>	<i>Nº de imágenes "función_apoyo"</i>
1 – SUSPENSIÓN	1.387
2 – AMARRE	1.045
3 – ANCLAJE	555
4 - PRINCIPIO O FIN DE LÍNEA	1.880
98 – PENDIENTE	5.054

Tabla 6. Resumen metadato para la función del activo

La Tabla 7 muestra el resumen de metadatos del cliente para la clasificación de la función especial del activo:

<i>Nombre de la clase</i>	<i>Nº de imágenes "funcion_especial"</i>
1- DERIVACION	772
2 - CT/CTC EN EL APOYO	601
3 - PASO AEREO A SUBTERRANEO	669

4 - CRUCE CON DOBLE AMARRE	712
5 - CRUCE CON VARILLA PREFORMADA	958
7 - SEGURIDAD REFORZADA Y NO DE CRUCE	1.281
8 - CRUCE CON DOBLE AISLADOR RIGIDO	275
9 - ELEMENTO DE MANIOBRA Y PROTECCION	677
10 - CRUCE CON AMARRE	909
98 – PENDIENTE	3.067

Tabla 7. Resumen metadato para la función especial del activo

5.2 DISEÑO DEL FLUJO DE EJECUCIÓN

Cuando un usuario sube una imagen al entorno, esta sigue un proceso automatizado dividido en dos etapas principales:

1. **Detección del activo con Grounding DINO:** Como se muestra en la parte superior izquierda de la Figura 3, la imagen original pasa primero por el modelo Grounding DINO, encargado de localizar el poste eléctrico dentro de la imagen. Este modelo genera un recorte centrado en el activo detectado, eliminando el fondo irrelevante.
2. **Clasificación con modelos YOLO:** El recorte obtenido se envía a los modelos de clasificación, como se observa en la parte inferior derecha de la Figura 3. Utilizando YOLO, se identifican distintas características del poste:
 - a. Tipo de material
 - b. Subtipo de material (si es metálico)
 - c. Función de apoyo
 - d. Función especial de apoyo

El resultado final consiste en una clasificación estructurada que puede ser utilizada directamente en tareas de inventario y gestión de activos. La Figura 3 refleja visualmente este flujo desde la imagen original hasta la obtención de etiquetas específicas.

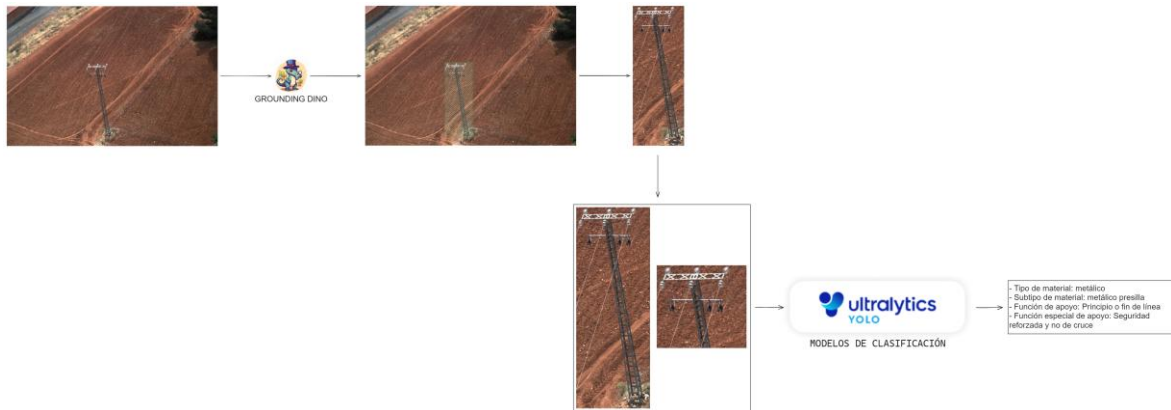


Figura 3. Flujo de ejecución completo para la clasificación de activos

5.3 DETECCIÓN DE ACTIVOS

El objetivo principal de este bloque de trabajo es el desarrollo de un **módulo de detección de postes eléctricos**, con el objetivo de eliminar el ruido de la imagen que no contenía ningún activo. Esta iniciativa ha permitido identificar con precisión la ubicación y la forma de los postes eléctricos. Al **eliminar el contenido no relevante**, se mejora la eficiencia y precisión de los clasificadores que se entrenan en el siguiente bloque de trabajo. En síntesis, este módulo de detección de postes eléctricos representa un avance fundamental en el desarrollo de capacidades de análisis de imágenes para aplicaciones en el ámbito de la distribución eléctrica.

Para ello, se ha utilizado la implementación oficial del modelo **Grounding DINO**, que combina una arquitectura de detección basada en Transformers (DINO) con un preentrenamiento orientado al uso de lenguaje natural. Esta combinación permite identificar objetos definidos mediante entradas humanas, como nombres de categorías o descripciones referenciales. El módulo desarrollado se encarga de localizar los postes eléctricos en cada imagen, devolviendo las coordenadas que indican su posición exacta. Este proceso de

detección se ha ejecutado una única vez por imagen del conjunto de datos, generando una tabla con el siguiente formato:

<i>Nombre de la imagen</i>	<i>Bounding box list</i>	<i>Logits list</i>
0001.jpg	[[x_min, y_min, width, height]]	[0.67]
0002.jpg	[[10, 12, 34, 20], [8, 2, 20, 19]]	[0.45, 0.79]

Tabla 8. Tabla resultante del módulo de detección de activos

La palabra utilizada para encontrar los postes eléctricos ha sido *electric pylon* y tal y como se ha mostrado en la Tabla 8, el modelo devuelve todos los postes que encuentra y la probabilidad que el modelo predice de que sea un poste eléctrico.

Con este modelo se puede recortar las imágenes de forma que los modelos de clasificación posteriores puedan aprender las características principales de los postes.

El segundo desarrollo realizado se ha centrado en la detección de imágenes duplicadas dentro del *dataset*, para ello se ha hecho uso de los datos devueltos por el modelo Grounding Dino. El procesamiento aplicado para detectar duplicados aprovecha la salida del modelo mostrada en la Tabla 8 para poder analizar los duplicados en la respuesta del modelo, sin tener que hacer un análisis procesando las imágenes. Tras haber realizado este análisis de duplicados se han detectado 3.068 imágenes duplicadas, un **28%** de los datos origen.

Tras detectar y eliminar las imágenes duplicadas del conjunto de entrenamiento, se procedió con el análisis de las *bounding box* generadas para cada imagen. Tras haber realizado este análisis, se pueden extraer las siguientes conclusiones:

- Había imágenes que no mostraban ningún activo.
- Había imágenes que mostraban numerosos activos diferentes entre sí.

Por lo tanto, se procedió a filtrar el conjunto de imágenes para poder seleccionar las imágenes con mejor calidad. El criterio de filtrado ha sido que sólo se seleccionan las imágenes que

tienen una única *bounding box* con una probabilidad mayor del 30%. Obteniendo un conjunto de entrenamiento final de 6.811 imágenes de las cuáles 477 no tienen etiqueta.

Tras este filtrado hay que actualizar las tablas de datos iniciales. A continuación se muestra un resumen de los datos con los que finalmente se ha entrenado el modelo de clasificación.

