



# MÁSTER EN INGENIERÍA INDUSTRIAL Y MÁSTER EN INDUSTRIA CONECTADA

## TRABAJO FIN DE MÁSTER APLICACIÓN DE REDES NEURONALES A LA TRAMITACIÓN DE EXPEDIENTES ADMINISTRATIVOS

Autor: Daniel Ortega Núñez

Director: Brais Chousa Arza

Madrid



Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título  
“Aplicación de redes neuronales a la tramitación de expedientes administrativos”  
en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el  
curso académico 2022/23 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.: Daniel Ortega Núñez

Fecha: 23/08/2023

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Brais Chousa Arza

Fecha: 23/08/2023





# MÁSTER EN INGENIERÍA INDUSTRIAL Y MÁSTER EN INDUSTRIA CONECTADA

## TRABAJO FIN DE MÁSTER APLICACIÓN DE REDES NEURONALES A LA TRAMITACIÓN DE EXPEDIENTES ADMINISTRATIVOS

Autor: Daniel Ortega Núñez

Director: Brais Chousa Arza

Madrid



# Agradecimientos

En este apartado debemos agradecer a la gente que nos ha apoyado, y pudiendo escribir en primera persona, entrar en sentimentalismos.

Primero, y ante todo, quería agradecer a mis padres por el apoyo constante, especialmente durante mis años de formación universitaria. También a mi jefe y director de este proyecto, por la confianza que ha puesto en mí durante este último año y medio, siendo siempre un ejemplo a seguir, y haciéndome aprender día a día. Por último, a todos mis compañeros de la universidad, que ayudándome estos últimos meses han hecho posible compaginar trabajo y estudios.





# APLICACIÓN DE REDES NEURONALES A LA TRAMITACION DE EXPEDIENTES ADMINISTRATIVOS

**Autor: Ortega Núñez, Nombre.**

Director: Chousa Arza, Brais.

Entidad Colaboradora: PricewaterhouseCoopers Asesores de Negocios S.L.

## RESUMEN DEL PROYECTO

En este proyecto se ha desarrollado una solución tecnológica aplicando una red neuronal convolucional (CNN) y reconocimiento óptico de caracteres (OCR) para detectar y leer áreas de interés de documentos. Con esto se propone utilizar el sistema para agilizar y evitar errores en la revisión manual de documentos de expedientes administrativos, específicamente el proyecto se ha centrado en el documento para acreditar la representación legal.

**Palabras clave:** YOLO, OCR, Inteligencia artificial, Expedientes, Red neuronal, CNN, Revisión documental, Representación legal.

### 1. Introducción

Este proyecto de fin de máster se centra en el uso y aplicación de inteligencia artificial y técnicas de OCR a la tramitación de expedientes administrativos, específicamente enfocado en el proceso de concesión de ayudas.

En la gestión de ayudas, un gran número de solicitudes puede traducirse en una acumulación de expedientes que resulta difícil de tramitar en tiempo y forma por parte del personal encargado de la revisión documental, ya que, es una tarea que puede requerir la revisión de una gran cantidad de documentos, lo cual, puede ocasionar problemas concretos en este tipo de procedimientos como:

- Retrasos en la resolución de las solicitudes y, por tanto, descontento por parte de los solicitantes debido a la demora en la concesión de la ayuda.
- Concesión de ayudas a solicitantes con documentación presentada errónea o inválida, debido a la incapacidad de revisar adecuadamente todos los expedientes en el plazo establecido.
- Pérdida de ayudas por parte de los potenciales beneficiarios en el caso de que existan limitaciones presupuestarias o temporales, al quedar expedientes sin revisar.
- Aumento de los costes de personal encargado de la revisión documental.

Teniendo en cuenta esta problemática, el proyecto persigue la agilización de la tramitación de expedientes mediante la aplicación de redes neuronales y tecnologías en el ámbito de la inteligencia artificial, desarrollando un sistema que sea capaz de revisar la documentación aportada por los solicitantes. De esta forma, se podrán solventar los problemas mencionados, y facilitar el trabajo a los revisores documentales a la vez que se mejoran los servicios ofrecidos a los ciudadanos.

## 2. Definición del proyecto

El proyecto se centrará en la creación de un sistema de revisión de documentos adaptado específicamente a los expedientes de los solicitantes en representación, es decir, al documento de representación legal necesario para permitir la presentación de una solicitud por parte de un representante. Para ello, la solución utilizará como tecnologías principales una red neuronal basada en YOLO para localizar los campos de interés del documento (entrenada para detectar NIF/NIE), recortarlos, y procesarlos mediante un OCR para digitalizar dicha información y compararla con la información real del solicitante.

## 3. Descripción del sistema

Para llevar esto a cabo, el sistema contará, en primer lugar, con una extracción de información relacionada con el expediente a analizar, es decir, datos como el identificador de expediente, NIF/NIE del solicitante y representante y la ubicación del documento a analizar.

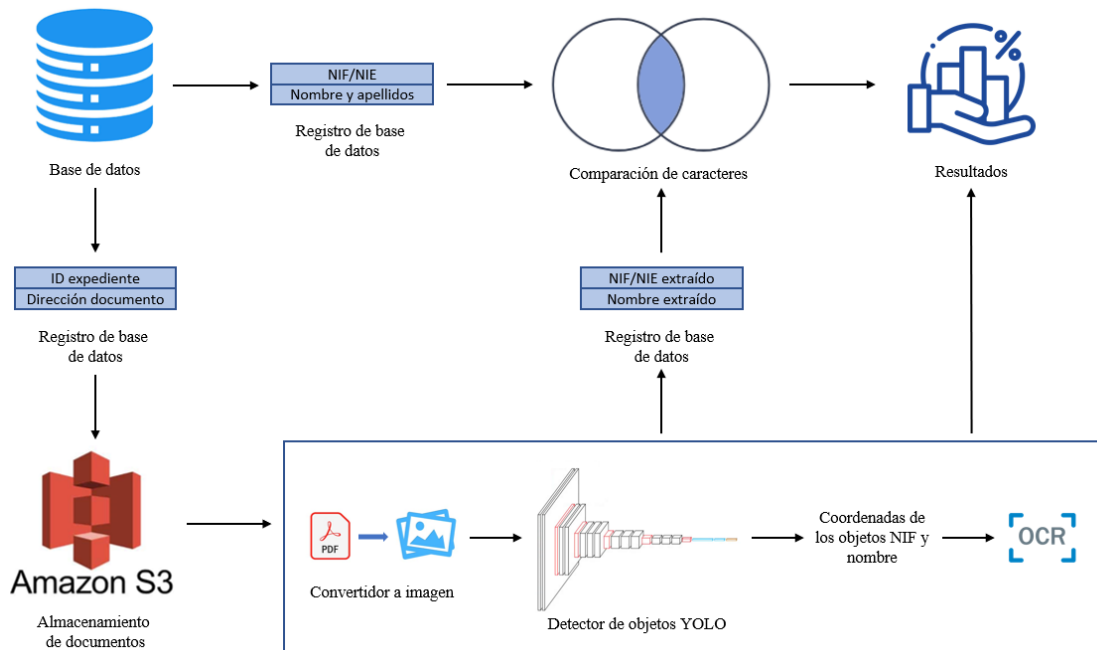
En segundo lugar, y a partir de la información extraída sobre la ubicación del documento, se ejecuta para este la solución implementada. Esta solución consiste en varios pasos:

- La transformación del documento en imagen.
- La detección de los campos de interés dentro del documento transformado mediante un detector basado en YOLO y entrenado para localizar NIF/NIE
- Obtener las coordenadas de las detecciones realizadas, la confianza de la detección, y la clase del objeto al que corresponde.
- A partir de las coordenadas, se recorta la imagen para obtener una nueva imagen de cada objeto detectado.
- Digitalizar la información de interés correspondiente a las imágenes recortadas mediante el OCR.

En tercer lugar, a partir de la información obtenida en este último paso, se realiza una comparación de caracteres con la información real de la base de datos correspondiente al expediente, y se obtiene un porcentaje como resultado. Mediante este porcentaje de similitud, se concreta un umbral a partir del cual, se decidirá si el resultado de la verificación del documento es positiva o negativa.

Por último, los resultados de la ejecución de YOLO, del OCR, y de la comparación de caracteres se extraerán con formato Excel junto con la información extraída del documento para registrarlos en base de datos cuando se considere adecuado.

A continuación, se muestra en la *Ilustración 1* un esquema general del diseño comentado:



*Ilustración 1: Diseño del sistema.*

Cabe destacar, que el desarrollo de este proyecto está enfocado a aportar valor adicional a una plataforma ya existente de tramitación de expedientes, por lo que, el diseño corresponde a una personalización específica para dicha plataforma. Se realizará una versión de prueba, para la que se entrenará un modelo simplificado con el que se realizará únicamente una revisión de NIF/NIE en el documento y comprobar si el funcionamiento resulta válido para desarrollar un sistema con especificaciones superiores.

#### 4. Resultados

Una vez desarrollada la solución, se han analizado los resultados obtenidos en relación con el modelo entrenado, y el umbral de verificación concretado.

Con respecto al modelo, por un lado se ha evaluado el tiempo de ejecución y se ha determinado que no es influyente en relación con el tiempo de una ejecución completa. Es relevante mencionar que, aunque el tiempo de ejecución del modelo podría tener un impacto en la escala de horas si el número de expedientes supera los cientos de miles, al no ser un proceso dependiente del tiempo (ya que la tramitación puede seguir realizándose durante la ejecución del proceso), se considera la posibilidad de usar arquitecturas más complejas en futuras mejoras aun sacrificando cierta velocidad para obtener mayor precisión en la detección.

Por otro lado se ha analizado el modelo a partir de los resultados del conjunto de datos de validación. Mediante esta evaluación, se ha podido observar que el modelo utilizado para la detección de objetos presenta overfitting. Esto se debe a que el modelo se ajusta excesivamente a una plantilla específica, lo que limita su adaptabilidad a diferentes formatos de documentos.

Para finalizar, se discute sobre el umbral de decisión para la verificación de documentos, ya que tiene un enfoque flexible con umbrales de similitud más o menos estrictos que permitirán declarar como positiva la verificación de un mayor o menor número de documentos según corresponda. Además, se destaca que el umbral actual no considera ciertos parámetros como la confianza del OCR.

## 5. Conclusiones

Tras la realización de este proyecto podemos constatar que el sistema prueba su funcionalidad al ser capaz de extraer y verificar correctamente el número NIF/NIE del documento. No obstante, para poder corroborar este requisito, es necesario adquirir información adicional, como los campos de firma. Esto es debido a la falta de validez que presenta un documento para demostrar la representación si no está debidamente firmado. Por lo tanto, es imprescindible desarrollar y entrenar un nuevo modelo que sea capaz de detectar firmas como elementos adicionales, permitiendo así ajustar el umbral de verificación en función de estas detecciones personalizadas.

Por ello, se propone como un área de desarrollo futuro la mejora en el entrenamiento del modelo. Como hemos mencionado anteriormente, la rapidez en la ejecución no es un factor clave en este proceso. En este nuevo planteamiento de entrenamiento, se consideraría implementar mejoras como: la incorporación de la detección de firmas como una categoría adicional, y el uso de un volumen de datos más amplio.

Adicionalmente, sería aconsejable utilizar una arquitectura de modelo más compleja que influyese en la velocidad de ejecución, pudiéndose realizar de forma independiente al proceso de revisión manual, y sin interrupciones.

## 6. Referencias

[1] Saarthi AI. (May 9, 2019). How to Build Your Own OCR. Medium. [En línea]. Disponible en: <https://medium.com/saarthi-ai/how-to-build-your-own-ocr-a5bb91b622ba>

[2] Dillon Reis, J., Kupec, J., Hong, J., & Daoudi, A. (May 17, 2023). Real-Time Flying Object Detection with YOLOv8. Disponible en: <https://arxiv.org/abs/2305.09972>

[3] Portal de Ayudas del Ministerio de Economía y Competitividad. [En línea]. Solicitudes. Disponible en: <https://portalayudas.mineco.gob.es/THD/solicitudes/Paginas/Solicitudes.aspx>

[4] EnriqueAV. (May 12, 2018). Detección de objetos con YOLO: implementaciones y cómo usarlas. Disponible en: <https://medium.com/@enriqueav/detecci%C3%B3n-de-objetos-con-yolo-implementaciones-y-como-usarlas-c73ca2489246>

# USE OF NEURAL NETWORKS TO AUTOMATE ADMINISTRATIVE DOSSIER PROCESSING

**Author: Ortega Núñez, Daniel.**

Supervisor: Chousa Arza, Brais.

Collaborating Entity: PricewaterhouseCoopers Asesores de Negocios S.L.

## ABSTRACT

In this project a technological solution has been developed by applying a convolutional neural network (CNN) and optical character recognition (OCR) to detect and read areas of interest in documents. With this, it is proposed to use the system to speed up and avoid errors in the manual review of documents in administrative dossiers, specifically the project has focused on the document to accredit legal representation.

**Keywords:** YOLO, OCR, Artificial intelligence, Dossiers, Neural network, CNN, Document review, Legal representation.

## 1. Introduction

This master's thesis project focuses on the use and application of artificial intelligence and OCR techniques to the processing of administrative files, specifically focused on the process of granting aid.

In the management of grants, a large number of applications can result in an accumulation of files that is difficult to process in a timely manner by the staff in charge of document review, since it is a task that may require the review of a large number of documents, which can cause specific problems in this type of procedures such as:

- Delays in the resolution of the applications and, therefore, dissatisfaction on the part of the applicants due to the delay in the granting of the aid.
- Granting of aid to applicants with erroneous or invalid documentation, due to the inability to adequately review all the files within the established deadline.
- Loss of aid to potential beneficiaries in the event of budgetary or time constraints, due to unreviewed files.
- Increased costs for personnel in charge of document review.

Considering this situation, the project aims to speed up the processing of files through the application of neural networks and technologies in the field of artificial intelligence, developing a system capable of reviewing the documentation provided by the applicants. In this way, the mentioned problems will be solved, and the work of document reviewers will be easier while improving the services offered to citizens.

