



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN CIENCIAS MATEMÁTICAS Y
DE LA ESPECIALIZACIÓN EN ESTADÍSTICA APLICADA

**Aprendizaje profundo para la identificación y localización de
daños por COVID19 en radiografías de rayos X.**

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN CIENCIAS

PRESENTA:
JONATHAN JAIR SÁNCHEZ CONTRERAS

DIRECTOR
GIBRAN FUENTES PINEDA IIMAS

CIUDAD DE MÉXICO FECHA.

Dedicatoria

Soy la convergencia de valiosas aportaciones provenientes de una infinidad de personas que han marcado mi vida con sus enseñanzas, consejos, amor, confianza, entre muchas otras cualidades que han influido en mi visión de la vida, en quién soy y en lo que creo. Siempre estarán presentes en mí, ustedes saben quienes son y lo que significan para mi. Este trabajo lo dedico a cada uno de ustedes.

En palabras de uno de los más grandes científicos de la historia:

"Si he logrado ver más lejos, ha sido porque he subido a hombros de gigantes".
Sir Isaac Newton (1643-1727)

Agradecimientos

Quiero expresar mis más sinceros agradecimientos al Dr. Gibran Fuentes Pineda, quien fungió como mi asesor, por su tiempo, dedicación, consejos, guía y calidad humana. Por la oportunidad de desarrollar el presente trabajo, el cual ha sido crucial para mi crecimiento profesional, personal e intelectual. También, extiendo mi agradecimiento a mis mentores y al Instituto de Matemáticas de la UNAM, quienes me brindaron los conocimientos y herramientas necesarios para entender y desarrollar esta tesis, así como para mi vida profesional diaria.

Al Dr. Adan Oswaldo Guerrero Cardenas, quien tiene mi total gratitud, ya que ha sido una de las figuras más influyentes en mi vida. No solo como el excelente científico que es, sino también por su calidad humana. Gracias por inspirarme, por las pláticas, los consejos, las experiencias y la excelente formación. Aprecio la oportunidad de desarrollarme como científico, por alentarme a mejorar y aprender constantemente, por mostrarme que se puede hacer de la ciencia un estilo de vida y ser un medio transformador de ideas. Agradezco la confianza inicial y la calidez con la que fui recibido como miembro del Laboratorio Nacional de Microscopía Avanzada. Tu apoyo fue fundamental para el desarrollo del presente trabajo.

Agradezco al Dr. Christopher David Wood, encargado del Laboratorio Nacional de Microscopía Avanzada del Instituto de Biotecnología UNAM, por su apoyo incondicional como miembro del laboratorio, actividad que me permitió concluir mi tesis.

Al financiamiento de la *Chan Zuckerberg Initiative* por hacer posible mi estancia de investigación en el Laboratorio Nacional de Microscopía Avanzada, lo cual fue fundamental para mi desarrollo y el de este trabajo así como la participación en otros proyectos ajenos.

También, agradezco a todos los compañeros del laboratorio, con quienes compartí grandes momentos y quienes tuvieron una influencia profesional y personal enorme.

A CONACYT, agradezco la beca de posgrado que me permitió concluir los créditos del programa de maestría.

Al Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica (PA-PIIT) IV100420 de la UNAM que impulso el desarrollo de la presente investigación.

Finalmente, un agradecimiento especial a mi familia, a mi mamá por ser la primera persona en apoyarme, amarme y creer en mí incondicionalmente, a mis amigos, quienes han sido una fuente incondicional de cariño, apoyo e inspiración.

Resumen

En un modelo para la localización y detección de objetos, la métrica mAP es de principal interés, ya que indica la capacidad de un modelo para identificar objetos de distintas clases presentes en una imagen. En el caso de la identificación y clasificación de daños en pulmones causados por COVID19, es relevante buscar la métrica mAP más alta posible, lo que se traduce en un modelo más confiable.

En la presente investigación se propone una metodología robusta que busca mejorar la métrica mAP para esta tarea, a través de la exploración de mejoras relativas a los parámetros utilizados en el entrenamiento, las anotaciones en los datos, el aprendizaje modular y la modificación de la arquitectura. Esto permite tener en cuenta muchos de los aspectos de mejora en el marco de trabajo del aprendizaje profundo. Lo anterior se aplica a los modelos de visión computacional YOLO y RetinaNet, lo que permite una comparación de la metodología propuesta en uno de los modelos más utilizados para esta tarea y en uno de los menos explorados.

De esta forma, se busca medir el impacto que cada una de las propuestas presentadas dentro de la metodología tiene sobre el desempeño de la métrica mAP en estos dos modelos, estableciendo un marco de trabajo que puede ser evaluado en otras tareas y modelos distintos a los presentados.

APRENDIZAJE PROFUNDO PARA LA
IDENTIFICACIÓN Y LOCALIZACIÓN DE
DAÑOS POR COVID19 EN
RADIOGRAFÍAS DE RAYOS X.

Índice general

Índice general	1
Introducción	1
Objetivos	2
Hipótesis	2
Justificación	3
1 Introducción histórica	4
1.1 Maria Skłodowska	4
1.2 Alan Turing	5
1.3 Wilhelm Röntgen	7
1.4 John McCarthy	7
2 Conceptos básicos	9
2.1 Biología y Medicina	9
2.1.1 Sistema respiratorio	9
Pulmón	9
Bronquios	9
Bronquiolos	9
Alvéolos	10
Pleura	10
2.1.2 Conceptos biomedicos	10
ADN	10
ARN	11
RT-PCR	11
Coronavirus	12
2.1.2.1 Opacidades	12
Consolidación	13
Derrame pleural	13

Neumonía	13
2.1.3 Imágenes médicas	14
2.1.3.1 Radiografías de rayos X	14
2.1.3.2 Tomografía computarizada	15
2.1.3.3 Resonancia magnética	15
2.1.3.4 Ecografía	15
2.1.3.5 Estándares de formato y transmisión de imágenes medicas	15
DICOM	15
HL7	16
2.2 Imágenes digitales	16
2.2.1 Estructura de una imagen digital	16
Píxel	16
Profundidad de color	17
Canal de color	18
Imagen digital	18
Interpretación fotométrica	19
2.3 Inteligencia artificial	21
2.3.1 Modelos analíticos vs inteligencia artificial	21
2.3.2 Conceptos clave	23
Conjunto de datos	23
Algoritmo	23
Entrenamiento	23
Modelo	23
Conjunto de entrenamiento	24
Conjunto de validación	24
Conjunto de prueba	24
Visión artificial	24
2.3.3 Tipos de errores	24
Error	25
Sesgo	25
Varianza	25
Ruido	25
2.3.4 Tipos de ajustes	26
Ajuste	26
Subajuste	26
Sobreajuste	26
2.3.5 Clasificación de algoritmos	26
2.3.5.1 Aprendizaje supervisado	26
Clasificación	27

Regresión	27
Segmentación	27
Detección de objetos	27
2.3.5.2 Aprendizaje no supervisado	28
Agrupación	28
Reducción de dimensiones	28
2.3.6 Evaluación de modelos	29
Matriz de confusión	29
Exactitud	30
Precisión	30
Sensibilidad	30
Especificidad	30
Puntaje F1	31
Índice Jaccard	31
Promedio de precisión	31
2.3.7 Redes neuronales artificiales	32
2.3.7.1 Conceptos básicos	32
Red Neuronal	32
Neurona	33
Capa	33
Conexión	34
Peso	34
Sesgo	34
Función de activación	34
Función de pérdida	34
Parámetro	34
Hiperparámetro	34
2.3.7.2 Funciones de activación	36
Identidad	36
Escalón binario	36
Logística, sigmoide o escalón suave	36
Tangente hiperbólica	36
Softplus	36
Gaussiana	36
GELU (Unidad de error gaussiano lineal)	36
ReLU (Unidad linealmente rectificada)	36
ELU (Unidad exponencial lineal)	36
SELU (Unidad exponencial lineal escalada)	37
Leaky-ReLU (Unidad linealmente rectificada modificada)	37

PReLU (Unidad linealmente rectificada parametrizada)	37
SiLU (Unidad lineal sigmoide)	37
Softmax (máximo suave)	37
Máxima salida	37
2.3.7.3 Funciones de pérdida o costo	37
Raíz cuadrada media (RMSE)	38
Error absoluto medio (MAE)	38
Error absoluto medio escalado MASE	38
Entropía cruzada categórica Categorical (Cross-Entropy)	39
Entropía cruzada binaria (Binary Cross-Entropy)	39
2.3.7.4 Optimizadores	40
Descenso estocástico de gradiente (SGD)	41
Descenso estocástico de gradiente con impulso (Momentum)	41
Algoritmo de gradiente adaptativo (AdaGrad)	42
Propagación de raíz cuadrática media (RMSProp)	43
Optimización de momento adaptativo (Adam)	43
AdaDelta	44
2.3.7.5 Retropropagación	45
2.3.7.6 Teoremas de aproximación.	47
2.3.7.7 Transferencia de conocimiento	49
2.3.7.8 Arquitecturas de redes neuronales	51
3 Estado del arte	52
3.1 Aprendizaje profundo para detección y clasificación de objetos	52
3.1.1 YOLO	52
3.1.2 Transformador de visión (Vision transformer)	56
3.1.3 Redes neuronales convolucionales	59
3.1.4 RetinaNet	64
4 Antecedentes	66
4.1 Aprendizaje profundo en detección COVID-19	66
4.1.1 Detección de COVID-19 en imágenes médicas mediante técnicas de aprendizaje profundo	66
4.1.2 Aprendizaje profundo y aprendizaje automático	69
4.1.3 IKONOS	72
4.2 Detección y localización de COVID-19 usando imágenes médicas	73
4.2.1 Red de discriminación y localización de lesiones COVID19	73
4.2.2 Aprendizaje profundo para la identificación y localización de COVID19	74
4.3 Conclusiones sobre los antecedentes	77

5 Materiales	78
5.0.1 Computadora usada	78
5.0.2 Conjuntos de datos utilizados	78
5.0.2.1 Presencia de opacidades	78
5.0.2.2 Análisis exploratorio	79
5.0.2.3 Opacidades Variadas	82
5.0.2.4 Análisis exploratorio	82
5.0.2.5 COVID19	93
5.0.2.6 Análisis exploratorio	95
6 Métodos	103
7 Discusión principal	108
7.0.1 Exploración de parámetros	108
7.0.1.1 Entrenamiento 1	109
7.0.1.2 Entrenamiento 2	111
7.0.1.3 Entrenamiento 3	113
7.0.1.4 Entrenamiento 4	115
7.0.1.5 Entrenamiento 5	117
7.0.1.6 Entrenamiento 6	119
7.0.1.7 Entrenamiento 7	121
7.0.1.8 Entrenamiento 8	123
7.0.1.9 Entrenamiento 9	125
7.0.1.10 Entrenamiento 10	127
7.0.1.11 Entrenamiento 11	129
7.0.1.12 Entrenamiento 12	131
7.0.1.13 Entrenamiento 13	133
7.0.1.14 Entrenamiento 14	135
7.0.1.15 Entrenamiento 15	136
8 Conclusiones	140
9 Repositorio	141
Bibliografía y referencias	142
Bibliografía y referencias	142
Lista de Figuras	149
Lista de Tablas	152

Introducción

El progreso de la tecnología ha proporcionado a la humanidad la capacidad para generar y almacenar grandes volúmenes de información que, al combinarse con disciplinas como las matemáticas, nos permiten el análisis masivo de datos, posibilitando el entendimiento de dinámicas complejas y patrones intrínsecos a los mismos datos. Esto nos da acceso a nuevas respuestas y cuestionamientos, impulsando avances significativos en diversas áreas, incluyendo la investigación científica, la optimización de recursos, la economía, la biología y la medicina, entre otras áreas que son fundamentales para el desarrollo y bienestar de nuestra especie.