La Tabla 9 muestra el resumen de metadatos del cliente para la clasificación del tipo de material tras el filtrado de duplicados:

<i>Nombre de la clase</i>	<i>Nº de imágenes "tipo_material"</i>
1 - HORMIGÓN	1.480
20 - METÁLICO CELOSIA	3.342
30 - METÁLICO CHAPA (TUBULAR)	531
40 - METÁLICO PRESILLA	981

Tabla 9. Resumen metadato para el tipo de material tras filtrar duplicados

La Tabla 10 muestra el resumen de metadatos del cliente para la clasificación del subtipo de material tras el filtrado de duplicados:

<i>Nombre de la clase</i>	<i>Nº de imágenes "subtipo_material"</i>
1 - HORMIGON ESTANDAR	711
10 - HORMIGON DOBLE	370
30 - HORMIGON PORTICO	399
80 - CELOSIA C	1.149
90 - CELOSIA S1	432

100 - CELOSIA S2	444
120 - CELOSIA 45T	433
140 - CELOSIA MAT	518
150 - OTRAS CELOSIAS	366
200 - CHAPA GALVANIZADA (TUBULAR)	531
210 - METALICO PRESILLA	981

Tabla 10. Resumen metadato para el subtipo de material tras filtrar duplicados

La Tabla 11 muestra el resumen de metadatos del cliente para la clasificación de la función de apoyo del activo tras el filtrado de duplicados:

<i>Nombre de la clase</i>	<i>Nº de imágenes "funcion_apoyo"</i>
1 – SUSPENSIÓN	898
2 – AMARRE	728
3 – ANCLAJE	422
4 - PRINCIPIO O FIN DE LÍNEA	979
98 – PENDIENTE	3.307

Tabla 11. Resumen metadato para la función de apoyo del activo tras filtrar duplicados

La Tabla 12 muestra el resumen de metadatos del cliente para la clasificación de la función especial del activo tras el filtrado de duplicados:

<i>Nombre de la clase</i>	<i>Nº de imágenes "funcion_especial"</i>
1- DERIVACION	449
2 - CT/CTC EN EL APOYO	318
3 - PASO AEREO A SUBTERRANEO	359
4 - CRUCE CON DOBLE AMARRE	430
5 - CRUCE CON VARILLA PREFORMADA	599
7 - SEGURIDAD REFORZADA Y NO DE CRUCE	868
8 - CRUCE CON DOBLE AISLADOR RIGIDO	215
9 - ELEMENTO DE MANIOBRA Y PROTECCION	424
10 - CRUCE CON AMARRE	581
98 – PENDIENTE	2.091

Tabla 12. Resumen metadato para la función especial del activo tras filtrar duplicados

Por lo tanto, haciendo uso de Grounding Dino se ha realizado un módulo de código destinado a la detección de activos para su posterior clasificación. En la Figura 4 se muestra un ejemplo visual de cuál sería el procedimiento realizado para cada una de las imágenes recibidas y cómo se ha recortado la imagen para introducirla posteriormente en los modelos de clasificación.

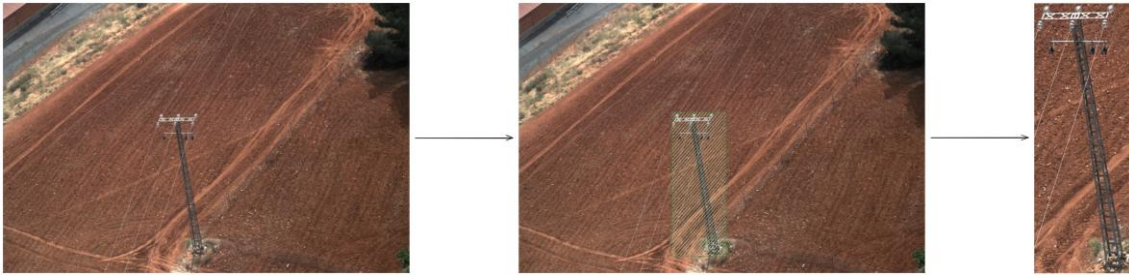


Figura 4: Detección de activos haciendo uso de Grounding Dino

5.4 CLASIFICACIÓN DE ACTIVOS

Este apartado describe el proceso de entrenamiento y utilización de cuatro modelos basados en la arquitectura YOLO, diseñados para clasificar los activos detectados en función de cuatro dimensiones: tipo de material, subtipo, función de apoyo y función especial de apoyo.

5.4.1 FINE-TUNING DE YOLO

El proceso de *fine-tuning* es una técnica que permite adaptar un modelo previamente entrenado para poder mejorar la precisión detectando un objeto en particular o para poder **detectar nuevos objetos**. En este proyecto se ha utilizado esta técnica para entrenar al modelo y poder clasificar los postes eléctricos previamente recortados con el modelo de Grounding DINO. Y para la clasificación de la función de apoyo, se entrenaba con la parte superior del poste eléctrico recortado.

Durante el proceso de *fine-tuning*, se reutilizan los pesos del modelo base como punto de partida y se ajustan en las capas finales de la red neuronal para que aprendan las nuevas clases. Al aprovechar el conocimiento general ya aprendido por el modelo, se mejora su capacidad de generalización incluso con un conjunto de datos reducido y específico. Esto ha sido muy beneficioso debido a la baja cantidad de datos con las que los modelos se han entrenado, en especial para la clasificación de la función de apoyo (ambas clasificaciones).

5.4.2 PARTICIONAMIENTO DE DATOS PARA EL ENTRENAMIENTO

En cuanto al método de particionamiento elegido para poder entrenar y validar los modelos, se optado por una estrategia de particionado estratificada. La cantidad de datos de cada conjunto ha sido la siguiente:

- 80% para entrenamiento
- 10% para validación
- 10% para test

Para cada modelo entrenado la división ha sido diferente por el hecho de que una imagen era posible que sólo tuviese la clasificación del tipo de material y no tuviese el resto. Además, la partición se ha realizado de forma **estratificada por clase**, lo que significa que se ha mantenido la misma proporción de cada categoría (por ejemplo, tipos de material, funciones de apoyo, etc.) en los tres subconjuntos, pero para cada modelo en concreto los conjuntos cambiaban radicalmente. De este modo, se asegura que tanto el entrenamiento como la validación y la prueba de los modelos se realizan sobre distribuciones representativas, evitando sesgos y mejorando la generalización del sistema.