## 2. Project definition

The project will focus on the creation of a document review system specifically adapted to the file of represented applicants, i.e., the legal representation document required to allow the presentation of an application by a representative. For this, the solution will use as its core technology a neural network based on YOLO to locate the document's

fields of interest (trained to detect NIF/NIE), cut them out, and process them through an OCR to digitize this information and compare it with the applicant's real information.

### **3. Description of the system**

To do this, the system will have, firstly, an extraction of information related to the dossier to be analyzed, i.e., data such as the dossier identifier, NIF/NIE of the applicant and representative and the location of the document to be analyzed.

Secondly, based on the extracted information about the location of the document, the implemented solution is executed for the document. This solution consists of several steps:

- The transformation of the document into an image.
- The detection of the fields of interest within the transformed document by using a detector based on YOLO and trained to locate NIF/NIE.
- Obtain the coordinates of the detections carried out, the confidence of the detection, and the class of the object to which it corresponds.
- From the coordinates, crop the image to obtain a new image of each detected object.
- Digitize the information of interest corresponding to the cropped images using OCR.

Thirdly, based on the information obtained in this last step, a comparison of characters is made with the actual information in the database corresponding to the dossier, and a percentage is obtained as a result. By this percentage of similarity, a threshold is established from which it will be decided whether the result of the document verification is positive or negative.

Finally, the results of the YOLO performance, OCR, and character comparison will be extracted in Excel format along with the information extracted from the document to be recorded in the database when it is considered appropriate.

A general outline of the annotated design is shown in *Illustration 1* below:

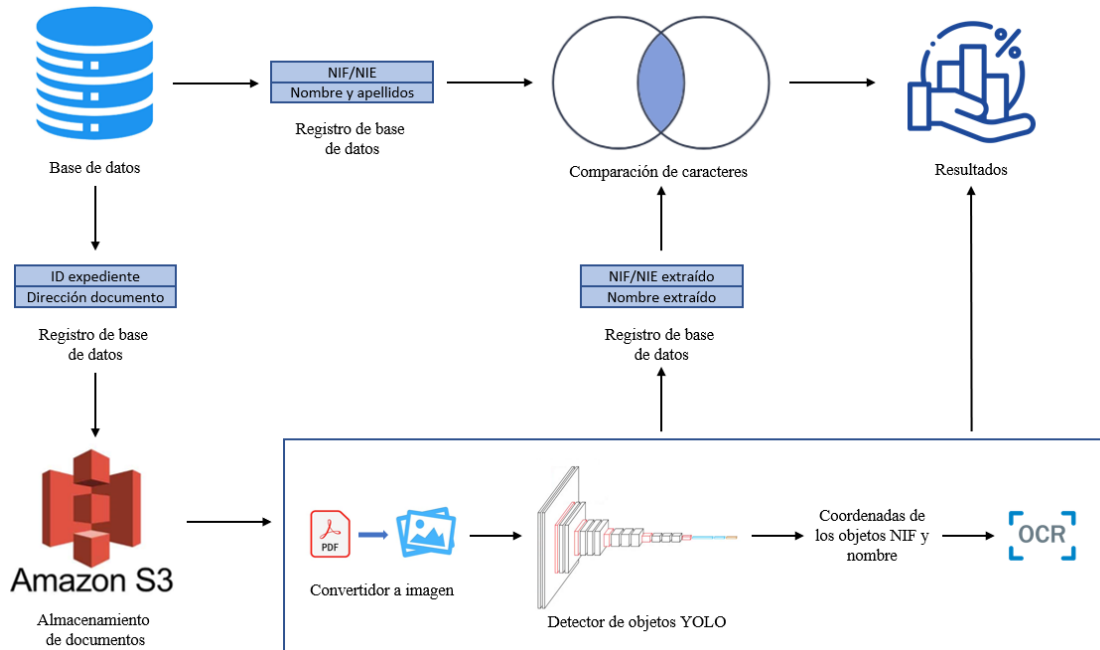


Illustration 1: System design.

It should be noted that the development of this project is focused on providing additional value to an existing platform for the processing of dossiers, so the design corresponds to a specific customization for that platform. A test version will be made, for which a simplified model will be trained with NIF/NIE detection in the document and checking if the functionality is valid for the development of a system with higher specifications.

#### 4. Results

Once the solution has been developed, the results obtained have been analyzed in relation to the trained model and the verification threshold.

Regarding the model, on one hand, the execution time has been evaluated and it has been determined that it is not relevant in relation to the time of a complete execution. It is important to mention that, although the execution time of the model could have an impact on the scale of hours if the number of files exceeds hundreds of thousands, as it is not a time-dependent process (since the procedure can continue to be carried out during the execution of the process), the possibility of using more complex architectures in future improvements is under consideration, even sacrificing some speed to obtain greater accuracy in detection.

On the other hand, the model has been analyzed from the results of the validation data set. Through this evaluation, it has been observed that the model used for object detection presents overfitting. This is due to the fact that the model adjusts excessively to a specific template, which limits its adaptability to different document formats.

Finally, the decision threshold for document verification is discussed, as it has a flexible approach with more or less strict thresholds of similarity that will allow declaring as positive the verification of a greater or lesser number of documents as corresponds. In

addition, it is noted that the current threshold does not consider certain parameters such as OCR confidence.

## 5. Conclusions

After the accomplishment of this project, we can see that the system proves its functionality by being able to correctly extract and verify the NIF/NIE of the document. However, in order to be able to corroborate this requirement, it is necessary to acquire additional information, such as signature fields. This is due to the lack of validity of a document to prove representation if it is not properly signed. Therefore, it is essential to develop and train a new model that is able to detect signatures as additional elements, allowing to adjust the verification threshold based on these customized detections.

Therefore, improvement in model training is proposed as an area for future development. As mentioned above, speed of execution is not a key factor in this process, so in this new training approach, it would be considered to implement improvements such as: the incorporation of signature detection as an additional category, and the use of a larger data volume.

Additionally, it would be beneficial to use a more complex model architecture that influences the speed of execution, which can be performed independently of the manual review process, and without interruptions.

## 6. References

- [1] Saarthi AI. (May 9, 2019). How to Build Your Own OCR. Medium. [En línea]. Disponible en: <https://medium.com/saarthi-ai/how-to-build-your-own-ocr-a5bb91b622ba>
- [2] Dillon Reis, J., Kupec, J., Hong, J., & Daoudi, A. (May 17, 2023). Real-Time Flying Object Detection with YOLOv8. Disponible en: <https://arxiv.org/abs/2305.09972>
- [3] Portal de Ayudas del Ministerio de Economía y Competitividad. [En línea]. Solicitudes. Disponible en: <https://portalayudas.mineco.gob.es/THD/solicitudes/Paginas/Solicitudes.aspx>
- [4] EnriqueAV. (May 12, 2018). Detección de objetos con YOLO: implementaciones y cómo usarlas. Disponible en: <https://medium.com/@enriqueav/detecci%C3%B3n-de-objetos-con-yolo-implementaciones-y-como-usarlas-c73ca2489246>



## *Índice de la memoria*

<b>Capítulo 1. Introducción.....</b>	<b>5</b>
1.1 Motivación del proyecto.....	6
1.2 Propósito del proyecto.....	6
<b>Capítulo 2. Descripción de las tecnologías .....</b>	<b>7</b>
2.1 YOLO (You Only Look Once) .....	7
2.2 Optical Character Recognition (OCR) .....	8
2.3 Principales herramientas utilizadas .....	9
2.3.1 Visual Studio Code.....	9
2.3.2 Google Colab.....	9
2.3.3 GitHub .....	9
2.3.4 PostgreSQL.....	10
2.3.5 Computer Vision Annotation Tool (CVAT) .....	10
2.3.6 Amazon S3 .....	10
<b>Capítulo 3. Estado de la cuestión .....</b>	<b>12</b>
<b>Capítulo 4. Definición del trabajo.....</b>	<b>15</b>
4.1 Justificación.....	15
4.2 Objetivos .....	16
4.3 Metodología.....	16
4.4 Planificación y estimación económica .....	17
<b>Capítulo 5. Sistema/modelo desarrollado .....</b>	<b>20</b>
5.1 Análisis del procedimiento actual .....	20
5.1.1 Sistema de revisión actual .....	20
5.1.2 Requerimientos del sistema actual en la solución .....	23
5.2 Diseño de la solución .....	24
5.2.1 Estructura del proyecto.....	26
5.2.2 Base de datos .....	30
5.2.3 Interfaz y casos de uso.....	32
5.3 Entrenamiento del modelo YOLO.....	34

5.3.1 Creación de dataset de entrenamiento .....	34
5.3.2 Entrenamiento del modelo YOLO.....	35
5.4 Implementación .....	37
5.4.1 Extracción y preparación de datos.....	37
5.4.2 Conversión de PDF a imágenes.....	37
5.4.3 Detección de posiciones de NIF con YOLO .....	38
5.4.4 Procesamiento de la imagen recordada .....	40
5.4.5 Extracción y OCR de NIF .....	41
5.4.6 Procesamiento de texto.....	41
5.4.7 Comparación de cadenas de caracteres y Verificación.....	43
5.4.8 Generación de resultados y almacenamiento.....	44
<b>Capítulo 6. Análisis de resultados.....</b>	<b>45</b>
6.1 Tiempos de ejecución de YOLO .....	45
6.2 Evaluación del modelo .....	46
6.3 Umbral de validación .....	47
<b>Capítulo 7. Conclusiones y trabajos futuros.....</b>	<b>49</b>
<b>Capítulo 8. Bibliografía .....</b>	<b>51</b>
<b>ANEXO I: Alineación con los Objetivos de Desarrollo Sostenible .....</b>	<b>53</b>

## *Índice de figuras*

Figura 1: Esquema de funcionamiento YOLO .....	8
Figura 2: Flujo de tramitación de expedientes.....	22
Figura 3: Diseño general de la solución. ....	26
Figura 4: Estructura del proyecto. ....	27
Figura 5: Diagrama de los archivos "data".....	28
Figura 6: Diagrama de los archivos "runs".....	29
Figura 7: Interfaz de revisión manual.....	33
Figura 8: Interfaz de verificación automática.....	33
Figura 9: Parámetros de las versiones de YOLO Ultralytics.....	36
Figura 10: Identificación en documento.....	38
Figura 11: Ejemplo de umbral binario.....	40
Figura 12: Imagen original.....	41
Figura 13: Imagen con escala de grises.....	41
Figura 14: Imagen con umbral binario.....	41
Figura 15: Formatos NIF/NIE.....	42
Figura 16: Archivo Excel de resultados.....	44
Figura 17: Consola con tiempos de ejecución de YOLO.....	45
Figura 17: Objetivos de desarrollo sostenible.....	53

## *Índice de tablas*

Tabla 1: Cronograma.....	17
Tabla 2. Especificaciones del equipo utilizado. ....	18
Tabla 3: Resumen de costes estimados.....	19
Tabla 4: Tabla en base de datos.....	32

## **Capítulo 1. INTRODUCCIÓN**

Este proyecto fin de máster se centra en el uso y aplicación de inteligencia artificial a la tramitación de expedientes administrativos, concretamente a un proceso de concesión de ayudas. Este tipo de procedimientos administrativos pueden variar dependiendo de las características y requisitos establecidos en las respectivas regulaciones de cada programa.

Un gran número de solicitudes de concesión puede generar un volumen de expedientes difícil de gestionar en tiempo y forma por parte del personal encargado de la revisión documental. De cara a que un expediente pueda ser resuelto de forma estimatoria, es necesario demostrar por parte del solicitante el cumplimiento de estos requisitos, ya sea mediante aportación de documentos o consultas telemáticas a Servicios de Administración Digital (previamente autorizadas por el solicitante).

Como se puede inferir, la tramitación de expedientes administrativos es una tarea que puede requerir la revisión de una gran cantidad de documentos, pudiendo ocasionar una acumulación de trabajo a los revisores y largos tiempos de espera a los solicitantes.

A continuación, se presentan problemas concretos que pueden darse en este tipo de procedimientos:

- Retrasos en la resolución de las solicitudes y, por tanto, descontento por parte de los solicitantes debido a la demora en la concesión de la ayuda.
- Concesión de ayudas a solicitantes con documentación presentada errónea o inválida, debido a la incapacidad de revisar adecuadamente todos los expedientes en el plazo establecido.
- Pérdida de ayudas por parte de los potenciales beneficiarios en el caso de que existan limitaciones presupuestarias o temporales, al quedar expedientes sin revisar.
- Aumento de los costes de personal encargado de la revisión documental.

Mediante la solución diseñada e implementada en este proyecto se podrán solventar los problemas mencionados, facilitando el trabajo a los revisores documentales a la vez que se mejoran los servicios ofrecidos a los ciudadanos.

## **1.1 MOTIVACIÓN DEL PROYECTO**

La motivación principal para la realización de este trabajo es poder afrontar y anticiparse a posibles situaciones adversas que puedan darse en la tramitación de un alto volumen de expedientes administrativos, solventando específicamente la problemática causada por el carácter manual del proceso de revisión documental y sus potenciales consecuencias negativas.

Otro aliciente para la formulación de este proyecto es la potencial mejora en la experiencia de usuario al simplificar y agilizar el proceso de tramitación de expedientes, evitando además errores en la revisión documental.

Por último, otra motivación del proyecto es poder llevar a cabo la implementación de esta solución como una nueva funcionalidad de una plataforma en producción, aportando un valor adicional a la misma, mientras se analizan las capacidades de diferentes tecnologías punteras, identificando sus ventajas e inconvenientes.

## **1.2 PROPÓSITO DEL PROYECTO**

El proyecto tiene como propósito el diseño e implementación de una solución que realice la revisión de documentos digitalizados de manera automatizada, agilizando la tramitación de expedientes administrativos mediante la aplicación de tecnologías en el ámbito de la inteligencia artificial.