En particular, en el campo de la biología y la medicina, los centros de investigación se han preocupado por generar numerosas bases de datos públicas que contienen una gran diversidad de información con anotaciones detalladas sobre distintas condiciones y características. Estas bases de datos se han convertido en valiosos recursos para científicos e investigadores, ya que permiten responder y plantear preguntas de manera precisa y eficiente.

Lo anterior no fue una excepción cuando, el 31 de diciembre de 2019, se notificó el brote del virus SARS-CoV-2, que posteriormente se convirtió en una pandemia, dando lugar a la crisis de salud global más importante de la actualidad [34]. El SARS-CoV-2, causante de la enfermedad infecciosa conocida como COVID19, se caracteriza por provocar problemas respiratorios y extrarespiratorios de gravedad variable, generando daño y dejando estragos principalmente en los pulmones [35].

El surgimiento de esta enfermedad propició la generación de bases de datos de imágenes médicas con anotaciones específicas realizadas por expertos sobre el daño causado por la infección en los pulmones. Dichas bases de datos han sido utilizadas para entrenar modelos en la detección de daño pulmonar haciendo uso de diversas técnicas de aprendizaje profundo, dando forma a los antecedentes que se discuten en el capítulo 3 del presente trabajo. Esta tesis se enfoca en explorar la posibilidad de mejorar el desempeño de los modelos en la detección e identificación de daños pulmonares a través del aprendizaje modular generado por transferencia de conocimiento, así como la variación de etiquetado de datos permitiendo la comparación de distintos modelos, entrenamientos y su correspondiente desempeño.

La mejora en el desempeño de los modelos brindaría a los expertos herramientas tecnológicas confiables y de acceso público que puede servir como auxiliar para la detección de daños causados o remanentes de la infección por SARS-CoV-2, permitiendo corroborar diagnósticos, evaluar posibles daños no detectados por el experto o dar seguimiento a pacientes en recuperación.

Así como la aportación de herramientas tecnológicas al acervo cultural que puede ser de ayuda en caso de un resurgimiento del virus o estudio de daños causados por nuevas variantes.

En caso de no demostrarse una mejora significativa para la detección e identificación de daños en pulmones para los modelos estudiados, con las técnicas propuestas y sus comparativas se estaría aportando un antecedente valioso para futuras investigaciones en la propuesta de nuevos modelos, su entrenamiento y evaluación.

En cualquiera de los dos posibles escenarios, se busca generar un trabajo de autocontenido, que permita entender el trabajo aquí desarrollado desde un enfoque multidisciplinario, así como el uso y puesta a disposición del repositorio con el código abierto que permite la replicación o, en su caso, la exploración de mejoras para los resultados obtenidos.

Objetivos

Objetivo general

Explorar la posibilidad de mejorar la métrica mAP de los dos modelos propuestos de aprendizaje profundo en la detección e identificación de opacidades pulmonares debidas al COVID19 a través del aprendizaje modular generado por transferencia de conocimiento.

Objetivos específicos

- Desarrollar análisis exploratorios de los conjuntos de datos seleccionados para entrenar modelos.
- Entrenar modelos para tareas específicas que permitan el aprendizaje modular, a través de transferencia de conocimiento en la tarea de detección e identificación de daño pulmonar.
- Explorar distintos parámetros, modificación de arquitectura, técnicas de etiquetado y entrenamiento.
- Evaluar y comparar el desempeño de los modelos propuestos.
- Reportar los resultados obtenidos y poner a disposición el repositorio con los códigos abiertos utilizados para la replicación de resultados o futuros desarrollos.
- Generar un trabajo de autocontenido, que facilite la comprensión del trabajo realizado, desde distintos enfoques.

Hipótesis

El aprendizaje modular generado por la transferencia de aprendizaje puede mejorar la métrica mAP en los dos modelos propuestos de aprendizaje profundo utilizados para la detección e identificación de daño en los pulmones ocasionado por COVID19.

Justificación

La comunidad científica ha comenzado a implementar herramientas de inteligencia artificial. Al trabajar de la mano con el rigor del método científico, es crucial que estas herramientas tecnológicas se adapten a él, garantizando la calidad de los resultados logrados. En el ámbito de la detección y diagnóstico de enfermedades, esto es fundamental. Es de suma importancia generar herramientas que demuestren su desempeño en estas tareas, así como contar con la documentación adecuada que muestre las evidencias cuantitativas que respaldan sus resultados. Esto destaca la necesidad de material veraz para fines de consulta, permitiendo a los expertos en el área, y sobre todo a los no expertos, tener un marco de referencia sobre las ventajas, desventajas, alcances y limitantes de estas nuevas tecnologías. Esto contribuye a generar una correcta sinergia entre la ciencia y la tecnología, aportando al progreso no solo de estas disciplinas, sino también de nuestra especie.

El COVID19 presenta una tasa de mortalidad más alta que otras enfermedades respiratorias comunes [81]. Se manifiesta de manera muy similar a otras neumonías virales y bacterianas en las radiografías de rayos X de tórax, lo que dificulta su diagnóstico [76]. Dada esta situación y motivado por la competencia pública **SIIM-FISABIO-RSNA COVID-19 Detection** propuesta en la plataforma **Kaggle** por la **Society for Imaging Informatics in Medicine** (<https://www.kaggle.com/c/siim-covid19-detection/leaderboard>) , donde el objetivo es la aplicación de modelos de redes neuronales evaluados con la métrica mAP y con el mismo propósito que el planteado en la presente investigación. Donde el modelo ganador obtuvo un mAP de 0.635, mientras que los 1304 restantes fueron inferiores. Cabe destacar que esta métrica es cercana a 0 para un mal desempeño y cercana a 1 en el mejor de los casos. Esto muestra que los resultados obtenidos en esta competencia no son los más deseados, especialmente considerando la delicada tarea de un diagnóstico médico.

Aunado a esto, muchos trabajos, como los presentados en el capítulo 3 y los comparados en el artículo titulado ***Advancement of deep learning in pneumonia/Covid-19 classification and localization: A systematic review with qualitative and quantitative analysis*** [67], muestra una comparación del desempeño de la aplicación de modelos de aprendizaje profundo, donde se intenta diagnosticar, clasificar y detectar el COVID19 a través de imágenes médicas como rayos X o tomografías computarizadas. Estos trabajos muestran desempeños diversos respecto a métricas como la precisión o la exactitud en un rango del 95 % al 96 %, lo cual representa un mejor desempeño en la tarea abordada. Sin embargo, ninguno resuelve el problema que se busca resolver en esta investigación mismo que se propone en la competencia. Sin embargo son un indicador positivo para la búsqueda de una mejora en la métrica mAP para el problema propuesto.

En este contexto, si bien la base de datos propuesta por el desafío no se conforma por una población de pacientes mexicanos y, hasta el desarrollo de esta investigación, no se cuenta con alguna base de datos pública con las anotaciones necesarias para una población mexicana, esto no representa un problema mayor. Dado que de resolver el problema positivamente, la transferencia de conocimiento permite realizar un reajuste de parámetros para adecuar el desempeño de la red a cualquier otra distribución de datos del mismo estilo, lo cual representa un precedente importante para futuras investigaciones y desarrollos.

Capítulo 1

Introducción histórica

En esta sección se hace una breve mención de los aportes de grandes científicos cuyas mentes brillantes, trabajos e investigaciones han dado lugar a múltiples áreas de investigación científica, incluyendo el desarrollo del presente trabajo.

Algunas de las personalidades aquí mencionadas fueron seleccionadas no solo por sus grandes contribuciones a los temas que se desarrollan en la presente investigación, sino también por su resiliencia ante las adversidades e injusticias sociales y académicas que enfrentaron. Son un ejemplo de las numerosas adversidades con las que los científicos y la sociedad aún se enfrentan y luchan por erradicar. Esto es un testimonio histórico de que la ciencia es un acto transformador necesario y disruptivo que demanda y ofrece libertad. Todo esto ha sido de suma influencia en la elección personal de realizar una carrera científica, resultado de los sueños que emanan de la admiración y profunda inspiración de muchas otras personalidades como las aquí expuestas.

1.1. Maria Skłodowska

Popularmente conocida como Marie Curie, es una de las figuras más prominentes en la historia de la ciencia, desempeñó un papel crucial en el descubrimiento de los rayos X y su aplicación en la radiografía. Nacida como Maria Skłodowska en Polonia en 1867, Curie se destacó por su trabajo pionero en el campo de la radiactividad, un término que ella misma acuñó. Junto con su esposo, Pierre Curie, llevó a cabo investigaciones fundamentales sobre elementos radiactivos, como el polonio y el radio.

Los rayos X, descubiertos por Wilhelm Conrad Röentgen en 1895, presentaban un fenómeno intrigante pero desconcertante para la comunidad científica. Fue Marie Curie quien, mediante su profundo conocimiento de la radiactividad y sus propiedades, proporcionó una comprensión más completa de la naturaleza de los rayos X. Su investigación sobre la radiactividad, que culminó en el aislamiento del radio, un elemento altamente radiactivo, proporcionó una base sólida para el estudio de los rayos X.

Además de su contribución teórica, Marie Curie también desempeñó un papel práctico en la aplicación de los rayos X en la medicina. Durante la Primera Guerra Mundial, organizó unidades móviles de rayos X para diagnosticar fracturas y cuerpos extraños en soldados heridos en el frente. Esta aplicación innovadora de

la tecnología de rayos X ayudó significativamente en el tratamiento médico de los soldados y demostró el valor práctico de esta tecnología en el campo de la medicina.

El legado de Marie Curie en el descubrimiento y aplicación de los rayos X sigue siendo fundamental en la ciencia moderna y la medicina. Su enfoque incansable en la investigación científica y su dedicación al avance del conocimiento no solo contribuyeron al descubrimiento de los rayos X, sino que también sentaron las bases para el desarrollo de la radiología y la medicina moderna. En reconocimiento a sus logros, Marie Curie se convirtió en la primera mujer en recibir un Premio Nobel, tanto en Física (1903, compartido con Pierre Curie y Henri Becquerel) como en Química (1911), premios que se le intentaron negar debido a prejuicios de género. Su influencia perdura como un faro de inspiración para generaciones de científicos y médicos, a pesar de haber enfrentado marginación por parte de otros científicos, falta de financiamiento e incluso negativas en el acceso a instituciones científicas por el simple hecho de ser mujer y presentar una mente adelantada a su época.



Figura 1.1: **Marie Skłodowska-Curie (1867 - 1934).**
Imagen de dominio público que muestra el rostro de Marie.

Para información detallada sobre la vida de Marie, véase [84].

1.2. Alan Turing

Alan Mathison Turing fue un genio matemático visionario cuyas contribuciones al campo de la computación y la inteligencia artificial siguen siendo fundamentales hasta el día de hoy. Nacido en 1912 en Gran Bretaña, Turing es mejor conocido por su trabajo pionero durante la Segunda Guerra Mundial, donde desempeñó un papel crucial en descifrar los códigos enemigos, especialmente el código alemán Enigma, lo que ayudó significativamente a los Aliados a ganar la guerra.

