



Universidad Politécnica
de Madrid

**Escuela Técnica Superior de
Ingenieros Informáticos**



Grado en Ciencia de Datos e Inteligencia Artificial

Trabajo Fin de Grado

**Sistema de detección y clasificación
automática de anomalías en radiografías
de tórax utilizando Deep Learning**

Autor: Mario Serrano Sánchez
Tutor: Ángel Mario García Pedrero

Madrid, 3 de junio de 2024

Este Trabajo Fin de Grado se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Grado

Grado en Ciencia de Datos e Inteligencia Artificial

Título: Sistema de detección y clasificación automática de anomalías en radiografías de tórax utilizando Deep Learning

3 de junio de 2024

Autor: Mario Serrano Sánchez

Tutor: Ángel Mario García Pedrero

Departamento de Arquitectura y Tecnología de Sistemas Informáticos

Escuela Técnica Superior de Ingenieros Informáticos

Universidad Politécnica de Madrid

Índice general

1. Introducción	3
1.1. Contexto	5
1.2. Objetivos	6
2. Marco teórico	8
2.1. Detección de objetos	8
2.2. Redes Neuronales Convolucionales	9
2.2.1. Operación de Convolución	10
2.2.2. Operación <i>pooling</i>	10
2.3. <i>Transformers</i>	11
2.4. Modelos que se emplean en este trabajo	12
2.4.1. Facebook DETR (<i>DEtection TRansformer</i>) ResNet 50	12
2.4.2. YOLOS (<i>You Only Look at One Sequence</i>)	13
2.5. Métricas	14
2.5.1. Intersección sobre Unión (IoU)	14
2.5.2. <i>Recall</i>	15
2.5.3. Precisión	15
2.5.4. <i>F1-Score</i>	16
2.5.5. Curva Precisión- <i>Recall</i>	16
2.5.6. Precisión promedio (<i>Average Precision</i> o <i>AP</i>)	16
2.6. <i>Non-maximum suppression</i>	17
2.7. Ensamble de modelos	17
2.8. <i>Weighted Boxes Fusion</i> (WBF)	18
2.9. Enfermedades a detectar en las radiografías	20
2.9.1. Aneurisma de aorta (<i>Aortic enlargement</i>)	20
2.9.2. Cardiomegalia (<i>Cardiomegaly</i>)	21
2.9.3. Engrosamiento pleural (<i>Pleural thickening</i>)	21
2.9.4. Fibrosis pulmonar (<i>Pulmonary fibrosis</i>)	21
2.9.5. Nódulo pulmonar (<i>Nodule/Mass</i>)	23
2.10 Interpretación de las radiografías de tórax	23
3. Metodología	25
3.1. Materiales	25
3.1.1. Conjunto de datos de imágenes radiográficas de tórax <i>VinDr-CXR</i>	25
3.2. Métodos	26
3.2.1. Preprocesamiento y análisis de los datos	26
3.2.2. Modelado	31
3.2.3. Técnica de ensamble	33

4. Experimentos y resultados	34
4.1. Diseño experimental	34
4.2. Resultados	35
4.2.1. Resultados de entrenamiento	35
4.2.2. Evaluación de los resultados	37
4.2.3. Resultados finales	38
4.2.4. Visualización de resultados	41
5. Conclusiones y trabajo futuro	49
5.1. Conclusiones	49
5.2. Trabajo futuro	50
6. Análisis de impacto del trabajo	51
6.1. Impacto general	51
6.2. Objetivos de Desarrollo Sostenible	52
Bibliografía	53

Resumen

El uso de rayos X para el diagnóstico de enfermedades está ampliamente extendido, debido a que la captura de una radiografía es una prueba rápida e indolora cuyos beneficios superan enormemente cualquier riesgo asociado a este tipo de radiación. No obstante, la interpretación de las imágenes radiográficas por parte de los radiólogos puede suponer un verdadero reto, especialmente cuando se trata de la zona torácica.

La detección de anomalías en estas imágenes de rayos X ayudada por computadores podría ofrecer una valiosa segunda opinión a los radiólogos, ayudando a reducir la presión que es ejercida sobre ellos e incluso ayudando a mejorar los diagnósticos en zonas rurales con pocos recursos.

Este Trabajo de Fin de Grado propone un sistema de detección automática de anomalías en radiografías de tórax mediante la utilización de técnicas de ciencia de datos e inteligencia artificial con el fin de ayudar en el diagnóstico de enfermedades como la dilatación aórtica, el engrosamiento pleural o la fibrosis pulmonar, entre otras. Para ello, se propone utilizar un ensamble de modelos de aprendizaje automático profundo para la detección de objetos, que son entrenados con un conjunto de imágenes radiográficas de tórax etiquetadas por expertos radiólogos.

La metodología comprende el preprocesamiento de tanto los datos médicos como las imágenes incluidas en ellos, el entrenamiento de los modelos de detección de objetos, la validación y evaluación de dichos modelos utilizando métricas estándar, y la interpretación de los resultados mediante análisis y casos de estudio.

El sistema ha demostrado ser capaz de detectar hasta cuatro patologías torácicas, especialmente mostrando un mejor rendimiento en la detección de la aneurisma aórtica y la cardiomegalia, logrando un *F1-Score* de entorno 0.84 en ambas clases.

Abstract

The use of X-rays for disease diagnosis is widely spread, as capturing a radiograph is a quick and painless test whose benefits far outweigh any associated risks with this type of radiation. However, the interpretation of radiographic images by radiologists can pose a real challenge, especially when it comes to the thoracic area.

Computer-aided detection of abnormalities in these X-ray images could provide valuable second opinions to radiologists, helping to alleviate the pressure on them and even improving diagnoses in rural areas with limited resources.

This Bachelor's Thesis proposes an automatic anomaly detection system in chest radiographs using data science and artificial intelligence techniques to aid in the diagnosis of diseases such as aortic enlargement, pleural thickening, or pulmonary fibrosis, amongst others. To achieve this, an ensemble of deep learning models for object detection is proposed, trained with a set of chest radiographic images labeled by expert radiologists.

The methodology includes preprocessing of both medical data and the images therein, training of the object detection models, validation and evaluation of the models using standard metrics, and interpretation of results through analysis and case studies.

The system has demonstrated the capability to detect up to four thoracic pathologies, particularly showing improved performance in detecting aortic aneurysm and cardiomegaly, achieving an *F1-Score* of around 0.84 in both classes.

Capítulo 1

Introducción

Aproximadamente 3.600 millones de exámenes médicos de diagnóstico, como rayos X, se realizan cada año en todo el mundo [1]. Solo en Estados Unidos, se toman más de 70 millones de radiografías de tórax al año [2]. La información estadística del gobierno del Reino Unido, proveniente del Servicio Nacional de Salud (NHS) en Inglaterra y Gales, indica que la radiografía de tórax continúa siendo la prueba de imagen más solicitada de manera consistente por parte de los médicos generales [3]. Este tipo de radiografía es comúnmente uno de los primeros procedimientos que se llevan a cabo si un médico sospecha la presencia de alguna enfermedad pulmonar o cardíaca [4].

La importancia de las imágenes de rayos X de esta zona yace en toda la información que se puede obtener de ellas, ya que proporciona una visión detallada de diversos aspectos dentro del cuerpo. Este tipo de imagen puede revelar el estado de los pulmones, identificando condiciones como cáncer, infecciones o acumulación de aire alrededor de un pulmón que pueda causar su colapso, además de mostrar enfermedades crónicas como enfisema o fibrosis quística. También puede detectar problemas pulmonares relacionados con el corazón, como líquido en los pulmones debido a insuficiencia cardíaca congestiva. En cuanto a lo que el corazón se refiere, las radiografías de tórax permiten evaluar su tamaño y contorno, identificando posibles signos de insuficiencia cardíaca, líquido alrededor del corazón o problemas en las válvulas cardíacas. Los grandes vasos sanguíneos cercanos al corazón, como la aorta y las arterias y venas pulmonares, son visibles en estas imágenes, lo que permite detectar aneurismas aórticos, otros problemas vasculares o enfermedades cardíacas congénitas. Asimismo, estas imágenes son capaces de mostrar fracturas de costillas o de la columna vertebral, así como otros problemas óseos [4].

No obstante, a pesar de que las radiografías de tórax son el tipo de imagen de rayos X más comúnmente usado, continúan siendo de las más difíciles de interpretar, debido a factores técnicos como la sub o sobre-exposición, o por la presencia de estructuras que se solapan, al concentrarse diversos órganos en un espacio limitado [5]. Es por esta misma razón que las interpretaciones erróneas de estas radiografías pueden tener consecuencias adversas significativas tanto para el paciente como para la carrera del clínico que las haya realizado.

En 2013, un estudio realizado por el *Royal College of Physicians* de Londres que pretendía evaluar la discrepancia en la interpretación de 50 radiografías de tórax entre

médicos generales y radiólogos [6], destacó los desafíos en la interpretación de estas imágenes. De las 50 radiografías, 21 eran consideradas normales, de las cuales el médico interpretó correctamente 20/21 (95%) como normales. El resto mostraban cambios anormales agudos (14/50, 28%) o cambios anormales crónicos (15/50, 30%). Solo 6/14 (43%) de las radiografías anormalmente agudas se interpretaron correctamente y 11/15 (73%) de los cambios crónicos se interpretaron adecuadamente. No hubo radiografías de tórax que los médicos consideraran como normales cuando había patología anormal. Esto sugiere que, aunque los médicos identificaron correctamente las radiografías normales, solo hicieron el diagnóstico correcto en el 50% de los casos patológicos graves, y especialmente tuvieron dificultades para diferenciar entre neumonía e insuficiencia cardíaca (Figura 1.1).



a Paciente con insuficiencia cardíaca. [7]



b Paciente con neumonía. [8]

Figura 1.1: Ejemplos de radiografías de tórax.

Así pues, se puede afirmar que, incluso para un médico experimentado, la evaluación de enfermedades pulmonares en radiografías de tórax puede llevar a un cierto grado de detección omitida o detecciones erróneas. Esto hace que los sistemas de diagnósticos asistidos por ordenador (DAO) sean un tema de investigación recurrente, ya que estos sistemas pueden ayudar a los médicos a identificar anomalías que pueden pasar fácilmente desapercibidas, mejorando así la precisión de su detección [9].

Esta idea no es nueva, ya en los años 60 se realizó el primer intento por establecer un sistema de detección asistida por computadora, y tras más de seis décadas, ahora existen varios productos comerciales desarrollados para su uso en aplicaciones clínicas. Sin embargo, a pesar de estos avances, la detección automática de enfermedades en radiografías de tórax sigue siendo un desafío debido a la complejidad de las imágenes. La mayoría de los sistemas DAO existentes se centran en la detección precoz del cáncer de pulmón, y hay relativamente pocos estudios dedicados a la detección automática de otros tipos de patologías [9].

Por esta razón, empresas como *Vingroup Big Data Institute* instan a los investigadores a desarrollar sistemas capaces de detectar múltiples enfermedades en imágenes rayos X de tórax a través de competiciones en la plataforma en línea *Kaggle* [10]. En esta competición se proporcionaba un conjunto de datos que cuenta con miles de radiografías de tórax anotadas por radiólogos expertos, y este es el que se ha usado

Introducción

para este proyecto. Esto alivia uno de los mayores desafíos que se presentan a la hora de desarrollar un sistema de este tipo, ya que obtener un conjunto de datos de este tamaño y calidad, con anotaciones precisas y exhaustivas, es extremadamente difícil y costoso.

1.1. Contexto

Históricamente, la interpretación de radiografías ha dependido en gran medida de la experiencia y habilidades de los radiólogos, lo que ha generado limitaciones en términos de precisión y eficiencia. Aunque se han realizado esfuerzos previos para mejorar este proceso mediante la introducción de sistemas de apoyo a la decisión basados en computadoras, son pocos los sistemas enfocados en las radiografías de tórax debido a la complejidad inherente a esta parte del cuerpo [9]. No obstante, ya existen algunos sistemas que abordan este problema obteniendo resultados muy prometedores, entre los que se encuentran los que se detallan a continuación.

- **CheXNet:** CheXNet [11] es una red neuronal convolucional de 121 capas desarrollada por investigadores del Hospital de Stanford, diseñada específicamente para la detección de hasta 14 enfermedades en radiografías de tórax. Esta red fue entrenada en un conjunto de datos masivo que contiene 112.120 imágenes de rayos X de tórax etiquetadas por radiólogos expertos.

Los resultados que arrojó este sistema fueron muy positivos, destacando especialmente que al comparar su rendimiento en la tarea de detección de la neumonía con la de cuatro radiólogos expertos, se obtuvo que CheXNet arrojaba una métrica F1 del 0,435, superior a la de los radiólogos con un 0,387 (Figura 1.2).

	F1 Score (95% CI)
Radiologist 1	0.383 (0.309, 0.453)
Radiologist 2	0.356 (0.282, 0.428)
Radiologist 3	0.365 (0.291, 0.435)
Radiologist 4	0.442 (0.390, 0.492)
Radiologist Avg.	0.387 (0.330, 0.442)
CheXNet	0.435 (0.387, 0.481)

Figura 1.2: Resultados de la comparación entre radiólogos y CheXNet. [11]

- **Estudio de Scripps Research Translational Institute:** A través de un artículo publicado en la revista médica *npj digital medicine* [12], el *Scripps Research Translational Institute* realizó un estudio donde se evaluaba el desempeño de diversos modelos de redes neuronales convolucionales profundas en la tarea de clasificación binaria para determinar la presencia o ausencia de anomalías en radiografías de tórax específicas. En este caso, se utilizó la métrica del área bajo la curva (AUC) para evaluar el rendimiento de los modelos, arrojando unos resultados muy positivos con una media del 0.97.
- **Estudio de Google Health:** Este estudio [13], publicado en la revista *Radiology* y realizado por Google Health en colaboración con *Stanford Healthcare, Palo Alto Veterans Affairs, Apollo Radiology International*, y *California Advanced Imaging*, discute también el uso de modelos de aprendizaje profundo para la detección de cuatro anomalías en radiografías de tórax: neumotórax, nódulo o masa, opacidad pulmonar (Figura 1.3) y fracturas, consiguiendo unas áreas bajo la curva

ROC (*Receiver Operating Characteristic*) de 0.95, 0.72, 0.91 y 0.86 respectivamente. En este caso, se utilizó el procesamiento del lenguaje natural (NLP), que permitió automatizar la generación de etiquetas a partir de los informes de radiología, optimizando así el proceso de etiquetado de imágenes y mejorando la calidad y cantidad de los datos de entrenamiento para los modelos.

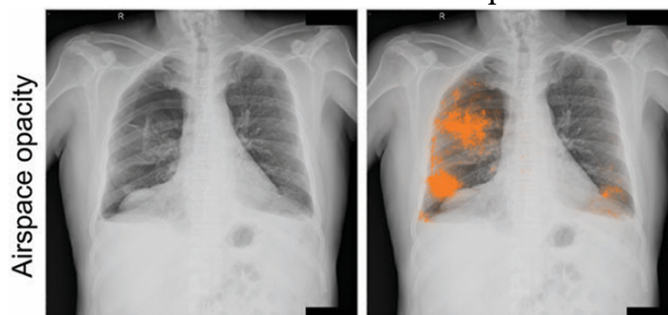


Figura 1.3: Ejemplo de la clase de opacidad pulmonar siendo clasificada correctamente por uno de los modelos entrenados en el estudio. [13]

Después de explorar las distintas aproximaciones, es relevante mencionar el contexto en el que se enmarca este trabajo. Uno de los puntos clave es el desafío de *Kaggle* [10], que tuvo lugar en el año 2021, del que se han extraído los datos utilizados en este trabajo. Este conjunto de datos también ha sido utilizado por múltiples equipos de investigación para sus desarrollos. Por ejemplo, un equipo a destacar participó en el desafío de *Kaggle*, y estaba conformado por Heyang Huang, Yijun Long y Yi Wei, todos del Departamento de Ciencias de la Computación y Estadísticas de la Universidad de Stanford [14]. Aplicaron técnicas de aprendizaje profundo para detectar y localizar anomalías en radiografías de tórax, mejorando significativamente el rendimiento del modelo YOLO empleando optimización de hiperparámetros y aumento de datos. Dicho modelo mejorado sería el que se usase posteriormente en la solución ganadora del desafío.

En el contexto de la detección de enfermedades, una técnica que se ha popularizado en los últimos años para mejorar la precisión de los resultados consiste en aplicar un enfoque basado en aprendizaje por ensambles [15], que consiste en combinar las predicciones de múltiples modelos con el fin de mejorar la precisión de un sistema. En 2015, el CEO de *Kaggle*, Anthony Goldbloom, reveló que los tres algoritmos más populares para problemas estructurados fueron el *XGBoost*, *random forest* y *gradient boosting*, todos ellos métodos de ensamble [16], por lo que es evidente que esta estrategia ha demostrado ser efectiva, y es por esto que se ha decidido aplicarla en este trabajo. En la sección 3.2.3 se explica con mayor detalle este tipo de estrategias.

1.2. Objetivos

El objetivo principal de este Trabajo de Fin de Grado es desarrollar un sistema de detección automática de anomalías en radiografías de tórax. Este objetivo se ha dividido en los objetivos secundarios que se detallan a continuación:

- **Comprender la estructura y contenido de la base de datos de radiografías:** Se ha escogido el conjunto de datos “*VinBigData Chest X-ray Abnormalities Detection*” disponible en la plataforma *Kaggle* [10].

Introducción

- **Implementar estrategias de preprocesamiento y normalización de las imágenes de rayos X:** Se procesan las imágenes para facilitar su posterior análisis y el entrenamiento de los modelos.
- **Implementar y ejecutar la técnica de ensamble en los modelos:** Se entrenarán los modelos y se aplicará la técnica de ensamble para producir una predicción más robusta y confiable.
- **Evaluar el rendimiento del sistema de clasificación:** Se valoran los resultados y se realizan los correspondientes ajustes necesarios.
- **Analizar cómo los resultados obtenidos se comparan con los de los modelos individuales:** El propósito es observar como se comporta cada modelo individualmente y el efecto que tiene la aplicación de la estrategia de ensamble.

Capítulo 2

Marco teórico

Este capítulo aborda los fundamentos esenciales relacionados con la detección de objetos y las técnicas de ciencia de datos y aprendizaje automático que se utilizan, así como conceptos médicos relacionados con el trabajo.

2.1. Detección de objetos

La detección de objetos es un área dentro de la visión por computadora que implica identificar y localizar la presencia de objetos concretos dentro de una imagen o un vídeo. En esta tarea, no solo se reconocen los objetos que se deseen localizar, sino que se determina su ubicación delimitándolos dentro de cajas delimitadoras o *bounding boxes* [17], en muchas ocasiones asignando una puntuación de confianza a dicha detección, como se puede observar en el ejemplo de la figura 2.1

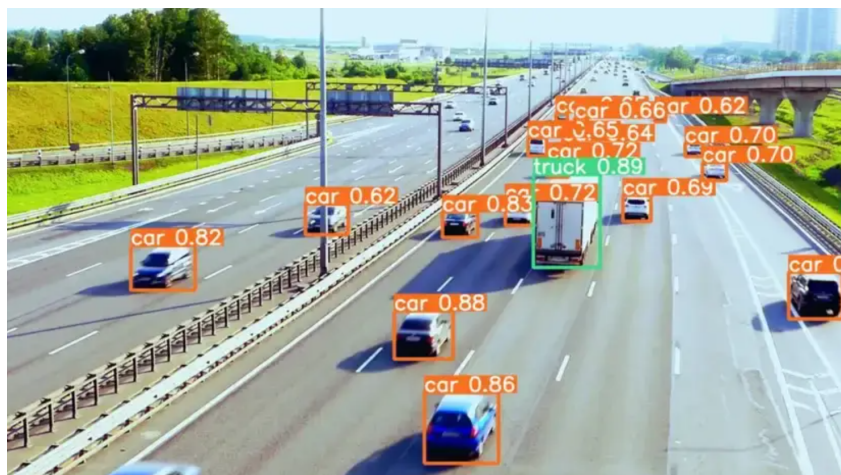


Figura 2.1: Ejemplo de detección de objetos. [18]

La detección de objetos es el pilar fundamental de un amplio abanico de aplicaciones de inteligencia artificial, como la conducción autónoma, la videovigilancia, y la interacción humano-computadora, entre otras. Para llevar a cabo un desarrollo que cuente con esta tecnología hay diversas técnicas disponibles, siendo los enfoques

más populares aquellos basados en el aprendizaje profundo. En general, existen dos enfoques para iniciar la detección de objetos mediante el aprendizaje profundo [17]:

- **Desarrollar y entrenar un detector de objetos personalizado:** Es necesario diseñar una arquitectura de red específica y recopilar una gran cantidad de datos etiquetados para entrenar la red neuronal desde cero, lo que requiere mucho tiempo y recursos de entrenamiento.
- **Afinación (*fine-tuning*) de un detector de objetos preentrenado:** Esta técnica implica iniciar con una red entrenada previamente con una gran cantidad de imágenes, que puede llegar incluso a los millones, y luego adaptarla para una aplicación específica. Algunos ejemplos son los modelos *YOLO*, *Detectron2* o *EfficientDet*. Este método proporciona resultados más rápidos y precisos, y es indicado para desarrolladores con recursos computacionales limitados. Este último enfoque es el que se ha escogido para este proyecto.

