

Desarrollar un OCR para leer la ficha técnica de un coche

Víctor Batista Medeiros

Resumen– En este proyecto se ha desarrollado un sistema OCR con el objetivo de automatizar la tarea de rellenar el documento oficial que te permite dar de baja un vehículo (certificado de destrucción). Para ello, en este documento se explica todo el pipeline efectuado, desde la detección de los campos de texto hasta su volcado en el documento oficial, pasando por el preprocesado y reconocimiento de estos.

Palabras clave– Ficha Técnica, Certificado De Destrucción, Visión Por Computador, Aprendizaje Profundo, Reconocimiento Óptico De Caracteres (OCR), YOLO, Faster R-CNN, Tesseract.

Abstract– In this project, an OCR system has been developed with the aim of automating the task of filling the official document that allows you to deregister a vehicle (certificate of destruction). For this purpose, this document explains the entire pipeline developed, from the detection of the relevant text fields to their transcription in the official document, including the preprocessing and recognition of these fields.

Keywords– Data Sheet, Certificate Of Destruction, Computer Vision, Deep Learning, Optical Character Recognition (OCR), YOLO, Faster R-CNN, Tesseract.

1 INTRODUCCIÓN

LA ficha técnica de un vehículo (Figura 1) es uno de los documentos obligatorios más importantes de un coche, ya que gracias a este documento se puede acreditar que el vehículo pasa las homologaciones técnicas correspondientes para circular por carretera (ITV).

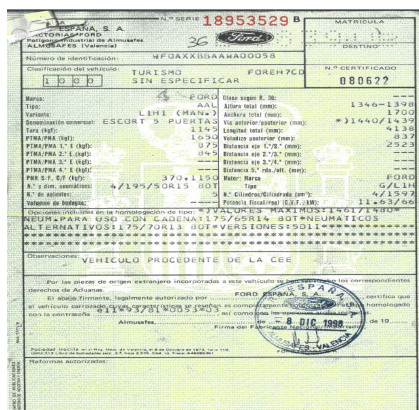


Fig. 1: Ficha Técnica

Cuando un vehículo llega al final de su vida útil es necesaria la ficha técnica para darlo de baja, y es por eso que en este proyecto se propone desarrollar un sistema OCR que permita automatizar la tarea de rellenar el documento oficial para dar de baja un vehículo (certificado de destrucción de anexos) a partir de una imagen escaneada de dicha ficha.

Para ello, lo primero que se realiza es la detección de los campos de texto que nos interesan mediante una red neuronal a partir del modelo YOLOV5 o Faster R-CNN. Posteriormente, se ejecuta un preprocesado de las imágenes, dado que estas suelen estar deterioradas y contener mucho ruido de fondo, mediante técnicas de binarización, eliminación de ruido, alineamiento... Y finalmente se reconoce su contenido gracias a un motor de OCR que emplea Tesseract.

El artículo está organizado de la siguiente manera: En la Sección 2 se describen los objetivos del proyecto. En la Sección 3 se indica cuál ha sido la metodología utilizada y la planificación establecida y posteriormente en la Sección 4 se explica el estado del arte, centrándose en dos casos similares a este proyecto y en su estructura y técnicas empleadas. Luego, en la Sección 5 se describe la planificación efectuada y en la Sección 6 el desarrollo general de esta. En la Sección 7 se explican los resultados obtenidos y finalmente en la Sección 8 las conclusiones pertinentes a estos y los posibles trabajos futuros.

2 OBJETIVOS

El objetivo principal de este proyecto consiste en diseñar un programa que, a partir de una imagen escaneada de una

- E-mail de contacto: victor.batista@e-campus.uab.cat
- Mención realizada: Computación
- Trabajo tutorizado por: Coen Antens (CVC)
- Curso 2022/2023

ficha técnica de un coche, permita extraer una determinada información de esta, para poder rellenar un conjunto de campos del formulario que es necesario para darlo de baja. Una vez definido esto podemos dividir el proyecto en el siguiente conjunto de objetivos y subobjetivos:

1. La detección de los campos de texto deseados mediante técnicas de Deep Learning para poder clasificarlos y segmentarlos. Dentro de este objetivo se entrenan dos modelos distintos que son YOLOV5 y Faster R-CNN para así poder comparar dos redes neuronales que siguen estrategias diferentes en cuanto a detección de objetos.
2. El preprocesamiento y reconocimiento del contenido de las imágenes segmentadas mediante Tesseract para obtener los caracteres o signos pertinentes.

Una vez realizado esto se analizan los resultados y se almacenan los datos en un formato adecuado y de fácil conversión, además del volcado de estos en el formulario pertinente para dar de baja el vehículo.

3 METODOLOGÍA

Para la realización de este proyecto empleo una metodología ágil [1]. Esta metodología se caracteriza por proporcionar un modelo iterativo e incremental capaz de adaptarse a cambios, con flexibilidad para incorporar nuevas fases en caso de ser necesario y con la finalidad de entregar valores de forma temprana y continua.

Para ello, se realizan reuniones semanales con el tutor con el objetivo de generar un feedback de las tareas realizadas para modificarlas si es necesario o definir las próximas a efectuar. Gracias a esto se pueden detectar problemas de manera temprana y generar un control de la evolución del proyecto.

En cuanto a la planificación de tareas a efectuar durante el transcurso del proyecto, se ha realizado un diagrama de Gantt mediante la herramienta Microsoft Project, consiguiendo de esta manera tener una estimación de la duración y plazos de las actividades, además de generar una visión general del proyecto. Este diagrama se puede encontrar en el apartado Gantt de anexos.

4 ESTADO DEL ARTE

En el caso de este proyecto partimos de un problema del que no se dispone ningún programa que realice una labor igual, pero sí que existen proyectos ciertamente parecidos o con objetivos similares, tales como el caso de la extracción de campos en tickets [2, 3] o la extracción de texto en imágenes que no siguen la estructura típica de página de texto, también conocidos como “text extraction from images in the wild” [4, 5], en los que encontramos ejemplos como el reconocimiento de matrículas, lectura de carteles, etc.

Según los casos de proyectos mencionados anteriormente, la extracción de campos de texto específico sigue un flujo típico como el que aparece en la Figura 2.

Siguiendo el pipeline, lo primero que se hace es preprocesar la imagen para eliminar el ruido, mejorar la claridad del texto, poner una correcta orientación, etc. Decir que este

paso es muy importante, ya que, dependiendo de la calidad del texto, la precisión para reconocerlo será mayor o menor y que en muchos casos suele ir después del paso de detección, dado que el background no es uniforme y nos interesa preprocesar el texto una vez ya lo tenemos segmentado, puesto que de esta manera suele ser más eficiente y menos difícil.

Seguidamente, se aplican redes de aprendizaje profundo para clasificar y segmentar las partes del texto deseadas y luego se aplican técnicas de OCR para reconocerlo.

Como último paso del pipeline se guardan los datos extraídos en el formato deseado o que más se adecue a nuestro proyecto.

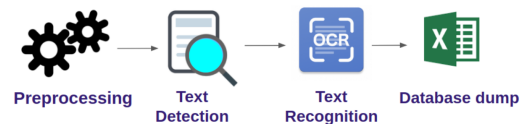


Fig. 2: Flujo extracción de texto

Respecto al paso de reconocimiento de texto, decir que existen una gran variedad de redes neuronales como la YOLOV5 [6], Faster R-CNN [7], EAST [8] ... Las cuales se diferencian principalmente por estar enfocadas a reducir el tiempo, obtener la precisión más elevada posible, detectar objetos pequeños ... Y es por ello por lo que no hay una sola estrategia posible y que cada proyecto emplea una distinta, ya que el modelo a seleccionar depende de que métrica se quiere optimizar y de la calidad o tipo de imagen de la que disponemos. Si nos centramos en los proyectos mencionados anteriormente podemos encontrar que para el caso de la detección de texto de matrículas se utiliza la red neuronal YOLOV5 consiguiendo un mean average precision (mAP) 5:95 del 75 % y para el problema de detección de texto en tickets mediante Faster R-CNN se consigue un mAP50 del 95 % dado que esta métrica es mucho más permisiva.

Finalmente, en cuanto al reconocimiento de texto, los proyectos suelen emplear librerías de OCR tales como Amazon Textract [9], Keras [10], Azure Cognitive Service [11] ... En función del tipo de imagen o texto, pero la técnica más empleada en la mayoría de estos es Tesseract, dado que es un motor de reconocimiento de código abierto que es compatible con muchos lenguajes de programación y que además proporciona buenos resultados (se consideran buenos cuando la precisión del OCR es superior al 90 %).

5 PLANIFICACIÓN

En esta sección se especifican todos los pasos seguidos para desarrollar el sistema de detección y reconocimiento de campos de texto, empezando por la creación del ground truth y acabando por el volcado de datos.