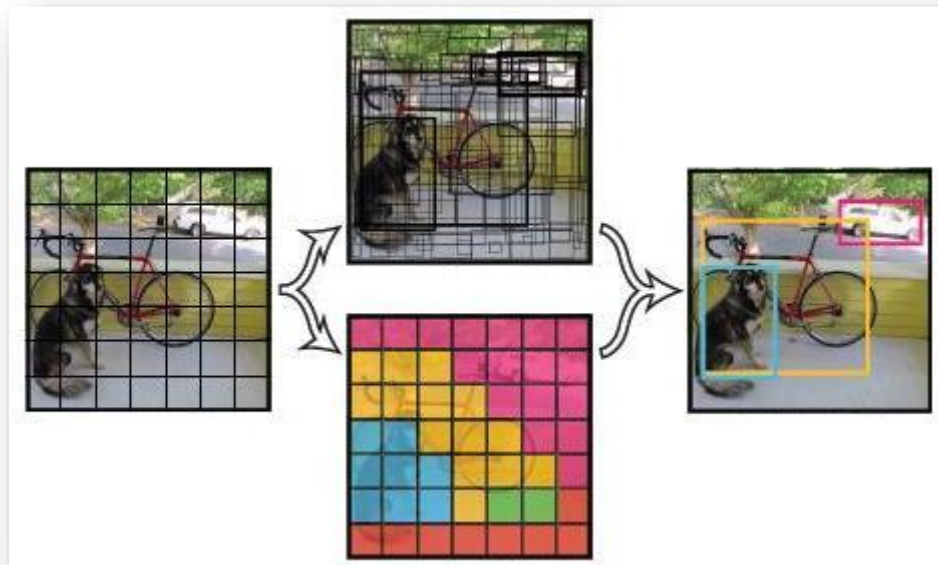
## **Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS**

Mediante la realización de este proyecto se va a desarrollar una solución basada en diferentes tecnologías, las cuales se explicarán brevemente en el presente capítulo. Así mismo, se van a presentar tanto las herramientas utilizadas y de interés, como los servicios que se han utilizado o se podrían utilizar en futuras mejoras de la solución.

### **2.1 YOLO (YOU ONLY LOOK ONCE)**

YOLO[2] es un algoritmo de detección de objetos en tiempo real, cuyo modelo se implementó como una red neuronal convolucional (en inglés, Convolutional Neural Network o CNN), que permite detectar en una imagen múltiples clases de objetos y delimitar sus ubicaciones. Como su nombre indica, esta detección la realiza en una única observación de la imagen, logrando así una velocidad de ejecución superior a la de otros algoritmos de identificación de objetos, a cambio de una reducción de su precisión.

El algoritmo divide la imagen en cuadrículas, a partir de las cuales predice posibles ubicaciones del objeto buscado, calculando el nivel de probabilidad de que se encuentre en cada conjunto de cuadrículas. Una vez obtenidas todas las predicciones, se eliminan los conjuntos que estén por debajo de un nivel de probabilidad determinado, dejando así únicamente la predicción que tiene mayor probabilidad de éxito o, lo que es lo mismo, menor incertidumbre.



*Figura 1: Esquema de funcionamiento YOLO[9][2].*

En el contexto de este proyecto, YOLO se emplea para identificar y delimitar regiones de interés en las imágenes digitalizadas de los documentos, obteniendo las coordenadas de las mismas para el tratamiento de la imagen.

## **2.2 OPTICAL CHARACTER RECOGNITION (OCR)**

OCR es una tecnología que permite digitalizar texto, tanto escrito a mano como digitalmente, a partir de imágenes que contienen datos tratables como una cadena de caracteres.

Esta tecnología identifica símbolos y caracteres de un determinado alfabeto y los compara con los patrones correspondientes. Los resultados de este tipo de digitalización pueden verse muy afectados por la calidad y por los diferentes niveles de grises debido a ruidos de la imagen.



En el contexto de este proyecto, OCR se utiliza para extraer la información de las regiones de interés de los documentos digitalizados, seleccionadas con la tecnología comentada anteriormente (YOLO), lo que permitirá el tratamiento ágil y sencillo de los datos.

## **2.3 PRINCIPALES HERRAMIENTAS UTILIZADAS**

A continuación, se presentan las herramientas y servicios utilizados durante la realización de este proyecto para desarrollar la solución planteada:

### **2.3.1 VISUAL STUDIO CODE**

Visual Studio Code[16] es un entorno de desarrollo integrado, se utiliza principalmente para escribir, depurar y ejecutar código de programación. Se trata de un editor de texto con amplia flexibilidad gracias a su compatibilidad con diferentes lenguajes.

Se ha utilizado principalmente para estructurar el proyecto y desarrollar diferentes scripts en Python.

### **2.3.2 GOOGLE COLAB**

Google Colab es una plataforma que proporciona un entorno gratuito con acceso a recursos de unidad de procesamiento gráfico (en inglés Graphics Processing Unit, GPU), donde se pueden llevar a cabo entrenamientos de modelos de inteligencia artificial. Se integra con Google Drive para facilitar el almacenamiento de los datos de entrenamiento y de los diferentes resultados obtenidos.

Durante el proyecto se ha utilizado para realizar varios entrenamientos de modelos para la solución.

### **2.3.3 GITHUB**

GitHub es una plataforma de almacenamiento de código que permite a los desarrolladores trabajar de forma conjunta en diferentes proyectos. Se emplea para el control de versiones

del código del proyecto, facilitando la colaboración, revisión y seguimiento de cambios. Además, permite compartir información de proyectos entre desarrolladores y exponerlos de forma gratuita.

Durante el proyecto se ha utilizado para obtener información de las diferentes arquitecturas de los modelos tratados de YOLO, así como documentación sobre los mismos.

### **2.3.4 POSTGRESQL**

PostgreSQL es un sistema de gestión de bases de datos relacional que utiliza el lenguaje Structured Query Language (SQL) para administrar y consultar datos. Es conocido por ser un sistema de código abierto, altamente extensible, enfocado en la integridad de los datos y con la capacidad de soportar datos complejos y relaciones.

Durante el desarrollo de la solución se ha utilizado para consultar información relevante sobre su funcionamiento y para la extracción de datos asociados con los resultados del proyecto.

### **2.3.5 COMPUTER VISION ANNOTATION TOOL (CVAT)**

CVAT[17] es una herramienta de anotación de imágenes y videos diseñada específicamente para tareas de visión artificial. Permite etiquetar objetos y regiones de interés en imágenes, lo que es necesario para llevar a cabo el entrenamiento de cualquier modelo de detección de objetos.

Esta herramienta se ha utilizado para etiquetar las regiones de interés seleccionadas con YOLO y generar un conjunto de datos propios para su entrenamiento.

### **2.3.6 AMAZON S3**

Amazon S3[1] es un servicio de almacenamiento de objetos en la nube con gran escalabilidad ofrecido por AWS (Amazon Web Services). Se utiliza para almacenar, llevar a cabo un control de versiones y recuperar datos, como imágenes y archivos, de manera eficiente.

Durante este proyecto se ha utilizado como repositorio de almacenamiento de todos los documentos de los expedientes administrativos.

## **Capítulo 3. ESTADO DE LA CUESTIÓN**

Durante este capítulo, se analizará el estado actual de las soluciones tecnológicas en el ámbito de la revisión automática de documentos administrativos, con el objetivo de conocer cómo se han abordado problemas similares, identificar oportunidades de mejora, así como estudiar tecnologías y herramientas que puedan aportar valor al presente proyecto.

El origen de la problemática planteada es el carácter manual del proceso de revisión documental de los expedientes administrativos. En este sentido, se pretende automatizar dicha revisión, abaratando costes de personal e incrementando la satisfacción de los solicitantes al darles respuesta en tiempo y forma.

Para solventar dicha problemática, se necesitan soluciones tecnológicas que sean capaces de: detectar textos dentro de un documento, extraerlos, y poder comparar la extracción realizada con la información verídica que corresponda. En esta línea, existen diferentes tecnologías y/o combinaciones de ellas que pueden ser válidas.

Una solución efectiva para la extracción de texto de documentos es la tecnología OCR, que permite convertir documentos digitalizados que contengan texto en cadenas de caracteres reconocibles por un sistema de información.

Las soluciones OCR extraen por defecto todo el texto que son capaces de reconocer dentro de un documento digitalizado, muchas veces sin seguir un orden lógico o estructurado. Cabe destacar que en múltiples ocasiones no es necesaria la extracción de todo el texto que contiene un documento, siendo únicamente necesario poner el foco en una zona específica donde se encuentran datos específicos. Para evitar esta ineficiencia y trabajar únicamente sobre las datos de interés, existen algoritmos basados en redes neuronales que permiten identificar y extraer de manera más precisa la información de los documentos digitalizados. En este sentido, se hace necesaria la detección de áreas o regiones de interés mediante alguno de los siguientes enfoques:

- Detectores basados en regiones: solución en dos pasos que, en primer lugar, identifica los campos o regiones de interés y, posteriormente, clasifica y determina la clase y posición de los datos. Algoritmos como Faster R-CNN[14] (Regions with Convolutional Neural Network features) y R-FCN[9] (Region-based Fully Convolutional Networks) adoptan este enfoque.
- Detectores de un solo disparo: solución que, como su nombre indica, realiza el proceso en un único paso, llevando a cabo la detección y clasificación mediante una única observación de la imagen. Esto se realiza mediante una red neuronal que predice tanto las dimensiones delimitadoras del dato de interés, como la clasificación del contenido. Permite una detección más rápida que los detectores basados en regiones. Un ejemplo de este enfoque es YOLO.

Estas técnicas de procesamiento de regiones pueden ser combinadas con tecnología OCR para obtener un resultado más ágil y preciso, no siendo necesaria la extracción de toda la información del documento y facilitando el tratamiento de los datos.

Existen soluciones que utilizan la combinación de estas tecnologías para la detección y lectura en tiempo real de, por ejemplo, matrículas de coche. Esto demuestra que, aplicando dichas tecnologías de forma correcta, se pueden utilizar en el marco del presente proyecto.

Por otro lado, comentar que actualmente existen servicios de Machine Learning (ML) que facilitan este tipo de tareas, como Amazon Textract, pero con un coste a partir de cierta cantidad de páginas analizadas. Este servicio está pensado para aportar una funcionalidad transversal y generalizada, permitiendo su utilización en cualquier tipo de documento, formulario, tabla, etc. Este hecho aporta flexibilidad al usuario, pero, en la práctica, este tipo de procesos requieren de un gran nivel de personalización, dependiendo del proyecto o la aplicación que se busca desarrollar. Por lo tanto, probablemente sea necesaria la contratación de servicios adicionales para conseguir un adecuado funcionamiento de la solución. Este tipo de soluciones puede llegar a ser más o menos rentable dependiendo del volumen documental con el que se esté tratando y el coste asociado a la ejecución del servicio.

Adicionalmente, existen otras soluciones que han aplicado también técnicas de procesamiento de lenguaje natural (NLP) y redes neuronales para analizar y entender el contenido de los documentos digitalizados. Entre estas soluciones se incluyen, por ejemplo, modelos de clasificación de documentos que su categorización y organización de manera ágil.

En resumen, existen diferentes soluciones tecnológicas que pueden ser consideradas para abordar el problema identificado en la revisión de expedientes administrativos, como son NLP, OCR, la combinación de este último con técnicas de procesamiento de regiones o Amazon Textract. Todas estas soluciones tienen diferentes ventajas y desventajas, por lo que es importante evaluarlas y compararlas en términos de precisión, velocidad, adaptabilidad y coste, de cara a escoger la más apropiada en base a los criterios de selección establecidos.

## **Capítulo 4. DEFINICIÓN DEL TRABAJO**

### **4.1 JUSTIFICACIÓN**

A partir de los problemas identificados en la introducción, es evidente que es necesario encontrar una solución alternativa o sistema de ayuda para agilizar la tramitación de expedientes administrativos en el contexto descrito.

Al evaluar y comparar las posibles soluciones tecnológicas para abordar la problemática planteada, la combinación de YOLO y OCR destaca como la elección más adecuada según los criterios de selección previamente establecidos.

- YOLO, al igual que la tecnología de OCR, cuentan con unos niveles satisfactorios en cuanto a precisión en la detección de objetos, lo cual es fundamental en el proceso de detección y verificación de información en expedientes.
- En términos de velocidad, la arquitectura de un solo disparo de YOLO permite una detección más ágil frente a otras tecnologías de detección de objetos.
- Esta combinación de YOLO y OCR destaca con respecto al resto de soluciones en cuanto a adaptabilidad debido a que YOLO puede ser entrenado para diferentes campos de interés, y el OCR ajustado para procesar diversos tipos de texto.
- En cuanto al coste, esta solución representa una inversión justificada en comparación con la contratación de servicios externos, ya que se adapta específicamente a los requerimientos del proyecto y evita costos recurrentes por volumen.

Teniendo esto en cuenta, la verificación de requisitos mediante un sistema de revisión documental basado en tecnologías de inteligencia artificial y OCR representa una vía prometedora para abordar los desafíos planteados. De esta forma se mejorarían los procedimientos administrativos, la experiencia de los ciudadanos, y además, se evitarían riesgos adicionales en la resolución de expedientes.

El proyecto se presenta como una oportunidad para contribuir en el ámbito de la tramitación de expedientes administrativos, para ello se propone la creación de un modelo de revisión de documentos adaptado a las particularidades de los expedientes para la concesión de ayudas, específicamente para los solicitantes en representación, considerando los modelos posibles del documento necesario en estos casos, y los requisitos para su verificación.

## **4.2 OBJETIVOS**

A continuación, se presentan los principales objetivos del proyecto:

- Crear una funcionalidad para agilizar el proceso de revisión documental de expedientes administrativos.
- Facilitar el trabajo de los revisores documentales involucrados en el proceso, reduciendo errores humanos.
- Reducir los costes de personal encargado de la revisión documental.
- Mejorar la experiencia de los solicitantes, reduciendo sus tiempos de espera.

## **4.3 METODOLOGÍA**

Para desarrollar el proyecto se definen las siguientes tareas:

1. Análisis/Búsqueda de información: se estudiarán otros proyectos que persigan objetivos similares, para entender en mayor profundidad su funcionamiento a nivel técnico y analizar las herramientas y servicios utilizados que sean potencialmente aplicables a este proyecto.
2. Diseño: se diseñará el sistema a nivel funcional y se establecerá la arquitectura técnica de la solución.
3. Desarrollo y pruebas de entrenamiento: se seleccionará la red neuronal, realizarán pruebas de entrenamiento, e implementará con el posterior sistema de análisis planteado.