2.2. Redes Neuronales Convolucionales

Las redes neuronales convolucionales o CNN (*Convolutional Neural Networks*), son un tipo de arquitectura de redes neuronales especialmente diseñadas para procesar datos estructurados en forma de matrices multidimensionales, por lo que se distinguen por su excelente rendimiento al tratar con imágenes. Este tipo de redes cuentan con tres tipos de capas principales [19]:

1. **Capa convolucional:** Es el componente fundamental de las redes convolucionales. Esta capa realiza operaciones de convolución en los datos de entrada para extraer características relevantes, y requiere algunos componentes, que son los datos de entrada y un filtro o *kernel*. La operación de convolución se realiza con el filtro y el resultado es un mapa de características. Después, se suele aplicar una transformación de Unidad Lineal Rectificada (ReLU) al mapa de características para introducir no-linealidad al modelo, aunque esto no siempre es así, ya que, por ejemplo, en muchas ocasiones se suelen apilar varias capas de convolución de manera consecutiva para explorar y capturar características más complejas y abstractas.
2. **Capa de *pooling*:** El objetivo de esta capa es reducir la dimensionalidad realizando la operación de *pooling*, es decir, reduce el tamaño espacial de los mapas de características, lo que disminuye el número de parámetros y computaciones en la red, al mismo tiempo que mantiene las características más relevantes. Esto es fundamental ya que el entrenamiento con imágenes es un proceso costoso computacionalmente debido al volumen de datos que se maneja.
3. **Capa completamente conectada:** En esta capa, como indica su nombre, cada nodo en la capa de salida se conecta directamente a un nodo en la capa anterior. Esta capa realiza la tarea de clasificación basada en las características extraídas a través de las capas anteriores y sus diferentes filtros. Generalmente utilizan la función de activación *softmax* para clasificar las entradas produciendo probabilidades de pertenencia en el rango de 0 a 1.

Estas tres capas se pueden apreciar en el diagrama simplificado de la arquitectura de una red convolucional de la figura 2.2.

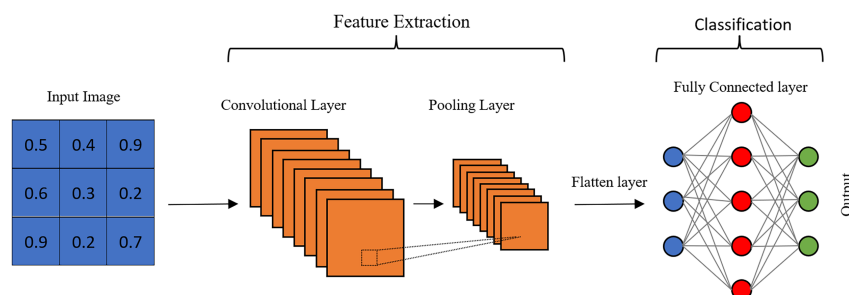


Figura 2.2: Arquitectura básica de una red convolucional. [20]

2.2.1. Operación de Convolución

Como se ha mencionado anteriormente, esta operación es la que se computa en las capas convolucionales de las redes neuronales. En matemáticas, la convolución (*) entre dos funciones produce una tercera función que expresa cómo una de las funciones se modifica por la otra [21]. En el contexto de las redes neuronales convolucionales, la convolución se utiliza para extraer características de los datos de entrada. Esto se logra aplicando un filtro o *kernel* a través de los datos de entrada para producir un mapa de características, que destacan aspectos importantes de la imagen original, como bordes o textura, y nos permiten observar lo que una red entrenada detecta en cada una de sus capas convolucionales [22]. En cuanto al *kernel*, este se trata de una matriz pequeña de dos dimensiones, usualmente cuadrada ($M \times M$), cuyo contenido se determina en función de las operaciones a realizar. Esta matriz se desplaza sobre la entrada (siguiendo un esquema de ventana deslizante) [21] y se aplica repetidamente a cada región de la misma mediante multiplicación y suma de matrices simples, obteniendo un mapa de características de dimensiones reducidas, lo que facilita su manipulación (figura 2.3). Este desplazamiento se denomina **stride**, y controla cuántas unidades se desplaza el filtro en cada paso a lo largo de la entrada [21].

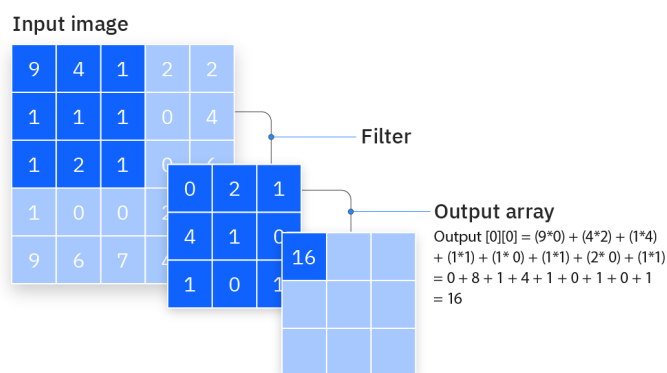


Figura 2.3: Operación de convolución. [19]

2.2.2. Operación pooling

El *pooling* [22] es una técnica utilizada en las redes neuronales convolucionales para reducir la dimensionalidad de los mapas de características tratando de mantener las características relevantes de manera más eficiente. Aunque hay diferentes tipos de

pooling, la idea principal es la misma: sustituir la salida actual en cierta ubicación con una estadística resumida de las salidas cercanas. Esto se logra típicamente tomando el valor máximo (*max pooling*) o promediando los valores (*average pooling*) dentro de una ventana. No obstante, a pesar de que reduce la dimensionalidad, esto se logra a costa de sacrificar detalles.

2.3. Transformers

Los modelos *transformers* son una clase de arquitecturas de redes neuronales diseñadas principalmente para tareas de procesamiento del lenguaje natural, aunque también se han aplicado con éxito en otras áreas, como la visión por computadora, especialmente tras el desarrollo de los derivados *Visual Transformers*, diseñados específicamente para trabajar con datos de imágenes. Lo que distingue a estos modelos del resto es que facilitan la captura de relaciones extensas entre elementos de una secuencia de entrada y permiten el procesamiento en paralelo de la secuencia, además de ser altamente escalables. Esto es posible gracias a que los *transformers* se basan en un mecanismo de “autoatención” que aprende las relaciones entre los elementos [23]. Esencialmente, dada una secuencia de elementos, el mecanismo de autoatención estima la relevancia de un elemento con respecto a otros elementos. El objetivo es capturar la interacción entre todas las entidades mediante la codificación de cada entidad en términos de la información contextual global, otorgando la capacidad de aprender las características tanto locales como globales. La diferencia principal entre la autoatención y la operación de convolución radica en que los filtros se calculan dinámicamente en lugar de ser estáticos, como en el caso de la convolución, donde los filtros permanecen iguales para cualquier entrada [23].

En el contexto de este trabajo, los *transformers* se han usado para la tarea de detección de objetos abordándolo como un problema de predicción de conjuntos y codificando las relaciones entre los elementos del conjunto, permitiendo la predicción de un conjunto de cajas delimitadoras en una sola instancia [23]. En la Figura 2.4 se puede visualizar una versión simplificada de la arquitectura de los *transformers*.

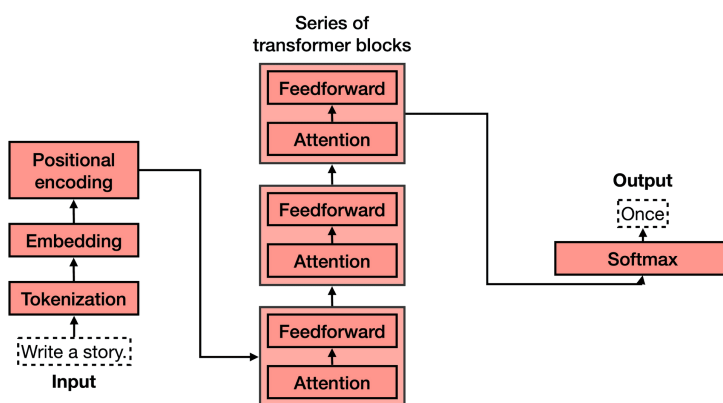


Figura 2.4: Simplificación de la arquitectura de los *transformers*. [24]

Como se puede observar en el diagrama, generalmente la entrada a la red pasa por un proceso de tokenización, donde se divide en unidades más pequeñas llamadas *tokens*. Acto seguido, cada *token* se convierte en un vector numérico de dimensiones fijas conocido como *embedding*, que captura su significado en un espacio dimensional.

2.4. Modelos que se emplean en este trabajo

Finalmente, se agrega información sobre la posición de cada *token* en la secuencia mediante el *positional encoding*, que asigna un vector único a cada posición en el texto o imagen para preservar el orden de los elementos. [24]

2.4. Modelos que se emplean en este trabajo

En este trabajo, se emplean dos modelos de detección de objetos a los que se les realiza *fine-tuning* o afinación, para posteriormente formar un ensamble con ambos.

2.4.1. Facebook DETR (*DEtection TRansformer*) ResNet 50

Es una arquitectura de red neuronal basada en *transformers* introducida por Facebook AI Research (FAIR) en 2020, cuenta con 41,6 millones de parámetros y fue entrenado con el conjunto de datos COCO 2017, compuesto por 118.000 imágenes anotadas [25]. El conjunto de datos COCO [26] se utiliza principalmente para entrenar y evaluar algoritmos de detección y segmentación de objetos, y también proporciona métricas de evaluación estándar que se utilizan para medir el rendimiento de los modelos en estas tareas.

Lo que diferencia al *DETR ResNet-50* es que consta de dos componentes principales [27]:

- **El codificador:** Utiliza la red neuronal convolucional ResNet50 (*Residual Network 50*) para extraer mapas de características de la imagen de entrada. Esta red está formada por 50 capas y cuenta con las características conexiones saltadas o *skip connections*, que permiten que la información se propague directamente desde capas anteriores a capas posteriores, sin necesariamente pasar por todas las capas intermedias, lo que evita el problema del desvanecimiento del gradiente durante el entrenamiento [28]. ResNet50 actúa como un extractor de características, analizando la imagen de entrada y generando mapas de características que capturan patrones y detalles complejos.
- **El decodificador:** El transformador DETR toma estos mapas de características y genera las predicciones de objetos en conjunto, es decir, predice simultáneamente las cajas delimitadoras y las clases a las que pertenecen.

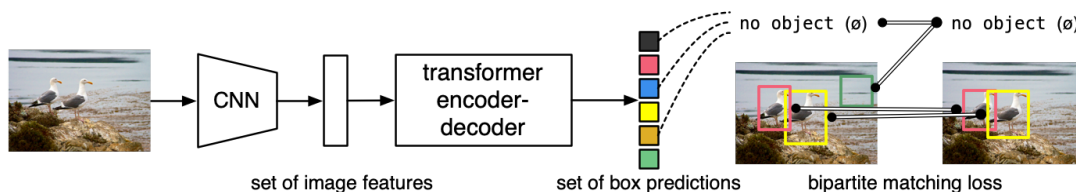


Figura 2.5: Arquitectura del modelo DETR ResNet50. [29]

El proceso de entrenamiento del modelo utiliza lo que se conoce como “coste de emparejamiento bipartito” o *bipartite matching loss*; dada una función de coste en concreto, encuentra la permutación de las predicciones que arroja el coste mínimo total, comparando las predicciones de clases y las coordenadas de las cajas delimitadoras para cada una de las hipótesis realizadas por el modelo sobre la presencia y ubicación de un objeto en la imagen con las anotaciones reales de la imagen [30].

Una vez que se tienen las predicciones y las anotaciones ajustadas, se utiliza el algoritmo de emparejamiento húngaro para establecer la mejor correspondencia uno a uno entre cada consulta y cada anotación. Este algoritmo busca minimizar el coste total de la correspondencia, asegurando así un mapeo óptimo entre las consultas y las anotaciones [25]. Por último, para optimizar los parámetros del modelo, se utilizan las siguientes métricas [25]:

- Para las clases, se utiliza la entropía cruzada estándar (*standard cross-entropy*) [31], que mide la discrepancia entre las distribuciones de probabilidad de las clases predichas y las clases reales.
- Para las cajas delimitadoras, se utiliza una combinación lineal de la función de coste L1 y el IoU (*Intersection over Union*, ver sección 2.5.1) generalizado. La función de coste L1 calcula la diferencia absoluta entre las coordenadas predichas y las coordenadas reales de las cajas delimitadoras, mientras que el IoU generalizado tiene en cuenta la superposición y la diferencia de áreas de las cajas.

El funcionamiento secuencial de este modelo se puede visualizar en la figura 2.5.

2.4.2. YOLOS (*You Only Look at One Sequence*)

YOLOS, es una serie de modelos enfocados a la detección de objetos y basados en la arquitectura del *Transformer* de Visión (ViT) con modificaciones mínimas. YOLOS fue propuesto por primera vez en un estudio publicado por la Escuela de EIC y el Instituto de IA de la Universidad de Ciencia y Tecnología de Huazhong junto con Horizon Robotics [32], de donde se ha obtenido la información contenida en este apartado. El objetivo del estudio descrito en [32], fue investigar si un *Transformer* de Visión puro, preentrenado solo con el conjunto de datos *ImageNet-1k*, que cuenta con 1000 clases de objetos y más de 1.2 millones de imágenes de entrenamiento anotadas por humanos [33], puede transferirse eficazmente a tareas de nivel de objeto y región, como la detección de objetos, sin conocimiento previo sobre la estructura espacial 2D de las imágenes.

El modelo YOLOS se diferencia de otros detectores de objetos al utilizar tokens de detección en lugar de los tradicionales tokens de clasificación utilizados en el *Visual Transformer*. Estos tokens de detección se utilizan para representar objetos potenciales en las imágenes, asociándose con una región específica de la imagen que podría contener un objeto, a diferencia de los tokens de clasificación, que representan contenido más general de una imagen o un texto. Concretamente, cada token de clasificación en el *Visual Transformer* es sustituido en YOLOS por cien tokens de detección, y cada uno aprende a especializarse en ciertas regiones y tamaños de las imágenes, como se puede observar en la figura 2.6.

Como se puede observar en la figura 2.7, en esta arquitectura, la imagen de entrada se divide en pequeños parches que se aplanan y se proyectan linealmente en vectores, denominados *Patch tokens* o tokens de parche. A estos tokens se les añade información sobre la posición espacial dentro de la imagen original. Estos tokens de parche, junto con los tokens de detección, se procesan mediante un codificador de *transformer*, que captura relaciones complejas entre los tokens. Por último, las salidas del codificador correspondientes a los tokens de detección se pasan a través de Perceptrones Multicapa, que generan las predicciones finales de clasificación y coordenadas

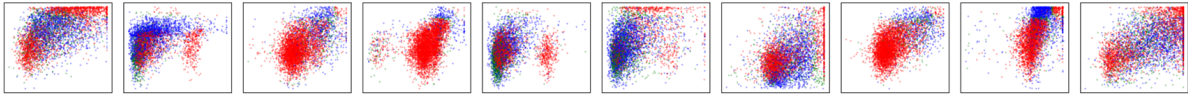


Figura 2.6: Visualización de todas las predicciones de cajas delimitadoras en todas las imágenes del conjunto de validación de COCO para los primeros diez tokens de detección. Cada predicción se representa como un punto cuyas coordenadas están normalizadas según el tamaño de la imagen a la que pertenezcan. Los puntos de color azul corresponden a objetos pequeños, los verdes a objetos mediano y los rojos a objetos de gran tamaño. Se observa como cada token aprende a especializarse en ciertas regiones y tamaños. [32]

de cajas delimitadoras para cada objeto en la imagen [32].

Durante el entrenamiento, al igual que el *DETR*, *YOLOS* emplea una función de coste de emparejamiento bipartito para la detección de objetos en un enfoque de predicción de conjuntos y el algoritmo de emparejamiento húngaro. La arquitectura *YOLOS*, incluso cuando se preentrena únicamente en *ImageNet-1k*, logra resultados competitivos, no obstante, la versión en concreto que se ha utilizado en este trabajo fue entrenado con el conjunto de datos de detección de objetos COCO 2017 por 150 *epochs* [34].

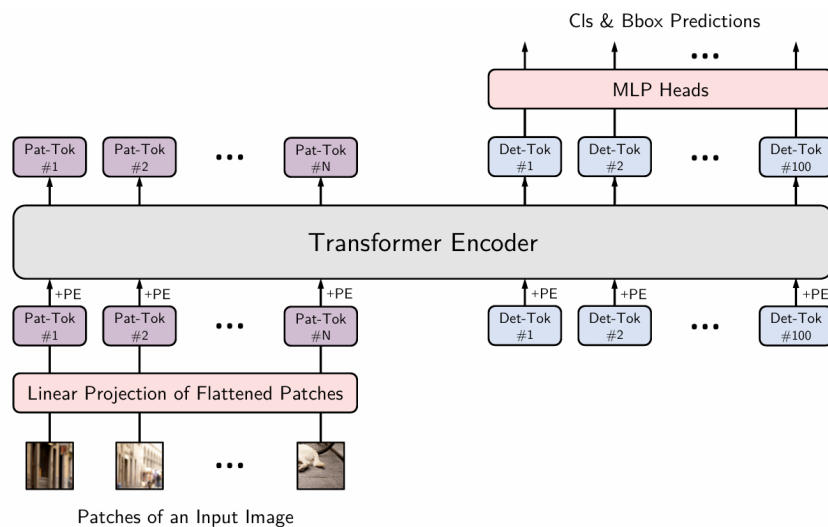


Figura 2.7: Diagrama de la arquitectura del modelo *YOLOS*. [32]

2.5. Métricas

2.5.1. Intersección sobre Unión (IoU)

Indica el solapamiento del área de la caja delimitadora predicha con el de la caja real. Cuanto más alto sea el IoU, más parecidas serán las coordenadas de ambas cajas. Esta métrica se computa de la siguiente forma:

$$IoU = \frac{|A \cap B|}{|A \cup B|} = \frac{I}{U}$$

Siendo A el área de la caja delimitadora predicha y B el área de la caja delimitadora verdadera, e I y U el área de intersección y unión, respectivamente, como se muestra a continuación en la figura 2.8.

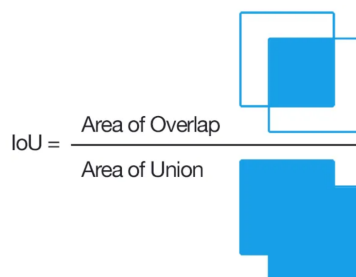


Figura 2.8: Visualización de la métrica IoU. [35]

Generalmente, se define un umbral de IoU para discernir entre los verdaderos y falsos positivos. Si la puntuación IoU de una predicción superase dicho umbral, se consideraría un verdadero positivo, y en caso contrario se consideraría un falso positivo, y por lo tanto, una predicción errónea [36].

2.5.2. Recall

El *recall* es una medida que evalúa cuántas veces un modelo logra identificar de manera correcta los verdaderos positivos, que son los casos correctamente identificados, de todos los casos positivos reales en el conjunto de datos [37]. Para calcular el *recall*, se divide el número de verdaderos positivos entre el total de instancias positivas, que comprende tanto los verdaderos positivos como los falsos negativos, que son los casos no detectados. Así pues, se calcula de la siguiente forma:

$$\text{Recall} = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Falsos Negativos}}$$

El *recall* se puede medir en una escala de 0 a 1 o como un porcentaje, y cuanto más alto, mejor.

2.5.3. Precisión

Similarmente al *recall*, la precisión es una medida que evalúa cuántas veces un modelo logra identificar correctamente la clase positiva, es decir, con qué frecuencia las predicciones positivas son correctas [37]. Para calcular la precisión, se divide el número de predicciones positivas correctas (verdaderos positivos) entre el número total de instancias que el modelo ha predicho como positivas, es decir, tanto los verdaderos como los falsos positivos. Así pues, se calcularía de la siguiente forma:

$$\text{Precision} = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Falsos Positivos}}$$

Al igual que el *recall*, la precisión se puede medir en una escala de 0 a 1 o como un porcentaje, y cuanto más alto, mejor.

2.5.4. F1-Score

El *F1-Score* se interpreta como la media armónica de la precisión y el *recall*, y alcanza su mejor valor en 1 y su peor valor en 0. La contribución relativa de la precisión y el *recall* al *F1-Score* es igual, y se calcula siguiendo la siguiente fórmula [38]:

$$F1-Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

2.5.5. Curva Precisión-Recall

Generalmente, en problemas de detección de objetos multiclase, en cada predicción el modelo estima la probabilidad de pertenencia a la clase predicha de cada caja delimitadora. Cuanto mayor sea esta probabilidad para una clase, mayor es la confianza en el modelo de que dicha caja contenga esa clase. El ajuste de un umbral de probabilidad permite balancear la precisión y el *recall* del modelo, dependiendo del contexto del problema y las prioridades. Un umbral de confianza de probabilidad bajo conduce a más detecciones por parte del modelo, lo que reduce las posibilidades de omitir verdaderos positivos y por tanto aumentando el *recall*. Por otro lado, un umbral de confianza alto aumenta la precisión, ya que reduce la probabilidad de detectar falsos positivos. Por lo tanto, existe un compromiso entre la precisión y el *recall* basado en este umbral. Una forma de visualizar este balance para diferentes valores del umbral es mediante la curva de precisión-*recall* (figura 2.9). Esto nos ayuda a determinar el umbral de confianza más adecuado [39].

