

Grupo 5			
Alumno/a:	Sara Rodriguez Galán	NIA:	100496119
Alumno/a:	Lucía Merinero López	NIA:	100496052
Alumno/a:	Jhonatan Barcos Gambaro	NIA:	100548615

1. Descripción

Describiremos las implementaciones y los cambios que han llevado a cabo cada miembro de este proyecto, donde discutiremos su finalidad y el resultado que hemos logrado tras su implementación.

1.1. Adaptación al nuevo problema(Jhonatan)

Nuestro principal reto en este proyecto ha sido la adaptación del notebook de referencia *detection_VOC4_tutorial_colab.ipynb*, en el cual se trabaja la detección de diferentes objetos (bottle, chair, dining_table y sofa). Para llevarla a cabo en un nuevo Dataset, nos hemos basado en el repositorio de github [OIDv4_ToolKit](#) proveniente de [Open Images Dataset V6](#). Así pues, hemos descargado 500 imágenes para el conjunto train y 100 imágenes para el conjunto test de las siguientes clases (Car, Bus, Truck y Motorcycle), abstrayendo así el proyecto base en un proyecto de Detección de vehículos, pudiendo aplicarse posteriormente para modelos de seguridad vial, monitorización de tráfico o control de accesos en la realidad.

Para poder llevar a cabo el proyecto con estos nuevos datos hemos tenido que modificar diferentes partes de la implementación. Por una parte, estas nuevas clases no venían con máscaras sino con anotaciones en formato de texto de la forma Class x_min, y_min, x_max, y_max, el formato estándar del método YOLO y que corresponden a las coordenadas normalizadas respecto al tamaño de la imagen.

Debido a este cambio en la interpretación de los datos, se ha modificado la clase principal myVOCDataset creando una nueva clase MyOIDDataset donde Dataset personalizado: Gestionamos las rutas dinámicamente, asociamos las imágenes con sus anotaciones y verificamos que los pares .jpg + .txt estén sincronizados.

Finalmente, reorganizamos el conjunto de datos descargado en las carpetas train/ y test/, conteniendo cada una imágenes y sus correspondientes .txt. Generamos las listas de rutas para ambos subconjuntos y las cargamos en DataLoaders independientes.

1.2. Mejoras propuestas

Efecto de la Modificación de Hiperparámetros en Faster R-CNN (Sara)

- Learning-rate: con valores altos como >0.01 , el modelo no convergía, y por el otro lado, valores muy pequeños <0.0001 el modelo demoraba mucho tiempo en aprender y se quedaba atascado en soluciones que no eran óptimas
- Batch size: probamos diferentes valores para ver qué ocurría, valores pequeños el entrenamiento era inestable, pero al colocar valores altos la GPU requería mucha memoria.
- Número de epochs: tener más epochs significa más aprendizaje, pero también colocar epochs en exceso nos lleva a que se sobreajuste el modelo, y deje de aprender, además de consumir tiempo innecesario. Los valores más adecuados encontrados son entre 5 y 7 epochs, ya que al colocar más de 10 el modelo empeoraba.

Uso de data augmentation (Sara)

Se aplicó con el objetivo de mejorar la capacidad del modelo adaptándose a las diferentes condiciones de las imágenes. Dicha técnica consiste en variar ligeramente la imagen para simular variaciones que pueden darse en casos reales. Probamos las siguientes transformaciones: Random Rotation, Random Crop, Horizontal Flip, además de aumentar el brillo y el contraste. Como resultado donde más mejora en los resultados fue al aplicar Horizontal Flip, y cambios de ajuste en el brillo y contraste.

Uso de anchors personalizados (Lucía)

Los *anchors* son rectángulos de referencia que un modelo como Faster R-CNN coloca sistemáticamente para detección de objetos. Cada *anchor* tiene un tamaño y relación de aspecto fijas que dan el punto de partida para que aprenda a adaptarlos a los objetos reales. Si extraemos los tamaños reales de las *boxes* de nuestra base de datos y usamos clustering para obtener anchors adaptados mejoraremos la eficiencia de nuestro detector y su precisión.

- Clustering: El objetivo es dividir un conjunto de puntos(ancho x alto) en K grupos, de manera que los puntos de cada grupo sean lo más parecido posible entre ellos. Utilizando el algoritmo Kmeans obtenemos K centroides que usaremos para calcular nuestras *anchors* y ratios personalizados.