4. Pruebas y simulación del sistema: se simulará el proyecto en un entorno de desarrollo para comprobar su correcto funcionamiento.
5. Revisión de errores: se buscarán los fallos en la simulación y comprobará el sistema para posibles mejoras.
6. Desarrollo de la memoria: se redactará el documento correspondiente a la memoria del proyecto.

#### **4.4 PLANIFICACIÓN Y ESTIMACIÓN ECONÓMICA**

Una vez identificadas y analizadas todas las tareas necesarias para la ejecución del proyecto, se debe definir una planificación realista para alcanzar los objetivos definidos. A continuación, se muestra un cronograma en el que se indican las tareas y tiempos asociados.

Tareas	Mayo	Junio	Julio	Agosto
Análisis/Búsqueda de información				
Diseño				
Desarrollo y pruebas de entrenamiento				
Pruebas y simulación del sistema				
Revisión de errores				
Desarrollo memoria				

*Tabla 1: Cronograma.*

En relación con los costes del proyecto, se ha realizado una estimación económica de los recursos necesarios:

- Costes de hardware: se utiliza un ordenador de sobremesa de uso personal, por lo que no se estiman gastos materiales en este ámbito. Las especificaciones técnicas del equipo utilizado se muestran en la siguiente tabla.

**Ordenador de sobremesa personalizado**

<b>Procesador</b>	Intel Core i7-4790 3.6 GHz
<b>Sistema operativo</b>	Windows 10 Pro
<b>Tarjeta gráfica</b>	GeForce GTX 970 4GB
<b>Memoria RAM</b>	24 GB DDR3-1600MHz
<b>Discos duros</b>	500GB SSD / 2x 120 GB SSD / 1TB HDD

*Tabla 2. Especificaciones del equipo utilizado.*

En caso de no disponer del hardware necesario, se estiman unos costes aproximados de 1.200,00 €.

- Coste de software: se utilizan múltiples herramientas y servicios gratuitos de código abierto (ver *Descripción de las tecnologías*), por lo que no se estiman gastos en este sentido.
- Costes de implementación: se han llevado a cabo los despliegues, configuraciones y pruebas de forma local, por lo que no se estiman gastos en este ámbito. En caso de llevar la solución a un entorno de producción, se estiman unos costes aproximados de 3.000,00 €.
- Costes de mantenimiento: en caso de llevar la solución a un entorno de producción bajo servicios en la nube, se estiman unos costes aproximados de 50,00 €.

<b>Conceptos de coste</b>	<b>Solución desarrollada en proyecto</b>	<b>Potencial solución en producción</b>
Hardware	0 €	1.200,00 €
Software	0 €	0 €
Implementación	0 €	3.000,00 €.
Mantenimiento	N/A	50,00 €

*Tabla 3: Resumen de costes estimados.*

A modo resumen, la estimación económica del proyecto contempla dos escenarios: el primero abarca el desarrollo y pruebas de la solución, donde no se generan costes significativos debido al uso de herramientas de código abierto y recursos hardware disponibles. Para el segundo escenario, que contempla la implementación en producción, se consideran costes estimados de 1.200,00 € para hardware, 3.000,00 € para la implementación y 50,00 € para el mantenimiento en un entorno de producción bajo servicios en la nube.

## **Capítulo 5. SISTEMA/MODELO DESARROLLADO**

El sistema desarrollado, como se ha mencionado previamente, consiste en una solución para la revisión automatizada de documentos, adaptada específicamente a los documentos de representación legal que presentan los solicitantes de ayudas durante el procedimiento administrativo correspondiente.

En este capítulo se llevará a cabo una descripción del procedimiento actual, estudiando sus pasos y requerimientos para llevar a cabo el diseño final de la nueva solución, así como su posterior entrenamiento e implementación para la obtención de los resultados.

### **5.1 ANÁLISIS DEL PROCEDIMIENTO ACTUAL**

Para realizar el diseño de una solución de revisión automatizada de documentos de un expediente, es preciso conocer el actual proceso de revisión manual, así como el flujo de un expediente dentro del procedimiento administrativo, de modo que se puedan especificar todos los requisitos a tener en cuenta:

#### **5.1.1 SISTEMA DE REVISIÓN ACTUAL**

En primer lugar, se van a comentar de forma simplificada y generalizada los pasos por el que pasa un expediente administrativo una vez se ha presentado una solicitud de forma satisfactoria dentro el programa de ayudas que corresponda.

La tipología del expediente puede diferir dependiendo del tipo de programa y a quien está dirigido, pero en cualquiera de los casos, cuando una solicitud se finaliza y registra, el expediente empieza su proceso de tramitación. En este punto, el expediente pasa a un estado de revisión, es decir, requiere de la verificación del cumplimiento de los requisitos del programa. Estos requisitos pueden ser verificados mediante consultas telemáticas a Servicios de Administración Digital (previamente autorizadas por el solicitante) o de forma manual

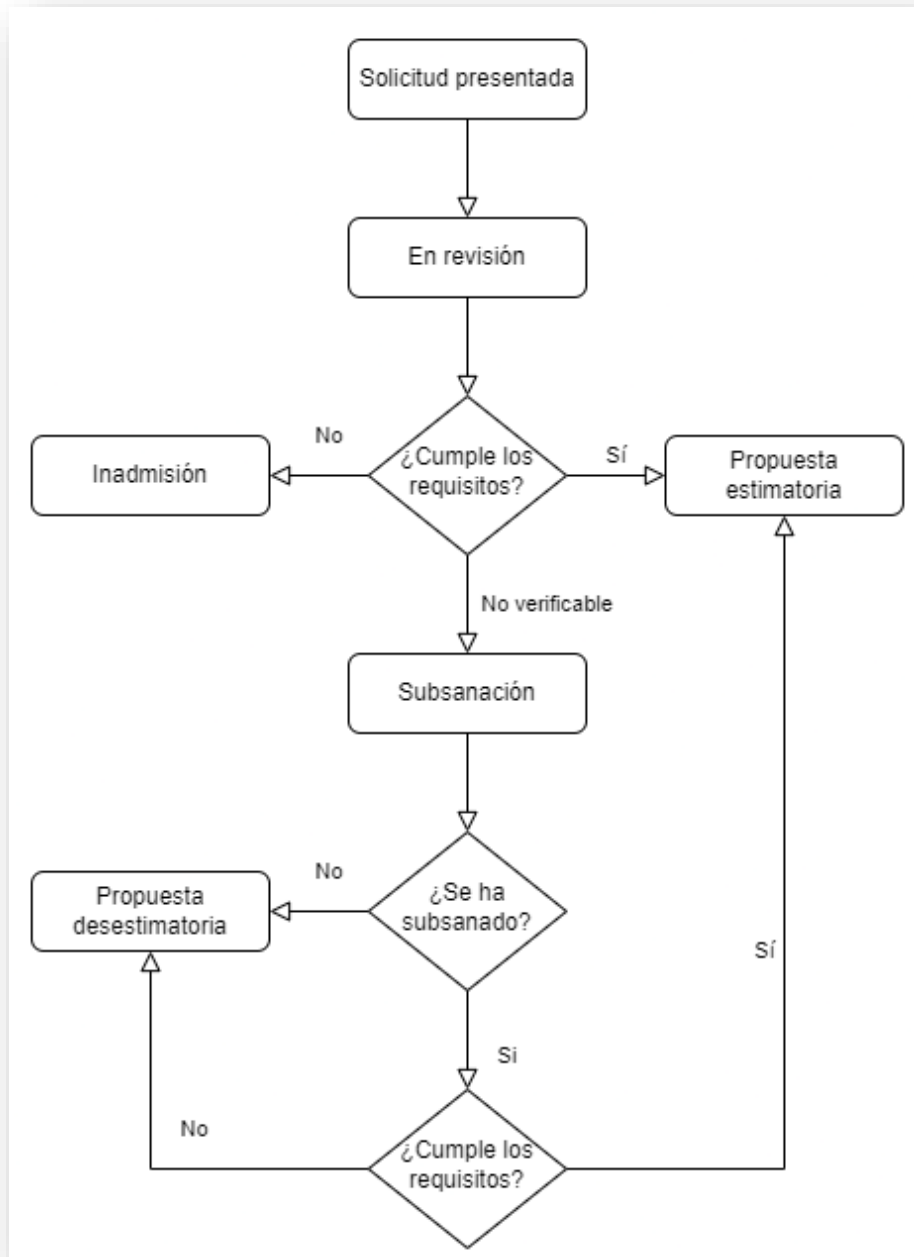
por parte de los revisores documentales, habiendo aportado el solicitante la documentación necesaria de manera previa.

Una vez el expediente ha sido revisado, puede pasar por distintos estados dentro del flujo de tramitación:

- Si los requisitos se han podido verificar con éxito y el solicitante cumple con los mismos, el expediente pasará a formar parte de una propuesta de estimación, quedando pendiente del organismo correspondiente para su aprobación definitiva y, por tanto, resolución final del expediente.
- Si los requisitos se han podido verificar con éxito y el solicitante no cumple con los mismos, el expediente será enviado a una resolución de inadmisión, para la cual el solicitante no dispondrá de la posibilidad de alegar.
- Si los requisitos no se han podido verificar con éxito debido a la entrega de documentación no válida, no legible o, incluso, que sea necesaria información adicional, el expediente pasará a un estado de subsanación, donde los revisores documentales requieren información al solicitante. Una vez el solicitante realiza la subsanación, el expediente pasa a ser revisado por segunda vez, si la subsanación no llega a realizarse por parte del solicitante en periodo de tiempo determinado o los requisitos siguen sin poder ser verificados, el expediente pasará a una propuesta de desestimación. En caso de propuesta de desestimación, el solicitante puede alegar el cumplimiento de los requisitos, aportando la información pertinente.

En última instancia, el solicitante tiene la posibilidad de presentar un recurso de alzada frente a cualquier resolución llevada a cabo.

A continuación, se muestra un diagrama de flujo simplificado, donde se muestran los estados por los que puede pasar un expediente administrativo desde su presentación hasta su resolución:



*Figura 2: Flujo de tramitación de expedientes.*

Una vez comprendido el flujo, destacar que el proyecto se va a enfocar en el primer estado de los expedientes administrativos, el correspondiente a la revisión de requisitos. Como se ha mencionado anteriormente, los requisitos pueden ser verificados mediante consultas

automáticas a Servicios de Administración Digital o de forma manual. La solución propuesta se enfoca en esta última casuística.

Para este proyecto es importante mencionar que las solicitudes pueden presentarse de dos formas: en nombre propio, en la que el interesado presenta y firma la solicitud; o en representación, en la que un representante presenta y firma la solicitud en nombre del interesado. En este último caso, es necesario aportar un documento acreditativo de representación legal, firmado tanto por el interesado como por el representante. La aportación de dicho documento es un requisito para cualquier solicitud realizada en representación.

El alcance concreto de la solución se enfoca en la revisión del documento acreditativo de representación legal, de cara a verificar la validez dicha evidencia de forma automatizada, sin necesidad de revisión manual.

Para más detalle, es importante especificar que la revisión manual del documento alcanzado consiste en verificar que el solicitante autoriza a un representante a presentar la solicitud en su nombre, indicando en dicho documento (tanto del interesado como del representante): nombre y apellidos, Número de Identificación Fiscal (NIF) o Número de Identidad de Extranjero (NIE) y firma manuscrita.

Si se verifica que el documento de representación legal contiene la información correcta del solicitante y representante y, además, se encuentra firmado por ambos, el requisito puede ser validado.

### **5.1.2 REQUERIMIENTOS DEL SISTEMA ACTUAL EN LA SOLUCIÓN**

Teniendo en cuenta el punto anterior y de cara a cumplir con las necesidades de un sistema de revisión válido para documentos de representación, se deberán cumplir los siguientes requisitos a la hora de diseñar la solución:

- Ser capaz de obtener y verificar los campos específicos del documento de representación legal, para garantizar la autenticidad de la información presentada.
- Ser escalable y capaz de manejar un volumen significativo de documentos.
- Ser flexible para permitir diferentes formatos y variaciones en la documentación, según las regulaciones específicas de cada programa y tipo de solicitud.
- Tener la capacidad de registrar y almacenar todas las acciones tomadas y resultados obtenidos durante el proceso de revisión. Además, debe manejar de manera adecuada los casos en los que se detecten errores o discrepancias en la información.
- Contar con una interfaz intuitiva y fácil de usar para los revisores documentales, presentando los resultados de la revisión automática de manera clara y siendo compatible con los flujos de trabajo actuales, de modo que se pueda integrar sin alterar en el proceso de revisión documental existente.
- Ser compatible e integrable con la plataforma tecnológica existente y estar diseñado de manera que permita un mantenimiento continuo y actualizaciones futuras.

## **5.2 DISEÑO DE LA SOLUCIÓN**

A partir de los requerimientos obtenidos a raíz del análisis del procedimiento actual realizado en el apartado anterior, se ha realizado un diseño de la solución por fases: extracción de base de datos, detección de objetos, comparación de datos.

En primer lugar, la extracción de la información necesaria correspondientes al expediente que se va a tratar directamente de la base de datos. La información necesaria es: el identificador del expediente, el NIF/NIE (tanto del interesado como del representante), sus nombres y apellidos, y la dirección lógica donde se encuentra guardado el documento a revisar dentro de la plataforma de revisión documental en producción (almacenado en Amazon S3).

En segundo lugar, a partir de la dirección lógica donde se encuentra guardado, se descarga el documento y se ejecuta la solución implementada. Concretamente, los pasos a seguir son:



- Transformación del documento descargado en una imagen.
- Detección de objetos de la imagen asociada al documento, mediante un modelo entrenado para localizar NIF/NIE, nombres completos, y firmas. En primera instancia, dentro del alcance del proyecto, sólo se entrenará el modelo para detectar NIF/NIE como objeto único.
- Obtención de las coordenadas de las regiones de interés detectadas u objetos y las clases de dichos objetos .
- A partir de dichas coordenadas, recorte de las imágenes para obtener una imagen individual de cada objeto detectado.
- Digitalización de la información de interés correspondiente a las imágenes recortadas.