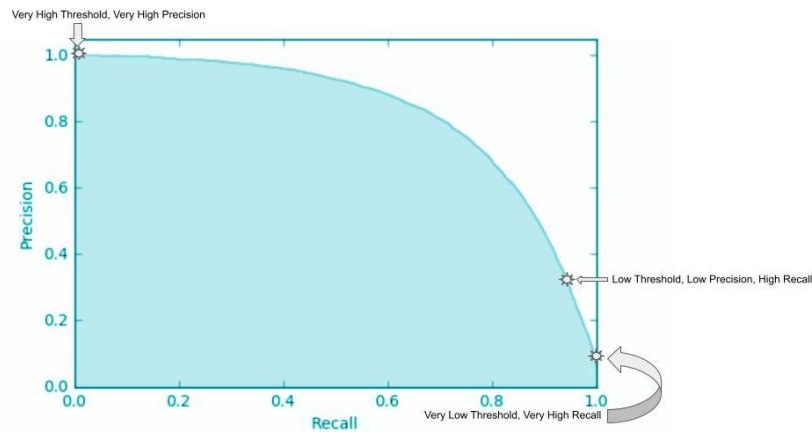


Figura 2.9: Visualización de la curva precisión-*recall*, donde se observa como el *recall* aumenta a medida que aumenta el umbral y disminuye la precisión. [40]

2.5.6. Precisión promedio (Average Precision o AP)

La precisión promedio reduce la curva precisión-*recall* a un valor escalar, y esta métrica se calcula usando el área bajo la curva de la siguiente forma [41]:

$$Average Precision (AP) = \int_0^1 p(r) dr$$

Siendo $p(r)$ la función dada por la curva precisión-*recall*. Así pues, la precisión promedio cuantifica la calidad general de un clasificador, teniendo en cuenta tanto la

precisión como el *recall*, y proporciona una medida resumida de su rendimiento. Cuanto mayor sea esta métrica, mejor será el rendimiento del clasificador. Esta medida se puede calcular para cada clase.

Derivada de esta, se calcula la métrica **precisión promedio media** (*Mean Average Precision* o *mAP*), que consiste en la media de todas las precisiones promedio de cada clase, por lo que cuantifica el rendimiento del algoritmo de detección y localización de objetos [42]. El *mAP* se calcula de la siguiente forma, siendo N el número de clases [42]:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i$$

2.6. Non-maximum suppression

En el área de la detección de objetos, es muy común toparse con instancias en las que existe una redundancia o solape de cajas delimitadoras pertenecientes al mismo objeto. En estos casos es común que se utilice el algoritmo de post-procesamiento *Non-maximum suppression*, cuyo objetivo es unificar todas las detecciones pertenecientes al mismo objeto, generando así un resultado más preciso y conciso [43], como se puede apreciar en la figura 2.10. Este algoritmo consta de los siguientes pasos [44]:

1. Se define un umbral de IoU (Intersección sobre Unión).
2. Se selecciona la caja delimitadora con mayor puntuación de confianza, que es un valor que indica la confianza en dicha predicción.
3. Se calcula el IoU de la caja seleccionada con el resto de cajas delimitadoras pertenecientes a la misma clase.
4. Si el IoU de dos cajas es mayor al umbral establecido, se elimina la predicción de la caja con menor confianza.
5. Se repite este proceso hasta haber procesado todas las cajas delimitadoras.



Figura 2.10: Ejemplo del resultado de aplicar *Non-maximum suppression*. [45]

2.7. Ensemble de modelos

El ensemble de modelos es una técnica de aprendizaje automático cuyo objetivo es obtener predicciones más precisas mediante la agregación de las predicciones de

múltiples estimadores o modelos. Esta técnica se basa en el principio de que una colectividad de modelos produce una precisión general mayor que uno individualmente.

Existen dos tipos principales de aprendizaje por ensamble [46]:

- **Métodos paralelos:** Los modelos base se entrenan en paralelo e independientemente del resto.
- **Métodos secuenciales:** En este caso, se construyen una secuencia de modelos de manera iterativa, donde cada nuevo modelo en la secuencia se entrena para corregir las deficiencias de los modelos anteriores (figura 2.11).

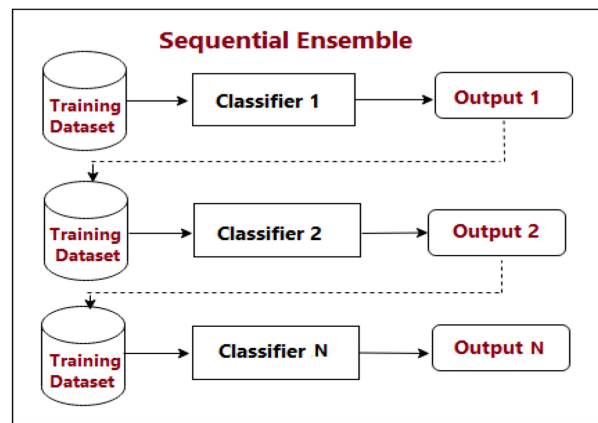


Figura 2.11: Marco general del ensamble secuencial [46].

Para consolidar las predicciones de los modelos base a una sola predicción, un método común es el de la votación. Este método cuenta con tres tipos de votación [46]:

- **Votación por mayoría:** En este método, cada modelo en el ensamble realiza una predicción para una instancia dada, y la clase que reciba más votos entre todos los modelos se selecciona como la predicción final del ensamble.
- **Votación promedio:** En este caso, la predicción final es el resultado de promediar todas las predicciones de los modelos individuales.
- **Votación promedio ponderada:** Es similar al método de votación promedio, pero en este caso se asignan pesos a cada modelo en función de su rendimiento o confianza en las predicciones y se promedian las predicciones ponderadas de acuerdo con estos pesos para obtener la predicción final.

2.8. Weighted Boxes Fusion (WBF)

Recientemente en el ámbito de la detección de objetos ha surgido un método de ensamble llamado *Weighted Boxes Fusion* (WBF). Este método generalmente se aplica para unificar las predicciones de varios modelos en una sola y consta de los siguientes pasos [47]:

1. Cada caja predicha de cada modelo se añade a una lista B , que está ordenada en orden decreciente de las puntuaciones de confianza.
2. Se declaran dos listas vacías, L y F , para conjuntos de cajas y para cajas fusionadas, respectivamente. Cada posición en la lista L puede contener un conjunto

de cajas o una sola, pero cada posición en F contiene una sola caja, que es la caja resultante de la fusión del conjunto de cajas correspondiente en L .

3. Se itera sobre las cajas predichas en B , intentando encontrar una caja coincidente en la lista F . Esta coincidencia se define como una caja con una superposición importante ($IoU > umbral$, ver sección 2.5.1) con la caja en cuestión.
4. Si no se encuentra esta coincidencia, se añade dicha caja de la lista B al final de las listas L y F como nuevas incorporaciones, y se procede a la siguiente caja en B .
5. Si se encuentra una coincidencia, se añade dicha caja a la lista L en la posición pos correspondiente a su caja coincidente en F .
6. Se recalculan las coordenadas de las cajas y su puntuación de confianza en la posición pos de la lista F , usando todas las cajas T acumuladas en la posición pos de L con las siguientes ecuaciones:

$$C = \frac{\sum_{i=1}^T C_i}{T} \quad (1)$$

$$X_{1,2} = \frac{\sum_{i=1}^T C_i * X_{1,2_i}}{\sum_{i=1}^T C_i}, \quad (2)$$

$$Y_{1,2} = \frac{\sum_{i=1}^T C_i * Y_{1,2_i}}{\sum_{i=1}^T C_i} \quad (3)$$

La puntuación de confianza de la caja fusionada C es la media de las confianzas C_i de las cajas que la forman (ecuación 1), y las coordenadas de la caja fusionada son las sumas ponderadas de las cajas que la forman (ecuación 2, ecuación 3), siendo X_1 e Y_1 las coordenadas X e Y de la esquina superior izquierda, y X_2 e Y_2 las coordenadas X e Y de la esquina inferior derecha. Así pues, las predicciones con mayor confianza contribuyen más a la caja fusionada que el resto.

7. Una vez todas las cajas en la lista B han sido procesadas, se re-escalán las puntuaciones de confianza en la lista F . Esto se hace multiplicando las puntuaciones C por el mínimo entre el número de cajas del conjunto T y el número de modelos N , y dividido por el número de modelos N (ecuación 4).

$$C = C * \frac{\min(T, N)}{N} \quad (4)$$

Este algoritmo utiliza todas las cajas sin excluir ninguna, como si pasa con el algoritmo *Non-maximum suppression*, permitiendo incluso corregir casos en los que todas las cajas han sido predichas incorrectamente por todos los modelos [47], como se ilustra en la figura 2.12.

2.9. Enfermedades a detectar en las radiografías

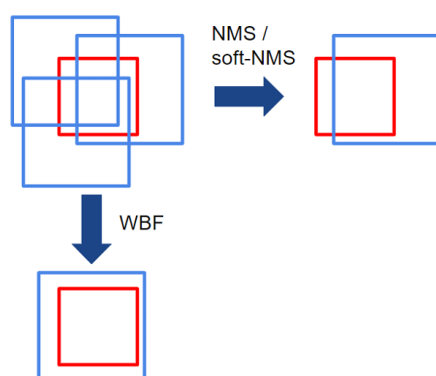


Figura 2.12: Ilustración que muestra la diferencia entre los resultados obtenidos por las variantes del algoritmo *Non-maximum suppression* y el algoritmo *Weighted Boxes Fusion* en un caso concreto. [47]

2.9. Enfermedades a detectar en las radiografías

En esta sección se explican brevemente las patologías que se tratará de detectar en este trabajo, y entre paréntesis se indica el nombre de las clases correspondientes a dichas enfermedades en la base de datos.

2.9.1. Aneurisma de aorta (*Aortic enlargement*)

La aneurisma de aorta se produce cuando una parte de la pared de la arteria aorta se debilita, lo que da lugar a que se expanda o dilate de manera irregular (Figura 2.13).

La aorta es la arteria principal que transporta sangre desde el corazón hasta el resto del cuerpo, por lo que una aneurisma aórtica es una emergencia médica que pone en peligro la vida. Esta enfermedad puede causar la ruptura de la aorta, por lo que la detección precoz es esencial [48].

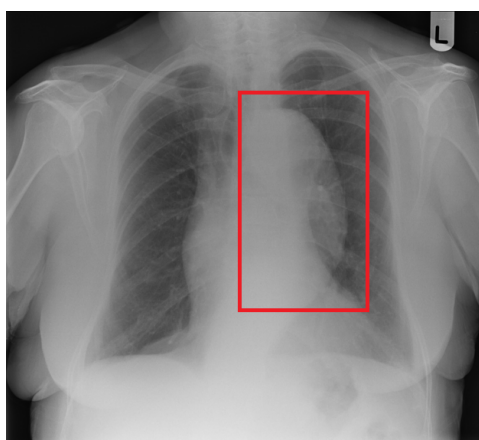


Figura 2.13: Radiografía de tórax donde se aprecia la expansión irregular de la aorta. [49]

2.9.2. Cardiomegalia (*Cardiomegaly*)

La cardiomegalia implica el agrandamiento del corazón. Esto se define generalmente por cuando el diámetro transversal de la silueta cardíaca abarca al menos un 50% del diámetro transversal del tórax cuando se observa en imágenes médicas como las radiografías de tórax (Figura 2.14). Regularmente, la cardiomegalia es un signo clínico de otro proceso patológico en desarrollo, y conduce a un espectro de síndromes clínicos de insuficiencia cardíaca. Aproximadamente, la mitad de las personas diagnosticadas con insuficiencia cardíaca fallece en los próximos cinco años al diagnóstico, por lo que la detección temprana de esta patología es muy importante, ya que muchos pacientes con este padecimiento son asintomáticos [50].

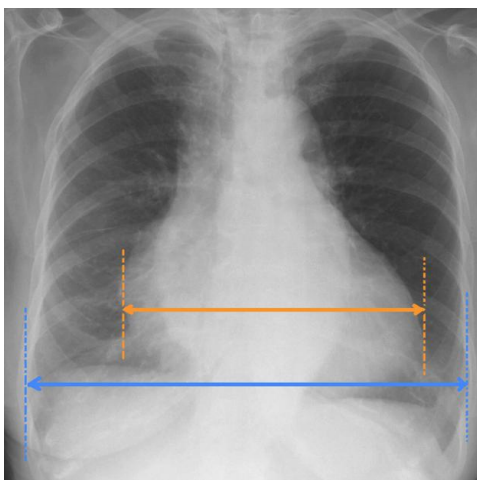


Figura 2.14: Radiografía de tórax donde se aprecia el excesivo diámetro de la silueta cardíaca. [51]

2.9.3. Engrosamiento pleural (*Pleural thickening*)

El engrosamiento pleural se refiere a un engrosamiento anormal de la membrana pleural, que es el revestimiento delicado alrededor de los pulmones y recubre el interior de la cavidad torácica [52] (Figura 2.15). Este engrosamiento puede ser una manifestación de diversas enfermedades pulmonares, incluyendo el cáncer de pulmón [53].

Una investigación llevada a cabo por Saito et. al. [53] donde se obtuvieron 28.727 radiografías de tórax de revisiones médicas anuales reveló que el engrosamiento pleural era la patología más comúnmente identificada.

2.9.4. Fibrosis pulmonar (*Pulmonary fibrosis*)

La fibrosis pulmonar sucede cuando el tejido pulmonar se daña y posteriormente cicatriza, engrosando así dicho tejido (Figura 2.16). Este tejido engrosado y rígido afecta al correcto funcionamiento de los pulmones, siendo el principal síntoma la dificultad de respirar. Normalmente, los médicos no son capaces de dar con la causa de esta enfermedad. La fibrosis pulmonar empeora con el tiempo, y, a medida que empeora, las personas se vuelven cada vez más cortas de aliento. Esta falta de aliento puede empeorar repentinamente, lo que se denomina exacerbación aguda y puede ser potencialmente mortal [55].

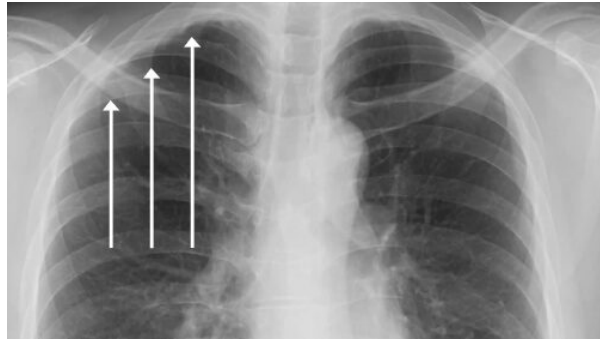


Figura 2.15: Radiografía de tórax donde se aprecia el engrosamiento anormal de la membrana pleural. [54]

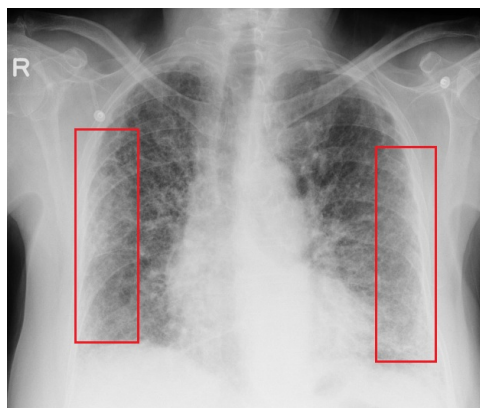


Figura 2.16: Radiografía de tórax donde se aprecia el engrosamiento del tejido pulmonar. [56]

2.9.5. Nódulo pulmonar (*Nodule/Mass*)

Un nódulo pulmonar es una masa o lesión redondeada que se forma en el pulmón y que se a veces se detecta en una radiografía de tórax o abdomen [57] (Figura 2.17). Estos nódulos son muy comunes, y la mayoría (95 %) son benignos o no cancerígenos [58].

Estas masas pueden tener diversas causas, pero generalmente se suelen formar a consecuencia de que una infección o enfermedad inflame el tejido pulmonar, formando así un pequeño grupo de células (granuloma) que con el tiempo puede endurecerse en el pulmón, formando así un nódulo pulmonar no canceroso [58].

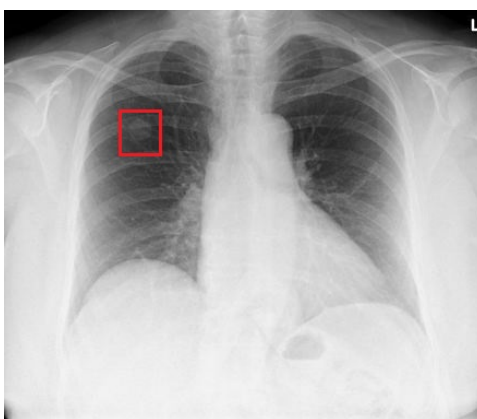


Figura 2.17: Radiografía de tórax donde se aprecia un nódulo pulmonar. [59]

2.10. Interpretación de las radiografías de tórax

La interpretación de las radiografías de tórax se basa en la capacidad de los rayos X atravesar objetos según su densidad. En una radiografía de tórax, se pueden identificar cuatro densidades principales [60]:

- **El hueso**, así como los metales, aparecen completamente blancos en la radiografía debido a que son densos y opacos a los rayos X, lo que los hace fáciles de identificar.
- **El agua** o los tejidos blandos aparecen en un color entre blanco y grisáceo en la radiografía. Los órganos y vasos sanguíneos tienen esta densidad, así como anomalías como acumulaciones de líquido en los pulmones.
- **La grasa** o los tejidos adiposos tienen un tono gris en la radiografía. Puede observarse en regiones como las mamas, los espacios entre las costillas y alrededor de las estructuras del tórax.
- **El gas** o el aire aparece negro en la radiografía, lo que ayuda a identificar fácilmente los pulmones, la tráquea y los bronquios.

La calidad de una radiografía de tórax varía según la exposición a los rayos X. Si la exposición es alta, la radiografía se ve más oscura, lo que dificulta la visualización de algunos aspectos pulmonares pero facilita la visión de zonas detrás del corazón (Figura 2.18), mientras que si la exposición es baja, la radiografía se ve más brillante

2.10. Interpretación de las radiografías de tórax

y puede ser difícil ver las estructuras blandas detrás del corazón (Figura 2.18), pero acentúa ciertas características de los pulmones [61]. Una exposición adecuada es esencial para una correcta interpretación de la radiografía por parte de los profesionales [60].

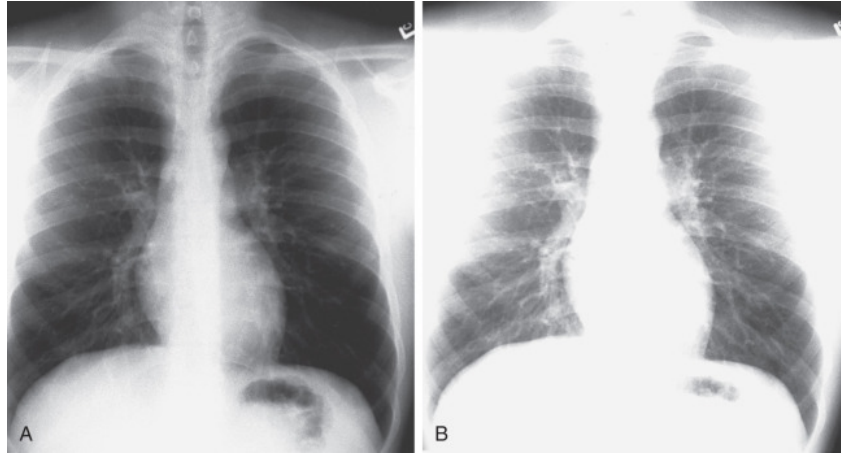


Figura 2.18: De izquierda a derecha, ejemplo de una radiografía con una exposición alta y otra con una exposición baja. [61]

Capítulo 3

Metodología

En este capítulo se explica la metodología seguida para implementar el módulo desarrollado para la detección automática de anomalías en radiografías de tórax.

3.1. Materiales

Esta sección está dedicada a los materiales que se utilizan en este trabajo.

3.1.1. Conjunto de datos de imágenes radiográficas de tórax *VinDr-CXR*

El conjunto de datos que se utiliza en este desarrollo es una versión del *VinDr-CXR*, que fue creado por Ha Q. Nguyen et. al. y colaboradores de instituciones como VinBigData JSC, VinUniversity, y dos hospitales en Vietnam [62]. Este conjunto de datos incluye 18.000 imágenes que fueron anotadas manualmente por un total de 17 radiólogos experimentados con 22 etiquetas locales de rectángulos que rodean anomalías y 6 etiquetas globales de enfermedades sospechosas, mientras que en la versión del reto de *Kaggle* de la que se hace uso en este trabajo cuenta con 14 etiquetas que corresponden a 14 enfermedades distintas y una etiqueta adicional que corresponde a la ausencia de anomalías (*No finding*).

Las imágenes fueron obtenidas entre 2018 y 2020 en el Hospital 108 (H108) y el Hospital Universitario Médico de Hanoi (HMHU), dos de los más importantes hospitales en Vietnam. Se utilizaron diversas máquinas de escaneo de fabricantes conocidos, como Phillips o Samsung, y se obtuvo la aprobación ética de las juntas de revisión institucionales. Para proteger la privacidad de los pacientes, se eliminó toda la información identificable antes de la publicación del conjunto de datos. Las imágenes están disponibles en formato DICOM [10] y se dividen en un conjunto de entrenamiento de 15.000 imágenes y un conjunto de prueba de 3.000, con anotaciones independientes de 3 radiólogos para el entrenamiento y el consenso de 5 radiólogos para el conjunto de prueba. Cabe destacar que en este trabajo solo se usa el conjunto de entrenamiento, ya que es el único evaluable debido a que en *Kaggle* solo se proporcionan las anotaciones de este conjunto.