Ground Truth:

La realización del Ground Truth del proyecto está dividida en dos partes, en que, por un lado, se realiza la anotación de las imágenes y por el otro la descripción del contenido de texto.

En cuanto al anotado de imágenes, decir que este se realiza con la herramienta Labelimg [12] y que estas se guardan en el formato PASCAL VOC XML, ya que es el formato más utilizado y compatible para entrenar redes neuronales.

Respecto la cantidad de información que debemos clasificar de cada imagen, es necesario extraer 7 campos, los cuales son tanto numéricos como de caracteres. Los campos a clasificar son los siguientes:

1. Marca
2. Denominación comercial
3. Tara (kg)
4. Número de identificación
5. Motor: Tipo
6. N°. Cilindros/cilindrada
7. Potencia fiscal/real (C.V.F/KW)

Finalmente, como hemos mencionado anteriormente, se realiza la identificación manual del contenido de los campos de texto para poder entrenar el modelo de OCR.

Detección de campos de texto:

Una vez tenemos generado el Ground Truth de anotaciones, se entrenan las redes neuronales YOLOV5 y Faster R-CNN mediante Python.

He seleccionado estas dos redes para poder comparar dos estrategias de segmentación distintas, ya que en este caso YOLO es un detector de una sola etapa en que una red convolucional predice los cuadros delimitadores y las probabilidades de clase para estos cuadros, mientras que Faster R-CNN es un modelo de dos etapas en que en la primera etapa una red convolucional proporciona un mapa de características y se identifican las propuestas de región y en la segunda etapa las propuestas de región se remodelan mediante una capa de agrupación ROI (region of interest) que luego se usa para clasificar la imagen dentro de la región propuesta y predecir los valores de compensación para los cuadros delimitadores. Esto se puede observar en la imagen Comparativa Arquitectura YOLO y Faster R-CNN de anexos.

La diferencia principal de estas dos técnicas se basa en el tiempo, dado que en el caso de la Faster R-CNN al tener dos etapas tarda más en realizar la predicción, pero los resultados proporcionados suelen ser más precisos y es por eso que es una buena alternativa para este problema dado que el tiempo no es un inconveniente y lo que se busca es conseguir la precisión más alta.

Para poder evaluar estas redes, la métrica que se emplea es el mAP (mean average precisión), ya que es la métrica más empleada para evaluar detectores de objetos o texto y que se basa en calcular la precisión (porcentaje predicciones correctas) promedia para un valor de Recall (porcentaje de posibles positivos encontrados) de 0 a 1.

Preprocesado:

El preprocesado es una de las partes más importantes, ya que la precisión del OCR depende considerablemente de como de bien se distingan los caracteres respecto al background. Para ello el preprocesado se divide en 2 partes:

1. Eliminación de las palabras pequeñas que aparecen de fondo con el texto de España. Dado que estas palabras aparecen en un color verdoso se realiza una segmentación de color y en caso de proporcionar malos resultados se prueba de realizar un filtrado y eliminado de ruido pequeño mediante blurring y contornos.
2. Se aplican 3 técnicas de morfología para mejorar la calidad del texto [13]. Estas técnicas son las siguientes:
 - a) Binarización
 - b) Corrección de inclinación
 - c) Eliminación de ruido

Estas técnicas se aplican dado que el OCR funciona mejor con texto, que está bien alineado, con los contornos bien definidos y sin ruido de fondo.

Detección y Reconocimiento de texto:

Para la parte de reconocimiento de texto se entrena Tesseract 4.1.0 con el modo LSTM, ya que en esta versión y modo emplea un enfoque de aprendizaje profundo caracterizado por tener conexiones de retroalimentación que permiten proporcionar una mayor precisión.

En este caso, la métrica que empleo para comprobar la calidad del modelo es la tasa de errores de caracteres (CER), la cual está basada en la distancia de Levenshtein [14] y que se emplea para analizar el número mínimo de operaciones a nivel de carácter para que el texto del Ground Truth y el generado coincidan, es decir, analiza como difieren los textos entre sí.

Con el modelo de OCR entrenado tenemos que separar la parte de texto redundante de la que realmente nos importa. Para ello se emplean 3 técnicas que son:

- **EAST:** Es un modelo de aprendizaje profundo que está entrenado para predecir directamente la existencia de texto a partir de una red convolucional que permite generar predicciones densas por píxeles de palabras, eliminando así pasos intermedios de modelos tradicionales como la propuesta de candidatos, formación de regiones de texto ...
- **Dilatation Method:** Aplicar técnicas morfológicas de dilatación para unir las letras y así detectar objetos con un área mínima (palabras).
- **Rules Method:** Consiste en diseñar reglas a partir del modelo Tesseract, reglas como el reconocimiento de solo valores numéricos, la posición de la palabra en la imagen ...

Volcado de datos:

Finalmente, se introducen los datos pertinentes en el documento PDF que nos permite dar de baja el vehículo y se guardan también en un archivo Excel para mantener un histórico de estos.

6 DESARROLLO

En esta sección se explica de forma detallada como se han implementado todos los pasos mencionados anteriormente en el apartado de planificación.

Ground Truth de anotaciones

Lo primero que he realizado ha sido crear el dataset pertinente para poder entrenar las redes neuronales, consiguiendo así que sean capaces de identificar los elementos deseados dentro de la ficha técnica.

Para ello he anotado 250 imágenes en formato PASCAL VOC XML, ya que es el formato más compatible y usado para entrenar redes neuronales, y he dividido el dataset generado en 80 % train, 10 % test y 10 % validation, priorizando el train dada la poca cantidad de imágenes disponibles.

YOLOV5

Una vez generadas las anotaciones, como he mencionado anteriormente, estas están en formato PASCAL VOC XML y, por lo tanto, las he convertido al formato compatible con YOLO mediante unas operaciones que consisten en:

1. Transformar las coordenadas con las siguientes operaciones:

$$X = \frac{(X_{min} + X_{max})}{2}$$

$$Y = \frac{(Y_{min} + Y_{max})}{2}$$

$$Width = X_{max} - X_{min}$$

$$Height = Y_{max} - Y_{min}$$

Donde Xmin, Ymin, Xmax y Ymax corresponden a las coordenadas de las posiciones superior izquierda e inferior derecha de la imagen.

2. Normalizar las coordenadas a partir de las dimensiones de la imagen.

Con los datos en el formato correspondiente he entrenado una red neuronal YOLOV5 con el modelo YOLOV5S usando 100 epochs y un batch size de 32 ya que al tener un dataset pequeño son los valores estándar que se utilizan, además, he configurado dicha red para que realice data augmentation del tipo escalado, ajuste de espacio de color, volteado y mosaico.

Este último se encarga de combinar imágenes de entrenamientos en ciertas proporciones y de forma conjunta, tal y como se puede apreciar en Lote de entrenamiento de anexos. Remarcar que esta modificación es muy importante, ya que gracias a esta se mejora la precisión de la red neuronal respecto a la detección de objetos pequeños.

Finalmente, se ha implementado un código que te permite cargar el modelo mediante la librería Torch.Hub para así poder usar la red neuronal en entornos Python.

Faster R-CNN

Para entrenar la red neuronal Faster R-CNN no he tenido que realizar una conversión de datos, ya que el formato que utiliza es Pascal VOC.

Para ello he usado un modelo de Faster R-CNN ResNet50 FPN [16] y lo he configurado para que sea capaz de detectar los campos de texto necesarios en mi problema. Para esta red he usado 100 epochs y un batch size de 32 y como técnicas de data augmentation he configurado el programa para que aplique rotaciones y volteados a las imágenes.

Finalmente, al igual que en el modelo YOLOV5 he utilizado la librería Torch.Hub para poder cargar los pesos y usar la red en Python.

Preprocesado

En cuanto al preprocesado de las imágenes, el objetivo principal de este se ha centrado en eliminar todos los artefactos que difieran del texto en cuestión, como sería el caso de las palabras de fondo que aparecen de España o las líneas verticales u horizontales de los bordes de la ficha mediante las siguientes técnicas [13]:

1. Binarización y eliminación de ruido:

Para la binarización de la imagen he utilizado un umbralización adaptativa la cual binariza la imagen por secciones en función de las características locales de cada una de estas, es decir, no se aplica un umbral fijo para toda la imagen sino que se calcula un umbral diferente en función de la iluminación de la zona.

Con la umbralización aplicada me he centrado en la eliminación del ruido de fondo mediante las siguientes técnicas:

- **Gaussian Blur:** Permite mezclar las intensidades de los objetos pequeños para que se mezclen con el fondo, es decir, los píxeles cercanos con intensidades similares se consideran para el desenfoque.
- **Median Blur:** Remplaza el elemento central en el área del kernel con el valor medio de los píxeles debajo del área del kernel. Es útil para quitar ruido salt and pepper.
- **Bilateral Filtering:** Al igual que el Gaussian Blur pero con intensidades similares al pixel central. Por lo tanto, conserva los bordes, ya que los píxeles en los bordes tienen una gran variación de intensidad.
- **Fourier Transform:** Nos permite identificar estructuras de frecuencias periódicas para así eliminarlas.
- **Color Segmentation:** Nos permite eliminar los elementos en un rango de colores predefinido.