Por último, a partir de los datos obtenidos con el modelo de detección, se realiza una comparación de caracteres uno a uno con respecto a la información del expediente extraída directamente de la base de datos. El resultado de dicha comparación, en términos de coincidencia, se muestra como porcentaje.

A continuación, se muestra el diseño de forma esquematizada para facilitar su comprensión.

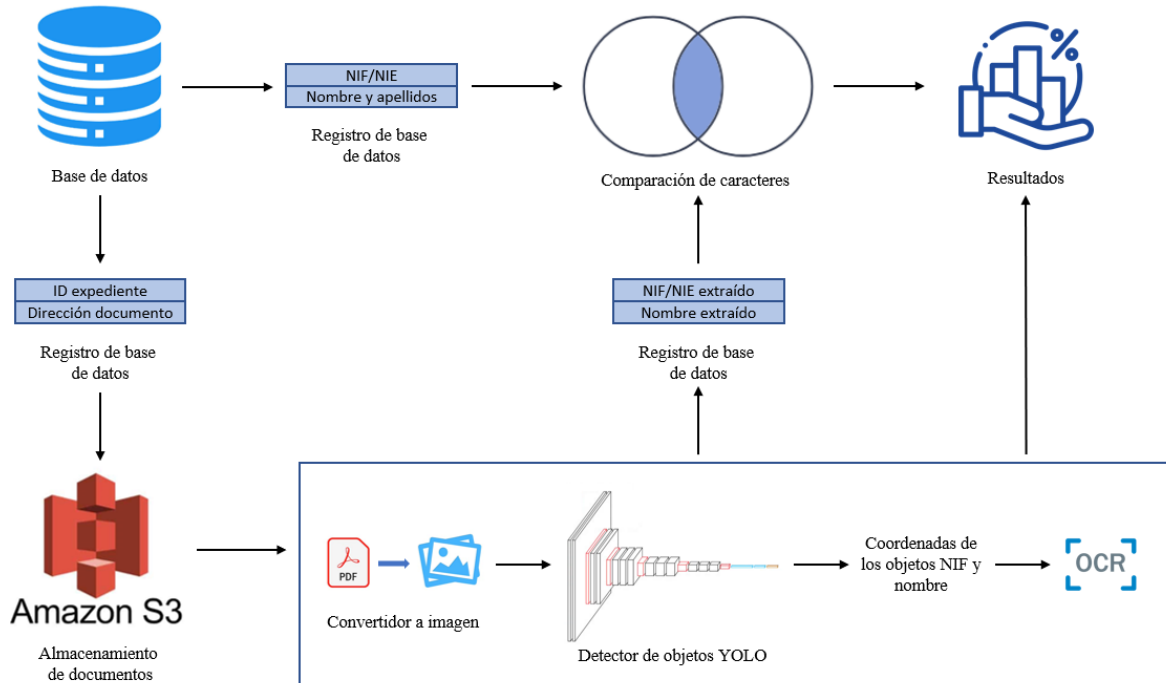


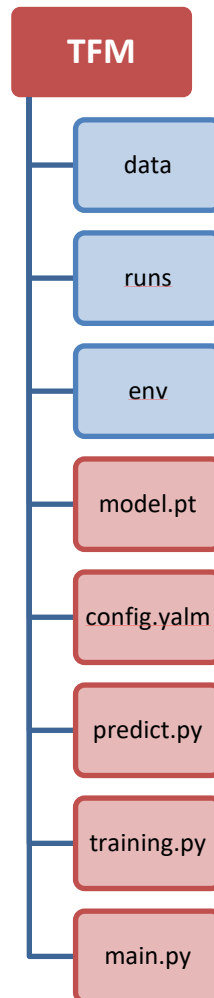
Figura 3: Diseño general de la solución.

Cabe destacar, que el desarrollo de este proyecto está enfocado para aportar valor adicional a una plataforma ya existente de tramitación de expedientes, por lo que se plantea como una versión de prueba. En este sentido, se entrenará un modelo simplificado para realizar únicamente revisión de NIF/NIE en el documento de representación legal y comprobar si el funcionamiento resulta válido para desarrollar una solución completa que alcance todas las especificaciones indicadas en el apartado *Requerimientos del sistema actual en la solución*.

### 5.2.1 ESTRUCTURA DEL PROYECTO

Con el entendimiento y diseño de la solución, se plantea una estructura del proyecto que permita tener organizada toda la información de manera clara, de modo que se facilite el acceso y administración a los archivos y recursos del mismo.

El proyecto se ha organizado en un directorio específico (a partir de un entorno de Python), que contiene diferentes carpetas para estructurar la información y las librerías utilizadas. Este directorio se estructura del siguiente modo:

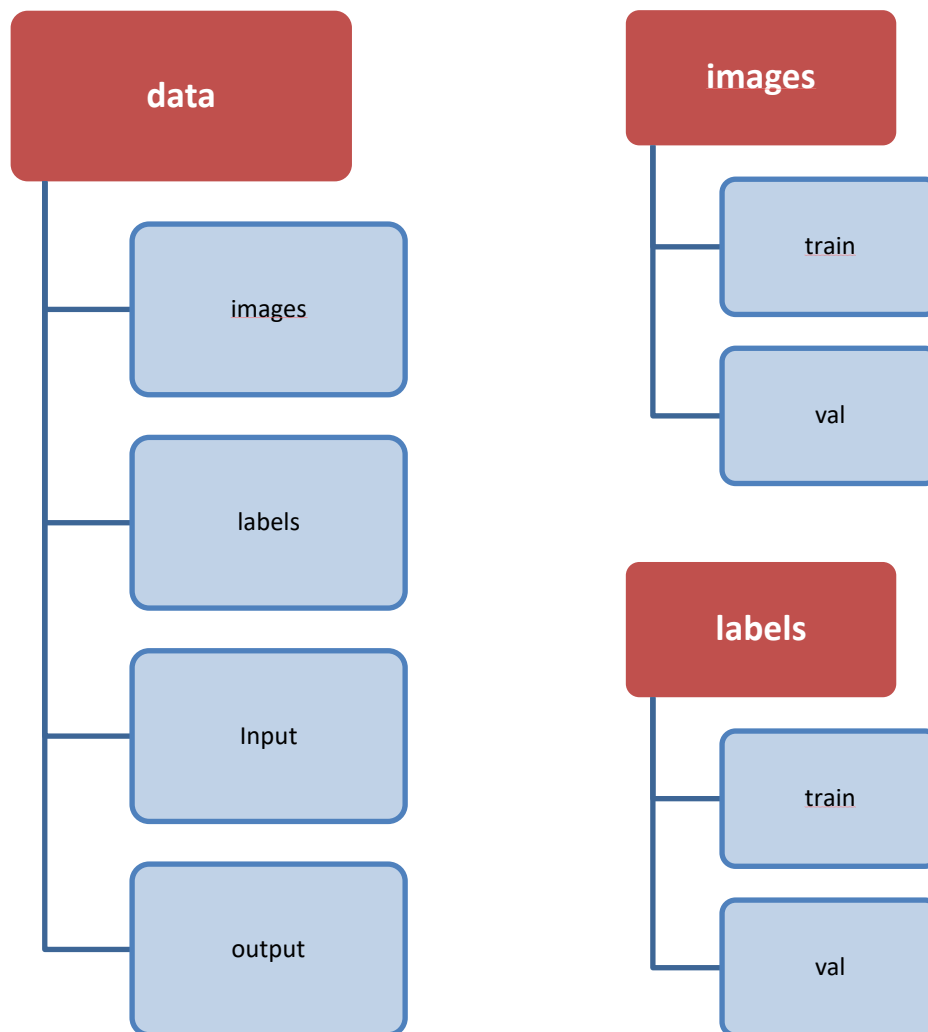


*Figura 4: Estructura del proyecto.*

Dentro de esta estructura se encuentran diferentes carpetas, de forma que dividen y agrupan diferentes tipos de datos y recursos, lo que contribuye a una mejor organización y mantenimiento del proyecto:

- La carpeta “data”, la cual se subdivide en 4 carpetas con dos propósitos distintos.
  - Las carpetas (“images” y “labels”) se utilizan para almacenar todos los datos necesarios para el entrenamiento del modelo de YOLO: “images” almacena todas las imágenes de entrenamiento, y “labels” las etiquetas que indican las cajas de contorno de los diferentes objetos dentro de cada imagen.

- Las otras dos carpetas (“input” y “output”) se utilizan durante la ejecución de la solución: “input” almacena los documentos pdf a analizar por parte del sistema, “output” almacena las imágenes recortadas a partir del documento transformado, y los resultados de la ejecución en formato Excel.



*Figura 5: Diagrama de los archivos "data".*

- La carpeta “runs” almacena los resultados de las ejecuciones de entrenamiento, donde se encuentran los diferentes modelos entrenados, así como las ejecuciones de predicción individuales del modelo de YOLO. La organización en esta carpeta facilita el seguimiento y la comparación de los resultados a lo largo del proyecto.

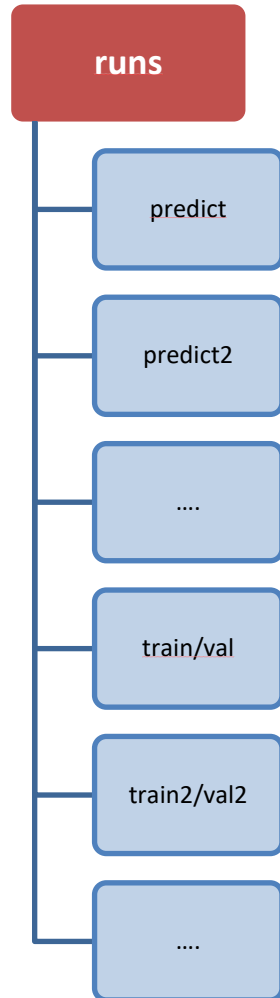


Figura 6: Diagrama de los archivos "runs".

- La carpeta “env” contiene los archivos y recursos necesarios para el entorno de Python, incluye las librerías instaladas, scripts de activación, y otros elementos esenciales para garantizar el correcto funcionamiento del proyecto.

Por último, como se muestra en la *Figura 4*, dentro del proyecto se almacenan los diferentes scripts desarrollados, y el modelo en uso:

- model.pt: Este archivo contiene el mejor modelo obtenido a partir de los diferentes entrenamientos, tanto en local como en Google Colab. Es el modelo en uso durante la ejecución de la solución.

- `config.yaml`: Este archivo configura los valores necesarios para el entrenamiento, compone en su interior las direcciones de los archivos de entrenamiento y validación, así como las diferentes clases que el modelo debe tener en cuenta.
- `predict.py`: Este script permite la ejecución del detector de objetos de forma individual. A partir de un modelo y una dirección o imagen, se ejecuta sobre las mismas guardando el resultado en “runs”, como se ha comentado. Se utiliza como una forma de probar el modelo entrenado y visualizar los resultados.
- `training.py`: Este script ejecuta el proceso de entrenamiento y evalúa el modelo a partir del set de validación. Es en este archivo donde se indica si partir de un modelo pre-entrenado, o realizar un entrenamiento de base.
- `main.py`: Este es el script principal, donde se encuentra todo el funcionamiento de la solución desarrollada, y la que proporciona la funcionalidad principal del proyecto, desde la conversión del documento pdf a imagen, hasta la obtención de los resultados.

La estructura y organización detallada de archivos y carpetas presentados aseguran un manejo sencillo y estructurado del proyecto, lo que facilita el desarrollo, pruebas y futuras actualizaciones.

### **5.2.2 BASE DE DATOS**

En la solución propuesta, debe existir una forma de comparar la información de los expedientes con la extraída de los documentos y, además, una forma de almacenar los datos relevantes sobre precisiones y resultados obtenidos durante la ejecución del sistema.

Para esto, por un lado, el sistema se va a diseñar de forma que pueda adaptar a cualquier método de extracción de información. Por otro lado, para el almacenamiento de los resultados, se va a crear una tabla que almacene toda la información relevante, para en un futuro poder agregarla a una base de datos existente según corresponda.

En primer lugar, se describen los datos a extraer que son necesarios para el funcionamiento de este sistema:

- Identificador del expediente: Identificador único del que dispone cada expediente.
- NIF/NIE del interesado: El número de identificación fiscal (NIF) o el número de identidad de extranjero (NIE) del interesado para el que se ha presentado la solicitud.
- NIF/NIE del representante: El número de identificación fiscal del representante que ha presentado la solicitud.
- Ubicación del documento: Dirección donde se encuentra almacenado el documento.
- Nombre del documento: Nombre con el que se ha descargado el documento.

En segundo lugar, se describe la estructura de la tabla de la base de datos sobre la que realizar el registro. En esta tabla, se guarda la información extraída de los documentos, pero de forma adicional se guardan los resultados esenciales de las detecciones realizadas por el modelo de YOLO y el OCR, para evaluar así el rendimiento del sistema y comprender su precisión en la extracción de información de los campos de interés en los documentos:

**Diseño de la tabla en base de datos**

<b>Campo</b>	<b>Tipo de Dato</b>	<b>Descripción</b>
id_expediente	UUID	Identificador único del expediente
applicant_nif	VARCHAR	NIF/NIE del interesado proveniente de la base de datos
representative_nif	VARCHAR	NIF/NIE del representante proveniente de la base de datos
applicant_extracted_nif	VARCHAR	NIF/NIE del interesado proveniente de la extracción del documento
representative_extracted_nif	VARCHAR	NIF/NIE del representante proveniente de la extracción del documento
applicant_box_confidence	FLOAT	Confianza de la detección por parte del modelo del NIF/NIE del interesado

representative_box_confidence	FLOAT	Confianza de la detección por parte del modelo del NIF/NIE del representante
applicant_ocr_score	FLOAT	Confianza de la lectura por parte del OCR del NIF/NIE del interesado
representative_ocr_score	FLOAT	Confianza de la lectura por parte del OCR del NIF/NIE del representante
applicant_similarity	FLOAT	Porcentaje de similitud del NIF/NIE del interesado extraído con el de la BD
representative_similarity	FLOAT	Porcentaje de similitud del NIF/NIE del representante extraído con el de la BD
verification_decision	BOOLEAN	Decisión para la verificación del documento a partir de los resultados obtenidos.