Como puede observarse en la figura 3.1, dentro de cada archivo DICOM se pueden encontrar atributos asociados a la radiografía de tórax correspondiente, como el sexo del paciente, el número de filas y columnas de píxeles de la imagen, o la

interpretación fotométrica de esta, que indica si la radiografía tiene los colores invertidos o no. No obstante, en los archivos DICOM no se hace referencia alguna a las enfermedades presentes en la radiografía o su ubicación en la misma. Esto es debido a que esta información se proporciona en un archivo CSV aparte, disponible también en *Kaggle* [10]. Este archivo cuenta con 67.914 filas, una por cada enfermedad detectada por cada radiólogo, o en su defecto por la ausencia de enfermedades. Por lo tanto, cada fila cuenta con el identificador de la imagen donde se ha realizado la detección, el nombre e identificador de la clase detectada, el identificador del radiólogo en cuestión, y las coordenadas de la esquina superior izquierda y la inferior derecha de la caja delimitadora que ubica en la enfermedad en la imagen.

```
(0010, 0040) Patient's Sex           CS: '0'
(0028, 0002) Samples per Pixel      US: 1
(0028, 0004) Photometric Interpretation CS: 'MONOCHROME2'
(0028, 0010) Rows                   US: 2430
(0028, 0011) Columns                 US: 1994
(0028, 0030) Pixel Spacing           DS: [0.175, 0.175]
(0028, 0100) Bits Allocated          US: 16
(0028, 0101) Bits Stored              US: 12
(0028, 0102) High Bit                US: 11
(0028, 0103) Pixel Representation    US: 0
(0028, 1050) Window Center           DS: '2047.0'
(0028, 1051) Window Width            DS: '4096.0'
(0028, 1052) Rescale Intercept       DS: '0.0'
(0028, 1053) Rescale Slope           DS: '1.0'
(0028, 2110) Lossy Image Compression CS: '00'
(7fe0, 0010) Pixel Data              OW: Array of 9690840 elements
```

Figura 3.1: Ejemplo de los atributos de un archivo DICOM de una instancia del conjunto de datos.

3.2. Métodos

En esta sección se detalla el procedimiento seguido a lo largo de la realización del trabajo. Este flujo de trabajo se sintetiza en el diagrama de la figura 3.2. El proceso incluye el preprocesamiento y análisis de los datos, seguido del modelado utilizando modelos preentrenados que se afinan para este proyecto específico utilizando el conjunto de imágenes radiográficas. Posteriormente, se aplican técnicas de ensamble para combinar los resultados de diferentes modelos y, finalmente, se realiza una evaluación exhaustiva de los resultados obtenidos. Cada uno de estos bloques será abordado en detalle en este trabajo.

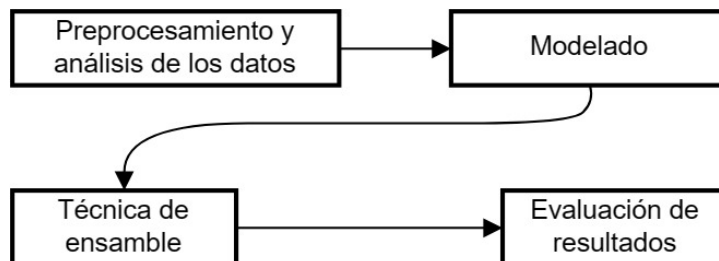


Figura 3.2: Flujo de trabajo o *pipeline* seguido en este proyecto.

3.2.1. Preprocesamiento y análisis de los datos

El proceso de preprocesamiento y análisis de datos es una fase fundamental que sienta las bases para una interpretación precisa de los resultados por parte de los

Metodología

modelos. En esta sección, se detalla cada etapa del proceso, incluidas las técnicas utilizadas para limpiar, transformar y explorar los datos, incluidas las herramientas empleadas para ello.

En primer lugar, se leyó el archivo CSV (correspondiente al conjunto de entrenamiento), que como se ha explicado en el apartado 3.1.1, contiene la información relacionada con las enfermedades presentes en las radiografías. Para leer este archivo y para el resto del proceso de tratamiento de datos, se ha utilizado **Pandas**, la biblioteca de código abierto de Python que proporciona estructuras de datos y herramientas de análisis de datos. Ya con el archivo CSV en un *DataFrame* de *Pandas*, se procedió a estudiar la distribución de las clases. Para ello, simplemente se realizó un conteo de la frecuencia de cada enfermedad en el *DataFrame* y se dividió entre 67.914; el número total de anotaciones. El resultado se representó gráficamente en el gráfico circular de la figura 3.3, y reveló factores importantes, como que la clase "No finding" suponía el 46,9% del total, y las siguientes cinco clases un 36,3%.

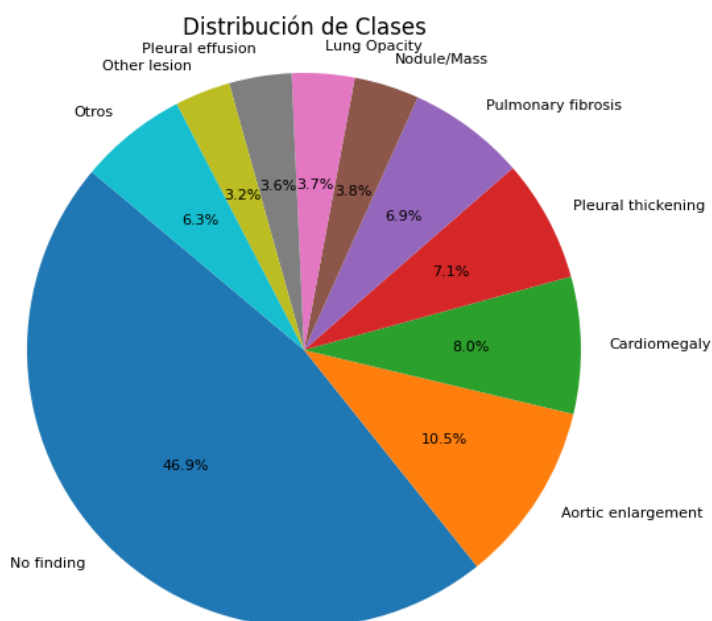


Figura 3.3: Diagrama circular que representa la distribución de clases presentes en el archivo CSV.

Con el fin de agilizar el entrenamiento de los modelos y de optimizar el uso de los limitados recursos computacionales disponibles, en este trabajo se ha decidido omitir la clase "No finding" y solamente conservar las siguientes cinco clases principales (*Aortic enlargement*, *Cardiomegaly*, *Pleural thickening*, *Pulmonary fibrosis* y *Nodule/Mass*). Tras la eliminación de las imágenes que contuviesen enfermedades que no estuviesen entre las cinco clases a conservar, se obtuvo un conjunto de datos de 1.662 instancias, que se redujo a su vez a la mitad, obteniendo así un conjunto final de 831 imágenes.

Ya con el conjunto de datos reducido definitivamente, el siguiente paso fue la eliminación de cajas redundantes. Dado que todas las imágenes de este conjunto fueron anotadas por tres radiólogos distintos, en todas ellas se producía múltiples solapamientos entre las anotaciones de cada experto. Por tanto, se recurrió al algoritmo de *Non-*

maximum suppression (2.6) para solucionar este problema. Para ello, se hizo uso de la biblioteca *Supervision*, que brinda funcionalidades para entrenar modelos de visión por computadora, especialmente en tareas de detección de objetos. Para aplicar el algoritmo de NMS, se debía determinar el umbral IoU (2.5.1), y las puntuaciones de confianza, que se determinaron todas en el valor máximo, 1, debido a que las anotaciones han sido realizadas por expertos. En cuanto al umbral IoU, tras realizar varias pruebas comparativas, se observó que un umbral de 0.4 era el que mejor equilibraba la eliminación de cajas delimitadoras sobrantes sin perder demasiadas detecciones verdaderas. Con un umbral de 0.5, aún se observaban solapamientos significativos en varias imágenes, mientras que con un umbral de 0.4, este problema se reducía considerablemente. No se quiso reducir el umbral por debajo de 0.4 para evitar eliminar cajas que correspondían a detecciones correctas.

Este algoritmo valoró con qué cajas delimitadoras quedarse y de cuáles se podía prescindir, como se puede observar en la figura 3.4.

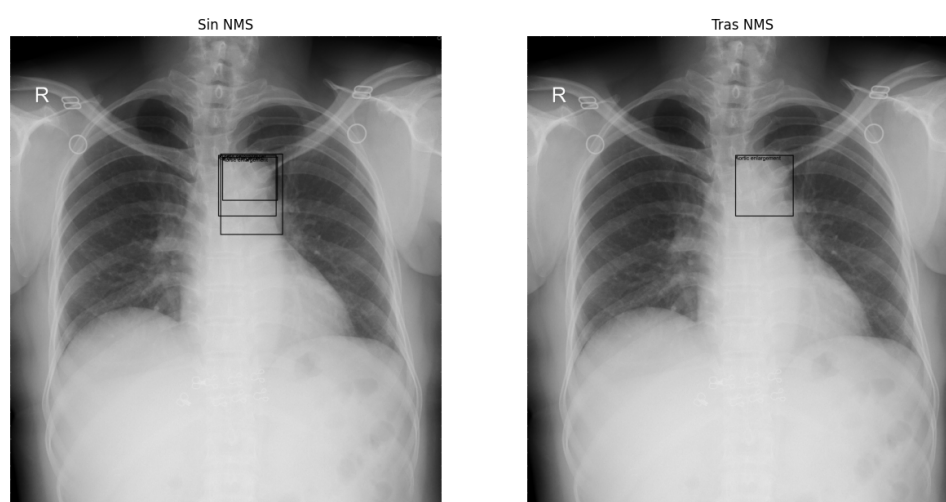


Figura 3.4: Visualización del efecto de aplicar *Non-maximum suppression* a una instancia del conjunto de datos en concreto.

Una vez completado este paso, el procesado de los datos restante se llevó a cabo siguiendo la estructura de los conjuntos de datos de *Hugging Face*, puesto que esta plataforma es la utilizada en este trabajo para afinar los modelos preentrenados. *Hugging Face* es una plataforma líder en la comunidad de inteligencia artificial que ofrece una amplia gama de recursos para el desarrollo y la implementación de modelos de aprendizaje automático. Proporciona bibliotecas, modelos preentrenados, conjuntos de datos, guías y otras herramientas para ayudar a los investigadores y desarrolladores en el campo de la inteligencia artificial.

Dicho esto, se decidió utilizar las librerías de *Hugging Face*, debido a su compatibilidad con los modelos preentrenados disponibles en la plataforma, y se replicó la estructura de los conjuntos de datos de detección de objetos de esta plataforma para facilitar la integración con estos modelos, asegurando así una implementación más eficiente. Esta estructura consta de los siguientes campos:

- **image_id**: El identificador numérico de la imagen.
- **image**: Objeto de tipo `PIL.Image.Image` que contiene una imagen. Este es un

objeto de imagen proporcionado por la biblioteca *Pillow* [63], una biblioteca de Python que permite la manipulación de imágenes de forma sencilla y eficiente.

- **width**: El ancho de la imagen.
- **height**: El alto de la imagen.
- **objects**: Un diccionario que contiene los metadatos de las cajas delimitadoras para los objetos en la imagen. Este diccionario contiene los siguientes campos:
 - **id**: El identificador numérico de la anotación.
 - **area**: El área de la caja delimitadora.
 - **bbox**: Las coordenadas de la caja delimitadora en formato *COCO*.
 - **category**: La clase o categoría del objeto.

En primer lugar, se accedió al conjunto de datos de archivos DICOM con la biblioteca de Python *Pydicom*, que proporciona herramientas para leer, escribir y manipular archivos DICOM de manera eficiente. Con los identificadores de las imágenes presentes en el *DataFrame*, que coinciden con los nombres de sus correspondientes archivos DICOM, se accede solo a los indicados, y se obtienen las imágenes y los campos correspondientes al ancho y alto de la imagen, `Columns` y `Rows`, respectivamente.

Al extraer las imágenes, se comprueba simultáneamente el atributo `Photometric Interpretation` del archivo DICOM, que describe cómo se representan los datos de la imagen en relación con la información de color y la profundidad de bits [64]. Si la interpretación fotométrica de la imagen DICOM es "MONOCHROME1", significa que los píxeles más oscuros se representan con valores numéricos más altos, es decir, los colores están invertidos, y por lo tanto se tiene que invertir la escala de intensidad de la imagen antes de introducirla en el *DataFrame*. Esto se logra restando el valor máximo de intensidad de píxel en la imagen de cada valor de intensidad de píxel en la imagen.

Después de incluir las imágenes y sus dimensiones (ancho y alto) en el *DataFrame*, se construye el diccionario con los metadatos de las cajas delimitadoras. En cuanto al identificador de la anotación, como no se proporcionaba en el archivo CSV, simplemente se ha asignado un número de manera incremental a cada uno. Por otro lado, como se ha mencionado anteriormente, el CSV si incluía coordenadas de las cajas, en formato $(x_{min}, y_{min}, x_{max}, y_{max})$. Estas coordenadas están en formato `pascal_voc`, donde x_{min} y y_{min} son las coordenadas de la esquina superior izquierda, y x_{max} y y_{max} las de la esquina inferior derecha [65]. No obstante, se requerían las coordenadas en formato *COCO*, que se expresan de la forma $(x_{min}, y_{min}, ancho, alto)$ y son las coordenadas de la esquina superior izquierda junto con el ancho y alto de la caja delimitadora [65]. Por lo tanto, se calculó el ancho y alto de las cajas delimitadoras siguiendo las siguientes sencillas fórmulas:

$$width = x_{max} - x_{min}, \tag{1}$$

$$height = y_{max} - y_{min} \tag{2}$$

Estos valores también permitieron calcular el área de cada caja delimitadora, completando así todos los campos necesarios.

Finalmente, se llevaron a cabo pequeñas transformaciones, como cambiar los identificadores de las imágenes de cadenas de caracteres a números enteros de manera incremental, y el mapeo de los identificadores de las clases a números comprendidos entre 0 y 4, debido a la reducción del número de clases. La correspondencia es la siguiente: 0: *Aortic enlargement*, 1: *Cardiomegaly*, 2: *Nodule/Mass*, 3: *Pleural thickening*, y 4: *Pulmonary fibrosis*.

Una vez finalizado el preprocesamiento, se procedió a dividir el conjunto de datos resultante en conjuntos de entrenamiento y prueba, asignando el 80% y 20% del total de datos a cada conjunto, lo que resultó en 665 y 166 instancias, respectivamente. Por último, ambos conjuntos se transformaron en un único conjunto de datos de *Hugging Face* utilizando la librería de Python *datasets*, que es una biblioteca de *Hugging Face* que facilita la carga, procesamiento y manejo de conjuntos de datos utilizados en el aprendizaje automático y la inteligencia artificial.

Este conjunto de datos final se exportó en formato *pickle*, que es un módulo de Python que convierte objetos en un formato binario para almacenamiento o transmisión y los reconstruye de nuevo cuando sea necesario, y se subió al servicio de almacenamiento en la nube *Google Drive* para poder posteriormente acceder a él de manera rápida y sencilla desde *Google Colab*, donde se realizó el resto del preprocesado. *Google Colab* es una plataforma en la nube desarrollada por Google que permite a los usuarios escribir y ejecutar código Python en recursos de cómputo de Google, incluidas GPU y TPU.

Para afinar un modelo, se deben preprocesar los datos para ajustarse al enfoque utilizado para el modelo preentrenado. Dentro de la biblioteca *transformers* de *Hugging Face*, existen herramientas que facilitan esta tarea, permitiendo generar instancias que procesan las imágenes según las necesidades del modelo preentrenado utilizado, simplificando así el proceso de preprocesamiento de los datos. No obstante, antes de procesar las imágenes, se aplican dos transformaciones al conjunto de datos. En primer lugar, para asegurarse de que el modelo no se sobreajusta a los datos de entrenamiento, se aplica la técnica de *data augmentation* o aumento de datos, que es una estrategia para aumentar la cantidad de datos de entrenamiento mediante la aplicación de transformaciones a los datos existentes. Para ello, se hace uso de *Albumentations*, la biblioteca de aumento de datos en Python diseñada específicamente para tareas de visión por computadora. Esta biblioteca garantiza que las transformaciones se aplican a la imagen y modifica las cajas delimitadoras acorde a dicha transformación [66]. En este caso, las transformaciones que se ha decidido aplicar a las imágenes son las siguientes:

- **Redimensión de las imágenes.** De esta manera se garantiza que los datos de entrada del modelo sean uniformes.
- **Rotación de las imágenes.** Al exponer el modelo a diferentes orientaciones de los objetos en las imágenes, se fomenta su capacidad para reconocer objetos desde diferentes perspectivas, lo que puede mejorar su capacidad de generalización a nuevos datos.
- **Cambio aleatorio de la luminosidad de la imagen.** Esto ayuda al modelo a ser

más robusto y capaz de reconocer objetos en diferentes entornos de iluminación, lo que es común en las radiografías al aplicarse diferentes niveles de exposición para destacar determinados elementos, como se explicó en la sección 2.10.

Seguidamente, como los modelos toman tensores como datos de entrada, se define una función que toma el conjunto de datos y, tras procesar las imágenes acorde a las necesidades del modelo, devuelve el conjunto de datos en formato tensor.

Por último, para definir cómo se deben agrupar y preprocesar los datos de entrada antes de alimentarlos al modelo durante el entrenamiento, se define una función agrupadora que toma los lotes de datos y los prepara. Esta función, de suma importancia, realiza las siguientes tareas:

1. Extrae los valores de píxeles de las matrices de *NumPy* correspondientes a cada imagen del lote y los almacena en una lista llamada `pixel_values`.
2. Después de recopilar todos los valores de píxeles, estos deben tener la misma longitud para poder formar un lote homogéneo. Para ello, se realiza un *padding*, es decir, se ajustan las matrices de píxeles de todas las imágenes a una longitud común, agregando píxeles vacíos extras en el caso de que sea necesario.
3. El resultado del *padding* se almacena en un diccionario que contiene los valores de los píxeles y la máscara de los píxeles, `pixel_mask`, que es una matriz binaria que indica qué píxeles son válidos y cuáles son rellenos. En el caso del modelo *YOLOS*, esta máscara no se utiliza.
4. Se recopilan las etiquetas que corresponden a las clases asociadas con cada imagen en el lote y se almacenan en una lista `labels`.
5. Finalmente, se devuelve un diccionario que representa un lote que contiene los valores de los píxeles (`pixel_values`), la máscara de píxeles (`pixel_mask`) y las etiquetas asociadas a cada imagen del lote (`labels`).

3.2.2. Modelado

La etapa de modelado abarca tanto el proceso de elección de los modelos como su entrenamiento.

Para este proyecto, se decidió utilizar modelos con arquitecturas basadas en *transformers*, ya que ofrecen la capacidad para capturar relaciones contextuales en datos secuenciales, lo cual es fundamental para la detección precisa de objetos en imágenes. Tras realizar una investigación exhaustiva sobre el estado del arte de los modelos *transformers* enfocados en la detección de objetos, se identificó el popular *DEtection TRansformer* (DETR), el cual destaca por su capacidad para unificar la detección y el reconocimiento de objetos en una única arquitectura, mejorando la precisión y reduciendo la complejidad del modelo [67]. A su vez, dentro de los modelos DETR, se identificó el *DETR-ResNet-50* de *Facebook* (2.4.1), que fue el finalmente escogido para afinar, ya que el uso de ResNet-50 en el sistema DETR mejora el rendimiento de la detección de objetos en imágenes en aproximadamente un 90% [68]. Este es un modelo que brinda muy buenos resultados, consiguiendo un *Average Precision* (2.5.6) de 42.0 en la validación del conjunto de datos COCO 2017 [25].

Dado que el primer modelo seleccionado fue una arquitectura *DETR*, se decidió incluir en el ensamble a la arquitectura *YOLOS*, la cual también se basa en *transformers*.

Algunas diferencias entre estas dos arquitecturas se listan a continuación [32]:

1. *YOLOS* utiliza una arquitectura de un solo codificador, mientras que *DETR* emplea una arquitectura de codificador-decodificador. Esto hace que *YOLOS* resulte más eficiente en términos de memoria y tiempo de cálculo, al aprender representaciones en una sola etapa.
2. *YOLOS* hereda directamente representaciones de un *Visual Transformer* preentrenado, a diferencia de *DETR* que entrena su codificador y decodificador desde la iniciación aleatoria.
3. *YOLOS* destaca por su simplicidad, ya que demuestra cómo se puede usar un *Visual Transformer* preentrenado para la detección de objetos con cambios mínimos en la arquitectura original.

Adicionalmente, esta arquitectura también obtuvo un *Average Precision* de 42.0 en la validación del conjunto de datos COCO 2017 [32]. Por todas estas razones, se llegó a la conclusión de que sería una muy buena elección para un ensamble de modelos junto con el *DETR*.