Una vez implementadas estas técnicas conjuntamente con Canny para detectar los bordes de las letras, he observado que la técnica que parece ofrecer mejor resultados es el Gaussian Blur tal y como se puede apreciar

en la Figura 3, pero que los resultados var an mucho en funci n de la imagen, ya que, dependiendo de la intensidad de las letras, los colores ... Se elimina bastante cantidad de estas.



Fig. 3: Preprocesado con filtro gaussiano

Decir tambi n que en este caso la segmentaci n de color [17] proporciona malos resultados dada la variedad de colores e intensidades de las im genes.

No conforme con estos metodos de preprocesado he buscado alguna otra alternativa que no fuera tan agresiva y he encontrado que la mejor forma es aplicar el m todo de binarizaci n de Otsu dado que en mi problema las im genes son bimodales, es decir, que la mayor a de p xeles se distribuyen en 2 regiones dominantes (el fondo verde y las letras negras)

Otsu nos permite encontrar el punto m nimo del histograma de la imagen, para poder separar las dos distribuciones predominantes, convirtiendo la imagen en una imagen en blanco (fondo) y negro (las letras) tal y como se muestra en la Figura 4.

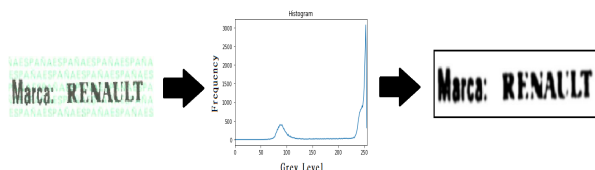


Fig. 4: Otsu Binarization

Agregar que en algunos casos la distribuci n de p xeles no es tan precisa por el hecho de que hay im genes muy deterioradas en que no hay una clara distinc n entre fondo y letras, y es por eso que aplicado un filtro gaussiano a posteriori para as  eliminar aquellos peque os fragmentos de fondo que quedan como ruido.

Finalmente, para la eliminaci n de l neas verticales y horizontales, lo que he hecho ha sido definir dos kernels, uno vertical (2,90) y uno horizontal (250,2), y he aplicado una detecci n de contornos para eliminar los elementos que sigan la distribuci n de los kernels.

2. **Skew Correction:** Para el Tesseract es muy importante que el texto est  alineado, es por eso que he implementado la t cnica Projection Profile [18] que rota la imagen en varios  ngulos y genera un histograma de p xeles a partir de la suma de p xeles a lo largo de las filas de la matriz de la imagen. Despu s se realiza una comparaci n con la diferencia m xima entre los picos (varianza) y se escoge el mejor  ngulo. Este proceso se puede ver reflejado en la Figura 5.

Reconocimiento De Texto

Con las im genes preprocesadas ha llegado el turno de entrenar el reconocedor de texto. Para ello primero he

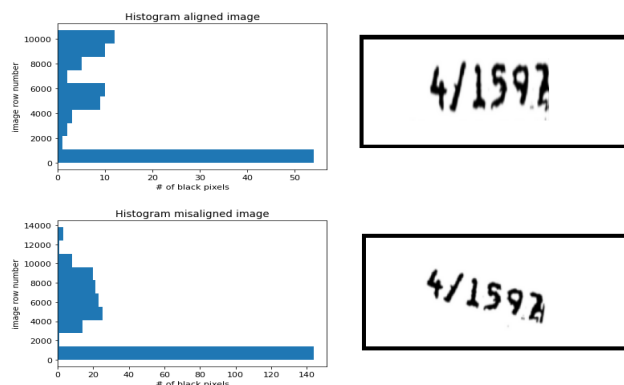


Fig. 5: Skew Correction

tenido que generar un nuevo Ground Truth y posteriormente he entrenado el OCR de Tesseract.

1. Ground Truth

Para la realizaci n del Ground Truth he cogido los labels generados anteriormente y los he recortado para eliminar las palabras de texto redundantes como son el caso de "marca", "tara" ... Remarcar que para realizar los recortes se ha implementado un programa que recorta los labels en funci n de un valor aproximado del tama o de la palabra redundante, dado que las im genes resultantes tienen proporciones similares y que en algunos casos se ha hecho manualmente.

Una vez con las im genes adaptadas he especificado el contenido de estas mediante archivos .box en los que encontramos escrito de forma manual el contenido de las im genes, consiguiendo generar de esta forma un dataset con 1700 im genes y 1700 archivos .box .

Debido a que para entrenar un reconocedor de texto se requieren miles de im genes, he implementado una funcionalidad que permite generar im genes de texto mediante la librer a PIL [19].

Para ello he tenido que buscar diversas fuentes de texto mediante herramientas de internet y he encontrado que la que m s se adapta es Calibri Bold. Adem s, para que estas im genes resultantes sean lo m s parecidas a las del dataset se les ha aplicado primero un filtro gaussiano para difuminarlas y posteriormente un poco de ruido gaussiano en forma de puntos, tal y como se muestra en la Figura 6.

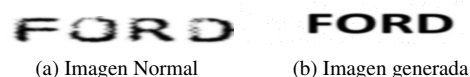


Fig. 6: Comparativa imagen normal y generada

2. Tesseract

Una vez con el Ground Truth generado se ha entrenado el OCR Tesseract 4.1.0 conjuntamente con Leptonica 1.8.0 (librer a de c digo abierto que incluye software para el procesamiento y an lisis de im genes). Para ello, Tesseract dispone de tres modos de entrenamiento basados en:

- a) **Training From Scratch:** Como bien indica el nombre, este modo de entrenamiento consiste en entrenar la red neuronal desde cero y por consiguiente se requiere un dataset representativo y grande. En este modo de entrenamiento cabe el peligro de sobre ajustar la red con los datos proporcionados.
- b) **Fine Tuning for Impact:** Consiste en entrenar una modelo de lengua ya existente a partir de unos datos adicionales sin cambiar ninguna parte de la red. Este modo de entrenamiento es útil en aquellos casos en que el texto del problema tiene similitudes con el del modelo entrenado y por consiguiente este solo se centra en refinar el modelo a nuestro tipo de datos.
- c) **Fine Tuning for \pm a few characters:** Este modo de entrenamiento es parecido al anterior, pero con la particularidad de que este se centra no en refinar el modelo completo, sino en hacer que este aprenda a detectar algún carácter que no se encuentre dentro de su rango de caracteres.

En este caso dado la naturaleza del problema y de sus datos, se han entrenado dos redes neuronales desde cero, una con el dataset preprocesado y otra sin, y 3 redes a partir del ajuste fino para el impacto, una sin el dataset preprocesado, otra con los datos preprocesados y la última con este mismo dataset más las imágenes generadas con la librería OpenCV.

Métricas de análisis:

Para poder evaluar el rendimiento y precisión de las redes neuronales y el OCR se han implementado las técnicas que permiten calcular el mAP (Mean Average Precision) y el CER (Character Error Ratio).

■ Character Error Ratio

Como bien indicamos anteriormente, el cálculo del CER está basado en la distancia de Levenshtein, la cual se utiliza para saber el número de operaciones necesarias para transformar una palabra en otra. Un ejemplo práctico de esto sería el caso en que Word te proporciona cuál es la palabra más probable a la tuya mediante el cálculo de esta distancia.

Una vez introducido esto, para poder calcular el CER se emplea la siguiente ecuación:

$$CER = \frac{S + D + I}{S + D + I + C}$$

En que S es el número de substituciones, D de eliminaciones, I de inserciones y C de caracteres correctos.

Para poder obtener los valores de estas variables primero se debe desarrollar la matriz de distancia de Levenshtein. Para ello se crea una matriz nxm en que n

y m son las longitudes de las cadenas de palabras y seguidamente se calculan los valores de dicha matriz utilizando la ecuación de la Figura 7 para cada casilla de la matriz.

$$\min \begin{cases} \max(i, j) & \text{si } \min(i, j) = 0, \\ \begin{matrix} lev_{a,b}(i-1, j) + 1 \\ lev_{a,b}(i, j-1) + 1 \\ lev_{a,b}(i-1, j-1) + 1_{a_i \neq b_j} \end{matrix} & \text{si } \min(i, j) \neq 0 \end{cases}$$

Fig. 7: Cálculo Posiciones Matriz de Levenshtein

A modo de entendimiento, que esta ecuación se centra en calcular cuál es la operación de coste mínimo necesaria para pasar de la letra de la palabra real a la calculada por el OCR, teniendo en cuenta que desplazarse en diagonal en la matriz equivaldría a realizar una substitución, en horizontal una inserción y en vertical una eliminación y que cada operación de estas tendría coste 1 a excepción de la substitución que puede tener coste 0 en caso de que las dos letras sean iguales.