*Tabla 4: Tabla en base de datos.*

La tabla anterior captura de manera estructurada la información sobre los resultados obtenidos durante la ejecución de la solución. Esto permite un análisis detallado de la decisión del sistema para la verificación del documento, así como la identificación de posibles mejoras en la detección y validación de los campos de interés.

### **5.2.3 INTERFAZ Y CASOS DE USO**

La decisión de verificación deberá mostrarse a los usuarios de la plataforma de revisión de forma sencilla e intuitiva. Para ello, se incorporarán los mismos mecanismos a nivel de front-end que para el resto de los requisitos, de forma que su flujo de revisión no se vea afectado y no sea necesario ningún desarrollo adicional para los diferentes resultados.

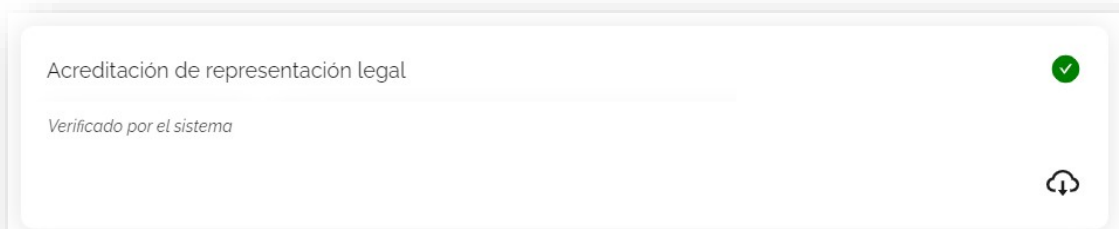
Actualmente, los revisores documentales acceden a la plataforma, y una vez han revisado el requisito a partir del documento, indican su estado: “Cumple”, “No válido”, “No legible”, “No aporta” o “No cumple”, e indican unas observaciones en caso de considerarlo necesario.





*Figura 7: Interfaz de revisión manual.*

Existen otros requisitos que se verifican de forma automática mediante consultas telemáticas a Servicios de Administración Digital, estos requisitos se muestran de la misma forma, pero se incluyen directamente con el estado “Cumple”, y marcados como “Verificados por el sistema”.



*Figura 8: Interfaz de verificación automática.*

Para describir las diferentes situaciones, se han definido distintos casos de uso basados en los resultados obtenidos durante el proceso de detección y la validación de los campos de interés en los documentos. Estos casos de uso reflejan los diferentes estados en los que puede encontrarse un expediente y como afecta esto a los revisores documentales. A continuación, se describen los posibles casos de uso:

1. Decisión de verificación del documento positiva:
  - Descripción: Cuando el campo “verification\_decision” tiene valor 1.
  - Interfaz: Interfaz para consultas automatizadas (*Figura 8*).
  - Acciones: No requeridas.
2. Decisión de verificación del documento negativa:
  - Descripción: Cuando el campo “verification\_decision” tiene valor 0.
  - Interfaz: Interfaz de pendiente de revisión manual (*Figura 7*).
  - Acciones: Revisar el requisito de forma manual.
3. Documentación no revisada:
  - Descripción: Cuando el sistema no se ha ejecutado para dicho expediente.
  - Interfaz: Interfaz de pendiente de revisión manual (*Figura 7*).
  - Acciones: Revisar el requisito de forma manual.

Cada expediente puede ser clasificado en uno de estos estados, lo que guía las acciones a tomar por parte de los revisores, sin afectar al flujo de un expediente. Mediante estos casos de uso se puede garantizar que cada expediente es tratado de acuerdo a los resultados obtenidos en la detección automática, sin cambiar el flujo de revisión existente y sin parar el proceso en caso de existir un expediente sin la revisión automática realizada.

## **5.3 ENTRENAMIENTO DEL MODELO YOLO**

En esta sección, se detalla el proceso de entrenamiento del modelo YOLO llevado a cabo para detectar las posiciones de los objetos en las imágenes de los documentos. El proceso se divide en varias fases:

### **5.3.1 CREACIÓN DE DATASET DE ENTRENAMIENTO**

En primer lugar, para llevar a cabo un entrenamiento satisfactorio, es necesario disponer de un conjunto de datos para el mismo. En este caso, como se va a entrenar para un conjunto de objetos particular, se ha tenido que desarrollar un dataset de forma manual, a partir de plantillas del documento de representación legal.

Con estas plantillas rellenas, se etiquetaron manualmente las posiciones de los NIF/NIE en las imágenes, utilizando la herramienta de etiquetado *Computer Vision Annotation Tool (CVAT)*. Esta herramienta permitía dibujar cuadros delimitadores alrededor de los NIF/NIE dentro de la imagen, y almacenaba las coordenadas de estos cuadros, en el momento de su exportación te permitía utilizar un formato adecuado para el entrenamiento del modelo YOLO.

Cada etiqueta incluye las coordenadas (x, y) de la esquina superior izquierda y la esquina inferior derecha del cuadro delimitador alrededor del NIF, y el número de la clase del objeto, en este caso 0 para el objeto NIF.

Estas imágenes y etiquetas se almacenan posteriormente en las carpetas “images” y “labels” que se ha mostrado en la *Figura 5*.

### **5.3.2 ENTRENAMIENTO DEL MODELO YOLO**

Para este proyecto se ha seleccionado como base el modelo YOLOv8[15] de Ultralytics debido a su capacidad para realizar detección de objetos con alta precisión y eficiencia, así como la gran cantidad de documentación de la que dispone.

El modelo se configuró para adaptarse a los requisitos específicos del proyecto. Para ello, a partir del archivo de configuración YALM (Yet Another Markup Language), se define la arquitectura del modelo. En este caso se ha seleccionado la versión YOLOv8n, que se trata de la red con menos parámetros como se puede ver en la *Figura 9*.

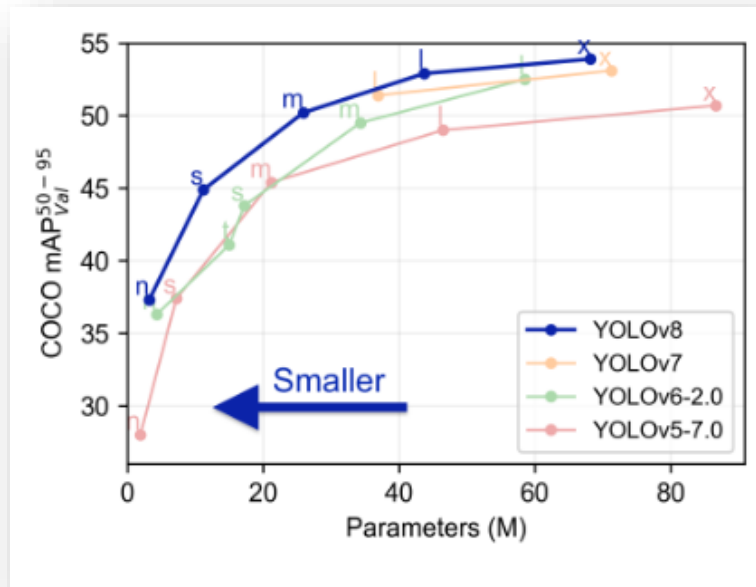


Figura 9: Parámetros de las versiones de YOLO Ultralytics[15].

Una vez indicada la arquitectura, a partir del archivo YALM comentado en el apartado *Estructura del proyecto*, en el que se indican las ubicaciones del conjunto de datos etiquetado y las clases para el modelo, se inició el proceso de entrenamiento del modelo YOLOv8.

Durante este proceso, el modelo se dedica a ajustar sus pesos y parámetros internos para aprender a detectar las clases de objetos indicadas en las imágenes. El entrenamiento se realizó iterativamente a lo largo de 100 épocas, y múltiples veces para distintos tamaños del conjunto de datos. Una vez el entrenamiento finaliza, se guarda el último modelo, y el mejor modelo.

El modelo utilizado ha sido un modelo entrenado desde Google Colab, con un reducido conjunto de 300 imágenes y únicamente detectando una clase de objeto.

## 5.4 IMPLEMENTACIÓN

En este apartado se proporcionará una descripción detallada la implementación del sistema de revisión documental automática, abarcando todos los puntos del proceso. A continuación se presentan las diferentes partes del proceso, y el funcionamiento en detalle de la ejecución del sistema:

### 5.4.1 EXTRACCIÓN Y PREPARACIÓN DE DATOS

La primera etapa del proceso implica la recopilación y preparación de los datos necesarios para la ejecución. En esta versión de prueba, no se ha implementado una comunicación directa con la base de datos, se ha creado una estructura de datos denominada “data\_mapping” que contiene la información de los documentos de prueba que se van a procesar. Esta información se introduce previamente e incluye para cada nombre de documento: el identificador de expediente, NIF del solicitante y del representante, así como direcciones de archivo asociadas.

### 5.4.2 CONVERSIÓN DE PDF A IMÁGENES

Para permitir el procesamiento automatizado, se lleva a cabo un proceso de conversión de documentos PDF a imágenes utilizando la biblioteca pdf2image.

Para esto, se declaran en primer lugar las direcciones de las carpetas de “input” y “output” (ver *Estructura del proyecto*), se obtiene la lista de todos los documentos con extensión PDF dentro de la carpeta “input”, y de forma individual gracias a una función desarrollada, se transforman todos los documentos a imagen. A continuación se expone el código de la función utilizada:

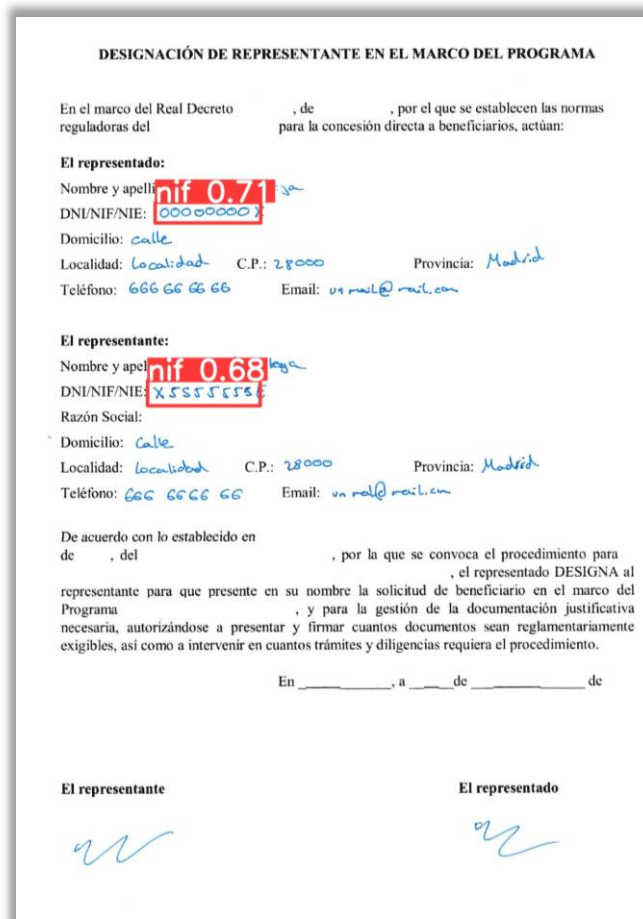
```
def convertir_pdf_a_imagen(archivo_pdf):
    imagenes = convert_from_path(archivo_pdf, first_page=1, last_page=1)
    nombre_archivo = os.path.splitext(archivo_pdf)[0]

    for i, imagen in enumerate(imagenes):
        ruta_guardado = f'{nombre_archivo}.jpg'
        imagen.save(ruta_guardado, 'JPEG')
```

Una vez se procesan los archivos PDF, las imágenes se almacenaron en una ubicación específica para su posterior procesamiento, en este caso se mantienen en la misma carpeta “input” que los documentos, ya que van a ser la entrada del sistema de detección.

### 5.4.3 DETECCIÓN DE POSICIONES DE NIF CON YOLO

A partir de las imágenes obtenidas en la conversión, y el modelo entrenado anteriormente, se procesa de forma individual y ordenada cada imagen. Por lo que, cada imagen pasará a continuación por el detector de YOLO entrenado para identificar la posición de los NIF en la imagen.



**DESIGNACIÓN DE REPRESENTANTE EN EL MARCO DEL PROGRAMA**

En el marco del Real Decreto \_\_\_\_\_, de \_\_\_\_\_, por el que se establecen las normas reguladoras del \_\_\_\_\_ para la concesión directa a beneficiarios, actúan:

**El representado:**  
Nombre y apellidos: **nif 0.71** *ya*  
DNI/NIF/NIE: **00000000**  
Domicilio: *calle*  
Localidad: *localidad* C.P.: *28000* Provincia: *Madrid*  
Teléfono: *666 66 66 66* Email: *va mail@mail.com*

**El representante:**  
Nombre y apellidos: **nif 0.68** *lega*  
DNI/NIF/NIE: **X.55555555**  
Razón Social:  
Domicilio: *calle*  
Localidad: *localidad* C.P.: *28000* Provincia: *Madrid*  
Teléfono: *666 6666 66* Email: *va mail@mail.com*

De acuerdo con lo establecido en \_\_\_\_\_, del \_\_\_\_\_, por la que se convoca el procedimiento para \_\_\_\_\_, el representado DESIGNA al representante para que presente en su nombre la solicitud de beneficiario en el marco del Programa \_\_\_\_\_, y para la gestión de la documentación justificativa necesaria, autorizándose a presentar y firmar cuantos documentos sean reglamentariamente exigibles, así como a intervenir en cuantos trámites y diligencias requiera el procedimiento.

En \_\_\_\_\_, a \_\_\_\_\_ de \_\_\_\_\_ de \_\_\_\_\_

El representante \_\_\_\_\_ El representado \_\_\_\_\_

Figura 10: Identificación en documento.