Una vez escogidos los modelos, se procedió a su entrenamiento. Para ello, primero se debían definir los hiperparámetros para configurar el entrenamiento de los modelos. A continuación se explican los hiperparámetros relevantes utilizados:

- `per_device_train_batch_size`: Determina el tamaño del lote de entrenamiento que se utilizará durante el entrenamiento.
- `num_train_epochs`: Indica el número de épocas completas que se utilizarán para entrenar el modelo.
- `fp16`: Habilita la precisión de 16 bits durante el entrenamiento en lugar de los convencionales 32 bits para representar valores numéricos, lo que se utiliza para reducir el uso de memoria y acelerar el entrenamiento del modelo en dispositivos de cómputos, en este caso una GPU T4, pero sacrificando precisión numérica [69].

Para el entrenamiento de los modelos se utiliza el optimizador *Adam*, que es un método eficiente y de bajo requerimiento de memoria para la optimización estocástica basada en gradientes. La optimización estocástica es una técnica que ajusta los pesos del modelo usando muestras aleatorias del conjunto de datos en lugar de usar todo el conjunto de datos a la vez. Esto hace que el entrenamiento sea más rápido y eficiente computacionalmente [70].

Los hiperparámetros utilizados para el optimizador *Adam* son los siguientes:

- `learning_rate`: Establece la tasa de aprendizaje inicial del optimizador *Adam*.
- `weight_decay`: Este parámetro controla la cantidad de decaimiento de peso en el optimizador *Adam* que se aplica a todos los pesos de la red, excepto los pesos de sesgo y los pesos de la capa normalización [69].

El entrenamiento se realizó con la clase `Trainer` de la biblioteca *transformers* que ha sido diseñada para facilitar el entrenamiento de modelos en *Pytorch*, que es una biblioteca de aprendizaje automático de código abierto que está diseñada para proporcionar flexibilidad y velocidad en la creación y el entrenamiento de modelos de aprendizaje profundo.

3.2.3. Técnica de ensamble

La técnica de ensamble utilizada ha sido la del algoritmo de *Weighted Boxes Fusion*, explicada en la sección 2.8. Se ha decidido escoger esta estrategia de ensamble ya que es la que mejores resultados ofrece en comparación con otros algoritmos de fusión de cajas [47] y tiene en cuenta la confianza en cada modelo, a diferencia de *Non-maximum suppression*, que está enfocado a predicciones realizadas por un mismo modelo. Para aplicar esta técnica, se ha hecho uso de la librería `ensemble_boxes`, que cuenta con una función específica para ello, que además ha sido desarrollada por el autor del propio método en sí, Roman Solovyev [71]. Este método de ensamble requiere los siguientes elementos:

- `boxes_list`: Una lista con las coordenadas de las predicciones de ambos modelos.
- `scores_list`: Una lista con las puntuaciones de confianza de cada predicción.
- `lables_list`: Una lista con las clases correspondientes a cada predicción de los modelos.
- `weights`: Una lista con los pesos asignados a cada modelo, es decir, la importancia que se e dará a las predicciones de cada modelo.
- `iou_thr`: El umbral de *IoU* que se utilizará para determinar si la superposición entre las cajas es suficiente o no como para considerarse su fusión.
- `skip_box_thr`: Este es un umbral para ignorar cajas con una puntuación de confianza muy baja durante el proceso de fusión de cajas. Las cajas con una confianza menor al umbral no se considerarán.

A partir de estos elementos, se obtienen tanto las coordenadas de las nuevas cajas fusionadas como sus respectivas puntuaciones de confianza y clases asociadas.

Así pues, la metodología consiste en obtener la predicción de ambos modelos por separado, y estructurar sus resultados en el formato correcto de listas para introducir las a la función junto con el resto de parámetros, obteniendo así las nuevas cajas resultantes de la fusión.

Capítulo 4

Experimentos y resultados

Este capítulo aborda el diseño experimental y los resultados obtenidos por el sistema de detección de anomalías en radiografías de tórax, indagando en los resultados individuales de cada modelo y del ensamble, así como en una comparación de los resultados de cada uno.

4.1. Diseño experimental

El diseño experimental del proyecto incluye todas las decisiones y configuraciones que contribuyeron al desarrollo y evaluación del sistema de detección automática de anomalías en radiografías de tórax. Estas configuraciones se pueden agrupar por las diferentes etapas de desarrollo del trabajo:

■ Preprocesamiento de datos

- Todas las imágenes radiográficas fueron redimensionadas a una resolución de 256x256 píxeles, para garantizar que las entradas a los modelos fuesen uniformes y para reducir la carga computacional.
- En el proceso de *data augmentation* o aumento de datos, se aplicaron transformaciones de rotaciones de imágenes en un rango de hasta 20 grados, de manera que se aumentase el volumen de datos y que los modelos pudiesen aprender los patrones que presentan las anomalías en las radiografías en distintas perspectivas.

■ Modelado

- Los hiperparámetros de entrenamiento de ambos modelos se definieron una única vez considerando cuáles serían los adecuados para la tarea de detección de objetos y no se volvieron a modificar (excepto el número de *epochs*, como se explica más adelante), ya que debido a los largos tiempos de entrenamiento y los limitados recursos, no era viable experimentar con estos valores para comparar resultados. A continuación se listan los valores escogidos para los hiperparámetros relevantes utilizados:

- `per_device_train_batch_size = 8`: Se ha escogido un valor de 8, que es común y proporciona un buen equilibrio entre la velocidad de

Experimentos y resultados

entrenamiento y la memoria requerida, además de ser adecuado para el número de instancias en el conjunto de entrenamiento.

- `num_train_epochs`: Este es el único hiperparámetro que se ha ido ajustando de acorde al tiempo disponible para entrenar en cada momento. Debido a los largos tiempos de entrenamiento y los límites de uso de Google Colab, los modelos se han entrenado por partes.
- `fp16 = True`: Se habilitó la precisión de 16 bits durante el entrenamiento para reducir el uso de memoria y acelerar el entrenamiento del modelo, utilizando una GPU T4 de Google Colab.

De igual manera, los hiperparámetros del optimizador *Adam* se definieron una única vez:

- `weight_decay = 1e-4`: El valor escogido es una elección prudente y estándar que proporciona una regularización efectiva sin comprometer significativamente el rendimiento del modelo en los datos de entrenamiento.
 - `learning_rate = 1e-6`: El valor de $1e-6$ (0.000001) escogido es bajo, ya que, a pesar de que el aprendizaje tomará más tiempo, se asegura que se aprenda adecuadamente, especialmente al tratar con un problema con datos de alta dimensionalidad como es la detección de objetos.
- En primera instancia, se decidió utilizar una versión reducida del modelo *YOLOS* [72] con el fin de reducir los tiempos de entrenamiento, pero tras las primeras 200 *epochs* el modelo dejaba de mejorar, por lo que finalmente se decidió cambiar al modelo *YOLOS* base [34].

▪ **Ensamble de modelos**

- Los pesos asignados a las predicciones de *DETR ResNet-50* y *YOLOS* fueron 1.5 y 1 respectivamente.
- Se escogió un valor de 0.1 para el umbral de ignorar predicciones, de manera que las cajas con una confianza menor al umbral no se considerarán.

▪ **Evaluación de los modelos**

- Para cada modelo y el ensamble, se ha calculado la métrica *F1-Score* para seis diferentes umbrales de IoU (0.2, 0.3, 0.4, 0.5, 0.6 y 0.7) para analizar la sensibilidad de cada uno a distintos niveles de solapamiento entre predicciones y las anotaciones reales.

4.2. Resultados

4.2.1. Resultados de entrenamiento

En primer lugar se entrenó el *DETR ResNet 50*. El entrenamiento se realizó por partes, ejecutando cierto número de *epochs* cada vez dependiendo del tiempo disponible para ello en el momento. Después de cada ejecución, se almacenaba el modelo resultante en el *Hugging Face Hub*, que es una plataforma en línea proporcionada por *Hugging*

Face que permite la publicación y distribución de modelos entrenados por los usuarios. Al retomar el entrenamiento, simplemente se descarga de nuevo el modelo y se continua la ejecución.

De esta forma, se entrenó el modelo *DETR ResNet 50* hasta las 310 *epochs*, finalizando en ese número al no notar mejoría a lo largo de 60 *epochs* en la función de pérdida, como se puede observar en la figura 4.1, concluyendo así con una pérdida de aproximadamente 0.8, frente a la pérdida de más de 7 que se obtenía al comenzar el entrenamiento. El tiempo de entrenamiento fue de aproximadamente 4 minutos por *epoch*, lo que se traduce en aproximadamente 21 horas en total.

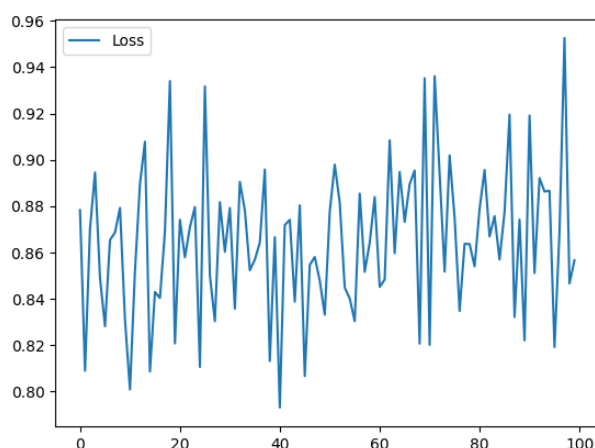


Figura 4.1: Visualización de la evolución de la función de pérdida en las últimas 60 *epochs* o 5000 pasos de entrenamiento del modelo *DETR ResNet-50*.

En cuanto al modelo *YOLOS*, los tiempos de entrenamiento fueron aproximadamente un 25% más rápidos que los del *DETR*, tardando algo menos de 3 minutos en completar una *epoch*. No obstante, a pesar de que inicialmente la pérdida decreció rápidamente, llegó un punto en el entrenamiento en el que se hizo evidente que este modelo tenía más dificultades en el proceso de aprendizaje, ya que la pérdida decrecía muy lentamente, a diferencia de como lo hacía el modelo *DETR* en la misma etapa, como se puede observar en la figura 4.2.

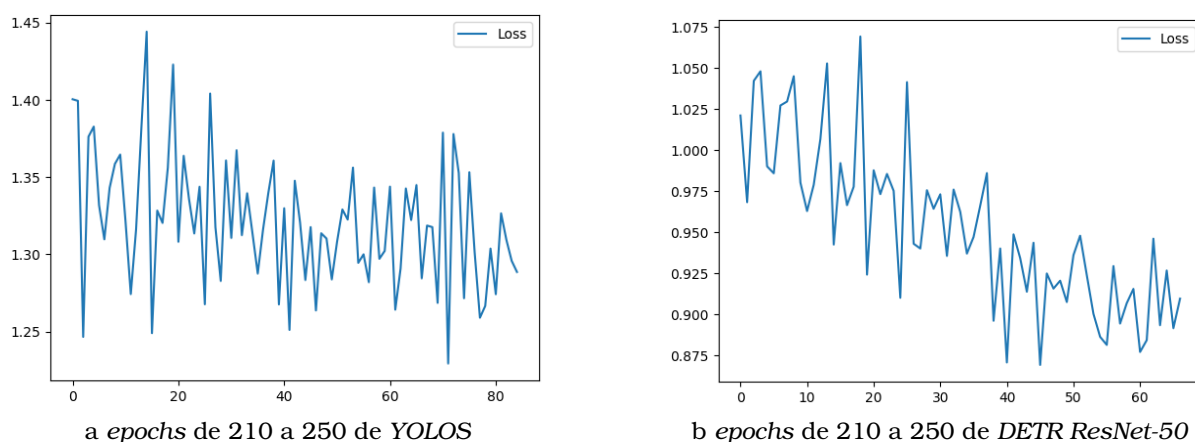


Figura 4.2: Evolución de la función de pérdida del entrenamiento de los modelos *YOLOS* y *DETR ResNet-50* en la misma etapa del entrenamiento.

Experimentos y resultados

Finalmente, tras 330 *epochs*, se decidió finalizar el entrenamiento del modelo *YOLOS*, al notar mínimas mejoras en la función de pérdida durante el entrenamiento, como se puede observar en la figura 4.3 de las últimas 30 *epochs*, donde destaca la poca mejoría respecto la figura 4.2, 80 *epochs* antes. En total, el entrenamiento de este modelo tomó aproximadamente unas 16 horas, y se finalizó con una pérdida considerablemente mayor a la del *DETR* (1.25 frente a 0.8).

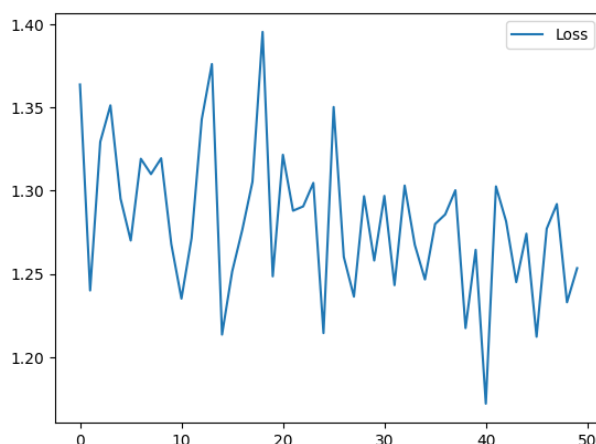


Figura 4.3: Visualización de la evolución de la función de pérdida en las últimas 30 *epochs* del entrenamiento del modelo *YOLOS*.

4.2.2. Evaluación de los resultados

La evaluación de los resultados se ha realizado de diversas formas. En primer lugar, se han utilizado las métricas de evaluación estándar de COCO, que es como suelen evaluarse comúnmente los modelos de detección de objetos [66]. Para ello, se utiliza una implementación de dichas métricas de la biblioteca de visión por computadora en *PyTorch* que proporciona una API de evaluación de métricas COCO, *torchvision*. Este método de evaluación cuenta con las siguientes 12 variantes de las métricas *Average Precision* y *Average Recall* [73]:

- **Average Precision (AP):** Las evaluaciones COCO no distinguen entre *mean Average Precision* y *Average Precision*, pero en realidad se trata de la primera, ya que se trata de la media de las precisiones promedio de todas las categorías. Se evalúa con la seis siguientes variantes:
 - Precisión promedio de diez valores de IoU en un rango de 0.5 a 0.95.
 - Precisión promedio cuando $IoU = 0.5$
 - Precisión promedio cuando $IoU = 0.75$
 - Precisión promedio para objetos pequeños, de área menor a 32 píxeles cuadrados.
 - Precisión promedio para objetos medianos, de área comprendida entre 32 y 96 píxeles cuadrados.
 - Precisión promedio para objetos grandes, de área mayor a 96 píxeles cuadrados.

- **Average Recall (AR).** Se trata del *Recall* promedio de todas las clases. Se evalúa con la seis siguientes variantes:
 - *Recall* promedio cuando se permite 1 detección por imagen, es decir, solo se conserva la detección con mayor puntuación de confianza de cada imagen.
 - *Recall* promedio cuando se permiten 10 detecciones por imagen.
 - *Recall* promedio cuando se permiten 100 detecciones por imagen.
 - *Recall* promedio para objetos pequeños.
 - *Recall* promedio para objetos medianos.
 - *Recall* promedio para objetos grandes.

No obstante, estos resultados no proporcionan métricas específicas para cada clase, lo que sería deseable para una evaluación exhaustiva del desempeño del modelo en distintas categorías. Adicionalmente, no se puede evaluar al ensamble de esta manera, ya que este se computa de manera manual, como se ha explicado en la sección 3.2.3. Por lo tanto, se decidió realizar evaluaciones adicionales para solventar estos problemas. Para ello, se ha partido de funciones de Python para calcular el IoU, la precisión y el *recall* en problemas de detección de objetos, disponibles en un artículo de *Analythics Vidhya* [74], que posteriormente se han modificado para ajustarlas a las necesidades específicas de este trabajo.

Iterando sobre el conjunto de datos de test, se realizan las predicciones del modelo y, con las anotaciones reales, se calcula la precisión, el *recall* y el *F1-Score* por cada imagen para poder posteriormente evaluar que tipo de radiografías son las que brindan mejores resultados y en cuáles el modelo se encuentra con más dificultades. Simultáneamente, los verdaderos y falsos positivos de cada clase son almacenados en un diccionario para poder posteriormente calcular las mismas métricas pero específicas para cada enfermedad. Adicionalmente, realizar esta evaluación de manera manual imagen por imagen tiene la ventaja de que se puede aplicar *Non-maximum suppression* a los resultados de las predicciones para asegurar de que no haya cajas redundantes, lo que puede causar errores en el proceso de evaluación, puesto que pueden aparecer más verdaderos y falsos positivos de los que deberían.

Por último, también se calculó estas métricas específicas de cada clase para diferentes valores de IoU, de manera que se pudiese observar como este factor afecta a los resultados y encontrar el valor óptimo.

4.2.3. Resultados finales

En primer lugar, se computaron los resultados con las métricas COCO para ambos modelos tras finalizar su entrenamiento, obteniendo los resultados que se exponen en el cuadro 4.1.

Estos resultados reflejan que el *DETR ResNet-50* generalmente supera a *YOLOS* en términos de precisión promedio (AP) en la mayoría de las condiciones de IoU y áreas, mientras que el *YOLOS* tiene un mejor desempeño en términos de *recall* promedio (AR) cuando se permiten más detecciones por imagen (10 y 100 detecciones). Adicionalmente, estos resultados evidencian que ambos modelos tienen problemas con la detección de objetos pequeños y medianos, lo que se refleja en sus bajas precisiones y

Experimentos y resultados

Modelo	Métrica	Valor
<i>DETR ResNet-50</i>	Average Precision (AP)	
	IoU = 0.50:0.95	0.155
	IoU = 0.50	0.345
	IoU = 0.75	0.106
	IoU = 0.50:0.95, objetos pequeños	-1.000
	IoU = 0.50:0.95, objetos medianos	0.000
	IoU = 0.50:0.95, objetos grandes	0.157
	Average Recall (AR)	
	IoU = 0.50:0.95, detecciones máx. = 1	0.204
	IoU = 0.50:0.95, detecciones máx. = 10	0.244
	IoU = 0.50:0.95, detecciones máx. = 100	0.257
	IoU = 0.50:0.95, objetos pequeños	-1.000
	IoU = 0.50:0.95, objetos medianos	0.005
	IoU = 0.50:0.95, objetos grandes	0.262
<i>YOLOS</i>	Average Precision (AP)	
	IoU = 0.50:0.95	0.145
	IoU = 0.50	0.321
	IoU = 0.75	0.090
	IoU = 0.50:0.95, objetos pequeños	-1.000
	IoU = 0.50:0.95, objetos medianos	0.000
	IoU = 0.50:0.95, objetos grandes	0.145
	Average Recall (AR)	
	IoU=0.50:0.95, detecciones máx. = 1	0.191
	IoU=0.50:0.95, detecciones máx. = 10	0.261
	IoU=0.50:0.95, detecciones máx. = 100	0.268
	IoU=0.50:0.95, objetos pequeños	-1.000
	IoU=0.50:0.95, objetos medianos	0.000
	IoU=0.50:0.95, objetos grandes	0.271

Cuadro 4.1: Resultados de métricas COCO para los dos modelos que conforman el ensamble.

recall promedios en esas categorías. Especialmente destaca que ambos modelos han obtenido valores negativos en el *Average Precision* (AP) y *Average Recall* (AR) para áreas pequeñas, lo que probablemente indica que los modelos no han sido capaces de detectar ningún objeto pequeño correctamente.

A la luz de estos resultados, se decidió asignar un peso de 1 al modelo *YOLOS*, y de 1.5 al modelo *DETR ResNet-50* para el ensamble.

Por otra parte, en el cuadro 4.2 se muestran los resultados específicos de la métrica *F1-Score* para cada clase obtenidos para cada modelo, así como para el ensamble de ambos con los pesos mencionados, para seis valores diferentes del umbral de IoU.

Modelo	IoU	Clase 1	Clase 2	Clase 3	Clase 4	Clase 5
<i>DETR ResNet-50</i>	0.2	0.916	0.846	0	0.112	0.034
	0.3	0.883	0.839	0	0.089	0.034
	0.4	0.838	0.839	0	0.067	0.034
	0.5	0.774	0.839	0	0.044	0.034
	0.6	0.696	0.771	0	0.022	0.034
	0.7	0.464	0.621	0	0.022	0.034
<i>YOLOS</i>	0.2	0.859	0.826	0	0.101	0
	0.3	0.859	0.826	0	0.075	0
	0.4	0.859	0.826	0	0.025	0
	0.5	0.815	0.771	0	0.025	0
	0.6	0.772	0.700	0	0	0
	0.7	0.592	0.488	0	0	0
Ensamble WBF	0.2	0.899	0.828	0	0.170	0.037
	0.3	0.865	0.820	0	0.127	0.037
	0.4	0.837	0.820	0	0.085	0.037
	0.5	0.795	0.804	0	0.042	0.037
	0.6	0.740	0.750	0	0.021	0.037
	0.7	0.560	0.578	0	0.021	0.037

Cuadro 4.2: *F1-Scores* de cada clase para diferentes IoUs, siendo las clases *Aortic enlargement*, *Cardiomegaly*, *Nodule/Mass*, *Pleural thickening* y *Pulmonary fibrosis*, en ese orden. En negrita se resaltan los mejores resultados de cada IoU para cada clase.