Con la matriz de Levenshtein completada, el siguiente paso consiste en encontrar el recorrido de coste mínimo desde la última casilla de la matriz (correspondiente al valor de distancia de Levenshtein) hasta el primer elemento de la matriz, consiguiendo de esta forma saber el número de operaciones de cada tipo. Para ello, los pasos a seguir consisten en seleccionar la casilla adyacente que tiene coste inferior y en función de si es desplazamiento horizontal, diagonal o vertical, incrementar en 1 el valor de dicha operación. Decir que en el caso de que el coste mínimo sea igual al actual se incrementaría en 1 la variable de caracteres correctos.

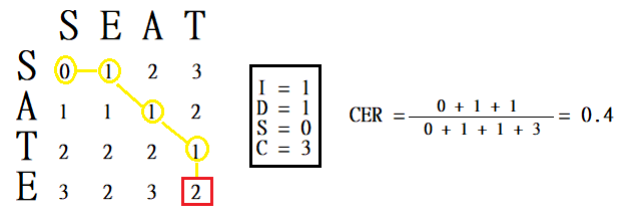


Fig. 8: Ejemplo cálculo CER

■ Mean Average Precision (mAP)

El mean average precision es la media del conjunto de áreas que se generan al representar la PR (precision-recall) curva de cada clase. Para poder representar esta curva es necesario calcular el Precision y el Recall tal y como se muestra en la Figura 9.

$$\begin{aligned} \text{Precision} &= \frac{TP}{TP + FP} & TP &= \text{True positive} \\ & & TN &= \text{True negative} \\ \text{Recall} &= \frac{TP}{TP + FN} & FP &= \text{False positive} \\ & & FN &= \text{False negative} \end{aligned}$$

Fig. 9: Cálculo Precision y Recall

En este caso, para considerar que una predicci3n es TP o FP se debe calcular el IOU (Intersection Over Union) que nos proporciona un valor de superposici3n entre la zona predicha y la real, sabiendo de esta manera como de pr3ximo es el 1rea de la predicci3n al 1rea real de la clase. Una vez calculada dicha intersecci3n, se establece un threshold a partir del cual se considera que una predicci3n es TP cuando el IOU es superior a este o FP cuando es inferior.

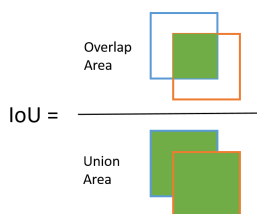


Fig. 10: C1culo IOU

Finalmente, incluir que en este caso el tipo de mAP que se ha calculado es el mAP@[.5:95] que se caracteriza por calcular el mAP promedio sobre diferentes thresholds que van de 0.5 a 0.95 con saltos de 0.05 en 0.05, consiguiendo de esta manera asegurar una avauaci3n m1s precisa de nuestro modelo.

Detecci3n y reconocimiento de palabras:

■ Detecci3n de palabras

Dado que los detectores nos proporcionan como resultado im1genes en que hay una serie de palabras que no nos interesan, como ser1a el caso de “marca”, “tipo”, “denominaci3n comercial” ... He implementado tres m1todos para detectar las palabras relevantes:

- **EAST:** Es un modelo de Deep Learning que te permite identificar palabras en entornos dif1ciles. Para ello he utilizado un modelo ya preentrenado (“frozen_east_text_detection”).
- **Dilatation Method:** Este m1todo consiste en juntar las letras mediante la dilataci3n de estas con un kernel horizontal. Gracias a esto podemos detectar objetos con un 1rea m1nima (palabras) y as1 separar las que no nos interesan.
- **Tesseract Rules:** Aprovechando el modelo de Tesseract ya entrenado, lo que he hecho ha sido aplicar una serie de “reglas” para poder detectar solo la parte relevante de los bounding boxes. Las reglas aplicadas han sido las siguientes:

1. **Reconocimiento num1rico:** En el caso de las clases que referencian a la tara, la potencia o la cilindrada, al tratarse de clases que 1nicamente contiene valores num1ricos y alg1n car1cter especial como “()” o “/” lo que he hecho ha sido restringir la lista de valores v1lidos del Tesseract para evitar que detecte letras o caracteres indeseados.

2. Eliminaci3n de palabras redundantes:

Para el resto de las clases lo que he hecho es almacenar el resultado del Tesseract en una lista de palabras y eliminar de esta la cantidad de palabras no deseadas. Un ejemplo de esto ser1a el caso de la clase n1mero de identificaci3n en que elimino las 3 primeras palabras de la lista. Hay que a1adir que en algunos casos el Tesseract junta palabras por equivocaci3n, como ser1a el caso de juntar la palabra “n1mero” y “de”, y es por ello que he generado una serie de condiciones en estos casos, teniendo en cuenta la cantidad de palabras de la lista.

3. Separaci3n por CER:

Como 1ltima regla, lo que he hecho para evitar los casos en que el Tesseract junta palabras es aplicar el c1culo del CER en funci3n de la palabra deseada para as1 saber a partir de que parte de la palabra separar.

Esto solo se aplica como 1ltima medida, dado que el Tesseract no est1 entrenado con las palabras redundantes y no es del todo precis3, pero al estar entrenado a partir del lenguaje espa1ol tiende a reconocer correctamente los caracteres.

Como ejemplo podemos encontrar el caso en que se nos generase una palabra con longitud superior a 20 caracteres, como ser1a el caso de la palabra “identificaci3nVVSFJX2SD34NSDJ2”. Para ello calculamos el CER respecto la palabra “identificaci3n” y todas las posibles palabras generadas al eliminar la 1ltima letra de esta. Finalmente, cortamos la palabra a partir de la posici3n en la que obtengamos el mejor CER.

- **Combinaci3n de Tesseract y dilatation method:** Como 1ltimo m1todo se ha realizado una combinaci3n de los dos m1todos anteriores en que primero se aplica el de dilataci3n y en caso de detectar una 1nica palabra porque estaban muy juntas se aplica el segundo m1todo. Remarca que se ha dado preferencia al m1todo de dilataci3n porque este al darte el contorno de la palabra te permite centrar el OCR en solo esa palabra sin tener en cuenta background consiguiendo as1 aumentar su precisi3n

Finalmente, analizando estos m1todos de forma superficial, he podido observar que el m1todo EAST con el modelo utilizado selecciona muchas zonas que no son palabras.

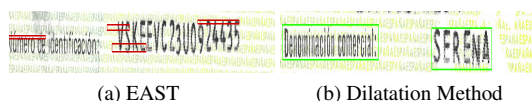


Fig. 11: Comparativa EAST y Dilatation Method

■ Reconocimiento de palabras

Para el reconocimiento de palabras he empleado el Tesseract ya entrenado, pero es importante especificar el método de segmentación de página (PSM), ya que en función de este los resultados variarán mucho.

En cuanto a métodos de segmentación de página [22], Tesseract dispone de 13 métodos que varían en función de la colocación y la cantidad de texto tal y como se puede observar en el apartado PSM de anexos. En mi caso he empleado el PSM 7 que se caracteriza por tratar la imagen como una única línea de texto.

Almacenamiento de datos:

Con el ciclo de localización y reconocimiento finalizado se han almacenado los datos tanto en el documento correspondiente de la DGT para dar de baja el vehículo como en un archivo Excel, el cual es requerido para este problema para mantener un registro de los datos.

1. Almacenamiento Documento DGT

Para almacenar los datos en el documento he empleado funciones de la librería fillpdf [20] las cuales te permiten acceder a los diversos campos del documento mediante un diccionario de palabras en el que se especifica el campo deseado como key y la consiguiente información que se desea almacenar como value.

2. Almacenamiento Excel

Para el almacenamiento de los datos en un documento Excel he utilizado funciones de la librería CSV [21]. Para ello primero se comprueba si existe un documento Excel, de lo contrario se crea con las categorías pertinentes (identificación, marca, tara ...) , y posteriormente se insertan los datos en su correspondiente categoría, tal y como se muestra en la Tabla Datos Vehículos de los anexos.

Remarcar que para que este proceso sea realizado correctamente, el Excel debe estar configurado con el delimitador “;” el cual viene por defecto.

7 RESULTADOS

Para la evaluación de los resultados he optado por analizar de forma independiente tanto la parte de detección como la parte de reconocimiento de texto. Seguidamente, se ha analizado todo de manera conjunta en un único pipeline, para así de este modo saber en qué puntos es mejor o peor mi sistema.

7.1. Detección de texto

Para evaluar el funcionamiento de las redes neuronales YOLOV5 y Faster R-CNN he calculado el mAp[5:95], tal y como mencioné anteriormente, empleando un 10 % del total de imágenes como test.