Si se han encontrado NIF en la imagen, se extrae la información de las coordenadas de la caja de contorno del objeto, y se recorta la imagen dos veces, para la detección del NIF del interesado y para el NIF del representante, guardándolas en la carpeta “output”. Si no se han encontrado detecciones, se muestra un mensaje detallando el suceso, se exporta como resultado de esta verificación una decisión negativa y se pasa a la siguiente imagen. A continuación se muestra el fragmento de código de este proceso:

```
# PRECEDIR CON YOLO LA POSICION DEL NIF
detections = nif_detector(ruta_img)[0]

if len(detections) == 0:
    print('No se encontraron NIFs')
    results.append({
        'petition_id': data['petition_id'],
        'dossier_id': data['dossier_id'],
        'applicant_nif ': data['applicant_NIF'],
        'representative_nif': data['representative_NIF'],
        'applicant_extracted_nif ': 'NOT_FOUND',
        'representative_extracted_nif': 'NOT_FOUND',
        'applicant_box_confidence': '-',
        'representative_box_confidence': '-',
        'applicant_ocr_score': '-',
        'representative_ocr_score': '-',
        'applicant_similarity': '-',
        'representative_similarity': '-',
        'verification_decision': 0
    })
    continue
detection = detections.bboxes.data.tolist()

# RECORTAR NIF A PARTIR DE LA PREDICCION
img = cv2.imread(ruta_img)

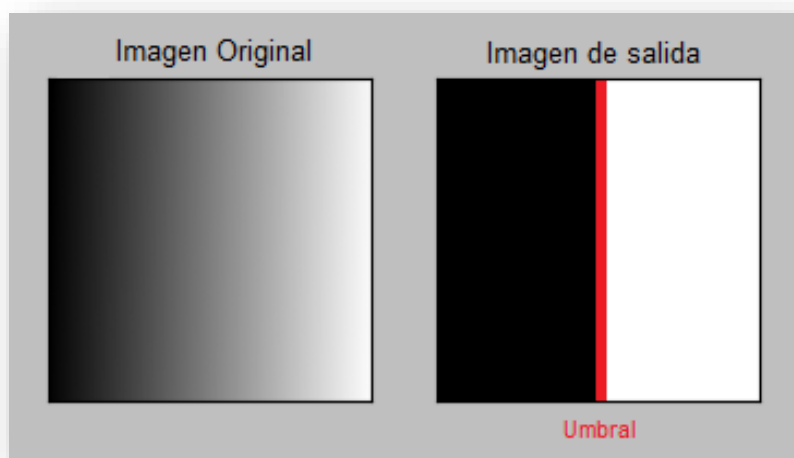
image_name, image_ext = os.path.splitext(image)
new_image_name = f"{image_name}_1{image_ext}"
image_path = os.path.join(save_path, new_image_name)
x1, y1, x2, y2, confidence1, type = detection[0]
nif1 = img[int(y1):int(y2), int(x1):int(x2)]
cv2.imwrite(image_path, nif1)

new_image_name = f"{image_name}_2{image_ext}"
image_path = os.path.join(save_path, new_image_name)
x1, y1, x2, y2, confidence2, type = detection[1]
nif2 = img[int(y1):int(y2), int(x1):int(x2)]
cv2.imwrite(image_path, nif2)
```

#### **5.4.4 PROCESAMIENTO DE LA IMAGEN RECORDADA**

En caso de haber realizado la detección de un NIF, se disponen de dos nuevas imágenes recortadas que contienen los caracteres de los diferentes NIF para cada imagen. Estas imágenes deben pasar por el OCR, pero previamente se va a realizar un tratamiento de estas imágenes.

En primer lugar, se transforma la imagen a escala de grises para facilitar la detección de bordes y detalles de la imagen. En segundo lugar se aplica un umbral binario para segmentar la imagen en blanco y negro, donde los píxeles que superan el umbral se vuelven blancos, y los que se encuentran por debajo se vuelven negros. A continuación se muestra una explicación del proceso:



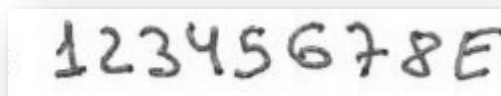
*Figura 11: Ejemplo de umbral binario.*

Mediante este proceso se resaltan los caracteres de la imagen para facilitar la lectura del OCR y evitar posibles fallos de lectura. A continuación se muestra un ejemplo de prueba:

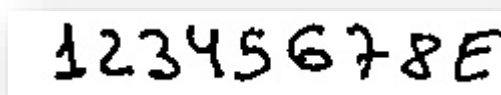




*Figura 12: Imagen original.*



*Figura 13: Imagen con escala de grises.*



*Figura 14: Imagen con umbral binario.*

#### **5.4.5 EXTRACCIÓN Y OCR DE NIF**

Las imágenes recortadas y tratadas se procesan mediante un OCR. Se ha utilizado la biblioteca EasyOCR para esta tarea, extrayendo con ella los contenidos de estas imágenes, es decir, los campos de interés (NIF del interesado y representante) en formato de texto entendible por el sistema como cadena de caracteres.

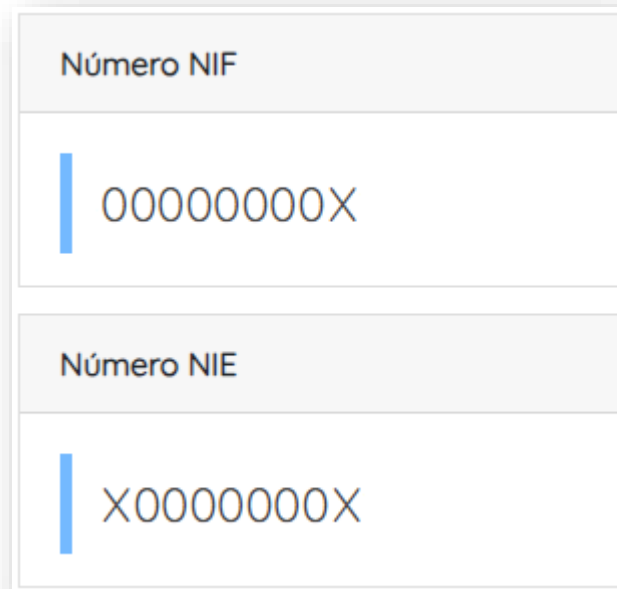
Este proceso proporciona además una puntuación de confianza para cada extracción de texto, la cual puede tenerse en cuenta para la decisión de validación del documento.

#### **5.4.6 PROCESAMIENTO DE TEXTO**

El siguiente paso del sistema es procesar el texto extraído del OCR, ya que debe tener el mismo formato que el almacenado en base de datos para una correcta comparación.

Para ello, en primer lugar, se convierte a mayúsculas todas las letras de las que disponga el texto y se eliminan espacios y símbolos como “-” o “.”.

En segundo lugar, todos los textos se procesan mediante una función desarrollada, la cual controla los formatos específicos de los NIF y NIE. A continuación se muestran los formatos tanto de NIF como de NIE para comprender la función:



The diagram illustrates the required formats for NIF and NIE numbers. It consists of two vertically stacked input fields. The top field is titled 'Número NIF' and contains the placeholder text '00000000X'. The bottom field is titled 'Número NIE' and contains the placeholder text 'X0000000X'. Each field has a blue vertical bar on the left side, indicating the start of the input area.

*Figura 15: Formatos NIF/NIE.*

Teniendo estos formatos en cuenta, se pretende controlar en cierta medida la correspondencia de los caracteres con relación a números o letras. Para ello:

- El primer carácter puede ser número o letra dependiendo si es NIF o NIE.
- Los caracteres intermedios deben ser siempre números.
- El último carácter debe ser siempre letra.

Se han declarado diccionarios con letras y números similares que el OCR puede haber confundido, para modificarlos con este criterio de formato según corresponda. A continuación se muestra la función desarrollada para esta tarea:

```
def format_nif(text):  
  
    formatted_nif = text[0]  
  
    for i in range(1, len(text) - 1):  
        if text[i] in dict_char_to_int.keys():  
            formatted_nif += dict_char_to_int[text[i]]  
        else:  
            formatted_nif += text[i]  
  
    if text[-1] in dict_int_to_char.keys():  
        formatted_nif += dict_int_to_char[text[-1]]  
    else:  
        formatted_nif += text[-1]  
  
    return formatted_nif  
  
dict_char_to_int = {'0': '0',  
                   '1': '1',  
                   'J': '3',  
                   'A': '4',  
                   'G': '6',  
                   '/': '1',  
                   'Z': '2',  
                   'S': '5'}  
  
dict_int_to_char = {'0': '0',  
                   '1': '1',  
                   '3': 'J',  
                   '4': 'A',  
                   '6': 'G',  
                   '2': 'Z',  
                   '5': 'S'}
```

#### **5.4.7 COMPARACIÓN DE CADENAS DE CARACTERES Y VERIFICACIÓN**

Finalizado el tratamiento de texto, es posible comparar con mayor seguridad la información extraída del OCR con la real. Por lo que, se utiliza la biblioteca *difflib* para comparar la similitud de los caracteres, obteniendo con ello un coeficiente de similitud. Esta comparación se lleva a cabo carácter a carácter, es decir, se compara el primer carácter de una cadena con el primer carácter de la otra cadena, el segundo con el segundo, y así sucesivamente.

Cuanto más alto es el coeficiente de similitud, más parecidos son las dos cadenas, por lo que a partir de esto se puede definir un umbral de verificación. Si la similitud de ambos NIF supera el umbral establecido (por ejemplo un 75%), se toma la decisión de verificación como positiva; de lo contrario, se marca como negativa.

### 5.4.8 GENERACIÓN DE RESULTADOS Y ALMACENAMIENTO

Una vez obtenido el coeficiente de similitud, se recopilan todos los datos que se han ido obteniendo durante el procesamiento de cada imagen (correspondiente a cada expediente), incluidos entre ellos los datos del expediente, los NIF extraídos, las puntuaciones de OCR, las puntuaciones de las cajas de contorno de YOLO, las similitudes y la decisión final de verificación.

Estos datos se almacenan en una estructura de datos llamada “results”, con la que finalmente se creará un “dataframe” de pandas para posteriormente extraerlos con formato Excel en la carpeta “output”.

	A	B	C	D	E	F	G	H	I	J	K	L
1	dossier_id	applicant_ni	representati	applicant_ex	representati	applicant_bd	representati	applicant_oc	representati	applicant_si	representati	verification
2	0d12fd51-69	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	0,70135242	0,64423931	0,65059325	0,63816121	1	0,88888889	1
3	3e853e38-47	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	0,71365523	0,52611822	0,66708325	0,20371572	1	0,88888889	1

*Figura 16: Archivo Excel de resultados.*

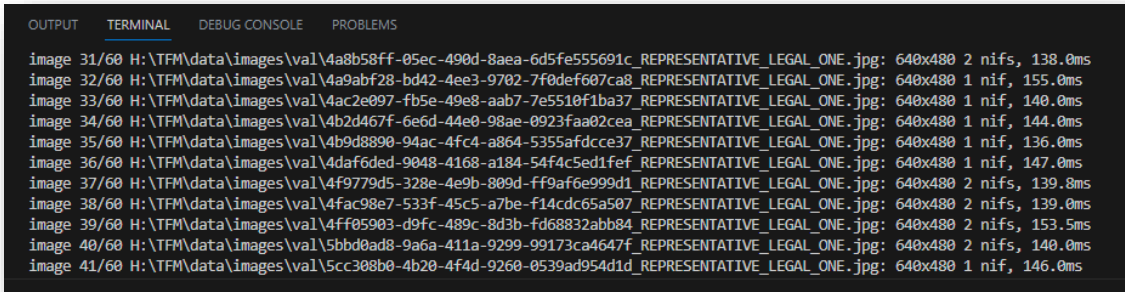
Fuera de la solución desarrollada, estos resultados se cargarán a la base de datos mediante un “INSERT” cuando se consideré adecuado, lo que aplicará los cambios de interfaz comentados en 5.2.3 *Interfaz y casos de uso* como corresponda, ya que son dependientes del valor en base de datos.

## Capítulo 6. ANÁLISIS DE RESULTADOS

En este capítulo se presentan y evalúan los resultados más relevantes del proyecto, de cara a evaluar la solución diseñada e implementada para la agilización del proceso de revisión documental de expedientes administrativos.

### 6.1 TIEMPOS DE EJECUCIÓN DE YOLO

Uno de los aspectos evaluados en profundidad es el tiempo de ejecución de la red neuronal utilizada para la detección de las posiciones de los NIF en cada documento. En este sentido, los tiempos requeridos para procesar una imagen y determinar si contiene un NIF, se muestran en la siguiente captura de pantalla de la consola tras ejecutar la solución.



```
OUTPUT  TERMINAL  DEBUG CONSOLE  PROBLEMS
image 31/60 H:\TFM\data\images\val\4a8b58ff-05ec-490d-8aea-6d5fe555691c_REPRESENTATIVE_LEGAL_ONE.jpg: 640x480 2 nifs, 138.0ms
image 32/60 H:\TFM\data\images\val\4a9abf28-bd42-4ee3-9702-7f0def607ca8_REPRESENTATIVE_LEGAL_ONE.jpg: 640x480 1 nif, 155.0ms
image 33/60 H:\TFM\data\images\val\4ac2e097-fb5e-49e8-aab7-7e5510f1ba37_REPRESENTATIVE_LEGAL_ONE.jpg: 640x480 1 nif, 140.0ms
image 34/60 H:\TFM\data\images\val\4b2d467f-6e6d-44e0-98ae-0923faa02cea_REPRESENTATIVE_LEGAL_ONE.jpg: 640x480 1 nif, 144.0ms
image 35/60 H:\TFM\data\images\val\4b9d8890-94ac-4fc4-a864-5355afdcc37_REPRESENTATIVE_LEGAL_ONE.jpg: 640x480 1 nif, 136.0ms
image 36/60 H:\TFM\data\images\val\4daf6ded-9048-4168-a184-54f4c5ed1fef_REPRESENTATIVE_LEGAL_ONE.jpg: 640x480 1 nif, 147.0ms
image 37/60 H:\TFM\data\images\val\4f9779d5-328e-4e9b-809d-ff9af6e999d1_REPRESENTATIVE_LEGAL_ONE.jpg: 640x480 2 nifs, 139.8ms
image 38/60 H:\TFM\data\images\val\4fac98e7-533f-45c5-a7be-f14cdc65a507_REPRESENTATIVE_LEGAL_ONE.jpg: 640x480 2 nifs, 139.0ms
image 39/60 H:\TFM\data\images\val\4ff05903-d9fc-489c-8d3b-fd68832abb84_REPRESENTATIVE_LEGAL_ONE.jpg: 640x480 2 nifs, 153.5ms
image 40/60 H:\TFM\data\images\val\5bbd0ad8-9a6a-411a-9299-99173ca4647f_REPRESENTATIVE_LEGAL_ONE.jpg: 640x480 2 nifs, 140.0ms
image 41/60 H:\TFM\data\images\val\5cc308b0-4b20-4f4d-9260-0539ad954d1d_REPRESENTATIVE_LEGAL_ONE.jpg: 640x480 1 nif, 146.0ms
```

Figura 17: Consola con tiempos de ejecución de YOLO.