Lo primero que destaca al visualizar la tabla de resultados, es que tanto *DETR ResNet-50* como *YOLOS* presentan dificultades significativas para detectar la clase 3 (*Nodule/Mass*) y la clase 5 (*Pulmonary fibrosis*) en todas las configuraciones del umbral de IoU. Esto es probable que se deba a la poca cantidad de datos de entrenamiento para estas clases, ya que a pesar de que se trató de que estuviesen balanceadas, no fue posible debido a la reducida representación de estas clases en el conjunto de datos. Por otro lado, a esto se le suma las características visuales que hacen que estas clases sean más difíciles de detectar y clasificar correctamente, puesto que, por lo que se ha observado en las radiografías del conjunto de datos, los nódulos pulmonares son comúnmente de tamaño pequeño y difíciles de localizar, así como la fibrosis pulmonar, que resulta desafiante delimitar en una caja.

En cuanto a las clases 1 y 2, se observa que el *DETR ResNet-50* produce mejores resultados en todos los valores de IoU para la clase 2. Sin embargo, a partir de un

Experimentos y resultados

IoU de 0.4, el YOLOs supera al *DETR ResNet-50* en la clase 1. Es importante destacar que el YOLOs también obtiene, en general, buenos resultados en ambas clases.

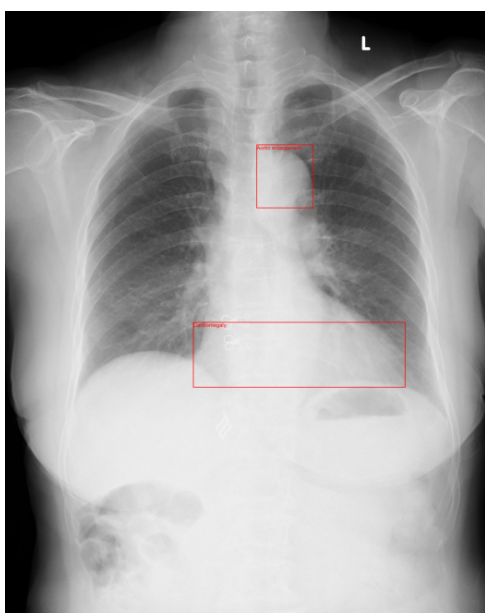
El ensamble WBF combina las predicciones de ambos modelos y muestra un mejor rendimiento en la clase 4 (*Pleural thickening*), con un *F1-Score* de 0.170 en IoU 0.2, en comparación con 0.112 (*DETR ResNet-50*) y 0.101 (YOLOS) en el mismo umbral de IoU. Esto indica que el ensamble puede aprovechar las fortalezas de ambos modelos y mejorar la detección en algunas clases difíciles. Adicionalmente, a pesar de que no logra superar los resultados del *DETR ResNet-50* en la clase 2 (*Cardiomegaly*), si lo consigue significativamente en la clase 1 (*Aortic enlargement*) para valores del umbral de IoU superiores a 0.4, gracias a la aportación del modelo YOLOs, que también obtuvo mejores resultados que el *DETR ResNet-50* en estas configuraciones.

Respecto a los umbrales de IoU, los valores de *F1-Score* disminuyen consistentemente con el aumento del umbral de IoU en todos los modelos. Este comportamiento es esperado, ya que los umbrales de IoU más altos requieren una mayor precisión en las predicciones de las cajas delimitadoras. Por ejemplo, la clase 1 en *DETR ResNet-50* cae de 0.916 en IoU 0.2 a 0.464 en IoU 0.7. De manera similar, YOLOs en la clase 1 muestra una disminución de 0.859 a IoU 0.2 a 0.592 en IoU 0.7. Un valor adecuado para este umbral sería de 0.5, donde se podría optar o bien por el *DETR ResNet-50* o por el ensamble WBF, puesto que obtienen un rendimiento similar.

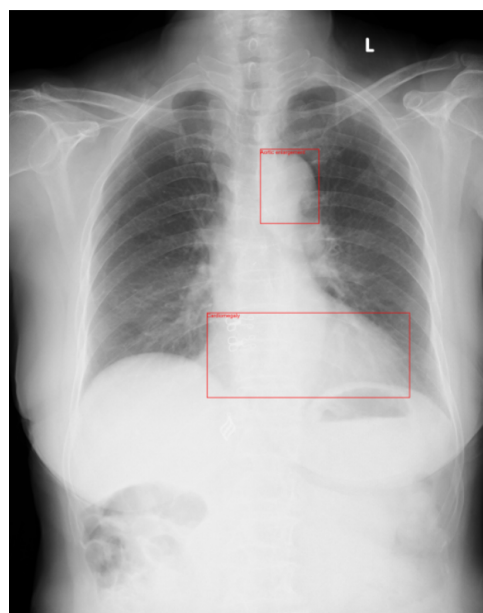
4.2.4. Visualización de resultados

Dado que se calculó la precisión, el *recall* y el *F1-Score* para cada modelo y el ensamble en cada imagen del conjunto de prueba, es posible visualizar las predicciones realizadas por los modelos en los casos con valores más o menos elevados para dichas métricas. Por ello, se ha decidido visualizar las predicciones donde se han obtenido los *F1-Score* más elevados ($F1-Score > 0.8$) y los más bajos ($F1-Score < 0.2$) para cada modelo y el ensamble, y compararlas con las predicciones de los otros modelos en esas mismas imágenes.

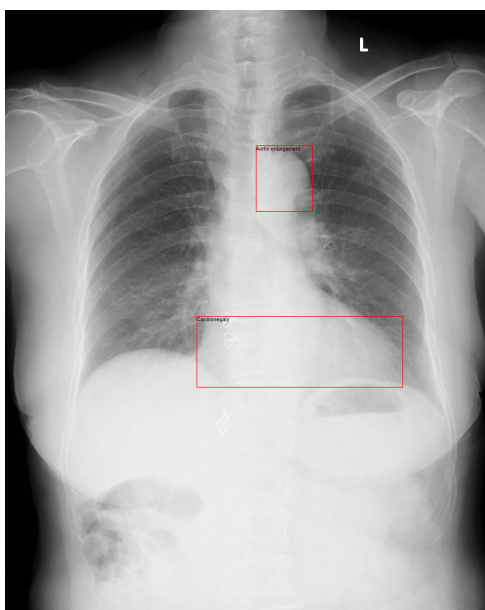
A continuación, se presentan las visualizaciones de las predicciones con *F1-Score* más elevados y más bajos. Cada figura muestra las predicciones realizadas por los modelos *DETR ResNet-50*, YOLOs, el ensamble y las anotaciones reales realizadas por los radiólogos, en ese orden. Primero, se muestran las predicciones con uno de los *F1-Score* más altos y uno de los más bajos para el modelo *DETR ResNet-50* (figuras 4.4 y 4.5). A continuación, se presentan las visualizaciones correspondientes para el modelo YOLOs (figuras 4.6 y 4.7), seguido por las predicciones del ensamble WBF (figuras 4.8 y 4.9), en cada caso comparándose con el resto.



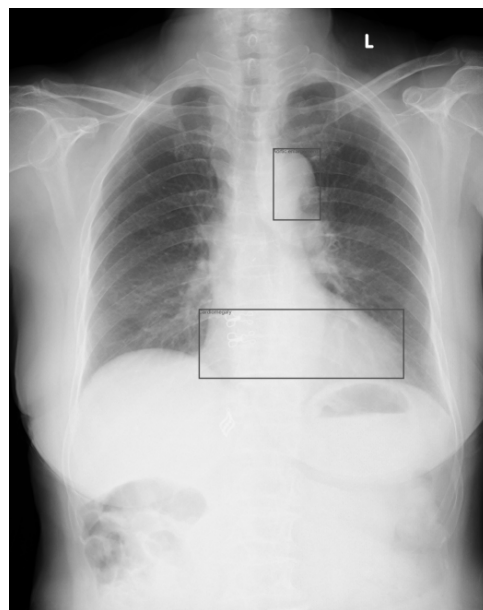
a Predicción realizada por el *DETR ResNet-50*.



b Predicción realizada por el *YOLOS*.

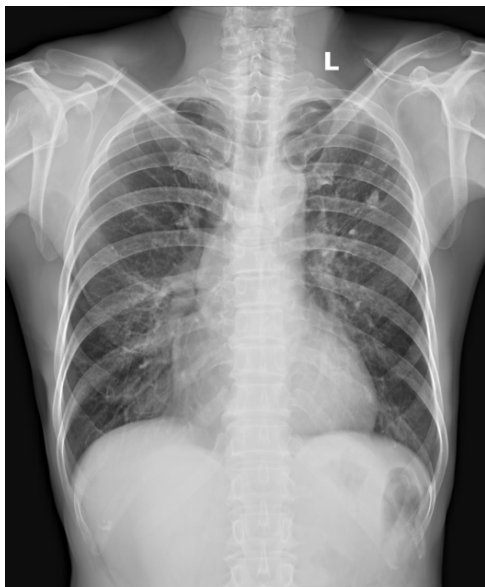


c Predicción resultado del ensamble.

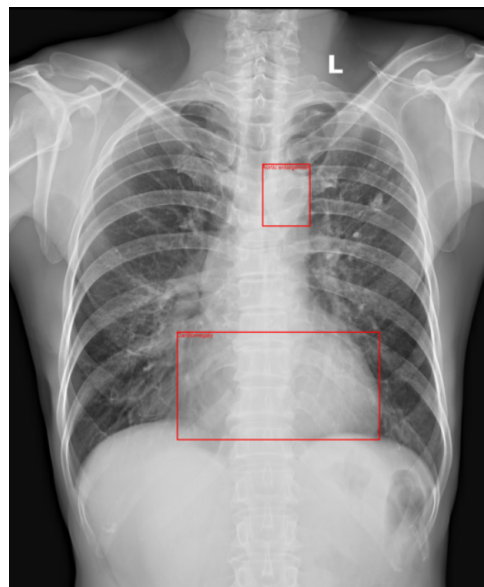


d Anotaciones reales realizadas por los radiólogos.

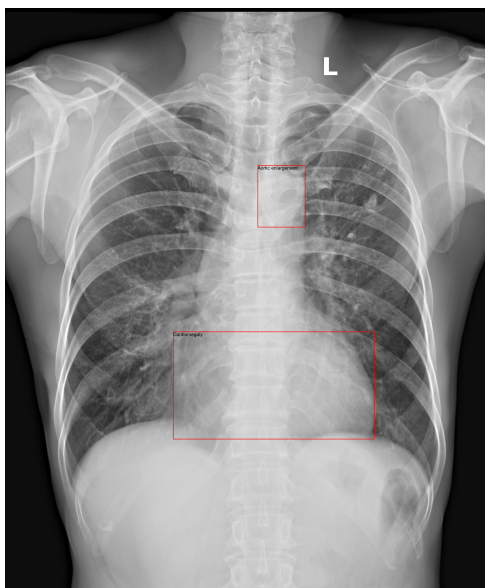
Figura 4.4: Visualización de los resultados obtenidos por los dos modelos entrenados y el ensamble en uno de los resultados obtenidos por el *DETR ResNet-50* con mayor *F1-Score*.



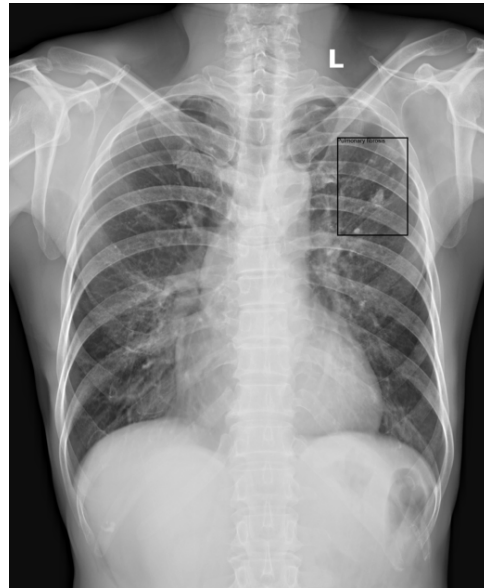
a Predicción realizada por el *DETR ResNet-50* (nula).



b Predicción realizada por el *YOLOS*.

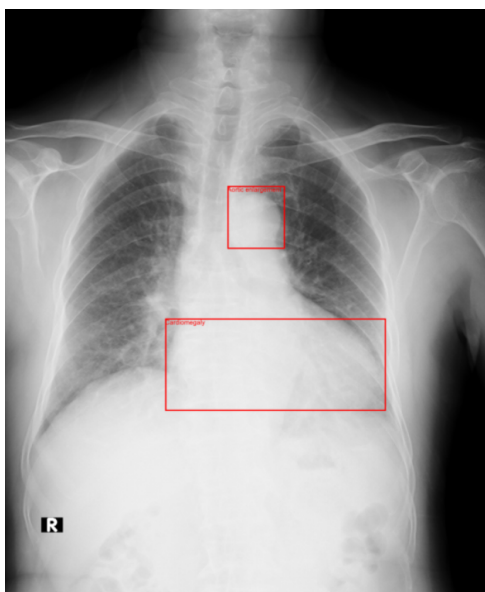


c Predicción resultado del ensamble.

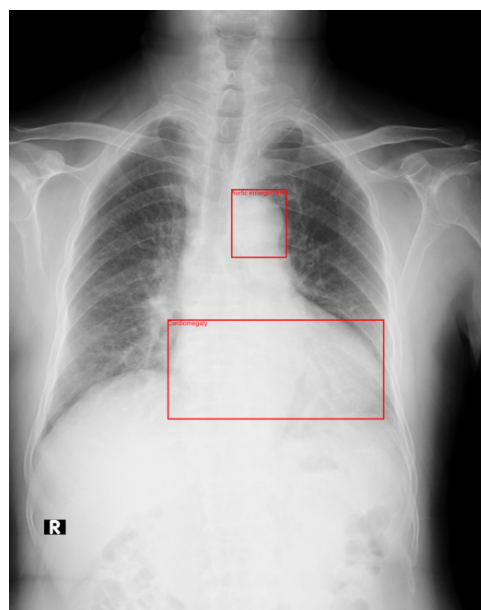


d Anotaciones reales realizadas por los radiólogos.

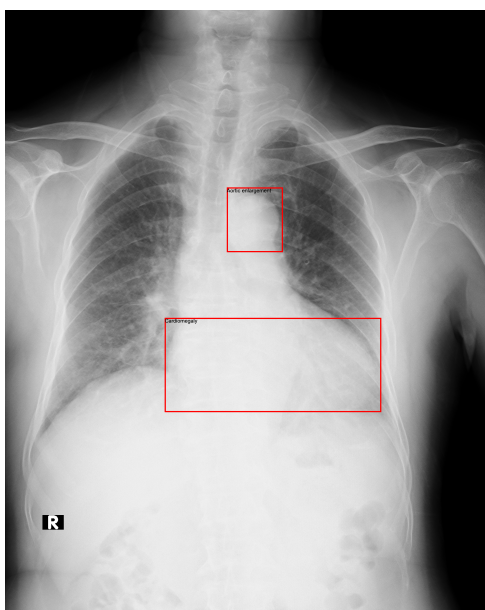
Figura 4.5: Visualización de los resultados obtenidos por los dos modelos entrenados y el ensamble en uno de los resultados obtenidos por el *DETR ResNet-50* con menor *F1-Score*.



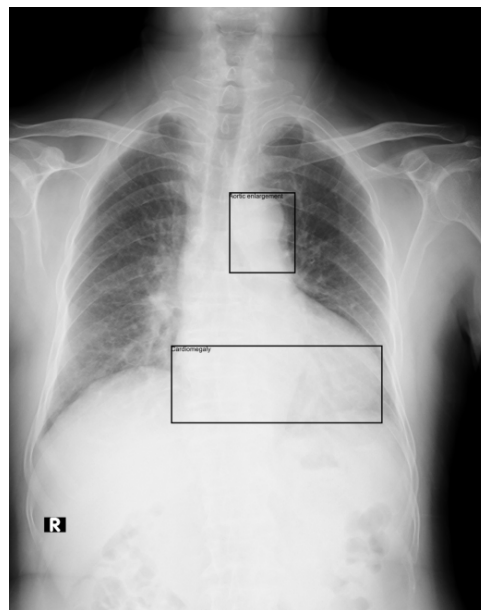
a Predicción realizada por el *DETR ResNet-50*.



b Predicción realizada por el *YOLOS*.

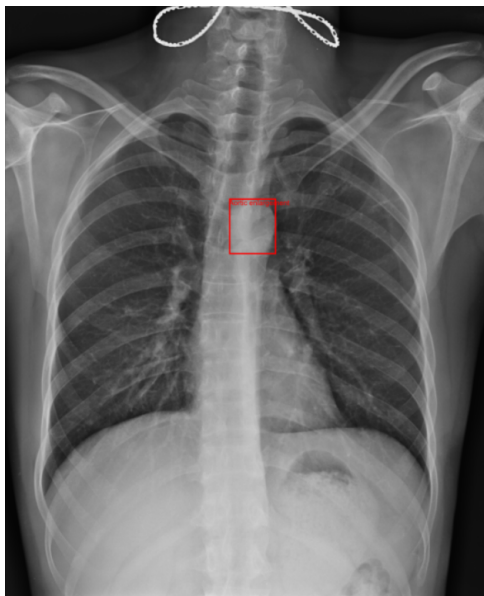


c Predicción resultado del ensamble.

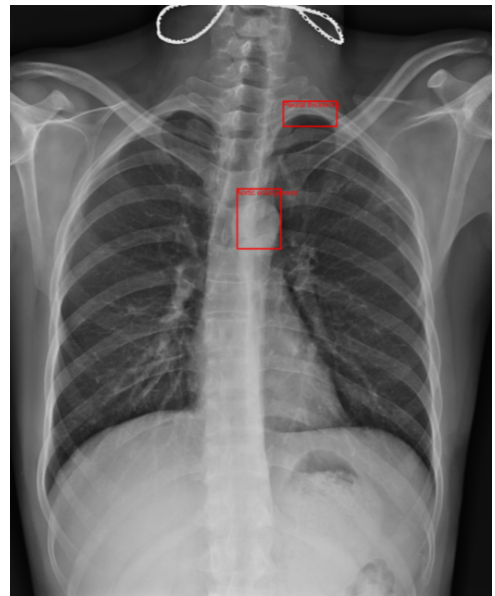


d Anotaciones reales realizadas por los radiólogos.

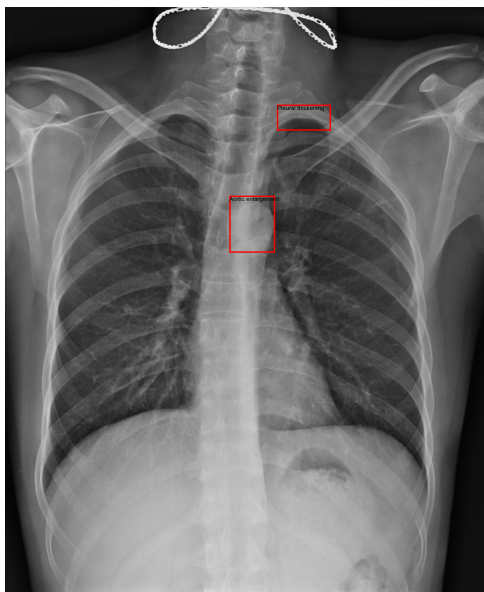
Figura 4.6: Visualización de los resultados obtenidos por los dos modelos entrenados y el ensamble en uno de los resultados obtenidos por el *YOLOS* con mayor *F1-Score*.



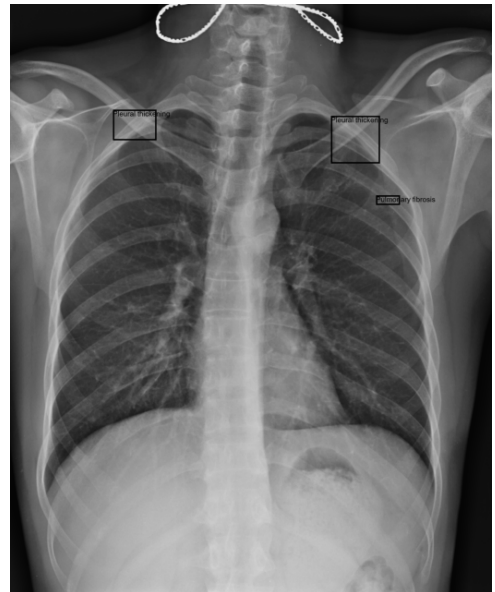
a Predicción realizada por el *DETR ResNet-50*.



b Predicción realizada por el *YOLOS*.

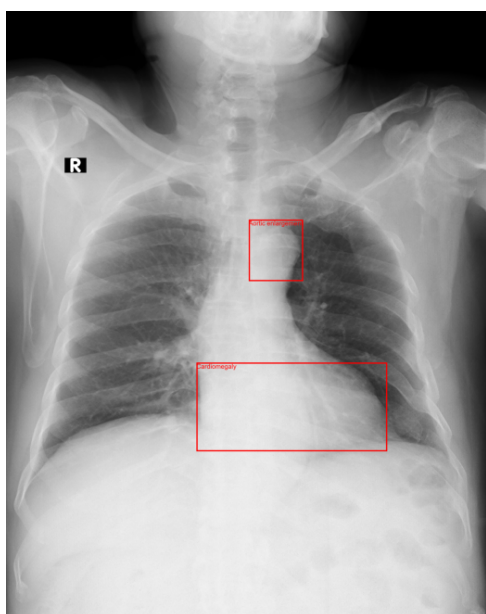


c Predicción resultado del ensamble.