5.4.3 HIPERPARÁMETROS DEL MODELO

Para el entrenamiento de los modelos de clasificación, se ha definido una configuración común de hiperparámetros, esta configuración es para poder entrenar los modelos haciendo uso de la librería Ultralytics (que ha sido la utilizada para realizar el *fine-tuning* de YOLOv8). La configuración se ha seleccionado de esta forma en base a la guía de buenas prácticas de la librería y también teniendo en cuenta el contexto del problema y de las imágenes futuras que se podrían recibir para definir bien las configuraciones de *image augmentation* que permite la librería en el entrenamiento. Además de la configuración, se han lanzado entrenamientos de un máximo de 100 *epochs* (debido a la baja cantidad de datos disponibles) con una implementación de *early stopping* de 3 *epochs* sin mejora. A continuación, se detalla la configuración utilizada:

<i>Parámetro</i>	<i>Valor</i>	<i>Descripción</i>
lr0	0.01	Tasa de aprendizaje inicial. Un valor moderado que equilibra la velocidad de convergencia y la estabilidad del entrenamiento.
warmup_epochs	3.0	Número de épocas de calentamiento. Incrementa gradualmente la tasa de aprendizaje al inicio del entrenamiento para estabilizar el proceso.
hsv_h	0.015	Rango de variación del matiz en la augmentación de datos. Introduce variaciones de color para mejorar la generalización del modelo.
hsv_s	0.7	Rango de variación de la saturación en la augmentación de datos. Simula diferentes condiciones de iluminación.
hsv_v	0.4	Rango de variación del valor (brillo) en la augmentación de datos. Ayuda al modelo a manejar diferentes niveles de exposición.
scale	0.5	Rango de escalado aplicado durante la augmentación. Permite al modelo detectar objetos de diferentes tamaños.
flipud	0.0	Probabilidad de voltear la imagen verticalmente durante la augmentación. No se aplica en este caso.
fliplr	0.5	Probabilidad de voltear la imagen horizontalmente durante la augmentación. Ayuda al modelo a ser invariante a la dirección de los objetos.
mosaic	1.0	Probabilidad de aplicar la augmentación tipo mosaico, que combina cuatro imágenes. Útil para mejorar la detección de objetos pequeños.
mixup	0.0	Probabilidad de aplicar la augmentación mixup, que combina dos imágenes. No se aplica en esta configuración.
copy_paste	0.0	Probabilidad de aplicar la augmentación copy-paste, que copia y pega objetos entre imágenes. No se aplica en esta configuración.

Tabla 13. Configuración de hiperparámetros de Ultralytics para YOLO

5.4.4 MODELOS ENTRENADOS

Los modelos que se han entrenado han sido los siguientes:

- Modelo de clasificación por **tipo de material**: Este modelo clasifica los activos según su tipo de material (Tabla 9). Las imágenes de entrada al modelo son recortes específicos de las detecciones realizadas previamente, centradas en el área del activo para eliminar cualquier contexto irrelevante.
- Modelo de clasificación por **subtipo de material**: Diseñado para identificar subcategorías dentro de cada tipo de material (Tabla 10). Las imágenes de entrada son los mismos recortes utilizados en el modelo anterior, pero el entrenamiento se enfocó en etiquetas más detalladas.
- Modelo de clasificación por **función de apoyo**: Este modelo clasifica los activos según su función principal de apoyo (Tabla 11). Las imágenes recortadas del activo detectado se utilizaron como entrada, asegurando que el modelo se centrara exclusivamente en las características visuales del apoyo, que se encuentra en la zona superior de la torre. Por lo que para estos modelos **se recorta la parte superior de la imagen**.
- Modelo de clasificación por **función especial de apoyo**: Diseñado para clasificar funciones especiales asociadas a los activos (Tabla 12). Las imágenes recortadas también se emplearon en este modelo **recortando la parte superior**, permitiendo capturar detalles clave necesarios para esta clasificación más específica.

Como se ha explicado en el apartado 5.2, el diseño del sistema garantiza que las imágenes que recibe los modelos de clasificación, son imágenes de los activos sin ruido de fondo, gracias al buen rendimiento del módulo de detección de postes eléctricos. Tras haber entrenado los modelos de clasificación, el módulo de clasificación recibe la clasificación de cada una de las clasificaciones mencionadas: tipo de material, subtipo de material, función de apoyo y función especial de apoyo. Como resultado, se obtiene para cada activo las cuatro clasificaciones. La Figura 5 muestra este flujo mostrando las imágenes recortadas de entrada y las clasificaciones generadas como salida.



Figura 5. Clasificación de activos haciendo uso de YOLO

5.5 INTERFAZ Y DESPLIEGUE

La aplicación desarrollada en Streamlit permite subir imágenes de activos (postes eléctricos), ejecutar un pipeline de detección y clasificación automática utilizando modelos de visión artificial, y visualizar tanto las imágenes procesadas como los resultados obtenidos, todo a través de una interfaz web sencilla. En la Figura 6 se muestra la interfaz realizada y cómo sería el proceso de detección y recorte del poste eléctrico. En la Figura 7 se muestran los resultados arrojados por los diferentes modelos de clasificación sobre la imagen subida a la App.

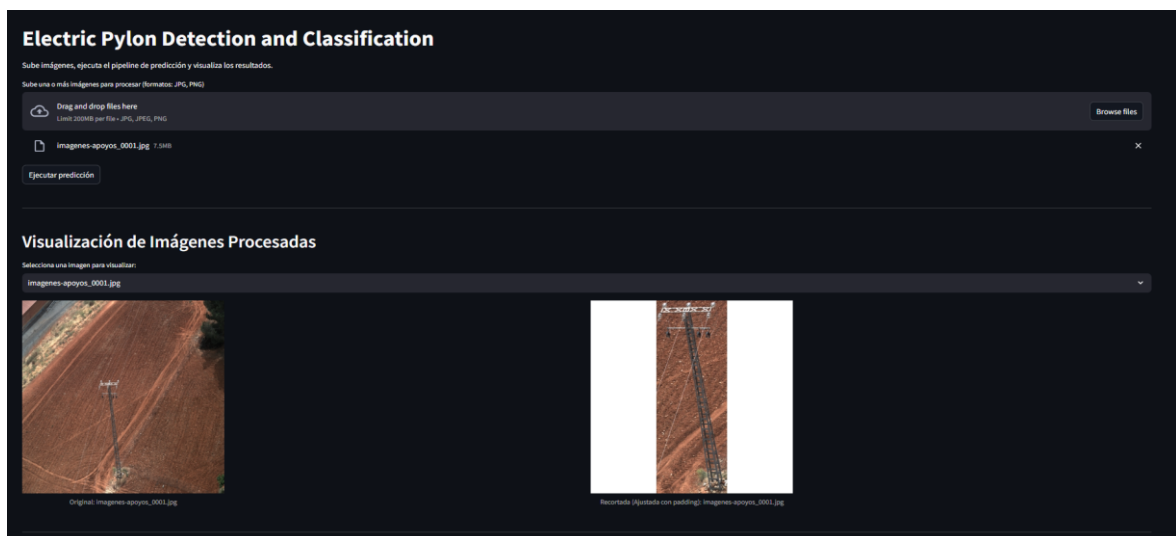


Figura 6. Streamlit App subida de una imagen y detección automática del poste eléctrico

Resultados de Predicciones

Predicciones realizadas para cada tipo de clasificación:

image_path	class	label	confidence
imagenes-apoyos_0001.jpg	funcion_especial_predictions	SEGURIDAD REFORZADA Y NO DE CRUCE	0.9565
imagenes-apoyos_0001.jpg	funcion_material_predictions	PRINCIPIO O FIN DE LINEA	0.9641
imagenes-apoyos_0001.jpg	subtipo_material_predictions	METALICO PRESILLA	0.9213
imagenes-apoyos_0001.jpg	tipo_material_predictions	METALICO PRESILLA	0.9198

Figura 7. Resultados mostrados en la interfaz de Streamlit

En la última fase del proyecto, se realizan todas las configuraciones necesarias para lanzar la aplicación basada en infraestructura de GCP. Para ello, se ha seleccionado un enfoque moderno que utiliza herramientas de infraestructura como código, automatización de despliegue y ejecución de la aplicación web. Esto incluye trabajos generales que permiten la carga de imágenes para activar un proceso de análisis visual utilizando modelos pre entrenados, presentando los resultados de manera interactiva en la App.