Modelo	mAP[5:95]
YOLOV5S	78.44
FASTER R-CNN	40.35

TABLA 1: MAP[5:95] MODELOS YOLOV5 Y FASTER R-CNN

Los resultados son bastante buenos teniendo en cuenta que la métrica evalúa thresholds muy altos y como podemos ver YOLOV5 es un 38.09 % más precisa que Faster R-CNN, ya que esta tiende a no seleccionar correctamente todo el bounding box e incluso a equivocarse completamente en alguna selección cosa que se puede ver reflejada en la imagen Comparativa Detección YOLOV5 y Faster R-CNN de anexos. Esto se debe a que la red neuronal Faster R-CNN requiere muchos más datos de entrenamiento para aumentar su precisión.

Por otro lado, vistos los buenos resultados de preprocesado he decidido entrenar otro modelo YOLOV5, pero con las imágenes completas previamente preprocesadas obteniendo los resultados que aparecen en la Tabla 2.

Modelo	mAP[5:95]
Modelo Sin Preprocesar	78.44
Modelo Preprocesado	75.32

TABLA 2: MAP[5:95] DE LOS MODELOS YOLOV5

De entre los dos métodos el modelo preprocesado proporciona un 3.12 % menos que el original. Esto se debe a que al aplicar la binarización Otsu a la imagen completa, como esta tiene muchos artefactos y elementos con frecuencias muy diversas, al generar el histograma y realizar la división se ven repercutidos en muchos casos las palabras de interés tal y como se muestra en el apartado Comparativa Imagen Preprocesada de anexos.

Con los modelos evaluados he realizado un estudio de los casos True Positives del modelo ganador para saber donde falla más, y como se puede ver en la Tabla 3 el modelo tiende a detectarlos todos bastante bien, pero tiene problemas con la clase identificación”. Esto se debe a que este parámetro suele estar en muchos casos mal colocados en cuanto a lo que se refiere a estar centrado dentro del recuadro delimitador.

Clase	True Positives
Marca	185
Denominación	190
Tara	189
Identificación	175
Motor:Tipo	204
Cilindrada	206
Potencia	205
TOTAL TP	1354
TOTAL CASES	1575

TABLA 3: TP DE CADA CLASE

Finalmente, he realizado la representación de la gráfica PR para poder observar dichos resultados de forma más visual.

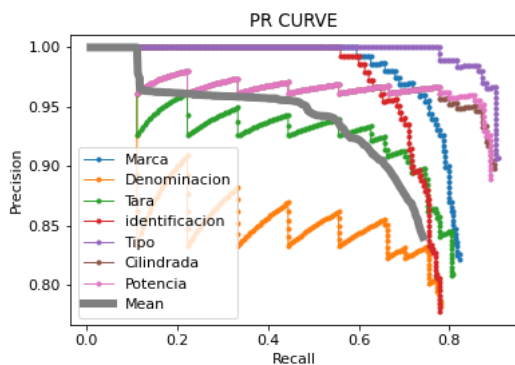


Fig. 12: Curva PR

7.2. Reconocimiento de texto

Como mencione anteriormente, he entrenado 6 modelos de OCR diferentes en función de si están o no preprocesadas las imágenes, si se ha entrenado desde cero o con fine tuning, o de si emplea las imágenes generadas con la librería PIL. Estos modelos han sido evaluados a partir de la precisión obtenida del CER y los resultados se pueden visualizar en la Tabla 4.

Modelo Tesseract	Precision (%)
Desde cero sin preprocesar	9.53
Desde cero preprocesado	21.69
Fine tuning (FT) sin preprocesar	88.71
Fine tuning (FT) preprocesado	92.43
FT sin preprocesar con imagenes PIL	87.04
FT preprocesado con imagenes PIL	90.42

TABLA 4: COMPARACIÓN TESSERACT

Como podemos observar en la Tabla, los dos modelos entrenados desde cero proporcionan muy malos resultados, lo cuales no superan el 30 % de precisión. Esto es debido a que para entrenar un modelo desde cero de forma eficiente se requieren grandes cantidades de datos o que el problema sea muy restringido. Por otro lado, en el caso de los entrenados a partir del ajuste fino para el impacto, vemos que estos ya proporcionan precisiones muy altas superiores al 85 %, y que de entre todos ellos el mejor es el que utiliza las imágenes preprocesadas pero sin las extensiones del dataset. El hecho de que al extender el dataset los resultados sean un poco peores es debido a que las imágenes generadas no son del todo parecidas a las normales y por consiguiente hacen que se ajuste la red de tal manera que genera ciertos fallos.

7.3. Pipeline completo

Para acabar se ha analizado el pipeline completo, es decir, desde que se detectan y preprocesan los bounding boxes de la imagen, hasta que se elimina la parte redundante y se reconoce el contenido de texto. Para analizar dicho pipeline se han tenido en cuenta los 3 métodos que se encargan de identificar la parte relevante de la imagen a partir de la técnica de dilatación (Método 1), el diseño de reglas utilizando Tesseract (Método 2) y la combinación de estos dos

métodos (Método 3), generando los resultados de la Tabla 5.

Método	Precision (%)
1	64.52
2	69.68
3	76.21

TABLA 5: PIPELINE COMPLETO

Si visualizamos los resultados de la Tabla podemos observar que estos no son tan buenos como en el análisis del Tesseract, ya que en ese caso las imágenes de Test eran imágenes buenas en las que solo se tenía la parte relevante, mientras que aquí en muchos casos el rendimiento se ve reducido por el hecho de que no se consigue distinguir correctamente las dos partes de la imagen, además de que en muchas situaciones las imágenes son de mala calidad.

Respecto a los resultados vemos que la peor precisión la tiene el Método 1, esto es debido en su mayoría a que en muchos casos la palabra relevante está muy próxima a la de no interés y por consiguiente al realizar la dilatación se junta y elimina por equivocación tal y como se puede ver en el apartado Problema Método Dilatación de anexos.

Por otro lado, vemos que el Método 2 genera mejores resultados que el anterior, pero este está muy limitado a que las imágenes cumplan dichas reglas.

Finalmente, observamos que el Método 3 es el que mejores resultados proporciona, ya que utiliza primero el Método 1 que es muy eficiente cuando las palabras no están juntas, y en los casos que ocurre esto utiliza el Método 2.

8 CONCLUSIONES

Como bien explicamos al principio del proyecto, este tiene dos grandes objetivos que son la detección de campos de texto y el preprocesado, reconocimiento y almacenamiento de los datos.

Respecto al primer objetivo se ha conseguido entrenar el modelo YOLOV5 y una vez acabado el minimum viable producto también se ha entrenado la red neuronal Faster R-CNN.

En el caso del primer modelo, como hemos analizado en los resultados, este proporciona un mAP muy elevado (78.44 %), teniendo en cuenta que los thresholds son muy altos y que a pesar de que el IOU no este dentro de este threshold no significa que el bounding box sea erróneo, ya que en ocasiones suele ser porque la longitud de este no es la exacta, pero sí que engloba todo el contenido de texto. Además, si lo comparamos con el caso de detección de texto de matrículas, podemos observar que hemos alcanzado unos porcentajes bastante similares.

Por otro lado, el segundo modelo no proporciona una precisión tan buena y como hemos mencionado anteriormente, esto se deba a la falta de datos, puesto que el número de epochs es el adecuado.

En el caso del segundo objetivo, si avaluamos de forma independiente el preprocesado y el OCR podemos ver que el preprocesado da buenos resultados, pero hay ocasiones en que la imagen está tan deteriorada o el texto se ve tan poco que este no funciona correctamente. Respecto al OCR

hemos visto que en el mejor de los casos nos da una precisión del 92.43 % lo que equivaldría a un CER de menos del 10 % indicando de esta manera que nuestro OCR es bueno reconociendo texto.

Finalmente, si evaluamos el pipeline completo vemos que los porcentajes respecto al OCR descienden, esto se debe en parte a que alguna detección del campo de texto no sea del todo correcta y sobre todo al factor preprocesado y detección de campos de texto relevantes, ya que la mayoría de veces en las que falla es porque la imagen está muy deteriorada o el texto es poco visible y al preprocesar se acaba de eliminar, o porque al utilizar el método de dilatación se unen palabras de interés con redundantes y se eliminan, haciendo de esta manera que el OCR no pueda reconocer contenido en la imagen.

8.1. Futuras Contribuciones

Como hemos podido observar en este proyecto, el principal problema es la detección del campo de texto relevante, y por consiguiente, sería interesante entrenar el modelo de Deep Learning para que sea capaz de detectar directamente solo este campo en vez del conjunto completo.

Otra aportación interesante sería generar un dataset más amplio para poder entrenar de forma más óptima el modelo Faster R-CNN y así poder comparar realmente los dos modelos.

Finalmente, una aportación interesante a tener en cuenta sería realizar un postprocesado del texto reconocido por el OCR, en que a partir de una base de datos con todos los modelos de coches, su correspondiente cilindrada, tipo de motor ... Se realiza una comparación para ver cuál es el texto más prometedor mediante la distancia de Levenshtein y se substituye por este, asegurando de esta manera que la palabra generada sea una que realmente exista.