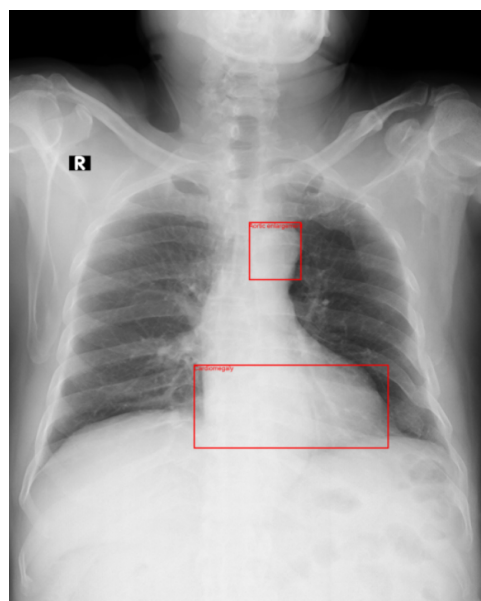


d Anotaciones reales realizadas por los radiólogos.

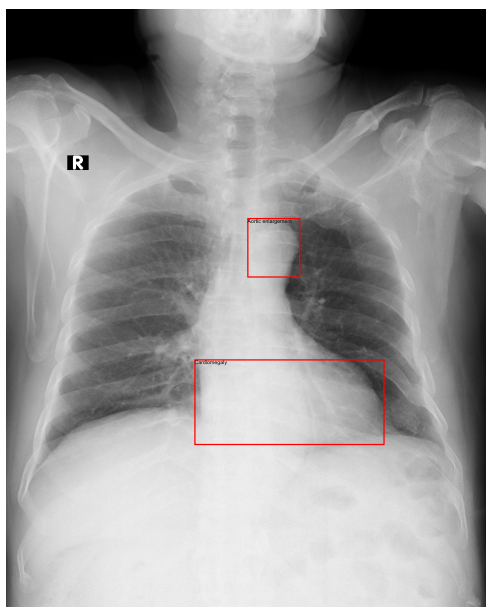
Figura 4.7: Visualización de los resultados obtenidos por los dos modelos entrenados y el ensamble en uno de los resultados obtenidos por el *YOLOS* con menor *F1-Score*.



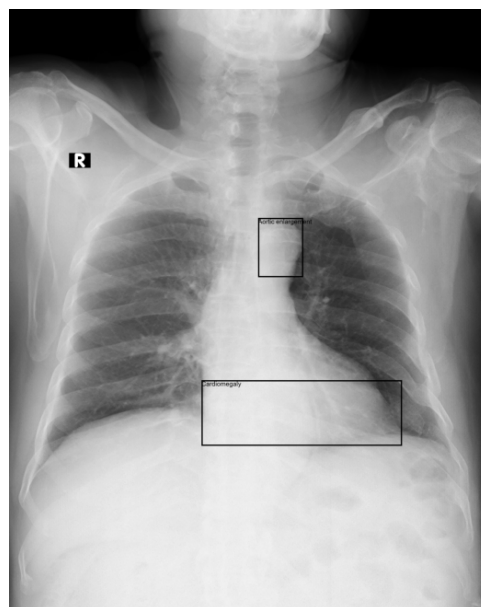
a Predicción realizada por el *DETR ResNet-50*.



b Predicción realizada por el *YOLOS*.

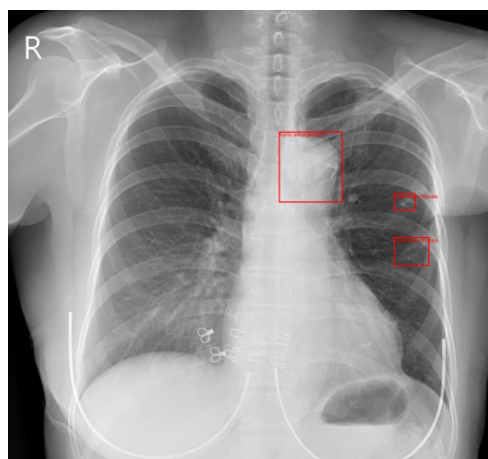


c Predicción resultado del ensamble.

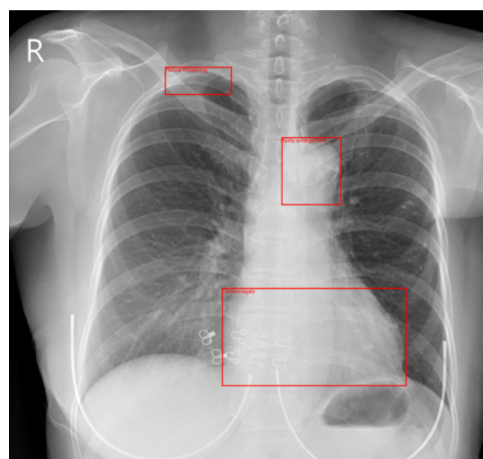


d Anotaciones reales realizadas por los radiólogos.

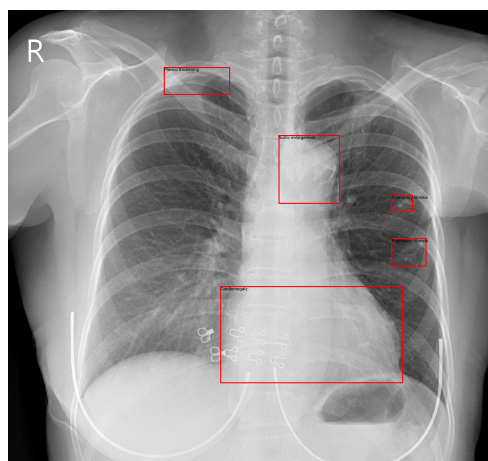
Figura 4.8: Visualización de los resultados obtenidos por los dos modelos entrenados y el ensamble en uno de los resultados obtenidos por el ensamble con mayor *F1-Score*.



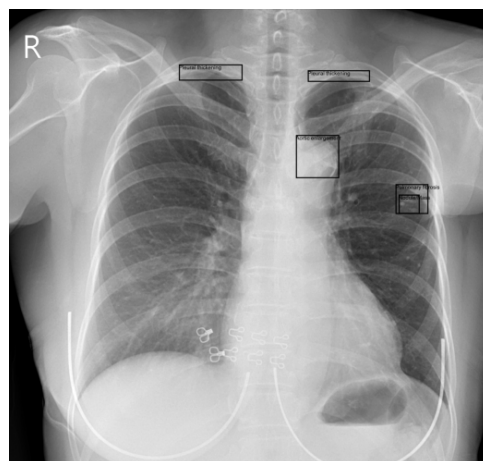
a Predicción realizada por el *DETR ResNet-50*.



b Predicción realizada por el *YOLOS*.



c Predicción resultado del ensamble.



d Anotaciones reales realizadas por los radiólogos.

Figura 4.9: Visualización de los resultados obtenidos por los dos modelos entrenados y el ensamble en uno de los resultados obtenidos por el ensamble con menor *F1-Score*.

Como se puede observar en las figuras 4.4, 4.6 y 4.8, las predicciones con mayor *F1-Score* siempre corresponden a las clases *Aortic enlargement* y *Cardiomegaly*, coincidiendo así con los resultados numéricos del cuadro 4.2, donde se evidencia el buen rendimiento de ambos modelos en estas dos clases. Por otro lado, en los resultados con menor *F1-Score* ambos modelos fallan en detectar alguna de las otras tres clases restantes. En concreto, en la figura 4.5, correspondiente a uno de los resultados obtenidos por el *DETR ResNet-50* con menor *F1-Score*, este no logra detectar ninguna clase, mientras que el *YOLOS* detecta las clases *Aortic enlargement* y *Cardiomegaly*, cuando realmente en las anotaciones correctas encontramos solamente un caso de fibrosis pulmonar o *Pulmonary fibrosis*. Algo similar ocurre en la figura 4.7, correspondiente a una de las predicciones del *YOLOS* con menor *F1-Score*, donde en las anotaciones reales encontramos una instancia de *Pleural thickening* o engrosamiento pleural en cada pulmón, además de fibrosis pulmonar en el pulmón izquierdo. En

este caso, a pesar de que el *YOLOS* logra detectar engrosamiento pleural en uno de los pulmones, no consigue coincidir exactamente con las anotaciones reales en la ubicación de la patología, y además identifica erróneamente una aneurisma aórtica o *Aortic enlargement*, al igual que el *DETR*. Por último, en la figura 4.9, correspondiente a uno de los resultados obtenidos por el ensamble con menor *F1-Score*, nos encontramos que el *DETR* logra detectar la fibrosis pulmonar y la aneurisma aórtica presente en las anotaciones reales, pero también detecta incorrectamente un caso de fibrosis pulmonar que no debería. En cuanto al *YOLOS*, este también localiza esta aneurisma, además de uno de los engrosamientos pleurales, pero detecta equivocadamente una cardiomegalia. Estas detecciones erróneas se acumulan en el ensamble, causando un empeoramiento de los resultados.

Capítulo 5

Conclusiones y trabajo futuro

5.1. Conclusiones

En este Trabajo de Fin de Grado se ha desarrollado un sistema de detección automática de anomalías en radiografías de tórax utilizando técnicas avanzadas de ciencia de datos e inteligencia artificial.

A pesar de los limitados recursos computacionales, se logró entrenar y validar modelos de detección de objetos con un rendimiento notable. El sistema ha demostrado ser efectivo en la identificación de diversas anomalías, especialmente en la aneurisma aórtica y la cardiomegalia, lo que sugiere que es viable desarrollar herramientas de apoyo al diagnóstico de enfermedades que podrían tener un impacto significativo en la práctica clínica, proporcionando una valiosa segunda opinión a los radiólogos, reduciendo la carga de trabajo y mejorando la precisión diagnóstica en zonas rurales con pocos recursos.

Por otro lado, el uso de un ensamble de modelos ha demostrado ser una estrategia eficaz en el área de la detección de anomalías en radiografías, mejorando los resultados individuales de cada modelo en tres de las cinco clases a detectar.

No obstante, se identificaron algunas limitaciones, como la necesidad de un mayor volumen de datos de entrenamiento para poder mejorar la detección en enfermedades menos representadas en el conjunto de datos disponible.

Personalmente, el desarrollo de este trabajo me ha servido en gran medida tanto para obtener nuevos conocimientos como para afianzar los que ya tenía. He aprendido en profundidad acerca de la detección de objetos y las técnicas, algoritmos y arquitecturas más populares hoy en día. Además, me he familiarizado con aspectos sobre las radiografías y enfermedades torácicas que desconocía, y gracias a este proyecto he conocido herramientas tan útiles como *Hugging Face*, e incluso mejorado en gran medida mis aptitudes de \LaTeX .

Habiendo finalizado el trabajo, se puede volver a los objetivos marcados en la introducción y verificar si han llegado a cumplirse:

- **Comprender la estructura y contenido de la base de datos de radiografía:** Se ha comprendido con éxito el formato de imágenes médicas DICOM, así como la interpretación de las radiografías y las enfermedades representadas en ellas.
- **Implementar estrategias de preprocesamiento y normalización de las imá-**

genes de rayos X: En efecto, se han implementado estrategias como el *Non-maximum suppression*, la redimensión de las imágenes o el *data augmentation*, cumpliendo así este objetivo con éxito.

- **Implementar y ejecutar la técnica de ensamble en los modelos:** Se ha logrado implementar la técnica de ensamble de *Weighted Boxes Fusion* de manera satisfactoria para fusionar las predicciones de ambos modelos.
- **Evaluar el rendimiento del sistema de clasificación:** Se ha evaluado el rendimiento tanto del sistema compuesto por el ensamble como de ambos modelos individualmente, obteniendo resultados detallados que proporcionan un desglose del desempeño de cada componente.
- **Analizar cómo los resultados obtenidos se comparan con los de los modelos individuales:** Se ha construido una tabla para poder comparar los resultados del ensamble con los de los modelos individuales en cada clase para distintos umbrales de IoU, y se han visualizado y analizado los mejores y peores resultados de cada uno.

5.2. Trabajo futuro

Este sistema podría beneficiarse de varias mejoras y extensiones en el futuro. Algunas de las áreas potenciales de desarrollo incluyen:

- **Experimentación con los hiperparámetros de entrenamiento:** Ajustar los hiperparámetros de los modelos podría llevar a mejoras significativas en el rendimiento del sistema. Esto incluye la optimización de parámetros como la tasa de aprendizaje o el tamaño del lote. Además, se podrían explorar técnicas avanzadas de ajuste de hiperparámetros, como el *grid search*, para encontrar las configuraciones óptimas.
- **Aumento del tamaño del conjunto de datos de entrenamiento:** La efectividad de los modelos podría mejorarse significativamente con un mayor volumen de datos de entrenamiento. Adicionalmente, también es importante asegurar que los datos estén balanceados en términos de las clases de enfermedades representadas. Esto ayudará a que el modelo no se sesgue hacia clases más representadas, como ha sucedido en este caso debido a la falta de datos de ciertas clases.
- **Adición de más modelos al ensamble:** El uso de la técnica de ensamble ha demostrado ser una estrategia efectiva, pero se podrían incorporar más modelos de detección de objetos al ensamble, incluyendo arquitecturas más recientes y avanzadas, lo que podría resultar en un sistema más robusto y preciso.

Capítulo 6

Análisis de impacto del trabajo

El desarrollo e implementación del sistema de detección automática de anomalías en radiografías de tórax presentado en este Trabajo de Fin de Grado tiene el potencial de causar un impacto significativo en diferentes ámbitos. En este capítulo se explora cómo este sistema puede mejorar la precisión diagnóstica en el ámbito médico, reducir la desigualdad en el acceso a servicios de salud o incluso servir como herramienta educativa. Además, se discute su contribución a los objetivos de salud y bienestar de la Agenda 2030.

6.1. Impacto general

En una visión general, este trabajo puede tener un potencial impacto en las siguientes áreas:

- **Ámbito médico:** Este sistema puede asistir a los radiólogos a identificar anomalías que podrían pasar desapercibidas, lo que aumentaría la precisión de los diagnósticos. Esto a su vez reduciría la carga de trabajo de los profesionales sanitarios, pudiendo automatizar la detección inicial de las anomalías, y permitiendo que se gestione un mayor número de casos en menos tiempo. Adicionalmente, este sistema puede servir como una valiosa herramienta de apoyo en regiones rurales o con recursos limitados en los que haya escasez de radiólogos, ayudando así a mejorar la calidad de la atención médica en estas zonas.
- **Social:** Al poder mejorar la calidad de la atención médica en ciertas regiones con escasos recursos, este sistema podría potencialmente reducir la desigualdad en el acceso a servicios médicos, contribuyendo a una atención más equitativa y justa, independientemente del nivel económico de los pacientes, y mejorando su calidad de vida.
- **Económico:** Al proporcionar una segunda opinión a los radiólogos y potencialmente mejorar la precisión de los diagnósticos, esto se podría traducir a una reducción de los costos asociados a diagnósticos erróneos y tratamientos incorrectos, optimizándose también así el uso de equipos y personal sanitario.
- **Educación:** Un sistema de estas características podría ser utilizado como herramienta educativa para la formación de radiólogos, ayudando a los estudiantes a

reconocer los patrones que muestran ciertas enfermedades en las radiografías de tórax.

- **Medioambiental:** El entrenamiento e implementación de un sistema como este puede tener un impacto ambiental significativo, debido al alto consumo de energía y recursos computacionales necesarios para entrenar arquitecturas como las *transformers* [75], por lo que es importante tratar de adoptar prácticas de software más eficientes que reduzcan la energía asociada al entrenamiento de estos modelos.

6.2. Objetivos de Desarrollo Sostenible

El desarrollo de un sistema de detección automática de anomalías en radiografías de tórax tiene un potencial impacto en varios Objetivos de Desarrollo Sostenible (ODS) de la Agenda 2030 [76], que se listan a continuación:

- **Salud y Bienestar (Objetivo de Desarrollo 3):** El sistema contribuye directamente a este objetivo de desarrollo, al potencialmente mejorar la capacidad de diagnóstico de enfermedades torácicas. Como se ha mencionado en el apartado anterior, la asistencia al diagnóstico puede reducir la carga sobre los radiólogos y mejorar la atención sanitaria en regiones con recursos limitados. Esto podría llevar a una detección más temprana de enfermedades, mejorando los tratamientos y las tasas de supervivencia.
- **Educación de Calidad (Objetivo de Desarrollo 4):** Alegando a lo dicho anteriormente, este sistema podría ser una excelente herramienta educativa para ayudar en la educación y capacitación de los profesionales de la salud.
- **Reducción de las Desigualdades (Objetivo de Desarrollo 10):** Tal como se señaló previamente, el acceso a herramientas de asistencia al diagnóstico puede reducir las desigualdades en la atención médica, especialmente en áreas rurales o en regiones con escasos recursos médicos, mejorando así la equidad en el acceso a diagnósticos de calidad.

Bibliografía

- [1] *World Health Organization To X-ray or not to X-ray?* <https://www.who.int/news-room/feature-stories/detail/to-x-ray-or-not-to-x-ray->. Accessed: 2024-04-7.
- [2] Lisa Iyeke et al. «Reducing Unnecessary 'Admission' Chest X-rays: An Initiative to Minimize Low-Value Care». En: *Cureus* 14.10 (2022).
- [3] *Radiopaedia.org Chest radiograph*. <https://radiopaedia.org/articles/chest-radiograph>. Accessed: 2024-04-8.
- [4] *Mayo Clinic Chest X-rays*. <https://www.mayoclinic.org/tests-procedures/chest-x-rays/about/pac-20393494>. Accessed: 2024-04-7.
- [5] RG Dreyer et al. «Assessing and comparing chest radiograph interpretation in the Department of Internal Medicine at the University of the Witwatersrand medical school, according to seniority». En: *African journal of thoracic and critical care medicine* 29.1 (2023), págs. 12-17.
- [6] I Satia et al. «Assessing the accuracy and certainty in interpreting chest X-rays in the medical division». En: *Clinical Medicine* 13.4 (2013), págs. 349-352. ISSN: 1470-2118. DOI: <https://doi.org/10.7861/clinmedicine.13-4-349>.
- [7] *Radiopaedia.org Heart failure X-ray*. <https://radiopaedia.org/articles/congestive-cardiac-failure>. Accessed: 2024-04-8.
- [8] *Radiopaedia.org Pneumonia X-ray*. <https://radiopaedia.org/cases/pneumonia-right-middle-lobe-1>. Accessed: 2024-04-8.
- [9] Chunli Qin et al. «Computer-aided detection in chest radiography based on artificial intelligence: a survey». En: *Biomedical engineering online* 17 (2018), págs. 1-23.
- [10] *Kaggle.com VinBigData Chest X-ray Abnormalities Detection*. <https://www.kaggle.com/competitions/vinbigdata-chest-xray-abnormalities-detection/data>. Accessed: 2024-04-8.
- [11] Pranav Rajpurkar et al. «Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning». En: *arXiv preprint arXiv:1711.05225* (2017).
- [12] Yu-Xing Tang et al. «Automated abnormality classification of chest radiographs using deep convolutional neural networks». En: *NPJ digital medicine* 3.1 (2020), pág. 70.
- [13] Gwilym S Lodwick, Theodore E Keats y John P Dorst. «The coding of roentgen images for computer analysis as applied to lung cancer». En: *Radiology* 81.2 (1963), págs. 185-200.
- [14] Heyang Huang, Yijun Long y Yi Wei. «Chest X-ray Abnormalities Detection». En: 0.

-
- [15] Maider Abad, Jordi Casas-Roma y Ferran Prados. «Generalizable disease detection using model ensemble on chest X-ray images». En: *Scientific Reports* 14.1 (2024), pág. 5890.
- [16] Gautam Kunapuli. *Ensemble methods for machine learning*. Simon y Schuster, 2023.
- [17] *MATLAB & Simulink What is Object Detection?* <https://www.mathworks.com/discovery/object-detection.html>. Accessed: 2024-04-9.
- [18] *Don't Repeat Yourself Object Detection with YOLO using PyTorch*. <https://dontrepeatyourself.org/post/object-detection-with-yolo-using-pytorch/>. Accessed: 2024-04-9.
- [19] *IBM What are Convolutional Neural Networks?* <https://www.ibm.com/topics/convolutional-neural-networks>. Accessed: 2024-04-9.
- [20] Ahmad Alsaleh y Cahit Perkgoz. «A space and time efficient convolutional neural network for age group estimation from facial images». En: *PeerJ Computer Science* 9 (2023), e1395.
- [21] *Medium Understanding "convolution" operations in CNN*. <https://medium.com/analytics-vidhya/understanding-convolution-operations-in-cnn-1914045816d4>. Accessed: 2024-04-10.
- [22] Hala Nelson. *Essential Math for AI*. "Reilly Media, Inc.", 2023.
- [23] Salman Khan et al. «Transformers in vision: A survey». En: *ACM computing surveys (CSUR)* 54.10s (2022), págs. 1-41.
- [24] *cohere.com What Are Transformer Models and How Do They Work?* <https://cohere.com/blog/what-are-transformer-models>. Accessed: 2024-04-20.
- [25] *Hugging Face facebook/detr-resnet-50*. <https://huggingface.co/facebook/detr-resnet-50>. Accessed: 2024-04-9.
- [26] *COCO Common Objects in Context*. <https://cocodataset.org>. Accessed: 2024-05-18.
- [27] *plugger.ai Enhancing Object Detection with Facebook DETR ResNet50: A Powerful AI Model*. <https://www.plugger.ai/models/facebook-detr-resnet50>. Accessed: 2024-04-9.
- [28] *Medium Skip Connection and Explanation of ResNet*. <https://chautuankien.medium.com/skip-connection-and-explanation-of-resnet-afabe792346c>. Accessed: 2024-04-9.
- [29] Nicolas Carion et al. «End-to-end object detection with transformers». En: *European conference on computer vision*. Springer. 2020, págs. 213-229.
- [30] Vivek Kalyan *End-to-End Object Detection with Transformers*. <https://www.vivekkalyan.com/end-to-end-object-detection-with-transformers>. Accessed: 2024-04-9.
- [31] *Machine Learning Mastery A Gentle Introduction to Cross-Entropy for Machine Learning*. <https://machinelearningmastery.com/cross-entropy-for-machine-learning/>. Accessed: 2024-05-27.
- [32] Yuxin Fang et al. «You only look at one sequence: Rethinking transformer in vision through object detection». En: *Advances in Neural Information Processing Systems* 34 (2021), págs. 26183-26197.
- [33] *paperswithcode.com ImageNet-1K*. <https://paperswithcode.com/dataset/imagenet-1k-1>. Accessed: 2024-05-13.
- [34] *Hugging Face hustvl/yolos-base*. <https://huggingface.co/hustvl/yolos-base>. Accessed: 2024-05-13.