$$\text{anchor_size} : s_i = \sqrt{w_i \cdot h_i}$$

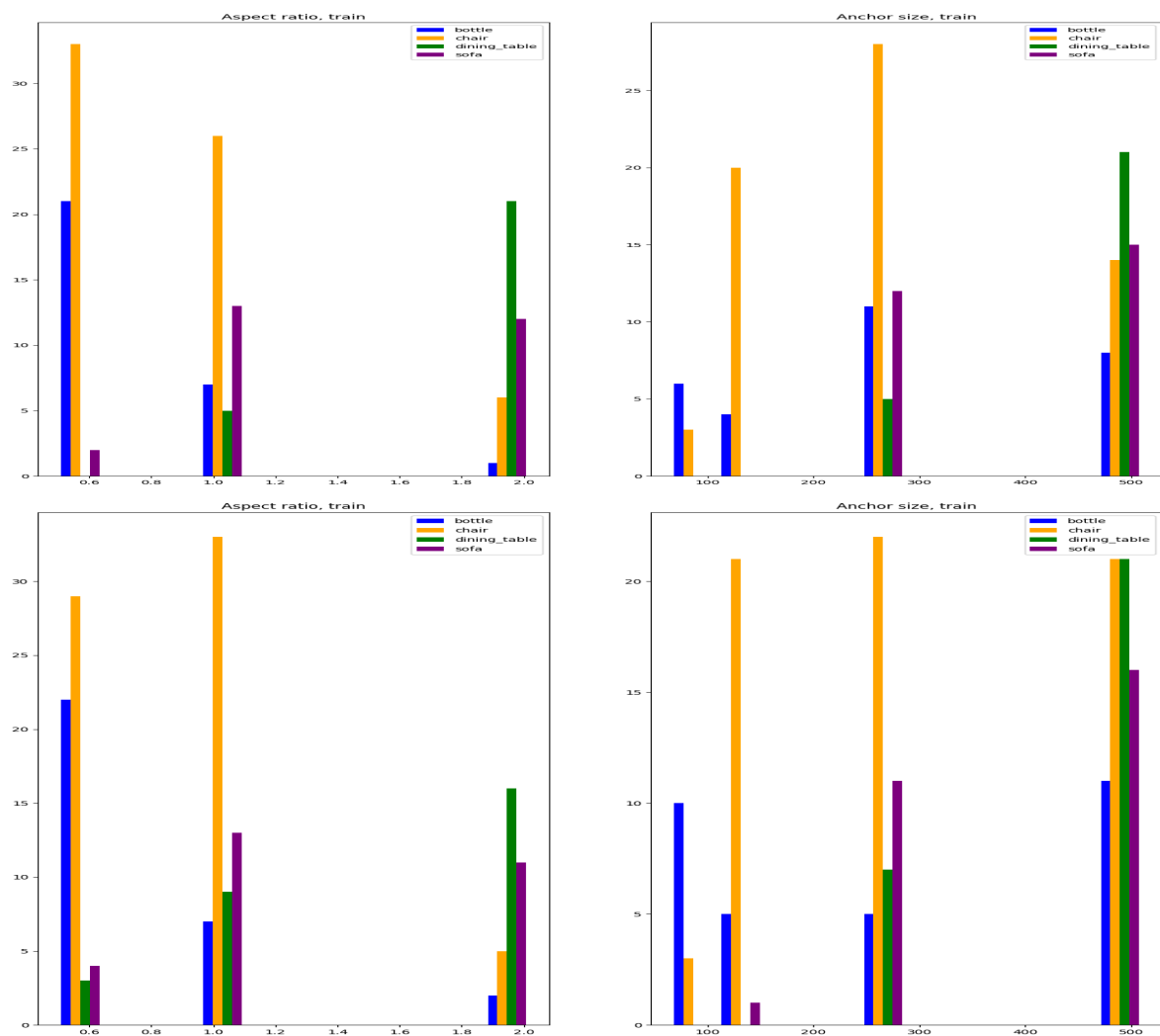
$$\text{aspect_ratio} : r_i = \frac{w_i}{h_i}$$

De esta manera, en lugar de dejar que el modelo haga sus anchors automáticamente les pasaremos estas métricas para mejorar resultados.

2. Material Extra

2.1. Evaluación de resultados

- Comparación de anchors iniciales vs personalizadas (Base de datos inicial)



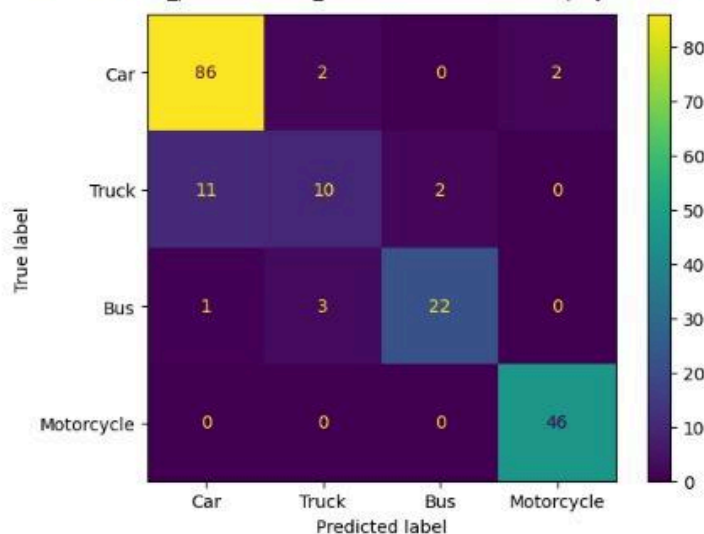
Observamos que al adaptar las anchors a la base de datos los tamaños se acercan más a los reales.