Sin embargo, los logros de Turing van mucho más allá de la guerra. Es considerado el padre de la informática teórica y sentó las bases de lo que hoy conocemos como ciencia de la computación. Su trabajo sobre las máquinas de Turing, un concepto teórico para entender el alcance y los límites de la computación, es fundamental

en este campo. Además, su concepto de la máquina universal sentó las bases para el diseño de los primeros computadores digitales.

Pero quizás uno de los aspectos más destacados de la vida de Turing fue su contribución al campo de la inteligencia artificial. En su famoso artículo *Computing Machinery and Intelligence* (Maquinaria computacional e inteligencia) [86], Turing propuso lo que ahora se conoce como la prueba de Turing, un criterio para determinar si una máquina puede exhibir un comportamiento inteligente equivalente o indistinguible del de un ser humano.

A pesar de sus impresionantes logros, Turing enfrentó enormes dificultades personales y profesionales debido a sus preferencias sexuales en una época en donde eran ilegales en el Reino Unido. En 1952, fue condenado por indecencia grave y se vio obligado a someterse a un tratamiento con hormonas para la castración química. Esta condena tuvo graves consecuencias en su vida y en su salud mental.

Trágicamente, en 1954, Turing falleció a la edad de 41 años en circunstancias aún no completamente esclarecidas. Se creen que su muerte fue un suicidio, aunque las circunstancias exactas siguen siendo motivo de debate.

A pesar de las injusticias que enfrentó, el legado de Alan Turing es innegable. Su trabajo sentó las bases de la informática moderna y la inteligencia artificial, y su contribución sigue siendo fundamental en la forma en que entendemos y desarrollamos la tecnología hoy en día. En 2009, el entonces Primer Ministro británico, Gordon Brown, emitió una disculpa oficial en nombre del gobierno británico por el trato injusto y la persecución sufrida por Turing. En 2013, la Reina Isabel II lo perdonó póstumamente, reconociendo así su invaluable contribución a la nación.



Figura 1.2: **Alan Turing (1912-1954).**
Imagen de dominio público que muestra el rostro de Alan.

Para información detallada sobre la vida de Alan, véase [85].

1.3. Wilhelm Röntgen

Wilhelm Conrad Röntgen, nacido el 27 de marzo de 1845 en Lennep, Alemania, es conocido principalmente por su descubrimiento de los rayos X en 1895, un logro que marcó un hito en la historia de la ciencia y la medicina. Röntgen realizaba experimentos con tubos de vacío cuando notó que un trozo de papel cubierto con platinocianuro de bario, colocado cerca de los tubos, comenzó a brillar. Descubrió que este fenómeno era causado por una forma de radiación hasta entonces desconocida, a la que llamó rayos X "debido a su naturaleza desconocida.

Sus investigaciones posteriores sobre los rayos X revolucionaron la medicina al permitir la visualización de estructuras internas del cuerpo humano sin cirugía. Este avance condujo al desarrollo de la radiografía, que se convirtió en una herramienta invaluable para el diagnóstico médico y la cirugía.

Röntgen recibió el primer premio Nobel de física en 1901 por su descubrimiento, continuó contribuyendo al campo de la física y la investigación hasta su muerte el 10 de febrero de 1923, dejando un legado perdurable en la historia de la ciencia.



Figura 1.3: **Wilhelm Conrad Röntgen (1845-1923)**.

Imagen de dominio público que muestra el rostro de Wilhelm junto a la primera radiografía de rayos X que muestra la mano de su esposa.

Para información detallada sobre la vida de Wilhem, véase [87].

1.4. John McCarthy

John Patrick McCarthy fue un influyente científico de la computación y pionero en el campo de la inteligencia artificial (IA). Nacido en 1927 en Boston, Massachusetts, McCarthy es conocido por sus numerosas contribuciones al desarrollo de la IA y por acuñar el término inteligencia artificial en 1956, durante una conferencia en Dartmouth College en Hanover, New Hampshire. Esta conferencia, referida como la conferencia de Dartmouth, marcó el inicio formal del campo de la inteligencia artificial.

Una de las contribuciones más destacadas de McCarthy fue el desarrollo del lenguaje de programación LISP (List Processing) [88], que se convirtió en un estándar

para la programación en IA debido a su capacidad para manipular símbolos y listas de manera eficiente.

McCarthy fue fundamental en la creación del concepto de "IA fuerte", que sostiene que las máquinas pueden ser tan inteligentes como los humanos, en contraposición a la "IA débil", que se enfoca en la simulación de comportamientos inteligentes sin pretender igualar la inteligencia humana.

A lo largo de su carrera, McCarthy enfrentó varias adversidades, incluyendo la resistencia inicial hacia la IA como un campo legítimo de estudio por parte de algunos colegas en la comunidad científica. Además, tuvo que lidiar con la falta de recursos y apoyo financiero en las etapas iniciales del desarrollo de la IA.

A pesar de estos desafíos, McCarthy perseveró y dejó un legado duradero en el campo de la IA, inspirando a generaciones de investigadores y contribuyendo significativamente al avance de esta disciplina. Su trabajo continúa siendo una influencia fundamental en la investigación y desarrollo de la inteligencia artificial en la actualidad. McCarthy falleció en 2011, dejando un impacto perdurable en el mundo de la ciencia y la tecnología.

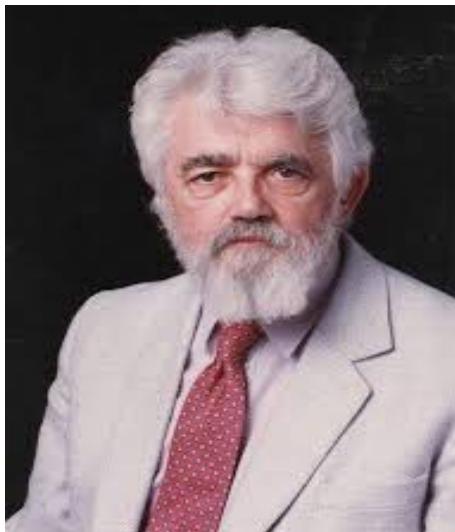


Figura 1.4: **John Patrick McCarthy (1927-2011)**.
Imagen de dominio público que muestra el rostro de John.

Para información detallada sobre la vida de John, véase [89].

Capítulo 2

Conceptos básicos

En este primer capítulo, se abordan y definen los conceptos e ideas clave que se utilizan a lo largo del presente trabajo. El objetivo es generar homogeneidad en las definiciones y proporcionar claridad, poniendo a disposición material de consulta dentro del mismo escrito.

2.1. Biología y Medicina

Con el propósito de facilitar la comprensión del texto, se proporcionan descripciones generales de los términos y conceptos relacionados con el ámbito médico y de la biología que se utilizan a lo largo del presente trabajo. No se pretende ofrecer una definición formal o rigurosa de los conceptos más allá de lo necesario para obtener una perspectiva clara y accesible del significado de dichos conceptos. Esto proporciona un sentido al escrito y permite generar una perspectiva adecuada para los lectores sin conocimientos específicos en estas áreas del saber.

2.1.1. Sistema respiratorio

Es el conjunto de órganos en el cuerpo humano que se ven involucrados en la respiración. Nos enfocaremos en los pulmones y sus componentes que conforman el modelo biológico, donde se presentan los daños de interés causados por el SARS-CoV-2.

Pulmón

Definición 2.1.1. *Es uno de los dos órganos ubicados en el tórax, que provee al organismo de oxígeno y extrae el dióxido de carbono del mismo.*

Cada pulmón se divide en secciones llamadas lóbulos, tres ubicados en el pulmón derecho y dos en el izquierdo.

Bronquios

Definición 2.1.2. *Son los conductos principales y más anchos de aire hacia los pulmones. Existe uno para el pulmón izquierdo y otro para el derecho, los cuales permiten el paso del aire hacia los pulmones.*

Bronquiolos

Definición 2.1.3. *Son las ramificaciones más pequeñas de los conductos aéreos que forman parte de los bronquios, encargadas de dirigir el aire hacia los alvéolos.*

Alvéolos

Definición 2.1.4. Son diminutos sacos de aire ubicados en los extremos de los bronquiolos; en los alvéolos se lleva a cabo el intercambio de oxígeno y dióxido de carbono entre los pulmones y la sangre durante el proceso de respiración.

Pleura

Definición 2.1.5. Es una membrana transparente y muy delgada que recubre los pulmones y, además, reviste el interior de la pared torácica. Permite que los pulmones se muevan suavemente durante la respiración.

Para información detallada sobre los conceptos aquí descritos, véase [43].

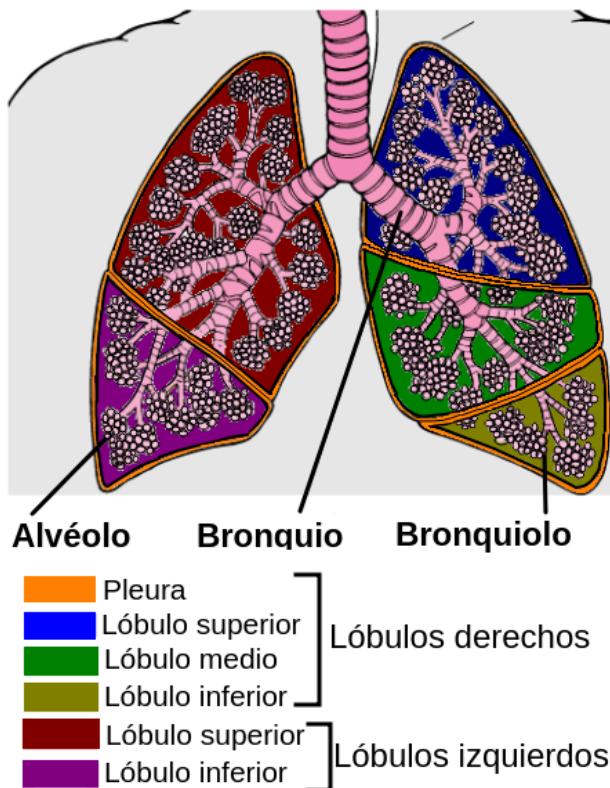


Figura 2.1: Esquema del aparato respiratorio.

Imagen modificada de [41], que presenta un esquema del sistema respiratorio, denotando con un código de colores las partes relevantes del sistema.

2.1.2. Conceptos biomedicos

ADN

Definición 2.1.6. Es la abreviación de ácido desoxirribonucleico, la denominación para el material que contiene la información hereditaria en gran parte de los organismos. Posee una estructura en forma de doble hélice, y está compuesto por pares de bases nucleicas que pueden ser adenina (A), timina (T), citosina (C) y guanina (G). Estas bases representadas por letras permiten que el ADN se modele por cadenas de caracteres que codifican toda la información necesaria para que un organismo funcione.

ARN

Definición 2.1.7. Es la abreviación de ácido ribonucleico. Posee una estructura de cadena única y se compone de las mismas bases nucleicas que el ADN, con la excepción del uracilo (U), que sustituye a la timina (T) presente en el ADN. Actúa como mensajero, llevando instrucciones desde el núcleo de la célula, donde se encuentra la información del ADN, hacia los ribosomas. Estos son mecanismos dentro de las células donde se producen proteínas. Este proceso, conocido como transcripción permite que los organismos desarrollen funciones y estructuras específicas.

Una cadena de caracteres que modela una parte del ADN se conoce como secuencia, mientras que el ADN completo, representado como secuencia, se denomina genoma. Por ejemplo, **TGG CCA GCG TGG** son los primeros 12 caracteres del genoma humano, el cual está compuesto por aproximadamente 3 mil millones.