La infraestructura se gestiona con Terraform, mediante el cual se puede aprovisionar una máquina virtual en GCP con el sistema operativo Debian 12. Esta instancia recibe una dirección IP pública estática, se configuran las reglas de firewall adecuadas para permitir el acceso HTTP/HTTPS y SSH remoto, y un registro DNS en Cloud DNS apunta a un dominio personalizado. En el mismo proceso, además, se automatizan componentes clave como Docker y Nginx, lo que permite clonar el repositorio del proyecto y descargar los modelos pre entrenados de un *bucket*¹¹ en Cloud Storage.

La aplicación ha sido desarrollada en Streamlit y empaquetada dentro de un contenedor Docker. Una vez desplegada, permite al usuario cargar imágenes directamente desde su navegador. Al subir una imagen, se invoca un script que ejecuta un pipeline de predicción

¹¹ Un *bucket* es un contenedor lógico para almacenar datos en Google Cloud Storage (GCS).

desarrollado en Python. Este pipeline procesa la imagen, aplica transformaciones necesarias, carga el modelo correspondiente y genera predicciones. Los resultados se presentan visualmente mediante imágenes recortadas que ilustran la clasificación realizada, junto con una tabla que resume las predicciones y niveles de confianza del modelo.

Para garantizar un flujo de despliegue y actualización eficiente, se ha diseñado una integración continua con Jenkins. A través de un script personalizado, el pipeline de Jenkins permite desplegar automáticamente la infraestructura, actualizar el código, reiniciar la aplicación y eliminar recursos temporales cuando sea necesario.

La seguridad del sistema se asegura mediante certificados SSL¹² gestionados por Nginx, así como mediante políticas estrictas de acceso por SSH restringidas a ciertas IPs y autenticación segura basada en claves. Gracias a esta arquitectura, el despliegue es totalmente reproducible, automatizado y escalable, lo que permite mantener la solución operativa y actualizada de forma sencilla y controlada.

¹² Un certificado SSL (*Secure Sockets Layer*) es un certificado digital que valida la identidad de un sitio web y cifra la comunicación entre el navegador y el servidor, garantizando una conexión segura.

Capítulo 6. ANÁLISIS DE RESULTADOS

Para poder mostrar las gráficas con los resultados de cada modelo entrenado, es necesario explicar qué significa cada métrica. Estas métricas son:

- *Precision*: mide el porcentaje de registros que han sido clasificados correctamente como un daño entre todos los registros clasificados como daños.

$$precision = \frac{TP}{TP + FN}$$

- *Recall*: mide el porcentaje de daños que han sido clasificados correctamente.

$$recall = \frac{TP}{TP + FP}$$

- *F1-score*: media ponderada de la métrica de *precision* y *recall*.

$$F1\ score = \frac{2 \times (recall \times precision)}{recall + precision}$$

También hay que saber interpretar lo que es una matriz de confusión, que es una herramienta para evaluar el rendimiento de un modelo de clasificación, especialmente en problemas multiclase. Se organiza en una tabla donde:

- Las filas representan las clases reales de las muestras (etiquetas verdaderas).
- Las columnas representan las clases predichas por el modelo.

Cada celda en la matriz indica el número de veces que una clase real fue clasificada como otra clase. En el caso ideal, los valores se concentran en la diagonal principal, lo que indica clasificaciones correctas. Los valores fuera de la diagonal representan errores de clasificación.

A continuación, se incluye una tabla con la nomenclatura típica para un problema multiclase:

<i>Clase Real / Predicha</i>	<i>Clase A</i>	<i>Clase B</i>	<i>Clase C</i>
Clase A	TP(A)	FP(A->B)	FP(A->C)
Clase B	FP(B->A)	TP(B)	FP(B->C)
Clase C	FP(C->A)	FP(C->B)	TP(C)

Tabla 14: ejemplo de una matriz de confusión

Leyenda:

- **TP(X)**: Verdaderos Positivos de la clase X (correctamente clasificados).
- **FP(X->Y)**: Falsos Positivos, donde la clase X fue clasificada erróneamente como Y.

Teniendo claro estos conceptos, se van a ir mostrando los resultados obtenidos tras entrenar los modelos de YOLO para clasificar cada una de las clases que se quieren realizar sobre los activos. Los gráficos son obtenidos directamente con YOLO tras un entrenamiento, y son los resultados obtenidos sobre el conjunto de validación con imágenes que no se han visto por los modelos.

6.1 RESULTADOS TIPO DE MATERIAL

Para evaluar el rendimiento del modelo que clasifica el tipo de material, se han generado dos representaciones gráficas clave: la curva Precision-Recall y la matriz de confusión. Estas herramientas permiten analizar en detalle la capacidad del modelo para realizar predicciones precisas en cada clase.

6.1.1 CURVA PRECISION-RECALL

La curva Precision-Recall proporciona una visión del balance entre la precisión (proporción de predicciones correctas sobre todas las predicciones realizadas para una clase) y el recall

(proporción de instancias correctamente identificadas sobre todas las instancias reales de la clase) a diferentes umbrales de confianza. En este modelo:

- Se obtuvo un mAP (mean Average Precision) de 0.961 para todas las clases a un umbral de IoU de 0.5.
- Las clases individuales presentan un rendimiento notable, con precision-recall equilibrados.