- Matriz de confusión de la base de datos de vehículos después de entrenarla con las transformaciones.

Vemos que coches y camiones se confunden porque entendemos que según como se miren, desde que perspectiva, pueden verse muy parecidos.

```
Evaluating...
Evaluating...
Evaluation complete in 1m 36s
Objectness-RPN. F1: 0.536673. Precision: 0.413870. Recall: 0.763096
Class: Car. Recall: 90/119
Class: Truck. Recall: 23/25
Class: Bus. Recall: 26/41
Class: Motorcycle. Recall: 46/62

Global classification: F1: 0.831337. Precision: 0.854804. Recall: 0.809123
Class: Car. F1: 0.914894. Precision: 0.877551. Recall: 0.955556
Class: Truck. F1: 0.526316. Precision: 0.666667. Recall: 0.434783
Class: Bus. F1: 0.880000. Precision: 0.916667. Recall: 0.846154
Class: Motorcycle. F1: 0.978723. Precision: 0.958333. Recall: 1.000000
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7af405fcc0d0>
```



3. Declaración de uso de IA Generativa.

Para llevar a cabo nuestro proyecto hemos utilizado herramientas de IA generativa como ChatGPT de OpenAI y Copilot en combinación con técnicas de Prompt Engineering para maximizar la utilidad y corrección de estas herramientas. En esta sección exponemos el uso de la IA generativa en este proyecto con el consecuente reflexionado sobre el uso de estas nuevas herramientas en la actualidad.

3.1. Aspectos de la práctica en los que hemos utilizado IA

A continuación se detallan los principales puntos en los que hemos utilizado esta herramienta para nuestro proyecto.

- Definición y comprensión de los objetivos de nuestro proyecto: En primer lugar, hemos utilizado inteligencia artificial generativa para garantizar una comprensión correcta y completa de los objetivos de la práctica a nivel técnico y matemático. Asimismo, hemos podido establecer el enunciado mediante ejemplos prácticos para poder desarrollar la práctica con total garantía a nivel de comprensión.
- Resolución de dudas sobre los problemas surgidos durante el desarrollo de la práctica: A medida que resolvemos el enunciado propuesto, surgieron diferentes problemas a nivel de código, como errores de ejecución, que solucionamos fácilmente mediante la depuración en VSCode con la ayuda de inteligencia artificial generativa, que nos permitió detectar dónde se encontraba el error en nuestro código para poder corregirlo.
- Redacción de código complejo guiada por los notebooks proporcionados y técnicas de prompt engineering para adaptar nuestro código base a nuevas modalidades, con el fin de simplificar nuestra implementación inicial y mejorar los resultados en función de los diferentes métodos de evaluación utilizados.
- Revisión y validación del código de nuestro modelo: Una vez finalizado nuestro código, realizamos un análisis exhaustivo para verificar que fuera correcto.

3.2. Hasta qué punto ha facilitado la realización de la práctica

En general, el acceso a estas nuevas herramientas de forma responsable y coherente ha sido una gran ventaja a la hora de realizar la práctica, especialmente en términos de tiempo, ya que ha facilitado distintos aspectos como:

- Comprensión rápida de conceptos complejos, para los cuales, en el pasado, tardamos mucho más en encontrar la información adecuada en internet, interpretarla y adaptarla a nuestras necesidades.

- Resolución de errores, que, al igual que en el punto anterior, anteriormente nos podía llevar mucho tiempo detectar y corregir correctamente.
- Obtención de explicaciones sobre métodos de librerías de Python que no conocemos en profundidad, de forma mucho más rápida y eficiente que consultando la guía oficial de la librería.
- Verificación del contenido de la memoria para asegurarnos de no dejar fuera ningún aspecto relevante para la comprensión total del trabajo desarrollado en esta práctica.

3.3. Cómo se ha verificado y adaptado la información obtenida

Como se menciona en el apartado anterior, consideramos el uso de estas nuevas herramientas como una gran ventaja frente al pasado, siempre que se utilicen de forma responsable y consciente. Por ello, hemos empleado un método de validación cruzada para asegurarnos de que la información proporcionada por la inteligencia artificial generativa fuera correcta y óptima. Para ello, seguimos la siguiente metodología:

- Comprobación de la validez de las respuestas a través de Internet y de los recursos proporcionados en los notebooks de referencia.
- Contraste de las explicaciones teóricas con fuentes en Internet y los notebooks de referencia, especialmente en la parte matemática, ya que las IAs generativas aún no están perfeccionadas en este ámbito.
- Uso de la IA como herramienta de apoyo y no como sustituto del pensamiento crítico para resolver los distintos problemas planteados.