Para información detallada sobre el ADN y ARN, véase [44].

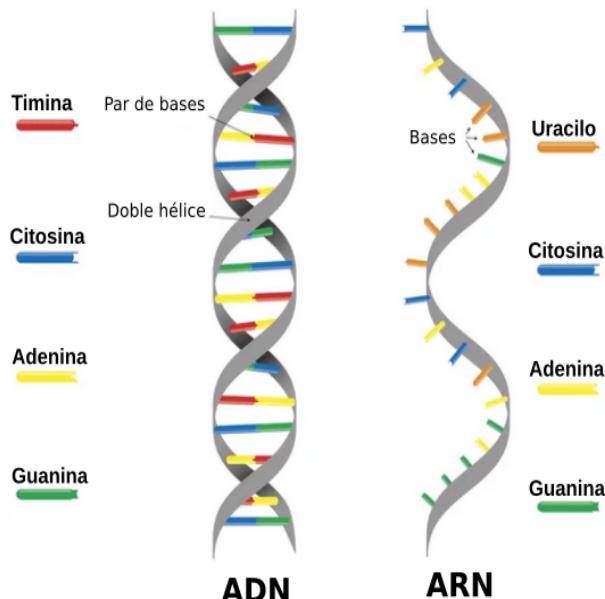


Figura 2.2: Esquema de la estructura del ADN y el ARN.

Imagen modificada de [42], que presenta un esquema de la estructura del ADN (izquierda) y ARN (derecha), denotando con un código de colores las bases respecto a cada uno.

RT-PCR

Definición 2.1.8. Son las siglas de reverse transcription-polymerase chain reaction (reacción en cadena de la polimerasa con transcriptasa inversa). Esta técnica representa una forma rápida y precisa para el diagnóstico de enfermedades de origen infeccioso, ya que es capaz de detectar la presencia de ADN o ARN del organismo causante de la enfermedad, permitiendo así la detección en las fases más tempranas de la infección.

La prueba se realiza tomando una muestra de material genético que es copiado múltiples veces (amplificación). De esta manera, si la muestra contiene organismos infecciosos, se vuelven más fáciles de detectar.

Este método fue uno de los principales utilizados para el diagnóstico del COVID-19.

Para información detallada sobre PCR, véase [45].

Coronavirus

Definición 2.1.9. *Los coronavirus son un tipo de virus, uno de ellos es el SARS-CoV-2, que produce la enfermedad infecciosa COVID-19. Se caracteriza por síntomas que van desde leves, como fiebre y tos, hasta formas más graves que pueden incluir dificultad respiratoria causadas por neumonia. La transmisión principal ocurre a través de gotas respiratorias al toser, estornudar o hablar. La enfermedad fue identificada por primera vez en Wuhan, China, en diciembre de 2019, provocando una pandemia global, dando lugar a importantes esfuerzos para controlar su propagación y desarrollar vacunas para prevenirla. [52]*

2.1.2.1. Opacidades

Se describe como el fenómeno físico y fisiológico que hace referencia a la capacidad relativa de la materia para obstaculizar la transmisión de energía radiante. Esto se traduce en áreas grises en las imágenes médicas, que son indicadores de la presencia de zonas altamente densas. Las áreas más densas son más opacas, variando en tonalidades grises, mientras que las zonas menos densas son más transparentes y aparecen en tonalidades oscuras. Estas opacidades son de causa variable y cuentan con distintas clasificaciones.

Las opacidades causadas por el COVID-19, presentes en radiografías de rayos X de pulmón, son los objetos de interés de la presente investigación, donde se busca la detección automática de estos elementos.

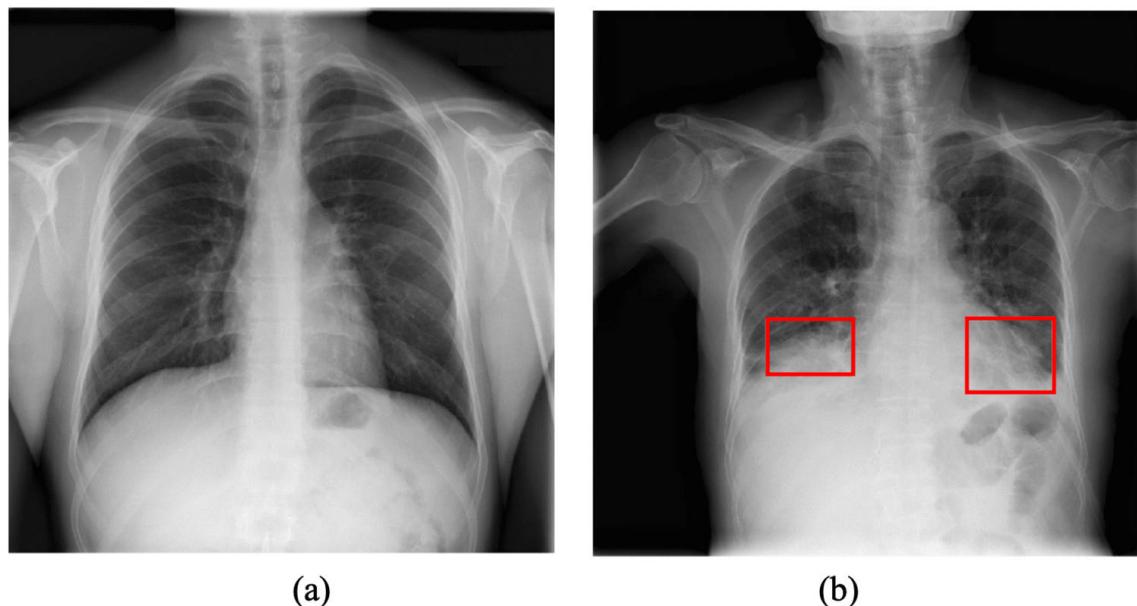


Figura 2.3: **Imagen sin presencia de opacidades frente a imagen con presencia de opacidades.**

Imagen modificada de [46], a) Muestra una imagen de una persona sana, sin presencia de opacidades. b) Imagen que destaca con rectángulos rojos la presencia de opacidades en los pulmones.

Consolidación

Definición 2.1.10. *Es un tipo de opacidad producida por la sustitución del aire en los alvéolos por líquido u otros materiales más densos como pus, sangre, proteínas, agua, etc. Esto es causado por una infección.*

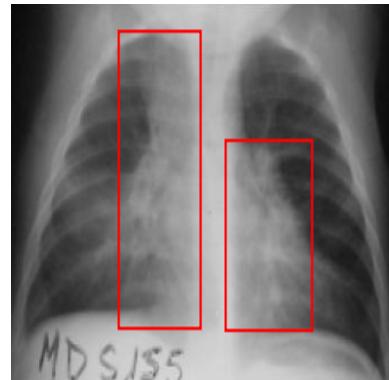


Figura 2.4: Ejemplo de opacidad causada por consolidación.

Imagen modificada de [1], que muestra en rectángulos rojos la presencia de una opacidad por consolidación en una radiografía de pulmón.

Derrame pleural

Definición 2.1.11. *Fenómeno que implica la acumulación anormal de líquido en el espacio pleural, el cual es el espacio entre las capas de la membrana que recubre los pulmones y la cavidad torácica. Este fenómeno se origina por diversas causas, tales como infecciones, insuficiencia cardíaca u otras enfermedades pulmonares.*

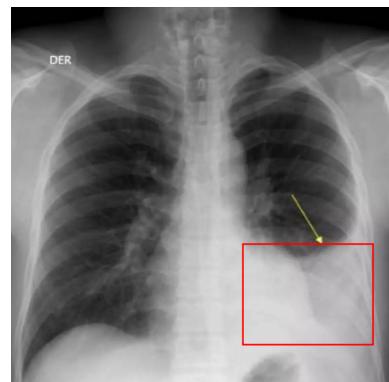


Figura 2.5: Ejemplo de opacidad causada por derrame pleural.

Imagen modificada de [38], que muestra en un recuadro rojo, opacidad generada por derrame pleural en el pulmón izquierdo.

Para obtener información detallada sobre los conceptos aquí expuestos, véase [47].

Neumonía

Definición 2.1.12. *Condición médica caracterizada por inflamación en los pulmones, donde el oxígeno en los alvéolos es sustituido por fluido, lo que vuelve a la respiración un proceso doloroso y evita el intercambio de oxígeno en la sangre. Puede ser causada por múltiples factores [39].*

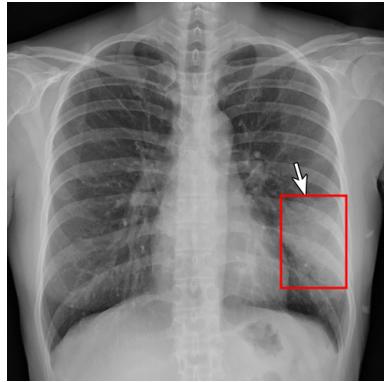


Figura 2.6: **Ejemplo de opacidad causada por neumonía.**

Imagen modificada de [40], que presenta en un recuadro rojo la opacidad de neumonía causada por la bacteria Chlamydia pneumoniae en el pulmón izquierdo.

2.1.3. Imágenes médicas

Los médicos recurren al diagnóstico mediante el uso de diversas imágenes del cuerpo humano, las cuales se obtienen con distintos métodos y con diferentes propósitos. En esta sección, se hará una breve mención de las más comunes.

2.1.3.1. Radiografías de rayos X

Se realizan mediante ondas electromagnéticas que generan una imagen del interior del cuerpo humano, representada en tonos que varían desde el blanco hasta el negro. Esto se debe a que los diferentes componentes del cuerpo tienen una capacidad única de absorción de dichas ondas. Los huesos, ricos en calcio, producen tonos claros, siendo los más visibles debido a su mayor capacidad de absorción. Por otro lado, el aire es el que menos radiación absorbe, generando colores oscuros.

Cabe destacar que hay dos tipos de radiografías de rayos X:

- **CR (*Computed Radiography*):** Radiografía computarizada, por su traducción al español, es un formato en el cual se emplea una placa de imágenes de fósforo para crear una imagen digital. Se utiliza un sistema basado en cassetes, similar al utilizado en las películas analógicas.
Es útil cuando se requiere una visión detallada de órganos y estructuras internas, como el cerebro, la columna vertebral, los vasos sanguíneos, entre otros.
- **DX (*Digital Radiography*):** Radiografía digital, por su traducción al español, representa la forma más avanzada en el campo de la radiografía. Emplea un detector de rayos X digital para adquirir imágenes automáticamente y transferirlas a una computadora para su visualización.
Es eficaz para exámenes rutinarios y de seguimiento destinados a visualizar estructuras menos complejas, como el tórax, las extremidades o los dientes, donde se requiere una imagen rápida y eficiente.

Para obtener información detallada sobre radiografías de rayos X, véase [53].

2.1.3.2. Tomografía computarizada

Se utiliza un equipo médico especial de rayos X para generar una imagen transversal del cuerpo. Su proceso consiste en combinar una serie de radiografías tomadas desde distintos ángulos alrededor del cuerpo, y mediante un procesamiento computacional se construyen imágenes transversales en 3D que proporcionan más información y claridad que una radiografía clásica.

Para obtener información detallada sobre tomografía computarizada, véase [54].