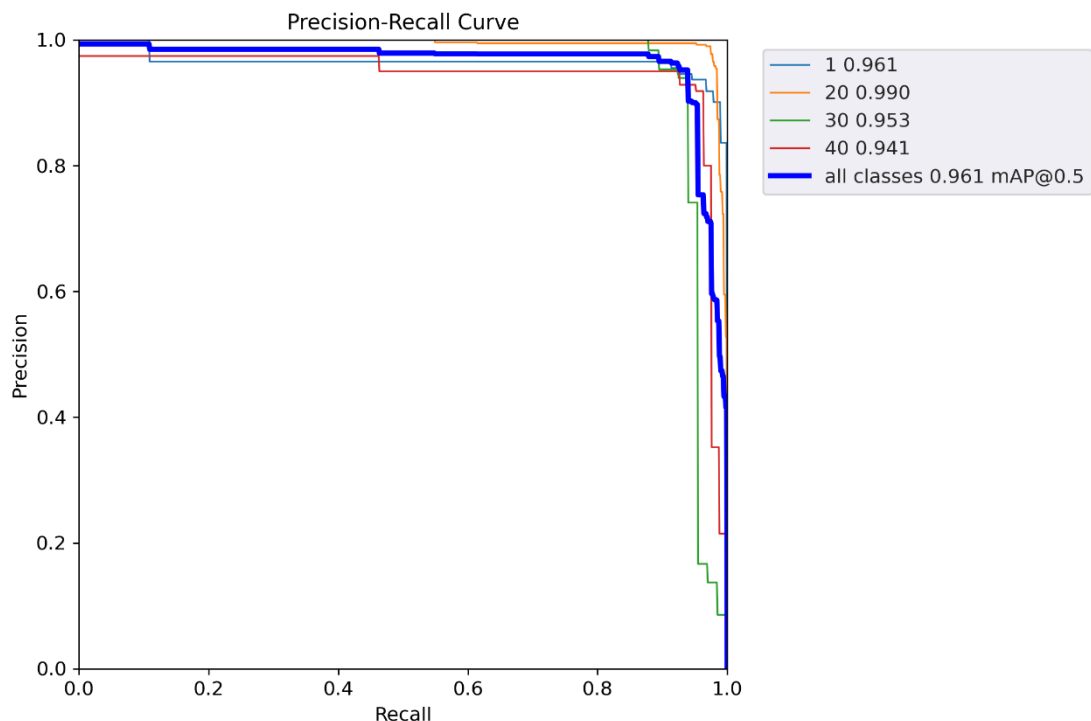


Figura 8. Curva de Precision-Recall para tipo de material

6.1.2 MATRIZ DE CONFUSIÓN

La matriz de confusión resume las predicciones realizadas por el modelo comparándolas con las etiquetas reales. En este caso el modelo ha demostrado una alta precisión en la mayoría de las clases, con algunos errores distribuidos en clases similares o en el fondo (*background*).

Estas visualizaciones destacan el rendimiento sólido del modelo para clasificar el tipo de material, siendo una herramienta clave para la validación y mejora del sistema de clasificación.

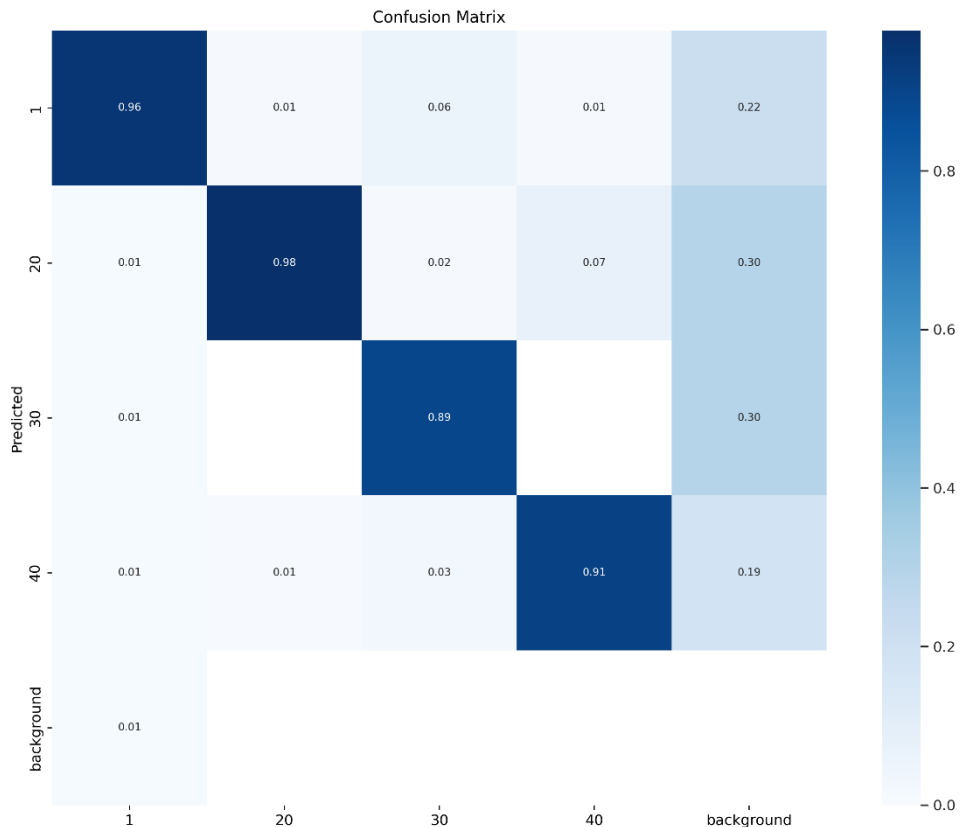


Figura 9. Matriz de confusión en tipo de material

6.2 RESULTADOS SUBTIPO DE MATERIAL

Para evaluar el rendimiento del modelo que clasifica el subtipo de material, se han generado dos representaciones gráficas clave: la curva Precision-Recall y la matriz de confusión.

6.2.1 CURVA PRECISION-RECALL

En este modelo:

- Se obtuvo un mAP (mean Average Precision) de 0.806 para todas las clases a un umbral de IoU de 0.5.
- Las clases individuales muestran un rendimiento consistente, con un balance adecuado entre precisión y recall, excepto en la clase 150.

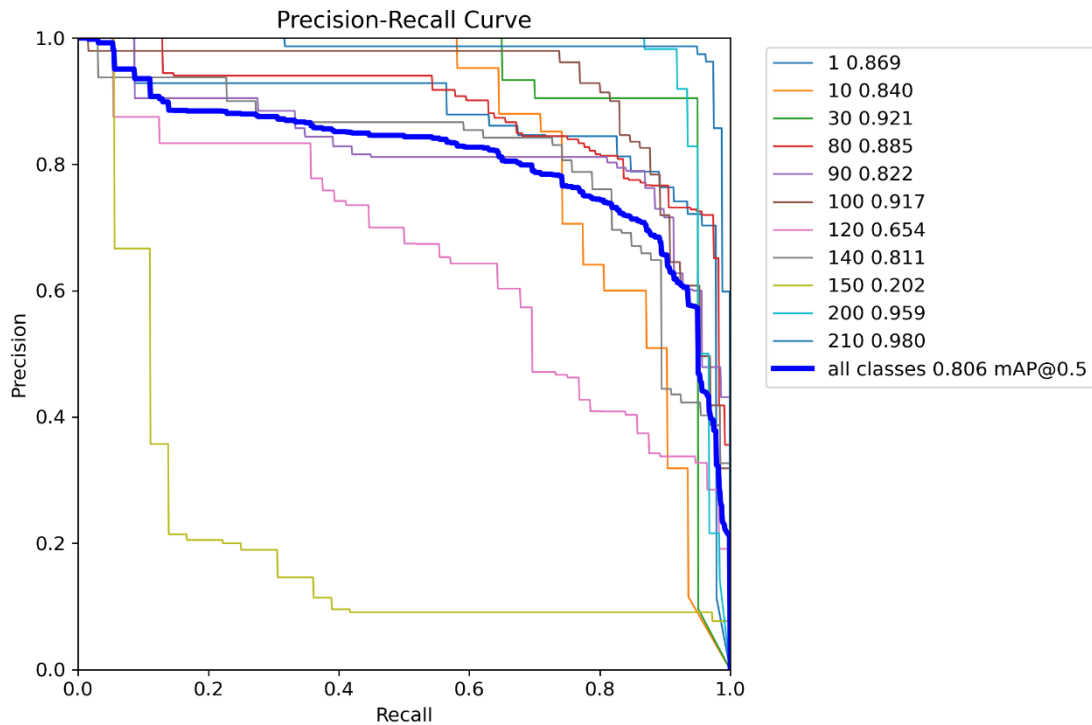


Figura 10. Curva de Precision-Recall para subtipo de material

6.2.2 MATRIZ DE CONFUSIÓN

La matriz de confusión obtenida destaca el sólido rendimiento del modelo para clasificar el subtipo de material, siendo una herramienta fundamental para la validación y mejora del sistema de clasificación.