Se puede observar que los tiempos oscilan entre 135 y 155 milisegundos, lo que resulta un tiempo razonable para considerar el modelo escalable y capaz de manejar un volumen significativo de expedientes y documentos.

Un punto importante a tener en cuenta, es que este proyecto no es altamente dependiente de la velocidad de ejecución de la solución, ya que no se requiere ninguna acción o análisis en tiempo real. Se trata de un proceso que se puede realizar de forma independiente y

desconectada, cargando los resultados en la plataforma de revisión documental a su finalización, ya que, de no haber realizado la revisión automática, es posible realizar la revisión manual de expedientes administrativos.

A modo de conclusión, hay que destacar que los tiempos de ejecución de YOLO son más que suficientes para llevar a cabo el proceso de manera adecuada. De todos modos, dado que los tiempos de ejecución no son un factor crítico, se podría utilizar un modelo de detección más complejo que permita, aun sacrificando velocidad de ejecución, una mayor precisión.

## **6.2 EVALUACIÓN DEL MODELO**

Otro de los aspectos evaluados de manera pormenorizada es el modelo diseñado y entrenado para la detección de regiones de interés dentro de los documentos digitalizados. En esta línea, se han realizado pruebas sobre múltiples documentos, de cara a determinar si el modelo de detección es adaptable a diferentes tipos de documentos o presenta *overfitting*, es decir, se ajusta demasiado a los documentos utilizados en el entrenamiento.

Se ha observado que, al haberse entrenado con un conjunto de datos muy reducido y basado principalmente en una plantilla específica de documento de representación, el modelo está sobreentrenado con dicha plantilla, mostrando problemas a la hora de detectar regiones de interés en otro tipo de documentos.

Este hecho tiene impacto en la precisión de la revisión documental y flexibilidad de la solución, ya que el modelo no puede utilizarse de forma generalizada para diferentes tipos de procesos administrativos y documentos. A pesar de ello, dentro del contexto del proyecto no supone un problema, ya que, para el proceso de revisión documental alcanzado, se utiliza prácticamente siempre la plantilla con la que se ha entrenado el modelo de detección.

A modo de conclusión, hay que destacar que el modelo presenta *overfitting*, por lo que en futuras implementaciones que aspiren a conseguir una solución más transversal que

contemple múltiples tipos de documentos, será esencial considerar una estrategia de entrenamiento con un mayor conjunto de datos.

### **6.3 UMBRAL DE VALIDACIÓN**

El último de los aspectos evaluados detalladamente es el umbral de validación utilizado para decidir si la detección de una región de interés dentro de los documentos digitalizados es correcta o errónea. Se trata de un componente crucial en el proceso de revisión documental, ya que dependiendo de lo estricto que sea dicho umbral, el número de documentos validados aumentará o disminuirá.

De cara a realizar un ajuste adecuado del umbral de validación, se ha seguido la siguiente metodología de ajuste de la precisión del modelo:

- Realizar un primer filtro con un umbral de validación lo más estricto posible, de modo que el texto extraído con la tecnología OCR sea exactamente igual que el NIF real.
- Dado que el porcentaje de documentos validados positivamente no resulta suficiente, reducir el umbral de validación hasta un 89%, que está asociado, en base a las pruebas realizadas, con la diferencia de un carácter diferente entre el texto extraído con la tecnología OCR y el NIF real. Esta diferencia se considera aceptable, ya que puede deberse a falta de precisión en la selección de la región de interés o en la lectura por parte del OCR.

En primera instancia, el umbral de validación utilizado no considera parámetros adicionales asociados con la ejecución de la solución como el coeficiente de confianza de YOLO o del OCR. De este modo, si se deseara llevar a cabo una validación más estricta, estos son parámetros que se podrían tener en cuenta para el desarrollo de un nuevo umbral.

A modo conclusión, hay que destacar que el umbral de validación es un parámetro ajustable, lo que permite tener flexibilidad a la hora de decidir el grado de similitud exigido para dar

por valido un documento. En base a las pruebas realizadas, se ha definido como umbral de validación idóneo un 89%.



## Capítulo 7. CONCLUSIONES Y TRABAJOS FUTUROS

En este capítulo, se exponen las principales conclusiones obtenidas al evaluar la solución desarrollada en el proyecto y los posibles trabajos futuros a tener en cuenta.

Por un lado, considerando el análisis realizado sobre los resultados obtenidos tras la ejecución de la solución podemos concluir lo siguiente:

- El sistema es capaz de obtener y verificar campos del documento, lo que demuestra la viabilidad de la solución en este sentido. No obstante, el modelo desarrollado en primera instancia únicamente detecta campos de interés que se corresponden con un NIF/NIE. En este sentido, de cara a poder llevar a cabo una validación íntegra del documento de representación legal, que incluya todos los requisitos de revisión, es necesario que se obtengan y verifiquen adicionalmente tanto los campos de nombre y apellidos como los de firmas manuscritas (un documento incompleto o sin firmar no es válido para acreditar la representación).
- Esta solución es escalable y puede manejar un volumen significativo de expedientes. Además, la solución se puede ejecutar por bloques de expedientes en caso de ser necesario y unificar posteriormente los resultados.
- Existe un *overfitting* del modelo entrenado de YOLO, haciendo que el modelo de detección sea válido únicamente para la estructura de documento utilizado en el caso de uso analizado. Esta situación hace que la solución no pueda ser utilizada para reconocer otros formatos y tipologías de documentos, sin antes ser entrenada para detectar los mismos.
- La solución recopila todos los parámetros, así como la decisión final de verificación, lo que aporta transparencia al proceso. Además, para los expedientes en los que no se detectan objetos en sus documentos, se añade un mensaje de error.
- La solución se puede integrar fácilmente con la plataforma de revisión documental en producción, siendo posible además la visualización de los resultados en una

interfaz intuitiva para los revisores documentales, sin alterar el proceso de revisión existente.

Por otro lado, se plantean los posibles trabajos futuros para llevar a cabo mejoras en la solución diseñada e implementada durante el proyecto:

- Utilizar un modelo con una arquitectura más potente, ya que utilizando un mayor volumen de parámetros se podría obtener una mayor precisión. Esto afectaría en la velocidad de ejecución del modelo, pero tiene un impacto reducido ya que se puede realizar de forma independiente al proceso de revisión manual sin interrumpir el mismo.
- Entrenar un nuevo modelo que detecte firmas como objeto adicional, de cara a revisar todos los requisitos necesarios dentro del documento de representación legal.
- Generar un conjunto de datos de entrenamiento más completo, que contenga diferentes plantillas e incluso documentos manuscritos con los que llevar a cabo un nuevo entrenamiento del modelo de detección.
- Probar nuevas soluciones que utilicen tecnología OCR para conseguir mejoras en la precisión de la lectura de caracteres.

En conclusión, se ha diseñado e implementado una solución que da cobertura a los objetivos y requerimientos definidos para el proyecto, si bien es cierto que durante su ejecución se han identificado potenciales acciones para su mejora, con las cuales la solución se podría implementar en cualquier tipo de plataforma de revisión documental, más allá de la utilizada como referencia para la realización del proyecto.

## Capítulo 8. BIBLIOGRAFÍA

- [1] Amazon S3. [En línea]. Disponible en: <https://aws.amazon.com/es/s3/getting-started/?nc=sn&loc=6&dn=1>
- [2] Dillon Reis, J., Kupec, J., Hong, J., & Daoudi, A. (May 17, 2023). Real-Time Flying Object Detection with YOLOv8. Disponible en: <https://arxiv.org/abs/2305.09972>
- [3] EasyOCR documentation. [En línea]. Disponible en: <https://www.jaided.ai/easyocr/documentation/>
- [4] EnriqueAV. (May 12, 2018). Detección de objetos con YOLO: implementaciones y cómo usarlas. Disponible en: <https://medium.com/@enriqueav/detecci%C3%B3n-de-objetos-con-yolo-implementaciones-y-como-usarlas-c73ca2489246>
- [5] Great Learning Team. (Jul 21, 2020). YOLO object detection using OpenCV. Disponible en: <https://www.mygreatlearning.com/blog/yolo-object-detection-using-opencv/>
- [6] Hui, J. (Mar 28, 2018). Object Detection Speed and Accuracy Comparison: Faster R-CNN, R-FCN, SSD, and YOLO. Disponible en: <https://jonathan-hui.medium.com/object-detection-speed-and-accuracy-comparison-faster-r-cnn-r-fcn-ssd-and-yolo-5425656ae359>
- [7] Hui, J. (Mar 18, 2018). Real-time Object Detection with YOLO, YOLOv2 and now YOLOv3. Disponible en: <https://jonathan-hui.medium.com/real-time-object-detection-with-yolo-yolov2-28b1b93e2088>
- [8] Hui, J. (Mar 28, 2018). Understanding Region-based Fully Convolutional Networks (R-FCN) for object detection. Disponible en: <https://jonathan-hui.medium.com/understanding-region-based-fully-convolutional-networks-r-fcn-for-object-detection-828316f07c99>

- [9] Jifeng Dai, Yi Li, Kaiming He, Jian Sun. (Jun 21, 2016). R-FCN: Object Detection via Region-based Fully Convolutional Networks. Disponible en: <https://arxiv.org/abs/1605.06409>
- [10] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. (May 9, 2016). You Only Look Once: Unified, Real-Time Object Detection. Disponible en: <https://arxiv.org/abs/1506.02640v5>
- [11] Joseph Redmon, & Farhadi, A. (Apr 8, 2018). YOLOv3: An Incremental Improvement. Disponible en: <https://arxiv.org/abs/1804.02767>
- [12] Portal de Ayudas del Ministerio de Economía y Competitividad. [En línea]. Solicitudes. Disponible en: <https://portalayudas.mineco.gob.es/THD/solicitudes/Paginas/Solicitudes.aspx>
- [13] Saarthi AI. (May 9, 2019). How to Build Your Own OCR. Medium. Disponible en: <https://medium.com/saarthi-ai/how-to-build-your-own-ocr-a5bb91b622ba>
- [14] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. (Jan 6, 2016). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Disponible en: <https://arxiv.org/abs/1506.01497v3>
- [15] Ultralytics YOLOv8 Documentation. [En línea]. Disponible en: <https://docs.ultralytics.com/>
- [16] Visual Studio Code. [En línea]. Disponible en: <https://code.visualstudio.com/>
- [17] Computer Vision Annotation Tool (CVAT). [En línea]. Disponible en: <https://www.cvat.ai/>
- [18] OpenCV Documentation. [En línea]. Disponible en: <https://docs.opencv.org/4.x/index.html>

## ANEXO I: ALINEACIÓN CON LOS OBJETIVOS DE DESARROLLO SOSTENIBLE

El presente anexo describe como la solución diseñada e implementada para la agilización del proceso de revisión documental de expedientes administrativos se encuentra alineada con los Objetivos de Desarrollo Sostenible (ODS).

En primer lugar, se muestran los ODS establecidos por la Organización de Naciones Unidas (ONU):



*Figura 18: Objetivos de desarrollo sostenible.*

La correspondencia entre los ODS y las áreas clave del proyecto se presentan en los siguientes puntos:

- **Objetivo 9: Industria, Innovación e Infraestructura:** Este objetivo está enfocado en construir infraestructuras resilientes, promover la industrialización sostenible y fomentar la innovación. Se trata de un objetivo importante para los proyectos de desarrollo de nuevas soluciones tecnológicas, a través de los cuales se permite un uso eficiente de recursos. El proyecto de revisión automatizada de documentos incorpora tanto tecnologías de procesamiento de imágenes como redes neuronales para mejorar la eficiencia en la verificación de documentos, el uso de estas tecnologías en auge contribuye a la promoción de la innovación tecnológica en el ámbito de la inteligencia artificial y el aprendizaje automático, impulsando el desarrollo y aplicación de las mismas en un ámbito administrativo como es la revisión de expedientes.
- **Objetivo 12: Producción y Consumo Responsables:** Este objetivo está enfocado principalmente en desvincular el desarrollo y crecimiento económico de la degradación medioambiental, además de aprovechar al máximo los recursos y promover un estilo de vida sostenible. La implementación de un sistema automatizado de revisión documental permite optimizar los procesos de verificación, reduciendo la necesidad de recursos, alineándose así con el objetivo de producción y consumo responsables, promoviendo prácticas más sostenibles en la gestión de documentos.
- **Objetivo 16: Paz, Justicia e Instituciones Sólidas:** Este objetivo está enfocado en promover sociedades justas, pacíficas e inclusivas. El sistema automatizado de revisión documental contribuye en la acreditación de representación legal, permitiendo garantizar la autenticidad de la información en los documentos de forma más ágil, lo que fortalece la confianza en las instituciones y promueve la justicia, aportando transparencia en los resultados. Esto se alinea específicamente con el objetivo 16, que consiste en crear a todos los niveles, instituciones eficaces y transparentes.

*ANEXO I: ALINEACIÓN CON LOS OBJETIVOS DE DESARROLLO SOSTENIBLE*

---

En conclusión, la solución diseñada e implementada persigue la eficiencia, precisión y sostenibilidad en los procesos de verificación documental, buscando alcanzar algunas de las metas propuestas por los ODS y los líderes mundiales.