2.1.3.3. Resonancia magnética

En esta técnica, se obtienen imágenes a través de campos magnéticos y ondas de radio. Mediante el uso de un potente imán, se genera un campo magnético con la fuerza suficiente para alinear los protones de los átomos de hidrógeno presentes en el cuerpo humano, induciéndolos a movimiento al ser expuestos a ondas de radio. Al interrumpir las ondas de radio, los protones regresan a su alineación original y recuperan su estado estático. Durante este proceso, se emiten ondas de radio de vuelta, las cuales contienen información captada por un detector especializado. Esta información permite reconstruir una imagen tridimensional compuesta por múltiples capas transversales.

Para obtener información detallada sobre resonancia magnética, véase [55].

2.1.3.4. Ecografía

Con esta técnica se obtienen imágenes a partir de ondas sonoras de alta frecuencia, las cuales rebotan en los tejidos del cuerpo humano. A través de sensores electrónicos, se construye una imagen.

Para obtener información detallada sobre ecografía, véase [56].

2.1.3.5. Estándares de formato y transmisión de imágenes médicas

Debido a la necesidad de almacenar y compartir información digital, ya sea con otros especialistas o con los pacientes, los especialistas se encargaron de establecer estándares internacionales para imágenes y texto digital que los centros deben cumplir.

A continuación, se mencionan algunos de los estándares más usados.

DICOM

Digital Imaging and Communications in Medicine, o, por su traducción al español, Imágenes y Comunicaciones Digitales en Medicina, es un estándar utilizado en el ámbito de la medicina para la gestión, almacenamiento, impresión y transmisión de información médica relacionada con imágenes. Las imágenes se producen en la extensión .dcm, un formato independiente del equipo médico utilizado. Este estándar consiste en una imagen de alta resolución diseñada para asegurar la interoperabilidad de las imágenes médicas entre diferentes sistemas y dispositivos médicos, además de un conjunto de metadatos que permite la inclusión de información clínica asociada a la imagen, como datos sobre el paciente, el procedimiento, el dispositivo de adquisición de imágenes, entre otros.

Para obtener información detallada sobre DICOM, véase [57].

HL7

Health Level 7, o, por su traducción al español, Salud Nivel 7, es una organización internacional que desarrolla estándares para la interoperabilidad y el intercambio de información en el ámbito de la atención médica. El estándar HL7 se utiliza para facilitar la comunicación entre sistemas de información de salud y promover la integración efectiva de datos clínicos y administrativos en entornos de atención médica. Define una estructura de mensajes estandarizada que permite la transferencia de datos entre diferentes sistemas de salud. Estos mensajes contienen información clínica y administrativa, como resultados de laboratorio, historias clínicas, órdenes de tratamiento, entre otros.

Para obtener información detallada sobre HL7, véase [58].

2.2. Imágenes digitales

Dado que las imágenes digitales son visualizadas, representadas y analizadas por computadoras, es fundamental comprender cómo son representadas por estos dispositivos.

En esta sección, se tratan los conceptos fundamentales e ideas que facilitan la comprensión de dicho panorama. Se utiliza tanto lenguaje matemático como coloquial, adaptándose a los lectores sin formación o interés en las matemáticas aquí presentadas.

2.2.1. Estructura de una imagen digital

Para comprender cómo las computadoras representan las imágenes, es necesario comenzar por entender la estructura básica que permite transformar nuestro concepto basado en la interpretación visual de una imagen al almacenamiento y representación de la información contenida en la misma.

Definición 2.2.1. Píxel

Es la unidad más pequeña de una imagen digital. Es la representación visual de un punto en una cuadrícula, y cada píxel codifica la información sobre su posición y color.

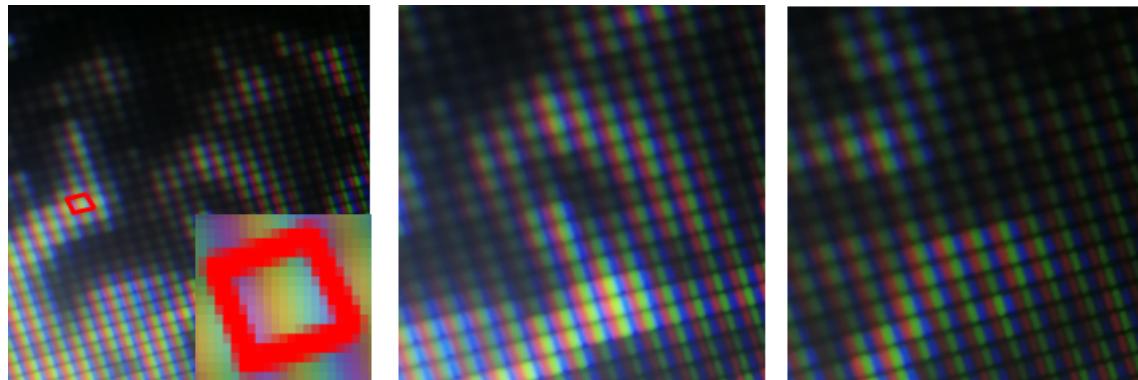


Figura 2.7: Fotos de un píxel.

Fotografías de la pantalla de una computadora, tomada con la adaptación de un lente tipo macro más el zoom de la cámara de un celular, donde se aprecia el arreglo de píxeles que conforman la imagen.

Respecto a la figura 2.7, resalta la ironía de que, al tratarse de una fotografía digital, esta se compone de píxeles. Al realizar un aumento del 1600 % al píxel resaltado en rojo, se puede apreciar un píxel representado por píxeles, como se muestra en la esquina de la primera imagen de la figura. Ahora, si este trabajo se está consultando en formato digital, es posible generar una captura de pantalla y repetir el proceso infinitamente, llevándonos a un ciclo recursivo de píxeles representados por píxeles, que a su vez son representados por más píxeles... Esto presenta una característica fractal interesante y curiosa.

Para el concepto de fractal, vease [75].

Definición 2.2.2. Profundidad de color

Se entiende como la cantidad de información que se puede almacenar en un píxel, lo cual está directamente relacionado con la cantidad de colores o tonalidades que se pueden representar. Esto depende de los bits (binary digit), o, por su traducción al español, dígito binario, que es la unidad mínima referente al almacenamiento de información, es decir, la memoria necesaria para almacenar un 0 o un 1. Esto se utiliza para representar la información de color contenida en una imagen. Si cada píxel puede almacenar N cantidad de bits, entonces puede representar 2^N colores. Esta relación está dada por la cantidad de combinaciones que se pueden dar usando N bits (una por cada color o intensidad).

Si una imagen se representa con 8 bits, cada píxel puede adoptar $2^8 = 256$ colores o intensidad; en cambio, si se representa con 16 bits, cada píxel tiene la capacidad de mostrar $2^{16} = 65536$ colores o intensidades.

Que un píxel “tome” un color o intensidad significa, en términos computacionales, que adquiere un valor numérico. Por ejemplo, en el caso de una imagen de 8 bits con 256 colores o intensidades disponibles, asignamos a cada color o intensidad un número mediante una enumeración simple, comenzando desde 0.



Figura 2.8: Etiquetado numérico para colores.

Representación gráfica de la asignación de intensidad de color en función de la asignación numérica.

Siguiendo este ejemplo, observamos que un píxel puede tener un valor en el intervalo $[0, 255]$. En términos generales, si una imagen está representada con N bits, sus píxeles adquieren valores en el intervalo real $[0, 2^N - 1]$. Comúnmente, este rango se representa con valores comprendidos entre cero y uno, lo que se denomina normalización y se puede realizar de distintas maneras.

Formalmente, se establece que la profundidad de color está representada por una función biyectiva $P : C \rightarrow [0, 2^N - 1]$, donde C es el conjunto de colores o intensidades a representar. La normalización se define mediante cualquier función biyectiva $N : [a, b] \rightarrow [0, 1]$, como por ejemplo $N(x) = \frac{x-a}{b-a}$. Dado que la composición de funciones biyectivas resulta en una función biyectiva, se tiene que $N \circ P : C \rightarrow [0, 1]$,

donde $N \circ P = N(P(x))$, describe un isomorfismo de conjuntos.

$$C(N) = \{x | x \in [0, 2^N - 1]\} \cong \{x | x \in [0, 1]\} \subset \mathbb{R}$$

Se puede obtener un color compuesto o secundario al mezclar una cantidad específica de colores base.

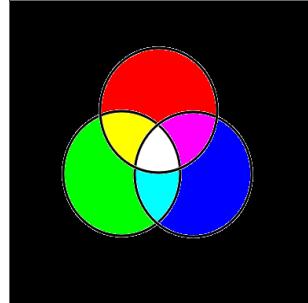


Figura 2.9: **Representación gráfica de la combinación de colores.**

Diagrama que ilustra las combinaciones posibles de colores utilizando los colores **rojo**, **verde** y **azul** como base.

Definición 2.2.3. *Canal de color*

*Codifica la cantidad e intensidad de colores básicos utilizados para formar colores compuestos en una imagen digital, siguiendo sistemas estandarizados. Uno de los más utilizados es el sistema **RGB** (**Red**, **Green**, **Blue**), que define colores combinando distintos tonos de **rojo**, **verde** y **azul**, como se muestra en la figura 2.9.*

De esta manera, si tenemos una base conformada por k colores $\mathcal{B} = \{c_1, c_2, \dots, c_k\}$, podemos entender un color compuesto C como una combinación lineal de estos elementos base:

$$C = \sum_{i=1}^k \alpha_i c_i$$

donde α_i representa la intensidad de cada color base utilizado, $0 \leq \alpha_i \leq 1$ para cada $1 \leq i \leq k$. Aquí, $\alpha_i = 0$ representa la ausencia del i -ésimo color base, mientras que $\alpha_i = 1$ indica la intensidad máxima del i -ésimo color base.

Con estas ideas, podemos profundizar en cómo un ordenador representa e interpreta una imagen digital.

Definición 2.2.4. *Imagen digital*

Una imagen digital representada en N bits, con una cantidad c de canales para el color, se conceptualiza como una superposición de rejillas de píxeles. Cada rejilla tiene dimensiones de n píxeles de alto por m píxeles de ancho, donde cada píxel toma valores en el rango de $[0, 2^N - 1]$. Relativo al tamaño, la imagen se caracteriza por sus dimensiones $n \times m \times c$, que geométricamente se puede visualizar como un volumen cúbico de n unidades de altura, m unidades de longitud y c unidades de anchura.

En términos formales, una imagen digital representada en N bits con c canales de color es una matriz tridimensional M con entradas reales, que se entiende como un elemento $M \in \mathbb{M}_{n \times m \times c}(\mathbb{R})$, donde cada elemento de la matriz, denotado por m_{ijk} , representa el píxel en la posición $(i, j, k) \in \mathbb{N}^3$. Por definición, se cumple que $m_{ijk} \in \mathbb{R}$, específicamente

$$m_{ijk} \in [0, 2^N - 1] \cong [0, 1] \text{ donde}$$

$$1 \leq i \leq n$$

$$1 \leq j \leq m$$

$$1 \leq k \leq c$$

Geométricamente, se puede observar en la figura 2.10 un esquema para imágenes de un solo canal, es decir, donde la matriz es un arreglo bidimensional de píxeles con $c = 1$, y la imagen representada está en escala de grises.

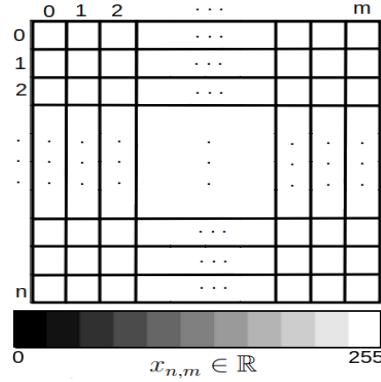


Figura 2.10: **Estructura geométrica de una imagen dada como matriz.**

Representación geométrica de una imagen con un solo canal de color $c = 1$, donde cada píxel toma valores en el rango $[0, 255]$. La profundidad de color es de $N = 8$ bits, y la variación del color abarca desde el blanco, pasando por escalas de grises, hasta llegar al negro.