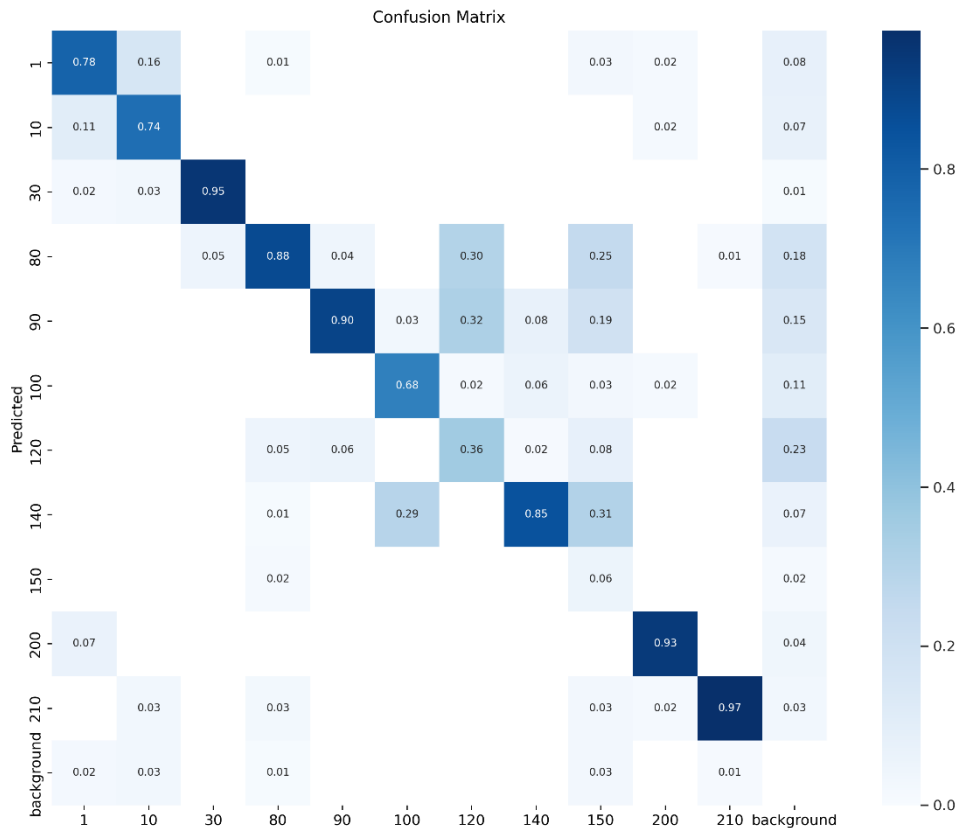


Figura 11. Matriz de confusión en subtipo de material

6.3 RESULTADOS FUNCIÓN DE APOYO

Para evaluar el rendimiento del modelo que clasifica la función de apoyo, se han generado dos representaciones gráficas clave: la curva Precision-Recall y la matriz de confusión.

6.3.1 CURVA PRECISION-RECALL

En este modelo:

- Se obtuvo un mAP (mean Average Precision) de 0.800 para todas las clases a un umbral de IoU de 0.5.

Las clases individuales muestran un rendimiento consistente, con un balance adecuado entre *precision* y *recall*, excepto en la clase 3.

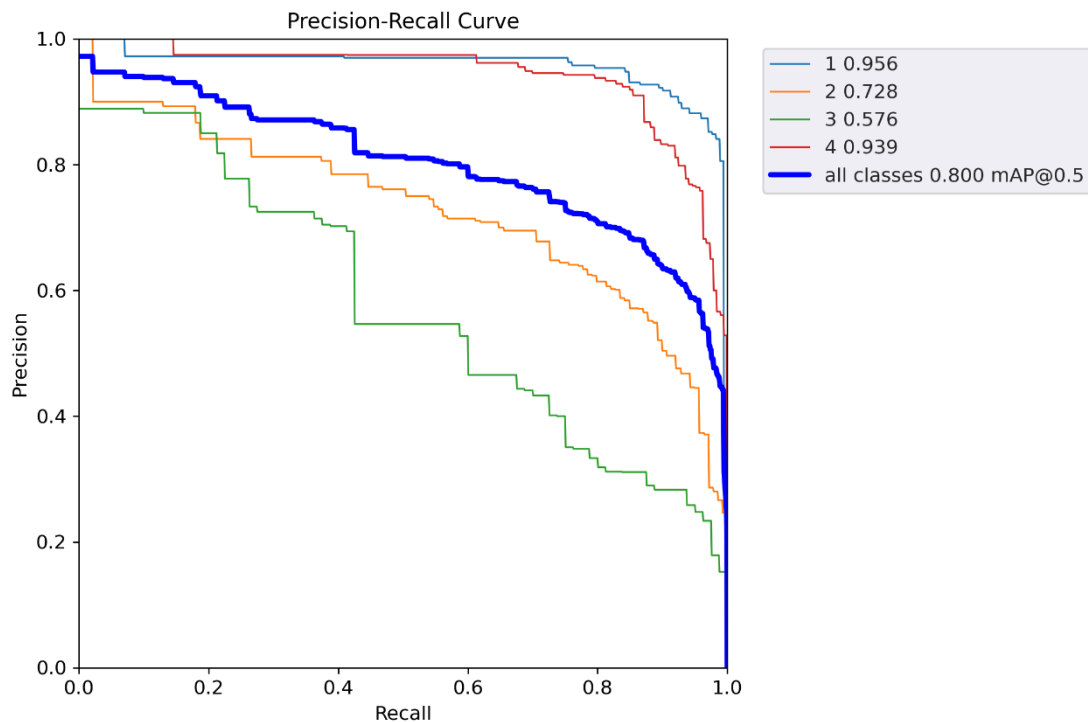


Figura 12. Curva de Precision-Recall para función de apoyo

6.3.2 MATRIZ DE CONFUSIÓN

La matriz de confusión obtenida muestra el sólido rendimiento del modelo para clasificar la función de apoyo de los activos, siendo una herramienta fundamental para la validación y mejora del sistema de clasificación.