9 AGRADECIMIENTOS

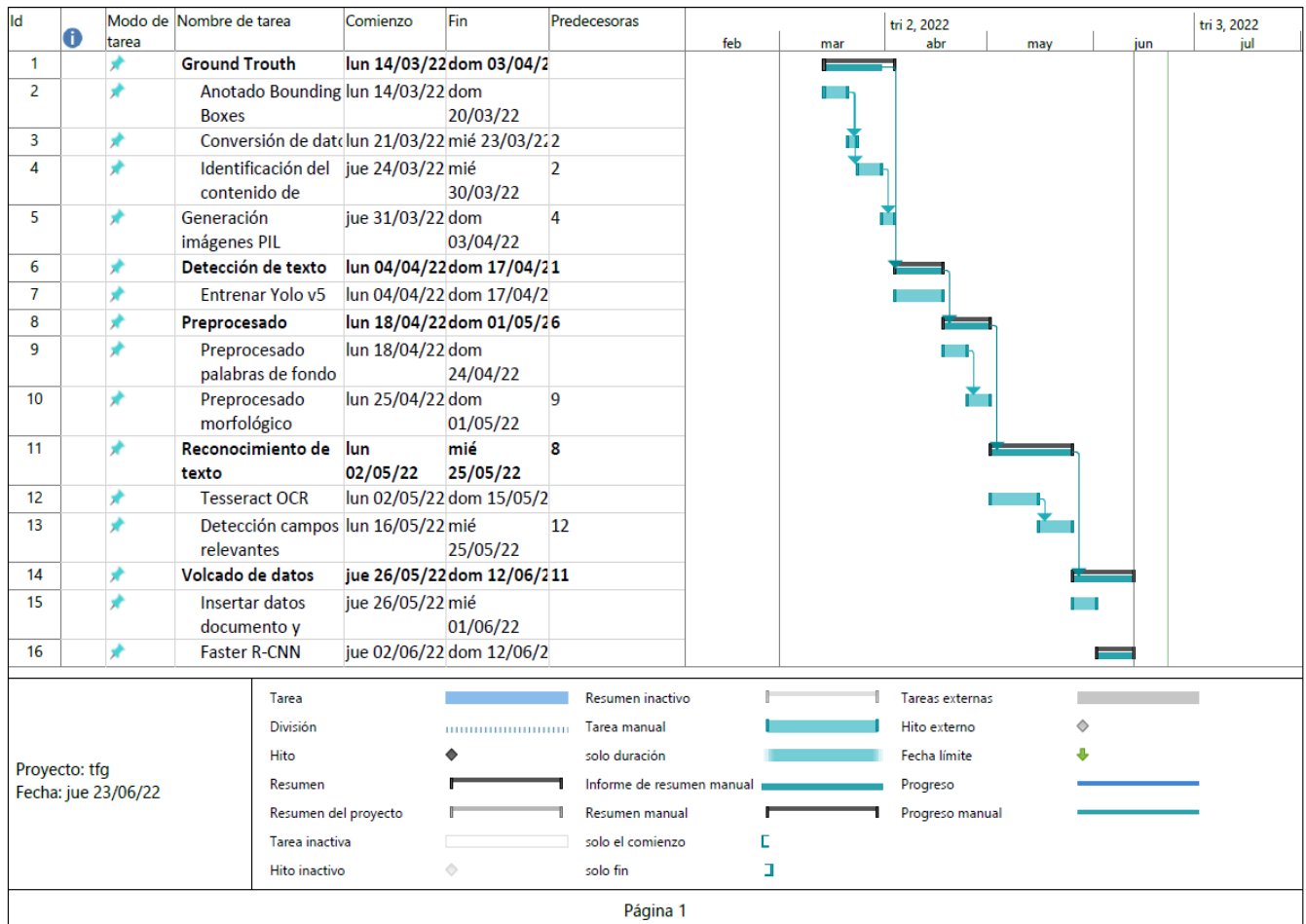
Agradecer al tutor del proyecto, Coen Antens, por guiarme durante todo el transcurso del TFG, aportándome conocimientos de diversos conceptos de visión por computador e incentivándome a realizar el mejor proyecto posible.

REFERENCIAS

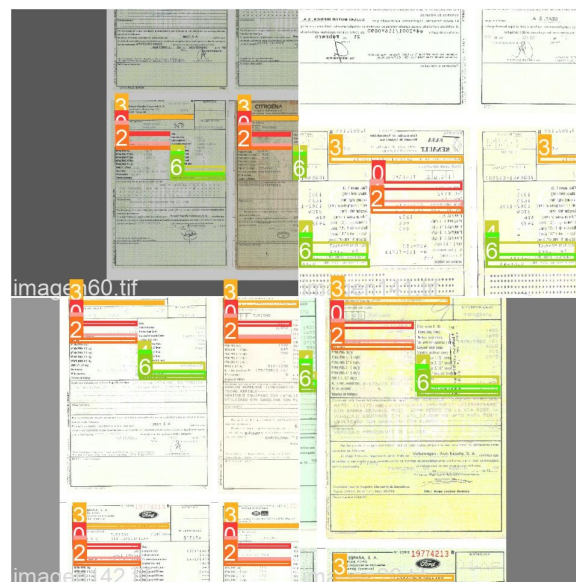
- [1] Gonçalves, L.(2021, 28 noviembre).Metodología Ágil. <https://adaptmethodology.com/es/que-es-la-metodologia-agil/>
- [2] Babu, S. C. (2022, 8 febrero). Digitize Receipts with Receipt OCR — Automated Receipt OCR. AI Machine Learning Blog. <https://nanonets.com/blog/receipt-ocr/preprocessing>
- [3] Tian,F.(2021, 5 enero).Research on Fast Text Recognition Method for Financial Ticket Image.<https://arxiv.org/abs/2101.01310>
- [4] Agarwal, R. (2021, 23 diciembre). Deep Learning Based OCR for Text in the Wild. AI and Machine Learning Blog. <https://nanonets.com/blog/deep-learning-ocr/>
- [5] C.A. Automatic vehicle number plate detection <https://medium.com/@ashok.1055/automatic-number-plate-detection-using-yolov5-and-extraction-of-text-using-ocr-2868668aa479>
- [6] Kathuria, A.(2021b, mayo 10). How to Train YOLO v5 on a Custom Dataset. Paperspace Blog. <https://blog.paperspace.com/train-yolov5-custom-data/>
- [7] faster rcnn: Ren, S. (s. f.). Faster R-CNN. <https://arxiv.org/abs/1506.01497>
- [8] EAST: An Efficient and Accurate Scene Text Detector. (2017, 11 abril). <https://paperswithcode.com/paper/east-an-efficient-and-accurate-scene-text>
- [9] Extracción inteligente de texto y datos con OCR - Amazon Textract - Amazon Web Services. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/textract/>
- [10] keras-ocr — keras ocr documentation. (s. f.). KERAS. <https://keras-ocr.readthedocs.io/en/latest/>
- [11] Computer Vision. (s. f.). Microsoft Azure. <https://azure.microsoft.com/es-es/services/cognitive-services/computer-vision/overview>
- [12] Build software better, together. (s. f.). GitHub. <https://github.com/tzutalin/labelIm>
- [13] Preprocessing OCR-Towards Data Science. Medium. <https://towardsdatascience.com/pre-processing-in-ocr-fc231c6035a7>
- [14] Levenshtein Distance. (s. f.). <https://people.cs.pitt.edu>
- [15] fillpdf.PyPI. <https://pypi.org/project/fillpdf/>
- [16] Rath, S. R. (2021, 28 noviembre). A Simple Pipeline to Train PyTorch Faster RCNN Object Detection Model. DebuggerCafe. <https://debuggercafe.com/a-simple-pipeline-to-train-pytorch-faster-rcnn-object-detection-model/>
- [17] Color Segmentation. (s. f.). Image Segmentation Using Color Spaces in OpenCV + Python. <https://realpython.com/python-opencv-color-spaces/>
- [18] Pre-Processing in OCR - Towards Data Science. Medium. <https://towardsdatascience.com/pre-processing-in-ocr-fc231c6035a7>
- [19] ImageDraw Module. (s. f.). PIL Imagedraw. <https://pillow.readthedocs.io/en/stable/reference>
- [20] fillpdf.(2022, 1 febrero).PyPI. <https://pypi.org/project/fillpdf/>
- [21] CSV File Reading and Writing—Python 3.10.4 documentation. <https://docs.python.org/3/library/csv.html>
- [22] Rosebrock, A. (2021, 7 septiembre). Tesseract Page Segmentation Modes (PSMs). <https://pyimagesearch.com/2021/11/15/tesseract-page-segmentation-modes-psms-explained-how-to-improve-your-ocr-accuracy/>

10 ANEXOS

10.1. Gantt



10.2. Lote De Entrenamiento



10.3. PSM

Page segmentation modes:

- 0 Orientation and script detection (OSD) only.
- 1 Automatic page segmentation with OSD.
- 2 Automatic page segmentation, but no OSD, or OCR.
- 3 Fully automatic page segmentation, but no OSD. (Default)
- 4 Assume a single column of text of variable sizes.
- 5 Assume a single uniform block of vertically aligned text.
- 6 Assume a single uniform block of text.
- 7 Treat the image as a single text line.
- 8 Treat the image as a single word.
- 9 Treat the image as a single word in a circle.
- 10 Treat the image as a single character.
- 11 Sparse text. Find as much text as possible in no particular order.
- 12 Sparse text with OSD.
- 13 Raw line. Treat the image as a single text line, bypassing hacks that are Tesseract-specific.