En la figura 2.11 se ilustra la misma representación geométrica, pero para el caso de 3 canales de color, en este caso la matriz es un volumen tridimensional con $c = 3$, y la imagen representada está a color.

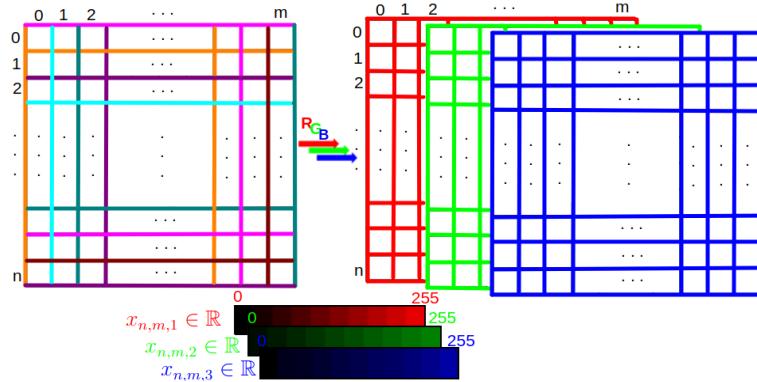


Figura 2.11: **Estructura geométrica de una imagen dada como matriz.**
Representación geométrica de una imagen con tres canales de color $c = 3$, donde cada píxel toma valores en el rango $[0, 255]$. La profundidad de color es de $N = 8$ bits, y la variación de intensidades se realiza en los colores rojo, verde y azul, cada uno representado por su canal respectivo. El color de cada píxel en la imagen se determina mediante la combinación lineal resultante de las intensidades de los colores base.

Definición 2.2.5. Interpretación fotométrica

Es un concepto referente al análisis e interpretación de la información relacionada con la intensidad de la luz en una imagen. Para nuestros fines, nos interesa la manera en que se asignan estos valores de intensidad lumínica. Para abordar esto, definiremos dos posibilidades:

MONOCROMO1: Es una asignación de la escala de intensidades. En este caso, indica que las intensidades van desde las tonalidades más brillantes hasta las más oscuras, describiéndose mediante valores de píxeles ascendentes. Es decir, a medida que el valor de la intensidad del píxel aumenta, el color que representa es más oscuro.

MONOCROMO2: Es la versión opuesta de MONOCROMO1; indica que las intensidades van desde las tonalidades más oscuras hasta las más brillantes. A medida que el valor de la intensidad del píxel aumenta, el color que representa es más claro.

Formalmente, la interpretación fotométrica esta dada por la característica de la normalización y la profundidad de color de ser definidas como una función biyectiva $N \circ P$, lo que permite reparametrizar la función inversa:

$$(N \circ P)^{-1}(1 - x)$$

esto modifica el sentido de la asignación de intensidades.

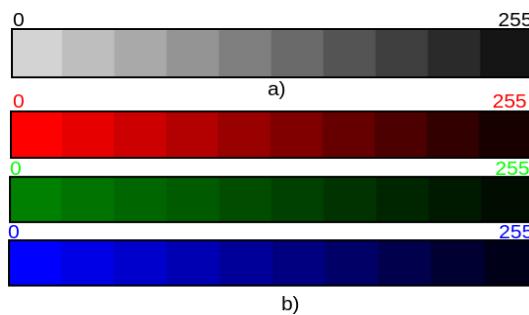


Figura 2.12: **MONOCROMO1.**

Representación gráfica de MONOCROMO1: a) muestra la asignación para la escala de grises y b) la asignación para **RGB** respectivamente.

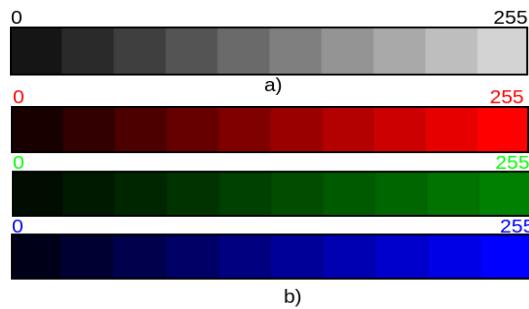


Figura 2.13: **MONOCROMO2.**

Representación gráfica de MONOCROMO2: a) muestra la asignación para la escala de grises y b) la asignación para **RGB** respectivamente.

Para obtener información detallada sobre imágenes digitales, véase [59].

Para obtener información detallada sobre conceptos de álgebra lineal, véase [61].

2.3. Inteligencia artificial

La inteligencia artificial (IA) es un campo de la informática centrado en el desarrollo de sistemas capaces de realizar tareas que normalmente requieren inteligencia humana. Estas tareas incluyen el aprendizaje, el razonamiento, la percepción visual, el reconocimiento del lenguaje natural y la toma de decisiones. El objetivo principal de la IA es crear máquinas que puedan imitar la inteligencia humana para resolver problemas complejos.

Dentro de la inteligencia artificial, el aprendizaje automático o *Machine Learning* (ML) en inglés es una rama esencial. Ofrece técnicas que permiten a las máquinas aprender patrones a partir de datos y mejorar su rendimiento sin intervención humana directa. Para evitar programar explícitamente todas las reglas, los algoritmos de aprendizaje automático permiten que el sistema aprenda de los datos y ajuste su comportamiento en consecuencia, lo anterior partiendo de un código de programación estático.

En resumen, la inteligencia artificial busca crear sistemas inteligentes que imiten la inteligencia humana. El aprendizaje automático, por otro lado, es una técnica dentro de la inteligencia artificial que permite a las máquinas aprender y mejorar a través de la experiencia. En este contexto, se entiende la inteligencia como la habilidad de aprender a partir de la experiencia y su aplicación en la adaptación de comportamiento. Sin embargo, no se pretende profundizar en los aspectos filosóficos, psicológicos, sociológicos y culturales que implican entender y definir lo que es la inteligencia.

En esta sección, se abordan los conceptos e ideas que facilitan la comprensión y aplicación de la totalidad de metodologías, técnicas y estrategias empleadas a lo largo de la presente investigación.

2.3.1. Modelos analíticos vs inteligencia artificial

La inteligencia artificial y el aprendizaje automático ofrecen un marco de trabajo que, de cierta manera, difiere del método deductivo que normalmente empleamos al encontrar relaciones o modelos analíticos.

Por ejemplo, suponiendo que se quiere estudiar el fenómeno de la trayectoria que describe el lanzamiento de un objeto, para esta finalidad se debe tener en cuenta parámetros específicos relacionados al fenómeno:

- Puntos que describen la posición en el espacio $(x, y) \in \mathbb{R}^2$.
- Velocidad inicial del objeto lanzado v_0 .
- Ángulo de lanzamiento inicial θ .
- Constante de aceleración de caída libre g .

En este caso, después de un tiempo de apelar a las relaciones físicas y matemáticas que vinculan estos parámetros o variables, se puede llegar a relacionarlos en un modelo analítico ya conocido y formulado por la ecuación de tiro parabólico.

$$y = x \tan(\theta) - \frac{x^2 g}{2v_0^2 \cos^2(\theta)}$$

Por otra parte, tomando como ejemplo uno de los problemas abordados en esta investigación, que es la clasificación de imágenes, se define el conjunto

$$\mathbb{I} = \{I | I \in M(\mathbb{R})_{n \times m \times c}\}, \text{ donde } |\mathbb{I}| = k.$$

El cual representa un conjunto de k imágenes, cada una con dimensiones de $n \times m \times c$ píxeles. Esto implica que para cada imagen, hay información contenida en $n \times m \times c$ píxeles. Si se desea abordar la clasificación con un enfoque analítico, es necesario tener en cuenta los $n \times m \times c$ parámetros necesarios para definir una función $f : [0, 1]^{n \times m \times c} \rightarrow \{c_1, \dots, c_r\}$, donde $\{c_1, \dots, c_r\}$ es el conjunto de las r posibles clases para las imágenes.

En el caso de imágenes de 300×300 píxeles con 3 canales de color, sería necesario considerar 270,000 parámetros y las relaciones físico-matemáticas que los vinculan. Esto implica una complejidad que crece exponencialmente en función del número de píxeles utilizados para representar las imágenes, lo cual resulta en un trabajo prácticamente inconcebible para un ser humano.

Para evitar estas complicaciones, la inteligencia artificial nos permite definir una función $F_0 : [0, 1]^{n \times m \times c} \rightarrow \{c_1, \dots, c_r\}$ que inicialmente asigna valores de manera aleatoria a cada parámetro necesario. Luego, haciendo uso de datos anotados que capturan la asignación de clases deseada, junto con análisis estadísticos y técnicas de optimización, los parámetros se actualizan, generando así una nueva función $F_1 : [0, 1]^{n \times m \times c} \rightarrow \{c_1, \dots, c_r\}$ que modela de una mejor manera al comportamiento de nuestra función objetivo f .

Este proceso se puede repetir durante un número i de iteraciones, dando lugar a una sucesión de funciones $\{F_j\}_{j=0}^i$ tal que $\lim_{j \rightarrow \infty} F_n = f$. Es decir, para un número suficientemente grande de iteraciones, se puede generar una aproximación muy precisa de la función buscada $f : [0, 1]^{n \times m \times c} \rightarrow \{c_1, \dots, c_r\}$.

Estos métodos no proporciona una función explícita que modele la relación deseada, pero genera una cantidad considerable de parámetros óptimos que permiten una aproximación precisa. Además, proporciona métricas estadísticas que cuantifican la calidad en el rendimiento del modelo. Al no ser un modelo analítico determinista, no puede garantizar un 100 % de certeza en el rendimiento, pero sí un porcentaje bastante cercano que depende de la calidad de la información representada por los datos, la elección del modelo y las estrategias de entrenamiento utilizadas.

Es importante destacar que no solo se requiere de datos cuya distribución caracterice la información de interés para el modelo, sino que también se necesita una capacidad de cómputo considerable. Esto se traduce en suficiente memoria para el almacenamiento de datos y capacidad de cálculo para su análisis.

Para obtener información detallada sobre el marco de trabajo aquí expuesto y sus aplicaciones, consulte [62].

2.3.2. Conceptos clave

Conjunto de datos

Definición 2.3.1. *Colección organizada de información utilizada para extraer características de interés. Debe ser altamente representativa de las relaciones que se desean aprender y modelar.*

El conjunto de datos desempeña un papel crucial en la implementación de modelos de inteligencia artificial, por lo que debe ser cuidadosamente seleccionado y anotado. Este proceso representa un porcentaje significativo en cuanto al buen desempeño del modelo. Por ende, se debe dedicar una gran parte del tiempo a realizar análisis estadísticos exploratorios para entender las distribuciones y características que están presentes en los datos.

Formalmente, un conjunto de datos anotados se puede entender como una colección de n datos $\mathbb{X} = \{x_1, \dots, x_n\}$ y su respectivo conjunto de etiquetas o anotaciones $\mathbb{Y} = \{y_1, \dots, y_n\}$ que describen una relación \mathcal{D} , es decir, $\mathcal{D} \subseteq \mathbb{X} \times \mathbb{Y}$, descrita por:

$$\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\}$$

Esta relación abstrae la función objetivo de interés.