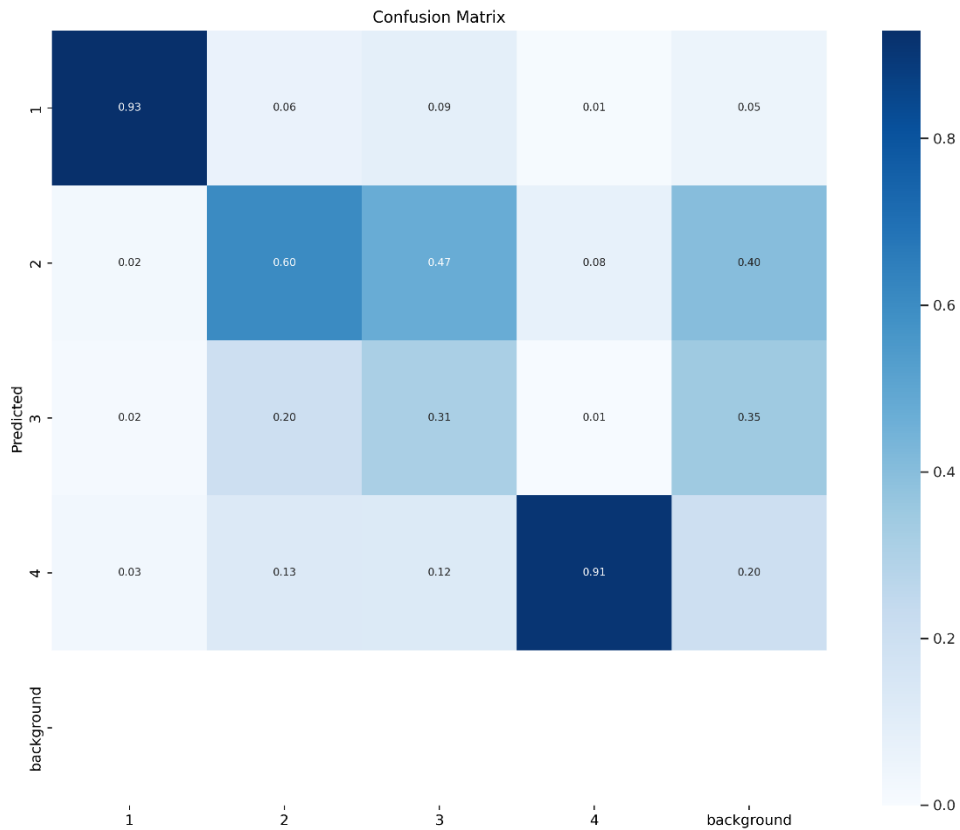


Figura 13. Matriz de confusión en función de apoyo

6.4 RESULTADOS FUNCIÓN ESPECIAL DE APOYO

Para evaluar el rendimiento del modelo que clasifica la función de apoyo, se han generado dos representaciones gráficas clave: la curva Precision-Recall y la matriz de confusión.

6.4.1 CURVA PRECISION-RECALL

En este modelo:

- Se obtuvo un mAP (mean Average Precision) de 0.741 para todas las clases a un umbral de IoU de 0.5.

Las clases individuales muestran un rendimiento consistente, con un balance adecuado entre precisión y recall, excepto en la clase 5.

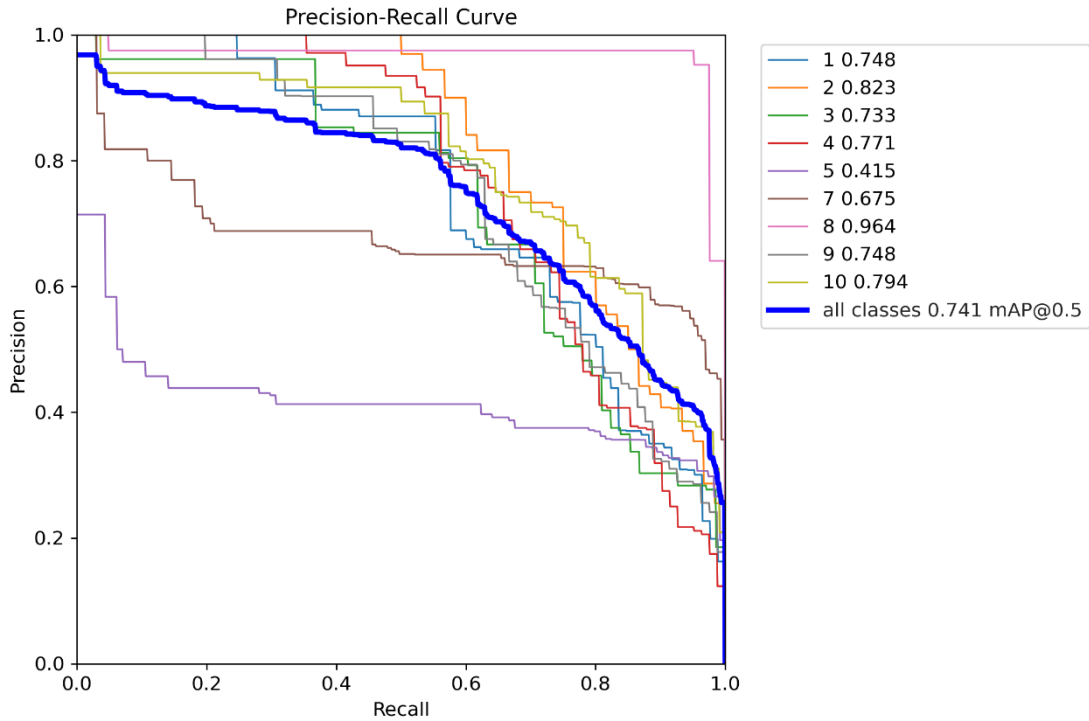


Figura 14. Curva de Precision-Recall para función especial de apoyo

6.4.2 MATRIZ DE CONFUSIÓN

La matriz de confusión resume las predicciones realizadas por el modelo comparándolas con las etiquetas reales. En este caso, el modelo ha demostrado una alta precisión en la mayoría de las clases, con algunos errores distribuidos en clases similares o en el fondo (background), en concreto la clase 5 tiene un error elevado.

Estas visualizaciones destacan el sólido rendimiento del modelo para clasificar la función especial de apoyo de los activos, siendo una herramienta fundamental para la validación y mejora del sistema de clasificación.