- [35] *pyimagesearch Intersection over Union (IoU) for object detection*. <https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>. Accessed: 2024-05-23.
- [36] Hamid Rezaatofghi et al. «Generalized Intersection over Union». En: (jun. de 2019).
- [37] *evidentlyai.com Accuracy vs. precision vs. recall in machine learning: what's the difference?* <https://www.evidentlyai.com/classification-metrics/accuracy-precision-recall>. Accessed: 2024-04-10.
- [38] *Scikit Learn Metrics: F1-Score*. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html. Accessed: 2024-05-20.
- [39] *kili-technology.com Mean Average Precision (mAP): A Complete Guide*. <https://kili-technology.com/data-labeling/machine-learning/mean-average-precision-map-a-complete-guide>. Accessed: 2024-04-10.
- [40] *DataCamp Precision-Recall Curve in Python Tutorial*. <https://www.datacamp.com/tutorial/precision-recall-curve-tutorial>. Accessed: 2024-04-10.
- [41] *Testing with Kolena Average Precision*. <https://docs.kolena.com/metrics/average-precision>. Accessed: 2024-05-23.
- [42] *Medium.com Breaking Down Mean Average Precision (mAP)*. <https://towardsdatascience.com/breaking-down-mean-average-precision-map-ae462f623a52>. Accessed: 2024-04-10.
- [43] *Medium.com Non Max Suppression (NMS)*. <https://medium.com/analytics-vidhya/non-max-suppression-nms-6623e6572536>. Accessed: 2024-05-06.
- [44] Jan Hosang, Rodrigo Benenson y Bernt Schiele. «Learning non-maximum suppression». En: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, págs. 4507-4515.
- [45] *LearnOpenCV.com Non Maximum Suppression: Theory and Implementation in PyTorch*. <https://learnopencv.com/non-maximum-suppression-theory-and-implementation-in-pytorch/>. Accessed: 2024-05-07.
- [46] Ammar Mohammed y Rania Kora. «A comprehensive review on ensemble deep learning: Opportunities and challenges». En: *Journal of King Saud University-Computer and Information Sciences* 35.2 (2023), págs. 757-774.
- [47] Roman Solovyev, Weimin Wang y Tatiana Gabruseva. «Weighted boxes fusion: Ensembling boxes from different object detection models». En: *Image and Vision Computing* 107 (2021), pág. 104117.
- [48] *heart.org Your Aorta: The Pulse of Life*. <https://www.heart.org/en/health-topics/aortic-aneurysm/your-aorta-the-pulse-of-life>. Accessed: 2024-05-06.
- [49] *Radiopaedia.org Thoracic aortic aneurysm*. <https://radiopaedia.org/articles/thoracic-aortic-aneurysm>. Accessed: 2024-05-07.
- [50] Hina Amin y Waqas J Siddiqui. «Cardiomegaly». En: (2019).
- [51] *Radiopaedia.org Cardiomegaly*. <https://radiopaedia.org/cases/cardiomegaly-3>. Accessed: 2024-05-07.
- [52] *Mesothelioma.com Pleural thickening*. <https://www.mesothelioma.com/asbestos-cancer/pleural-thickening/>. Accessed: 2024-05-06.
- [53] Akira Saito et al. «Pleural thickening on screening chest X-rays: a single institutional study». En: *Respiratory research* 20 (2019), págs. 1-7.

- [54] *Pleural Mesothelioma Center Pleural Thickening and Asbestos*. <https://www.pleuralmesothelioma.com/asbestos/related-diseases/pleural-thickening/>. Accessed: 2024-05-07.
- [55] *mayoclinic.org Pulmonary fibrosis*. <https://www.mayoclinic.org/diseases-conditions/pulmonary-fibrosis/symptoms-causes/syc-20353690>. Accessed: 2024-05-06.
- [56] *Radiopaedia.org Pulmonary fibrosis*. <https://radiopaedia.org/articles/pulmonary-fibrosis>. Accessed: 2024-05-07.
- [57] *American Cancer Society Lung Nodules*. <https://www.cancer.org/cancer/types/lung-cancer/detection-diagnosis-staging/lung-nodules.html>. Accessed: 2024-05-06.
- [58] *Cleveland Clinic Pulmonary Nodules*. <https://my.clevelandclinic.org/health/diseases/14799-pulmonary-nodules>. Accessed: 2024-05-06.
- [59] *Radiopaedia.org Solitary pulmonary nodule (an approach)*. <https://radiopaedia.org/articles/solitary-pulmonary-nodule-an-approach>. Accessed: 2024-05-07.
- [60] William Pezzotti. «Interpretación de la radiografía de tórax». En: *Nursing* (2014).
- [61] *Radiology Key The Normal Chest Image, Technical Considerations*. <https://radiologykey.com/chest-11/>. Accessed: 2024-05-06.
- [62] Ha Q Nguyen et al. «VinDr-CXR: An open dataset of chest X-rays with radiologist's annotations». En: *Scientific Data* 9.1 (2022), pág. 429.
- [63] *PyPI.org Pillow*. <https://pypi.org/project/pillow/>. Accessed: 2024-05-20.
- [64] *Innolitics Photometric Interpretation Attribute*. <https://dicom.innolitics.com/ciods/rt-dose/image-pixel/00280004>. Accessed: 2024-05-11.
- [65] *Albumentations Documentation Bounding boxes augmentation for object detection*. https://albumentations.ai/docs/getting_started/bounding_boxes_augmentation/. Accessed: 2024-05-11.
- [66] *Hugging Face Computer vision task guides: Object detection*. https://huggingface.co/docs/transformers/tasks/object_detection. Accessed: 2024-05-10.
- [67] Tahira Shehzadi et al. «2d object detection with transformers: a review». En: *arXiv preprint arXiv:2306.04670* (2023).
- [68] Endang Suherman et al. «Implementation of ResNet-50 on End-to-End Object Detection (DETR) on Objects». En: *Sinkron: jurnal dan penelitian teknik informatika* 8.2 (2023), págs. 1085-1096.
- [69] *Hugging Face Transformers: Training Arguments*. https://huggingface.co/docs/transformers/v4.40.2/en/main_classes/trainer. Accessed: 2024-05-13.
- [70] Diederik P Kingma y Jimmy Ba. «Adam: A method for stochastic optimization». En: *arXiv preprint arXiv:1412.6980* (2014).
- [71] *GitHub ZFTurbo/Weighted-Boxes-Fusion*. <https://github.com/ZFTurbo/Weighted-Boxes-Fusion>. Accessed: 2024-05-15.
- [72] *Hugging Face hustvl/yolos-tiny*. <https://huggingface.co/hustvl/yolos-tiny>. Accessed: 2024-05-13.
- [73] *COCO Detection Evaluation*. <https://cocodataset.org/#detection-eval>. Accessed: 2024-05-16.
- [74] *Analytics Vidhya Measuring Success: Comprehensive Evaluation Metrics for Object Detection Models*. <https://www.analyticsvidhya.com/blog/2024/02/>

BIBLIOGRAFÍA

- evaluation-matrix-for-object-detection-using-iou-and-map/. Accessed: 2024-05-16.
- [75] Emma Strubell, Ananya Ganesh y Andrew McCallum. «Energy and policy considerations for deep learning in NLP». En: *arXiv preprint arXiv:1906.02243* (2019).
- [76] *Naciones Unidas Objetivos de Desarrollo Sostenible*. <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>. Accessed: 2024-05-27.

ORIGINALITY REPORT

17%

SIMILARITY INDEX

14%

INTERNET SOURCES

6%

PUBLICATIONS

10%

STUDENT PAPERS

PRIMARY SOURCES

1	hdl.handle.net Internet Source	2%
2	Submitted to Universidad Politécnica de Madrid Student Paper	1%
3	Submitted to Universitat Politècnica de València Student Paper	<1%
4	Submitted to Universidad Rey Juan Carlos Student Paper	<1%
5	oa.upm.es Internet Source	<1%
6	diagnosticar-ips.com Internet Source	<1%
7	riunet.upv.es Internet Source	<1%
8	Submitted to University of South Australia Student Paper	<1%
9	www.coursehero.com	

Internet Source

<1 %

10

export.arxiv.org

Internet Source

<1 %

11

uvadoc.uva.es

Internet Source

<1 %

12

Submitted to Universidad de Málaga - Tii

Student Paper

<1 %

13

Submitted to University of Liverpool

Student Paper

<1 %

14

openarchive.usn.no

Internet Source

<1 %

15

drpress.org

Internet Source

<1 %

16

Submitted to Universidad de Alicante

Student Paper

<1 %

17

huggingface.co

Internet Source

<1 %

18

www.mayoclinic.org

Internet Source

<1 %

19

Junhua Luo, Shujing Wang, Qixiang Wang, Shaojun Liu. "A Lung Lesion Detection Algorithm Based on YOLOv7 and Self-

<1 %

Attention Mechanism", 2023 42nd Chinese Control Conference (CCC), 2023

Publication

20 Submitted to Universidad de Valladolid <1 %
Student Paper

21 exploredoc.com <1 %
Internet Source

22 www.igi-global.com <1 %
Internet Source

23 Submitted to Indian Institute of Information Technology, Lucknow <1 %
Student Paper

24 ouci.dntb.gov.ua <1 %
Internet Source

25 Submitted to University of Greenwich <1 %
Student Paper

26 vjol.info.vn <1 %
Internet Source

27 Submitted to Politecnico di Milano <1 %
Student Paper

28 Submitted to University of Sydney <1 %
Student Paper

29 orbilu.uni.lu <1 %
Internet Source

Submitted to University of Cyprus

30	Student Paper	<1 %
31	Submitted to Charles University Student Paper	<1 %
32	Submitted to Gazi University Student Paper	<1 %
33	Submitted to Universidad Internacional de la Rioja Student Paper	<1 %
34	Submitted to Universitas Bengkulu Student Paper	<1 %
35	Submitted to Visvesvaraya National Institute of Technology Student Paper	<1 %
36	redes-linux.all-inone.net Internet Source	<1 %
37	Submitted to Loughborough University Student Paper	<1 %
38	dspace.mit.edu Internet Source	<1 %
39	pure.tue.nl Internet Source	<1 %
40	link.springer.com Internet Source	<1 %

41	prezi.com Internet Source	<1 %
42	Submitted to Cardiff University Student Paper	<1 %
43	Sounak Mondal, Suraj Modi, Sakshi Garg, Dhruva Das, Siddhartha Mukherjee. "ICAN: Introspective Convolutional Attention Network for Semantic Text Classification", 2020 IEEE 14th International Conference on Semantic Computing (ICSC), 2020 Publication	<1 %
44	Submitted to Unviersidad de Granada Student Paper	<1 %
45	repositorio.unbosque.edu.co Internet Source	<1 %
46	zwoogle3.informatik.hs-kl.de Internet Source	<1 %
47	Submitted to Queensland University of Technology Student Paper	<1 %
48	Submitted to University of Nottingham Student Paper	<1 %
49	arxiv.org Internet Source	<1 %
50	repositorio.uam.es Internet Source	<1 %

<1 %

51

riull.ull.es

Internet Source

<1 %

52

Lazar Dašić, Ognjen Pavić, Tijana Geroski, Dragan Milovanović, Marina Petrović, Nenad Filipović. "Transfer Learning with Deep Convolutional Neural Networks for Respiratory Disease Classification in X-Ray Images", 2023 IEEE 23rd International Conference on Bioinformatics and Bioengineering (BIBE), 2023

Publication

<1 %

53

Submitted to Universidad Anahuac México Sur

Student Paper

<1 %

54

Submitted to Universidad TecMilenio

Student Paper

<1 %

55

es.scribd.com

Internet Source

<1 %

56

iris.polito.it

Internet Source

<1 %

57

www.mdpi.com

Internet Source

<1 %

58

www.researchgate.net

Internet Source

<1 %

59	www.slideshare.net Internet Source	<1 %
60	Submitted to Monash University Student Paper	<1 %
61	chnola.staywellsolutionsonline.com Internet Source	<1 %
62	creativecommons.org Internet Source	<1 %
63	pa.bibdigital.ucc.edu.ar Internet Source	<1 %
64	web.archive.org Internet Source	<1 %
65	D. Ferrando Blanco, Ó. Persiva Morenza, L.B. Cabanzo Campos, A.L. Sánchez Martínez et al. "Utility of artificial intelligence for detection of pneumothorax on chest radiopgraphs done after transthoracic percutaneous transthoracic biopsy guided by computed tomography", Radiología (English Edition), 2024 Publication	<1 %
66	Submitted to Istanbul Gelisim University Student Paper	<1 %
67	Submitted to The Open University of Hong Kong Student Paper	<1 %

68

acikerisim.sakarya.edu.tr

Internet Source

<1 %

69

developer.android.com

Internet Source

<1 %

70

www.aiplusinfo.com

Internet Source

<1 %

71

dspace.ut.ee

Internet Source

<1 %

72

etd.aau.edu.et

Internet Source

<1 %

73

[Submitted to Canterbury Christ Church University](#)

Student Paper

<1 %

74

[Submitted to Ilerna Online](#)

Student Paper

<1 %

75

[Submitted to Infile](#)

Student Paper

<1 %

76

[Submitted to Universidad Nacional de Educación a Distancia](#)

Student Paper

<1 %

77

[Submitted to Universidad de Chile](#)

Student Paper

<1 %

78

[Submitted to University of Glamorgan](#)

Student Paper

<1 %

79	article.auone.jp Internet Source	<1 %
80	patientnavigatortraining.org Internet Source	<1 %
81	Submitted to Universidad Catolica San Antonio de Murcia Student Paper	<1 %
82	Wajahat Saeed, Muhammad Sohail Saleh, Muhammad Naqash Gull, Hassan Raza, Rafaqat Saeed, Tahira Shehzadi. "Geometric features and traffic dynamic analysis on 4-leg intersections", International Review of Applied Sciences and Engineering, 2023 Publication	<1 %
83	bibdigital.epn.edu.ec Internet Source	<1 %
84	mx.php.net Internet Source	<1 %
85	www.math.cinvestav.mx Internet Source	<1 %
86	www.silared.com Internet Source	<1 %
87	Neeraj Menon, Pooja Yadav, Vinayakumar Ravi, Vasundhara Acharya, V Sowmya. "Unsupervised generative learning-based	<1 %

decision-making system for COVID-19
detection", Health and Technology, 2024

Publication

88

Submitted to Swinburne University of
Technology

Student Paper

<1 %

89

Sánchez Salguero, Xavier, Universitat
Autònoma de Barcelona. Departament de
Medicina i Cirurgia Animals. "Nuevas medidas
radiográficas para la detección del
incremento del atrio izquierdo en perros y
nuevo método para la obtención del Vertebral
Heart Scale", [Barcelona] : Universitat
Autònoma de Barcelona,, 2013

Internet Source

<1 %

90

Submitted to University of Canterbury

Student Paper

<1 %

91

juansensio.com

Internet Source

<1 %

92

riaa.uaem.mx

Internet Source

<1 %

93

serwiss.bib.hs-hannover.de

Internet Source

<1 %

94

(3-13-16)

http://150.214.191.180/Documentos/tesis_dpto/139.pdf

Internet Source

<1 %

95	Submitted to ITESM: Instituto Tecnológico y de Estudios Superiores de Monterrey Student Paper	<1 %
96	docplayer.es Internet Source	<1 %
97	isc.utp.edu.co Internet Source	<1 %
98	portalinvestigacion.consorciomadrono.es Internet Source	<1 %
99	repositorio.unal.edu.co Internet Source	<1 %
100	www.ijeei.org Internet Source	<1 %
101	www.uca.edu.sv Internet Source	<1 %
102	celiacosdemexico.org.mx Internet Source	<1 %
103	ciateq.repositorioinstitucional.mx Internet Source	<1 %
104	de.slideshare.net Internet Source	<1 %
105	enciclonetdidactica.blogspot.com Internet Source	<1 %
106	Submitted to imfice	

<1 %

107 issuu.com
Internet Source

<1 %

108 munin.uit.no
Internet Source

<1 %

109 revistas.unal.edu.co
Internet Source

<1 %

110 www.amboss.com
Internet Source

<1 %

111 www.cochrane.org
Internet Source

<1 %

112 www.frontiersin.org
Internet Source

<1 %

113 www.hospitalbaciliogomez.com
Internet Source

<1 %

114 www.ibm.com
Internet Source

<1 %

115 (9-26-14)
<http://189.2.156.229/pesquisa/download/cadernos/PRC>
Internet Source

<1 %

116 MARTA SOLIVA SÁNCHEZ. "Estudio de filigranas en documentos administrativos del s.XVIII en el Archivo Histórico Municipal de Alicante. Análisis morfológico y puesta en

<1 %

valor.", 'Universitat Politecnica de Valencia',
2017

Internet Source

117 Mary Ann Torres, Sofia Gruskin, Kent Buse, Taavi Erkkola, Victoria Bendaud, Tobias Alfvén. "Monitoring HIV-Related Laws and Policies: Lessons for AIDS and Global Health in Agenda 2030", AIDS and Behavior, 2017
Publication

118 es.wikipedia.org
Internet Source

119 ia904706.us.archive.org
Internet Source

120 ntnuopen.ntnu.no
Internet Source

121 openaccess.uoc.edu
Internet Source

122 pesquisa.bvsalud.org
Internet Source

123 quirodoc.com
Internet Source

124 radiopaedia.org
Internet Source

125 repositorio.upch.edu.pe
Internet Source

126	repository.udistrital.edu.co Internet Source	<1 %
127	www.adobe.com.mx Internet Source	<1 %
128	www.repositorio.usac.edu.gt Internet Source	<1 %
129	www.uia.mx Internet Source	<1 %
130	"Utilización de micelio en el desarrollo de un sistema constructivo: componentes compuestos para la edificación modular de vivienda colectiva", Pontificia Universidad Católica de Chile, 2024 Publication	<1 %
131	89douner.tistory.com Internet Source	<1 %
132	cicese.repositorioinstitucional.mx Internet Source	<1 %
133	e-spacio.uned.es Internet Source	<1 %
134	frontdb.ugr.es Internet Source	<1 %
135	idus.us.es Internet Source	<1 %

136	investigaciones.uniatlantico.edu.co Internet Source	<1 %
137	moam.info Internet Source	<1 %
138	patents.google.com Internet Source	<1 %
139	pdffox.com Internet Source	<1 %
140	redicces.org.sv Internet Source	<1 %
141	repositorio.uan.edu.co Internet Source	<1 %
142	repositorio.ug.edu.ec Internet Source	<1 %
143	ri-ng.uaq.mx Internet Source	<1 %
144	sedici.unlp.edu.ar Internet Source	<1 %
145	tesisnred.net Internet Source	<1 %
146	web-newslist.com Internet Source	<1 %
147	www.analesderadiologiamexico.com Internet Source	<1 %

148	www.bdigital.unal.edu.co Internet Source	<1 %
149	www.dropbox.com Internet Source	<1 %
150	www.elconfidencial.com Internet Source	<1 %
151	www.issea.gob.mx Internet Source	<1 %
152	www.tamps.cinvestav.mx Internet Source	<1 %
153	www.ticbeat.com Internet Source	<1 %
154	Ana Maria Jimenez Pastor. "Aprendizaje profundo y biomarcadores de imagen en el estudio de enfermedades metabólicas y hepáticas a partir de resonancia magnética y tomografía computarizada", Universitat Politecnica de Valencia, 2023 Publication	<1 %
155	Erdi Çallı, Ecem Sogancioglu, Bram van Ginneken, Kicky G. van Leeuwen, Keelin Murphy. "Deep learning for chest X-ray analysis: A survey", Medical Image Analysis, 2021 Publication	<1 %

aws.amazon.com

156

Internet Source

<1 %

157

www.cacic2016.unsl.edu.ar

Internet Source


<1 %

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off

Este documento esta firmado por

	Firmante	CN=tfgm.fi.upm.es, OU=CCFI, O=ETS Ingenieros Informaticos - UPM, C=ES
	Fecha/Hora	Mon Jun 03 18:07:08 CEST 2024
	Emisor del Certificado	EMAILADDRESS=camanager@etsiinf.upm.es, CN=CA ETS Ingenieros Informaticos, O=ETS Ingenieros Informaticos - UPM, C=ES
	Numero de Serie	561
	Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)