Algoritmo

Definición 2.3.2. *Es un conjunto de instrucciones o reglas lógicas ordenadas, legibles y finitas diseñadas para realizar una tarea específica. Pueden traducirse a un lenguaje de programación que, al ser compilado, permite a la computadora procesar y ejecutar estas instrucciones.*

En el contexto de la inteligencia artificial, los algoritmos definen las instrucciones que permiten a la computadora procesar datos e implementar rutinas de entrenamiento, entre otros procesos que conducen al desarrollo de modelos entrenados en tareas específicas.

Entrenamiento

Definición 2.3.3. *Nombre que se utiliza para especificar el algoritmo que dicta las instrucciones necesarias para que el modelo aprenda las características de interés del conjunto de datos. Esto incluye desde el procesamiento de datos, su estructuración para alimentar el modelo, hasta las técnicas de optimización y evaluación utilizadas para actualizar los parámetros necesarios. Este entrenamiento tiene la finalidad de dotar y medir el aprendizaje del modelo.*

Modelo

Definición 2.3.4. *Es el resultado buscado al aplicar un algoritmo de aprendizaje automático. Este abstrae las relaciones representadas en el conjunto al que se aplica, con la finalidad de predecir o generalizar estas relaciones a nuevos datos no etiquetados.*

Se compone de la arquitectura utilizada y los parámetros asociados a la misma.

Formalmente un modelo M con parámetros θ se entiende como una aproximación funcional dada por

$$\mathcal{D} \approx M(x|\theta) : \mathbb{X} \rightarrow \mathbb{Y}$$

Conjunto de entrenamiento

Definición 2.3.5. *Porción del conjunto de datos seleccionado específicamente para entrenar un modelo. El conjunto de entrenamiento funciona como ejemplos que permiten al modelo ajustar parámetros.*

Conjunto de validación

Definición 2.3.6. *Porción del conjunto de datos que permite la evaluación del modelo durante el proceso de entrenamiento. Este subconjunto se utiliza después de cada reajuste de parámetros para evaluar la convergencia en el desempeño del modelo y cuantificar la calidad de los parámetros establecidos hasta ese momento.*

Conjunto de prueba

Definición 2.3.7. *Es una porción del conjunto de datos utilizada para cuantificar el desempeño final del modelo y su capacidad para generalizar las relaciones aprendidas durante el entrenamiento. Por ende, es importante que ningún elemento de este subconjunto sea utilizado para entrenar o validar el modelo.*

Formalmente, el conjunto de datos está dado por la relación $\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\}$, donde se define $\mathcal{E}, \mathcal{V}, \mathcal{P} \subset \mathcal{D}$ como la familia de subconjuntos $P = \{\mathcal{E}, \mathcal{V}, \mathcal{P}\}$, que cumple con las siguientes condiciones:

- $\emptyset \notin P$ el conjunto vacío no pertenece a la familia de conjuntos.
- $\bigcup_{A \in P} A = \mathcal{D}$ la union de la familia de subconjuntos genera el conjunto de datos.
- $\forall A, B \in P \text{ tq } A \neq B \implies A \cap B = \emptyset$ los subconjuntos son mutuamente excluyentes.

Es decir, el conjunto de entrenamiento \mathcal{E} , el de validación \mathcal{V} y el de prueba \mathcal{P} forman una partición del conjunto de datos. El tamaño de cada uno de estos subconjuntos se define acorde a la cantidad de datos disponibles y la complejidad de la relación que representan.

Visión artificial

Definición 2.3.8. *También denominada **visión computarizada** es una rama de la inteligencia artificial y la informática que se ocupa del desarrollo de sistemas capaces de interpretar y analizar visualmente el mundo a través de imágenes o videos. El objetivo principal es permitir que las computadoras obtengan información valiosa a partir de datos visuales, de manera similar a cómo lo hacen los seres humanos.*

2.3.3. Tipos de errores

En el aprendizaje automático, se estudian y clasifican los tipos de errores que puede presentar un modelo según su naturaleza. Aquí, mencionamos los más comunes.

Error

Definición 2.3.9. Al ser una aproximación a una función objetivo, los modelos de aprendizaje automático no son 100% exactos o precisos. Todos presentan un grado de inexactitud, equivocación o error, que se evalúa a través de métricas que describen qué tan acertados son los modelos en el desempeño de una tarea específica. Uno de los principales objetivos al desarrollar un modelo de aprendizaje automático es minimizar dicho error.

Sesgo

Definición 2.3.10. *Bias* por su traducción al inglés, es una cuantificación resultante de la diferencia entre un resultado arrojado por el modelo entrenado y el resultado esperado, que debe ser previamente conocido.

Formalmente, dado un modelo $M(x|\theta)$ y un conjunto de datos $\mathcal{D} \subseteq \mathbb{X} \times \mathbb{Y}$, el sesgo es el valor dado por la métrica $\mu_s : M(\mathbb{X}|\theta) \times \mathbb{Y} \rightarrow [0, 1]$ que cumple:

$$\mu_s(\hat{y}_i, y_i) \approx 1 \text{ cuando } \hat{y}_i \approx y_i$$

$$\mu_s(\hat{y}_i, y_i) \approx 0 \text{ cuando } \hat{y}_i \not\approx y_i$$

donde $\hat{y}_i = M(x_i|\theta)$ es la predicción del modelo para el dato x_i .

Varianza

Definición 2.3.11. Error que mide la sensibilidad del modelo a las fluctuaciones en los datos de entrenamiento, es decir, la variabilidad de las predicciones del modelo respecto a diferentes conjuntos de datos.

Formalmente, siguiendo la notación dada en la definición 2.3.10, la varianza es el valor dado por la métrica $\mu_v : M(\mathbb{X}_1|\theta) \times M(\mathbb{X}_2|\theta) \rightarrow [0, 1]$, que cumple:

$$\mu_v(\hat{y}_i^1, \hat{y}_i^2) \approx 1 \text{ cuando } \hat{y}_i^1 \approx \hat{y}_i^2$$

$$\mu_v(\hat{y}_i^1, \hat{y}_i^2) \approx 0 \text{ cuando } \hat{y}_i^1 \not\approx \hat{y}_i^2$$

donde $\hat{y}_i^1 = M(x_i^1|\theta)$ y $\hat{y}_i^2 = M(x_i^2|\theta)$ son las predicciones del modelo para distintos conjuntos de datos tales que $x_i^1 \in \mathbb{X}_1$ y $x_i^2 \in \mathbb{X}_2$.

Un valor alto de varianza sugiere que el modelo es sensible a pequeñas variaciones. Reducir la varianza es esencial para lograr modelos confiables.

Estos errores están relacionados de tal forma que al aumentar el sesgo, se disminuye la varianza y viceversa. Por ende, un objetivo fundamental es la búsqueda de valores óptimos.

Ruido

Definición 2.3.12. Denominado *error irreducible*, es el error siempre presente en cualquier modelo de aprendizaje automático debido a la naturaleza no determinista de la aproximación. Este error se debe en parte a las limitaciones computacionales, como redondeos y representaciones numéricas, así como a las discrepancias asociadas a los datos, como errores en los mecanismos de medición, recolección, anotación, entre otros.

Dada la inevitable presencia de este error, siempre debe contemplarse, siendo fundamental definir mecanismos para su minimización.

Formalmente, el ruido está dado por $\hat{y}_i + \varepsilon_i = M(x_i|\theta) + \varepsilon_i$, donde ε_i pertenece a una distribución que depende de la naturaleza de los datos.

2.3.4. Tipos de ajustes

Ajuste

Definición 2.3.13. *Es la capacidad de un modelo para adaptarse de manera efectiva a los datos, de modo que pueda realizar predicciones o tomar decisiones precisas en situaciones diversas.*

Subajuste

Definición 2.3.14. *Ocurre cuando un modelo no logra capturar de manera adecuada las complejidades presentes en los datos. Esto suele suceder cuando el modelo es demasiado simple para representar la relación subyacente entre los datos y las etiquetas correspondientes.*

Sobreajuste

Definición 2.3.15. *Ocurre cuando un modelo se ajusta demasiado a los datos, capturando incluso el ruido y las fluctuaciones aleatorias en esos datos. Esto provoca que el modelo no generalice resultados a datos nuevos.*

En ambos casos, la capacidad de generalización de resultados para el modelo se ve afectada. En el subajuste, el modelo no es capaz de hacer predicciones factibles incluso en el conjunto de datos con el que fue entrenado. En el sobreajuste, el modelo puede tener buenos resultados solo en el conjunto de datos con el que se entrenó.

Para obtener información detallada sobre lo aquí expuesto, consulte [48].

2.3.5. Clasificación de algoritmos

Los algoritmos de aprendizaje automático pueden ser clasificados según características, siendo las más comunes el tipo de aprendizaje o el tipo de tarea que desempeñan. En esta sección, se hace referencia a las más comunes.

2.3.5.1. Aprendizaje supervisado

En este enfoque, se requiere un conjunto de datos previamente etiquetado. Para los fines de esta investigación, se hace énfasis en cuatro tareas que bajo este enfoque de aprendizaje son importantes: la clasificación, la regresión, la segmentación y la detección de objetos.

Formalmente, para este tipo de aprendizaje requerimos un conjunto de datos, tal como se describe en la definición 2.3.1, el cual está dado por

$$\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\}$$

Clasificación

Consiste en algoritmos para asignar categorías a los elementos de un conjunto de datos, con el propósito de hacer predicciones discretas.

Formalmente, la clasificación consiste en algoritmos que permiten aprender la aproximación funcional

$$\mathcal{D} \approx M(x|\theta) : \mathbb{X} \rightarrow \mathbb{Y} \text{ donde } x_i \in \mathbb{X} \text{ y } y_i \in \mathbb{Y} \subseteq \mathbb{N}.$$

Regresión

Consiste en algoritmos que permiten aproximar una función entre una variable dependiente y una o más variables independientes, con el propósito de hacer predicciones y estimaciones numéricas continuas.

Formalmente, la regresión consiste en algoritmos que permiten aprender la aproximación funcional

$$\mathcal{D} \approx M(x|\theta) : \mathbb{X} \rightarrow \mathbb{Y} \text{ donde } x_i \in \mathbb{X} \text{ y } y_i \in \mathbb{Y} \subseteq \mathbb{R}.$$

Segmentación

Son algoritmos referentes a la visión por computadora que se encargan de la tarea de dividir una imagen en subregiones de interés o **regions of interest (ROI)**, por su traducción al idioma inglés, basadas en ciertos criterios, como la pertenencia a una clase específica. El objetivo de estos algoritmos es asignar una etiqueta a cada píxel de la imagen, indicando a qué clase o categoría pertenece ese píxel, generando máscaras que representan las ROI. La segmentación de regiones de interés es fundamental en aplicaciones que requieren comprender la estructura y contenido visual de las imágenes.

Existen dos tipos principales de segmentación:

- **Segmentación Semántica:** El objetivo es asignar una etiqueta a cada píxel de la imagen que represente la clase a la que pertenece.
- **Segmentación de Instancias:** Además de asignar una etiqueta a cada píxel, se busca identificar y distinguir cada instancia individual de un objeto dentro de una misma clase.

Detección de objetos

Son algoritmos referentes a la visión por computadora que implican localizar y clasificar objetos específicos dentro de una imagen o un video, proporcionan información sobre la ubicación precisa de cada objeto mediante una caja delimitadora alrededor de cada objeto presente en la imagen.