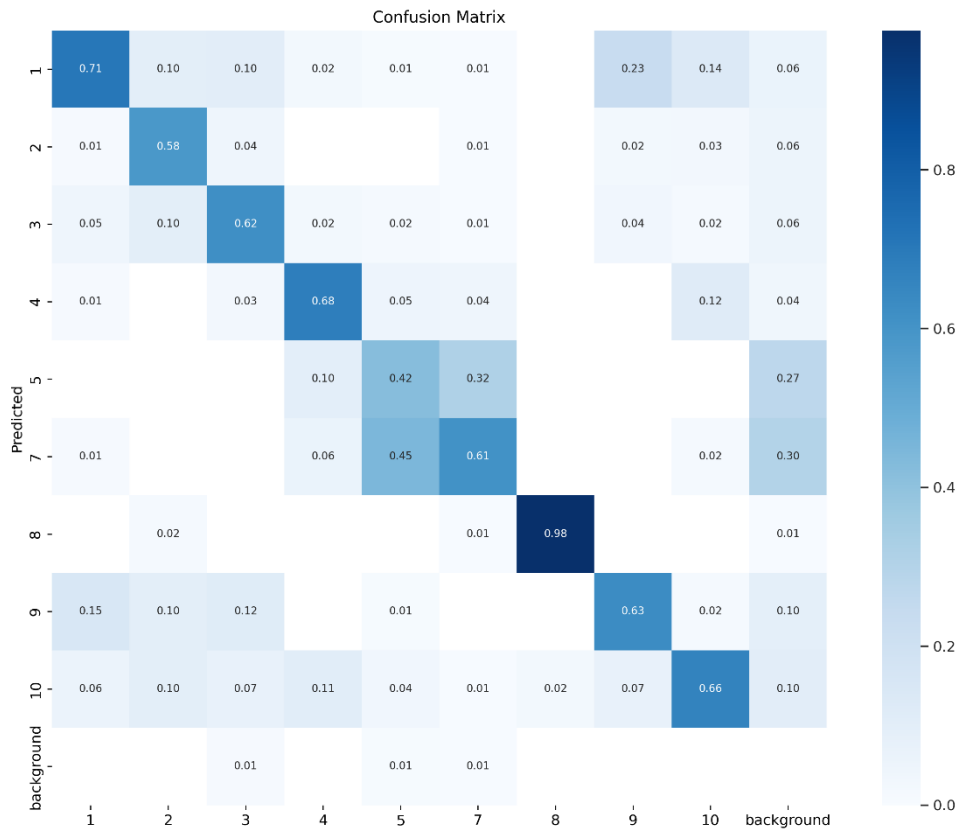


Figura 15. Matriz de confusión en función especial de apoyo

Capítulo 7. CONCLUSIONES Y TRABAJOS FUTUROS

7.1 CONCLUSIONES

El presente proyecto ha demostrado la viabilidad y alto impacto de aplicar técnicas de *computer vision* al problema de la gestión del inventario de activos en redes de distribución. Se ha validado el uso de Grounding DINO para la detección de activos y el *fine-tuning* de modelos YOLO para su clasificación, en concreto, se ha logrado desarrollar un sistema robusto capaz de identificar el activo y, posteriormente, clasificar los diferentes materiales y características de apoyo con alta precisión y eficiencia.

Se pueden destacar los siguientes aspectos en cuanto a los resultados tan satisfactorios obtenidos:

- **Alto rendimiento de los modelos:** Las métricas alcanzadas, como el mAP y las distribuciones favorables en las curvas *Precision-Recall* y matrices de confusión, validan la fiabilidad de los modelos entrenados y el diseño de la solución.
- **Automatización:** El sistema desarrollado proporciona una solución automatizada para la clasificación de activos, reduciendo la intervención manual y aumentando la fiabilidad en la gestión de inventarios en redes de distribución.
- **Potencial comercial:** la herramienta puede aportar gran valor a empresas como Iberdrola en términos de eficiencia operativa y optimización de recursos.

Se puede afirmar que el objetivo del proyecto de **desarrollar una solución basada en técnicas de visión artificial capaz de procesar imágenes de activos físicos para extraer información relevante de inventario** se ha cumplido con éxito.

Los resultados obtenidos consolidan el diseño de la solución técnica implementada y reafirma el valor de la colaboración entre la inteligencia artificial y el sector energético,

destacando cómo este enfoque puede mejorar la gestión de activos y optimizar los recursos en un contexto industrial en constante evolución.

7.2 TRABAJOS FUTUROS

Tras la validación del prototipo, el siguiente paso es evolucionar la solución hacia una herramienta comercializable y fácilmente integrable en entornos reales. Para ello, se proponen las siguientes líneas de desarrollo:

- **Ampliación del modelo:** Incorporar más tipos de activos y características, mejorando la precisión y adaptabilidad. Será clave aumentar el volumen y la diversidad de imágenes etiquetadas.
- **Mejora del modelo:** Durante el desarrollo del modelo no se ha recibido *feedback* ni conocimiento de los técnicos especialistas, desarrollando una implementación muy sencilla del modelo YOLO. Con la correcta transmisión de conocimiento técnico, se podría adaptar la solución y entrenar modelos más específicos y centrados en las partes idóneas del poste para poder clasificar de forma correcta.
- **Interfaz de usuario avanzada:** Rediseñar la interfaz actual para ofrecer una experiencia más intuitiva, con soporte para carga masiva, visualización de resultados y gestión de inventarios. Incluso poder realizar una integración con los sistemas del cliente.
- **Refuerzo del *backend*:** Desarrollar una infraestructura más robusta y escalable, con mecanismos de autenticación, almacenamiento eficiente y trazabilidad.

La base tecnológica actual —basada en Python, librerías *open-source*, contenedores Docker y despliegue en Google Cloud— proporciona un entorno flexible para escalar y adaptar el sistema a distintos operadores del sector energético. Estas mejoras consolidarán la herramienta como un sistema fiable para la gestión visual automatizada de activos.

Capítulo 8. BIBLIOGRAFÍA

- [1] J. D. T. D. J. M. Ross Girshick, "Rich feature hierarchies for accurate object detection and semantic segmentation," *UC Berkeley*, p. 21, 2014.
- [2] D. M. D. F. E. T. Guillaume Maître, «Autonomous inspections of power towers with an UAV,» *Institutional Repository - Research Portal Dépôt Institutionnel - Portail de la Recherche*, 2021.
- [3] S. M. D. F. a. J. V. Anicetus Odo, «Towards the Automatic Visual Monitoring of Electricity Pylons from Aerial Images,» *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2020.
- [4] S. Fang, C. Haiyang, L. Sheng y W. Xiaoyu, «A Framework of Power Pylon Detection for UAV-based Power Line Inspection,» *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, 2020.
- [5] Z. Z. T. R. F. L. H. Z. J. Y. C. L. J. Y. H. S. J. Z. L. Z. Shilong Liu, "Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection," 2023.
- [6] S. D. R. G. A. F. Joseph Redmon, *You Only Look Once: Unified, Real-Time Object Detection*, Washington: Allen Institute for AI, 2016.

ANEXO I - STACK TECNOLÓGICO UTILIZADO

A continuación, se detalla el conjunto de tecnologías y herramientas empleadas en el desarrollo del proyecto, tanto para la implementación de los modelos de visión artificial como para la construcción del entorno web de análisis y su despliegue automatizado en la nube.

Lenguaje de programación y librerías principales

- **Python 3.10:** Lenguaje principal del proyecto, empleado para el desarrollo de modelos, scripts de procesamiento y la aplicación web.
- Librerías de visión artificial y deep learning:
 - Ultralytics YOLO (versión 8): para la clasificación de imágenes.
 - Grounding DINO: modelo de detección open-set.
 - OpenCV: manipulación y procesamiento de imágenes.
 - torch, torchvision: base del entrenamiento y uso de modelos.
- Librerías auxiliares:
 - pandas, numpy: manipulación de datos.
 - matplotlib, seaborn, plotly: visualización de resultados.
 - scikit-learn: particionamiento de datos y evaluación.
 - tqdm: seguimiento de procesos largos.
 - logging: trazabilidad del sistema.
- Desarrollo de la interfaz web
 - Streamlit: Framework ligero para construir interfaces interactivas en Python. Permite al usuario cargar imágenes, ejecutar inferencias y visualizar resultados.

Infraestructura y despliegue

- Docker: Contenerización de la aplicación para facilitar su despliegue y portabilidad.
- Terraform: Automatización de la infraestructura en Google Cloud Platform (GCP).
- Google Cloud Platform (GCP):
 - Compute Engine: para alojar la aplicación.
 - Cloud Storage: almacenamiento de imágenes.
 - Cloud DNS: gestión del dominio personalizado.
 - Cloud Logging: monitorización.
- Nginx: Servidor web utilizado como proxy inverso para la aplicación.
- Jenkins: Herramienta de integración y despliegue continuo (CI/CD).
- GitHub: Repositorio del código fuente y gestión del control de versiones.