10.4. Tabla Datos Vehículos

	A	B	C	D	E	F	G
1	Identificació	Marca	Denominació	Tara (KG)	Motor_tipo	Cilindrada	Potencia
2	VF7N1RHYF36670746	SEAT	IBIZA	1145	(D) 1Y	4/1597	8,77/37
3	VF71CORG3408SH734	FORD	ESCORT 5 PUERTAS	1150	G/L1H	4/1340	11.63/66
4	WZSD6763FS763F673	LANCIA	LANCIA Y	860	840A3.000	4/1.242	10,00/44
5	WZ9384HJDHUR498532	SEAT	IBIZA	1145	(D) 1Y	4/1597	8,77/37

10.5. Comparativa Detección YOLOV5 y Faster R-CNN

EMPRESA: FORD ESPAÑA, S. A.
FACTORY FORD
Polígono Industrial de Almussafes
ALMUSAFES (Valencia)

N.º SERIE: 18719621 B

MATRICULA: 10002

Número de identificación: WFOBXXWPRBWU56605

Clasificación del vehículo: TURISMO FORKV35S SIN ESPECIFICAR

N.º CERTIFICADO: 039345

Marca: FORD

Tipo: RBT

Variantes: J401 (MAN.)

Denominación comercial: KA 3 PUERTAS

Tara (kg): 908

PTMA (kg): 1265

PTMA/PMA 1.ª E (kg): 710

PTMA/PMA 2.ª E (kg): 600

PTMA/PMA 3.ª E (kg): 600

PTMA/PMA 4.ª E (kg): 600

PMA S.F. G/F (kg): 600

N.º y dim. neumáticos: 4/165/65R13 72R

N.º de asientos: 4

Volumen de bagaje: 10.28/44

Opciones incluidas en la homologación de tipo: NEUMATICOS ALTERNATIVOS: 155/70R13 72R*VERSIONES: 4A11,4C11,4E11

Observaciones:

Por las piezas de origen extranjero incorporadas a este vehículo se han abonado los correspondientes derechos de Aduanas.

El abajo firmante, legalmente autorizado por FORD ESPAÑA, S. A., certifica que el vehículo carrozado cuyas características se reseñan es completamente conforme con el tipo homologado con la contraseña: e9*95/54*0019*03, así como con las opciones arriba indicadas.

Almussafes, 01 JUL 1998

Firma del Fabricante Nacional/Importador:

Sociedad inscrita en el Reg. Merc. de Valencia, nº 9 de Octubre de 1972, tomo 110, folio 217, Libro de Societades sect. 3.ª, tomo 2.575, Cód. 16, Folio 44060361

Reformas autorizadas:

EMPRESA: FORD ESPAÑA, S. A.
FACTORY FORD
Polígono Industrial de Almussafes
ALMUSAFES (Valencia)

N.º SERIE: 18719621 B

MATRICULA: 10002

Número de identificación: WFOBXXWPRBWU56605

Clasificación del vehículo: TURISMO FORKV35S SIN ESPECIFICAR

N.º CERTIFICADO: 039345

Marca: FORD

Tipo: RBT

Variantes: J401 (MAN.)

Denominación comercial: KA 3 PUERTAS

Tara (kg): 908

PTMA (kg): 1265

PTMA/PMA 1.ª E (kg): 710

PTMA/PMA 2.ª E (kg): 600

PTMA/PMA 3.ª E (kg): 600

PTMA/PMA 4.ª E (kg): 600

PMA S.F. G/F (kg): 600

N.º y dim. neumáticos: 4/165/65R13 72R

N.º de asientos: 4

Volumen de bagaje: 10.28/44

Opciones incluidas en la homologación de tipo: NEUMATICOS ALTERNATIVOS: 155/70R13 72R*VERSIONES: 4A11,4C11,4E11

Observaciones:

Por las piezas de origen extranjero incorporadas a este vehículo se han abonado los correspondientes derechos de Aduanas.

El abajo firmante, legalmente autorizado por FORD ESPAÑA, S. A., certifica que el vehículo carrozado cuyas características se reseñan es completamente conforme con el tipo homologado con la contraseña: e9*95/54*0019*03, así como con las opciones arriba indicadas.

Almussafes, 01 JUL 1998

Firma del Fabricante Nacional/Importador:

Sociedad inscrita en el Reg. Merc. de Valencia, nº 9 de Octubre de 1972, tomo 110, folio 217, Libro de Societades sect. 3.ª, tomo 2.575, Cód. 16, Folio 44060361

Reformas autorizadas:

(a) YOLOV5

(b) Faster R-CNN

10.6. Comparativa Imagen Preprocesada

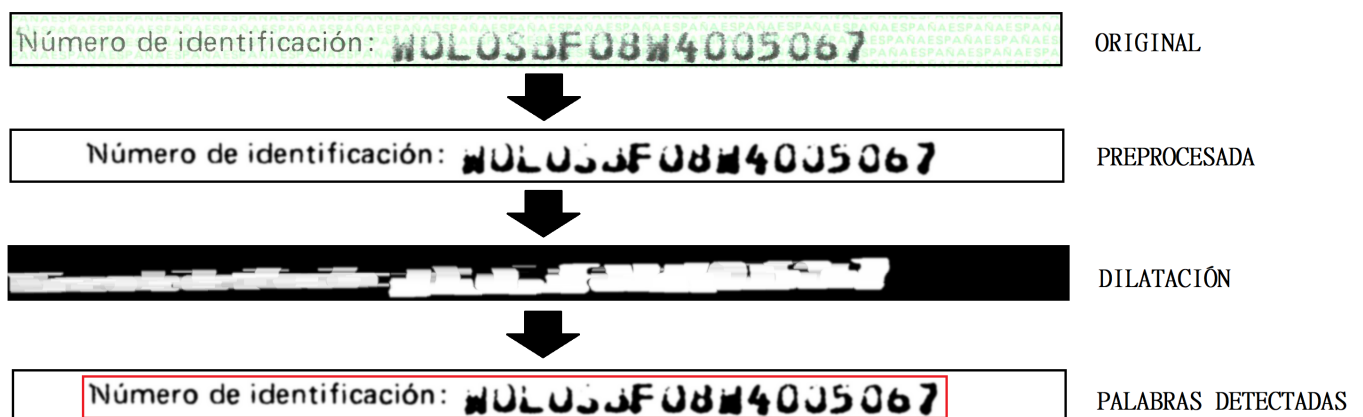
Nº SERIE 23152918 B		MATRÍCULA	
HONDA MOTOR CO., LTD			
Número de identificación: 3HMEK3760Y0204358			
Clasificación del vehículo: 786 R		Nº CERTIFICADO 77192117	
Marca: HONDA Tipo: E43 Variante: 1138/SK7,04526 Denominación comercial: Civic Sp. 1.5i 13 Tara (kg): 1060 MTMA/MMA (kg): 1500 MTMA/MMA 1º E (kg): 850 MTMA/MMA 2º E (kg): 780 MTMA/MMA 3º E (kg): MTMA/MMA 4º E (kg): MMR S/F, c/F (kg): 500, 1000 Neumáticos: 8: 195/60 R14 82H + 771 Nº de asientos: 5 Volumen de bodega:	Clase: Altura total (mm): 1375 Anchura total (mm): 1695 Via anterior/posterior (mm): 1478/1460 Longitud total (mm): 4170 Voladizo posterior (mm): 720 Distancia eje 1º/2º (mm): 2420 Distancia eje 2º/3º (mm): Distancia eje 3º/4º (mm): Distancia 5ª rueda/últ. (mm): Motor: Marca: HONDA Tipo: Nº Cilindros/Cilindrada (cm³): 4: 01520 Potencia fiscal/real (C.V./KW): 44/32.04	Opciones incluidas en la homologación de tipo: NEUMÁTICOS: 175/65 R14 82H VEHICULO EQUIPADO CON CATALIZADOR NO APTO GASOLINA CON PLOPO Observaciones: (*) 1 NEUMATICO DE USO TEMPORAL 175/70 R14 82H vehículo importado. con D.R.R. nº 0011-E-001707 de 06/04/2000	
Por las piezas de origen extranjero incorporadas a este vehículo se han satisfecho los correspondientes derechos de Aduanas. El abajo firmante, legalmente autorizado por HONDA MOTOR EUROPE (SAATCHI), S.A., certifica que el vehículo carrozado cuyas características se reseñan es completamente conforme con el tipo homologado con la contraseña 25192/01/0007, así como con las opciones arriba incluidas. F. Real de Llobregat, s.l. de Firma, Junio de 2000 Sociedad inscrita en Francia con la identificación: D. Roger Solergibert Mora Reformas autorizadas:			

(a) Imagen Sin Preprocesar

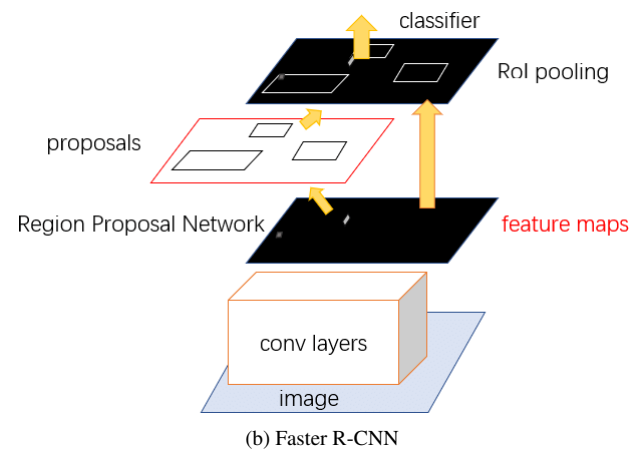
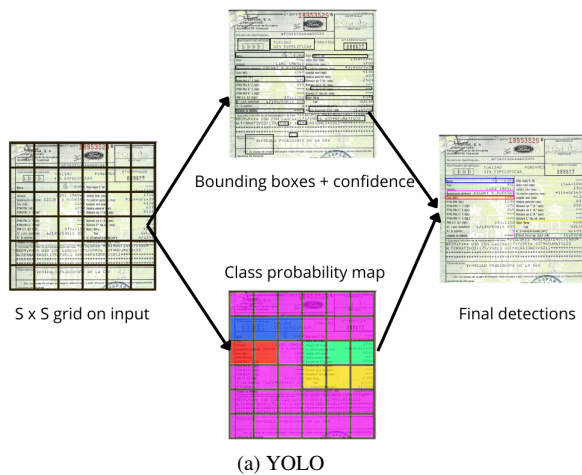
Nº SERIE 23152918 B		MATRÍCULA	
HONDA MOTOR CO., LTD			
Número de identificación:			
Clasificación del vehículo:		Nº CERTIFICADO	
Marca: Tipo: Variante: Denominación comercial: Tara (kg): MTMA/MMA (Kg): MTMA/MMA 1º E (kg): MTMA/MMA 2º E (kg): MTMA/MMA 3º E (kg): MTMA/MMA 4º E (kg): MMR S/F, c/F (kg): Neumáticos: Nº de asientos: Volumen de bodega:	Clase: Altura total (mm): Anchura total (mm): Via anterior/posterior (mm): Longitud total (mm): Voladizo posterior (mm): Distancia eje 1º/2º (mm): Distancia eje 2º/3º (mm): Distancia eje 3º/4º (mm): Distancia 5ª rueda/últ. (mm): Motor: Marca: Tipo: Nº Cilindros/Cilindrada (cm³): Potencia fiscal/real (C.V./KW):	Opciones incluidas en la homologación de tipo: Observaciones: Por las piezas de origen extranjero incorporadas a este vehículo se han satisfecho los correspondientes derechos de Aduanas. El abajo firmante, legalmente autorizado por , certifica que el vehículo carrozado cuyas características se reseñan es completamente conforme con el tipo homologado con la contraseña , así como con las opciones arriba incluidas. de Firma, de D. Roger Solergibert Mora Reformas autorizadas:	

(b) Imagen Preprocesada

10.7. Problema Método Dilatación



10.8. Comparativa Arquitectura YOLO y Faster R-CNN



10.9. Certificado Destrucción

COMUNITAT AUTÒNOMA DE CATALUNYA
DEPARTAMENT DE MEDI AMBIENT I HABITATGE

Generalitat de Catalunya
Departament de Medi Ambient i Habitatge
Agència de Residus de Catalunya

Dr. Roux, 80
08017 Barcelona Tel. 93 567 33 00

Còpia NP 1,2,3 i 4

Certificat de destrucció del vehicle al final de la seva vida útil Certificado de destrucción del vehículo al final de su vida útil			
1. Dades del vehicle per descontaminar Datos del vehículo a descontaminar			
1.1 Marca	1.2 Model	1.3 Tipus	
1.4 Matricula	1.5 Data de matriculació	1.6 País de matriculació	1.7.1 Gasolina 1.7.2 Gas-oil 1.7.3 Altre
1.8 Número d'identificació (Bastidor) Número de identificación (Bastidor)			
1.9 Estat del vehicle	1.9.1 Sencer	1.9.2 Parcialment desmuntat	1.9.3 Sinistrat
1.10 Altres dades d'interès: Otros datos de interés			
2. Dades del titular del vehicle lliurat Datos del titular del vehículo entregado			
2.1 Nom i cognoms del titular o denominació de la Raó Social Nombre y apellidos del titular o denominación de la Razón Social		Declaro que he lliurat el vehicle indicat en aquesta instal·lació de Recepció / Declaro que he entregado el vehículo reseñado en esta Instalación de Recepción	
2.2 CIF / DNI / NIE / Altre	2.3 Nacionalitat	2.9 Nom i cognoms: Nombre y apellidos:	
2.4 Direcció	2.6 Codi postal	2.10 CIF / DNI / NIE / Altre	Signatura del titular o representant Firma del titular o representante 2.12 Data
2.5 Municipi	2.7 País	2.11 Concepte de representació	
2.8 País	2.12 Data		
3. Dades de la instal·lació de recepció de vehicles Datos de la instalación de recepción de vehículos			
3.1 Nom o Raó social		Instal·lació de recepció (Segell i signatura)	
3.2 CIF	3.3 Adreça	Instalación de recepción (sello y firma)	
3.4 Municipi	3.5 Codi postal	3.8 Data	
3.6 Província	3.7 País		
4. Dades del centre autoritzat de tractament de vehicles (A complimentar pel propi centre) Datos del centro autorizado de tratamiento de vehículos (A cumplimentar por el propio centro)			
4.1 Nom o Raó social		4.2 CIF	
4.3 Codi d'autorització de gestor de residus perillosos		4.4 R.E.I.C.	
4.5 Direcció centre gestor		4.6 Municipi	
4.7 Codi postal	4.8 Província	4.9 Telefon	4.10 Fax
4.11 Persona responsable		4.12 E-mail	
CERTIFICAT PER LA PRESENT: haver verificat sobre el mateix vehicle que el seu número de sèrie, reproduït en el xasis, així com la seva placa de matriculació, corresponen al vehicle lliurat. Procedint a complir amb l'apartat de l'art. 8.1 del R.D. 1383/2002, realitzant les operacions de descontaminació, complint així amb lo establert en l'Annex III de manera que s'afavoreixi la reutilització i el reciclatge, per aquest ordre.		La persona que signa es fa responsable de l'acceptació del residu descrit per al seu tractament apropiat, d'acord amb el Real Decret 1383/2002, de 20 de desembre, sobre gestió de vehicles al final de la seva vida útil.	
CERTIFICADO POR LA PRESENTE: haber verificado sobre el mismo vehículo que su número de serie, reproducido en el chasis, así como su placa de matriculación, corresponden al vehículo entregado. Procediendo a cumplir con el apartado del art. 8.1 del R.D. 1383/2002, realizando las operaciones de descontaminación, cumpliendo así con lo establecido en el Anexo III de manera que se favorezca la reutilización y el reciclaje, por este orden.		La persona que firma se hace responsable de la aceptación del residuo descrito para su tratamiento apropiado, de acuerdo con el Real Decreto 1383/2002, de 20 de diciembre, sobre gestión de vehículos al final de su vida útil.	
DECLARO DECLARO			
5.1 <input type="checkbox"/> Que s'adjunta el corresponent document de matriculació	Gestor (Segell i signatura)		
5.2 <input type="checkbox"/> La inexistència del document de matriculació	Gestor (Sello y firma)		
5.3 Data		5.3 Data	

Ejemplar per a: Agència de Residus de Catalunya / Dirección General de Residuos / Centro Gestor Autorizado / Inst. Recepció o Tracte / Ejemplar para: Agencia de Residuos de Catalunya / Dirección General de Residuos / Centro Gestor Autorizado / Inst. Recepción o Tratamiento

ver. 1.3