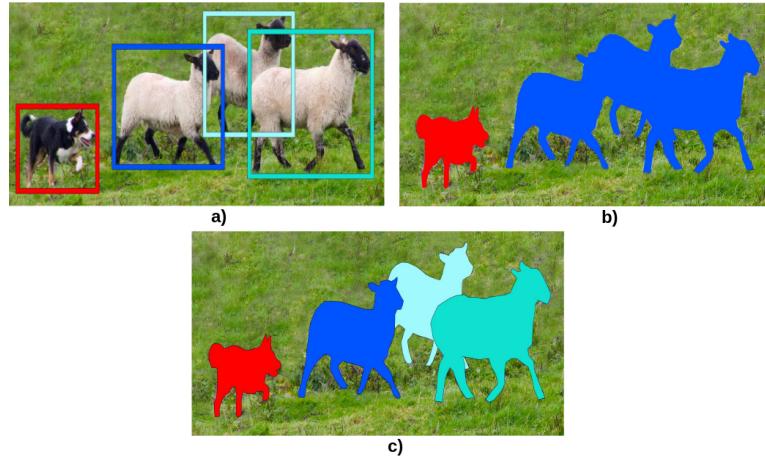


Figura 2.14: Segmentación y detección de objetos.

Figura tomada de [50], donde se destaca la diferencia entre la segmentación y la detección de objetos. a) Muestra recuadros delimitadores de colores diferentes para la detección de animales en la imagen. b) Muestra la segmentación semántica, donde se observan máscaras azules y rojas para cada clase de animal presente en la imagen. c) Muestra la segmentación de instancias, donde se observa una máscara para cada animal presente en la imagen.

2.3.5.2. Aprendizaje no supervisado

En este enfoque, se requiere un conjunto de datos que no cuenta con un etiquetado definido. Lo que se busca es descubrir patrones inherentes a la naturaleza de los datos. Se destacan dos tareas muy comunes que se pueden realizar con este enfoque de aprendizaje: la agrupación y la reducción de dimensiones.

Formalmente, para este tipo de aprendizaje requerimos un conjunto de datos, que a diferencia de la definición 2.3.1, no describe ninguna relación explícita:

$$D = \{x_1, \dots, x_n\} = \mathbb{X}$$

Agrupación

Consiste en algoritmos que permiten encontrar patrones y características de similitud en los datos, lo que permite particionar el conjunto de datos original en subconjuntos determinados por el mismo modelo.

Formalmente, la agrupación consiste en algoritmos que permiten aprender la asignación dada por:

$$M(x|\theta) : \mathbb{X} \rightarrow \mathcal{C} \text{ donde } x_i \in \mathbb{X} \text{ y } \mathcal{C} = \{C_1, \dots, C_j\} \text{ tal que } C_k \subset D \text{ para } 1 \leq k \leq j.$$

Reducción de dimensiones

Consiste en algoritmos que permiten acotar las características representativas de un conjunto de datos. No todas las características o variables relacionadas respecto a los datos son significativas, por lo tanto, los algoritmos buscan extraer las características principales. Es decir, buscan las mejores variables descriptivas para las relaciones inherentes a los datos.

Formalmente, si tenemos un conjunto de datos de alta dimensionalidad \mathbb{X} donde cada dato está descrito por m características, es decir, $x \in \mathbb{R}^m$, los algoritmos de

reducción de dimensiones buscan aprender una incrustación

$$M(x|\theta) : \mathbb{X} \subseteq \mathbb{R}^m \hookrightarrow \mathbb{X}' \subseteq \mathbb{R}^k \text{ donde } m < k.$$

Para obtener información detallada sobre la clasificación de modelos aquí expuesta, consulte [49].

2.3.6. Evaluación de modelos

Para determinar la calidad de las predicciones de un modelo de aprendizaje, necesitamos métricas que cuantifiquen su desempeño a la hora de generalizar resultados. En esta sección, se hace mención de algunas de las métricas más usadas que proporcionan una evaluación del desempeño de un modelo.

Matriz de confusión

Definición 2.3.16. *Es una herramienta intuitiva que permite determinar distintas métricas respecto a las predicciones de un modelo, tomando en cuenta las etiquetas verdaderas. Se implementa cuando el modelo predice distintas clases, representándose mediante una matriz de $n \times n$, donde n es el número de clases posibles. Esta matriz contiene los conteos de las predicciones posibles.*

Verdadero positivo (VP) y Verdadero negativo (VN):

También conocidos como **TP** y **TN** por sus respectivas abreviaciones en inglés, son los casos donde la clase predicha coincide con la clase real.

Falso positivo (FP) y Falso negativo (FN)

También conocidos como **Error tipo 1** y **Error tipo 2** respectivamente, son los casos donde la clase fue predicha falsamente.

Formalmente, una matriz de confusión es un elemento $C \in \mathbb{M}_n(\mathbb{N})$ donde $C_{i,j}$ es el número de instancias de la clase $1 \leq i \leq n$ que fueron clasificadas como clase $1 \leq j \leq n$.

		Clase predicha	
		Clase1	Clase2
Clase real	Clase1	VP	FN
	Clase2	FP	VN

Figura 2.15: **Matriz de confusión para $n = 2$ clases.**

Representación gráfica de una matriz de confusión para evaluar un modelo de clasificación binaria, donde **VP**, **FN**, **FP** y **VN** están dadas por las abreviaciones descritas en la definición 2.3.16.

A partir de la matriz de confusión, se definen distintas métricas $\mu : \mathbb{M}_n(\mathbb{N}) \rightarrow [0, 1]$ que sirven como indicadores de la calidad de las predicciones del modelo, donde el

valor 0 significa un pésimo desempeño y el 1 un excelente desempeño respecto a la métrica definida. Estas están dadas mediante las siguientes formulaciones.

Exactitud

Definición 2.3.17. *Conocida como **accuracy** por su nombre en inglés, representa la proporción de instancias correctamente clasificadas respecto al total de instancias en el conjunto de datos. En otras palabras, mide la fracción de predicciones correctas realizadas por el modelo. Sin embargo, la exactitud puede ser engañosa en situaciones donde las clases están desequilibradas, es decir, cuando alguna de las clases es exageradamente grande o pequeña en comparación con las otras. En esos casos, la exactitud no proporciona una medida representativa del rendimiento del modelo.*

$$\text{exactitud} = \frac{VP + VN}{VP + FP + FN + VN}$$

Precisión

Definición 2.3.18. *Mide la proporción de instancias correctamente clasificadas como positivas entre todas las instancias clasificadas como positivas por el modelo. En otras palabras, la precisión se centra en la calidad de las predicciones positivas del modelo. Proporciona información sobre la proporción de instancias positivas predichas que realmente son positivas. Es útil cuando la importancia de los falsos positivos es alta.*

$$\text{precisión} = \frac{VP}{VP + FP}$$

Sensibilidad

Definición 2.3.19. *Conocida como **Recall** por su nombre en inglés o como **tasa positiva verdadera**, mide la proporción de instancias positivas correctamente identificadas por el modelo en relación con el total de instancias que son realmente positivas. En otras palabras, la sensibilidad proporciona una medida de la capacidad del modelo para capturar todas las instancias positivas presentes en el conjunto de datos. Es especialmente útil en situaciones donde la importancia de falsos negativos es crítica.*

$$\text{sensibilidad} = \frac{VP}{VP + FN}$$

Especificidad

Definición 2.3.20. *Conocida como **tasa negativa verdadera**, mide la proporción de instancias correctamente clasificadas como negativas entre todas las instancias que son realmente negativas. Se centra en evaluar la capacidad del modelo para identificar correctamente las instancias negativas. Esta métrica es especialmente relevante en situaciones donde la importancia de los falsos positivos es alta y se busca minimizar la probabilidad de clasificar incorrectamente instancias negativas como positivas.*

$$\text{especificidad} = \frac{VN}{VN + FP}$$

Puntaje F1

Definición 2.3.21. Es una métrica que combina la precisión y la sensibilidad en un solo valor, proporcionando así una medida más equilibrada. Es particularmente útil cuando hay un desequilibrio entre las clases en el conjunto de datos. Se define como la media armónica de la precisión y la sensibilidad, penalizando más fuertemente los casos en los que una de las dos métricas es baja.

$$F1 = \frac{2 \text{Precisión} * \text{Sensibilidad}}{\text{Precisión} + \text{Sensibilidad}}$$

Para obtener información detallada sobre las métricas aquí expuestas, consulte [?].

Índice Jaccard

Definición 2.3.22. También conocido como **IoU** (Intersection over Union) o **Intersección sobre la Unión** por su traducción al español, es una métrica que mide el grado de similitud entre dos conjuntos, sea cual sea el tipo de elementos, es utilizada para evaluar la precisión de las predicciones en problemas de detección de objetos.

Dada una predicción del modelo $M(x_1|\theta) = \hat{y}_1$ y la respectiva etiqueta y_1 para el dato x_1 , donde $M(x|\theta)$ es un modelo para la detección o segmentación de objetos en imágenes, el índice de Jaccard está dado por:

$$IoU = \frac{|\hat{y}_1 \cap y|}{|\hat{y}_1 \cup y|}$$

Formalmente, $IoU : M(\mathbb{X}|\theta) \times \mathbb{Y} \rightarrow [0, 1]$, donde $IoU=0$ indica un resultado deficiente y $IoU=1$ indica una coincidencia exacta.

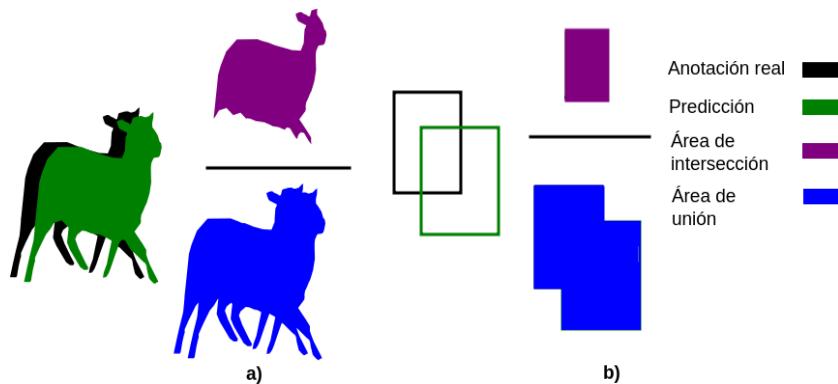


Figura 2.16: Índice Jaccard.

Representación gráfica del índice Jaccard. a) Muestra la ilustración del índice para el caso de una segmentación. b) Muestra la ilustración del índice para el caso de una detección.

Promedio de precisión

Definición 2.3.23. Denotado como **mAP** (mean average precision) por su connotación en el idioma inglés, es una métrica utilizada para evaluar el rendimiento de modelos en tareas de detección de objetos, especialmente en problemas en los que se detectan múltiples objetos en una imagen. Se utiliza para evaluar la precisión de la localización y la clasificación conjuntamente.

Para calcular esta métrica se procede de la siguiente manera:

- **Cálculo de la Precisión y el Recall por Clase:** Para cada clase, se calcula la precisión y el recall en diferentes umbrales de confianza.
- **Construcción de la Curva Precision-sensibilidad:** Para cada clase, se construye una curva que representa la relación entre la precisión y la sensibilidad al variar el umbral de confianza.
- **Cálculo del Área Bajo la Curva PR:** Se calcula el área bajo la curva para cada clase.
- **Promedio de las Áreas Bajo la Curva:** Se promedian las áreas bajo la curva para todas las clases para obtener la mAP.

Formalmente, la métrica $mAP : M(\mathbb{X}|\theta) \times \mathbb{Y} \rightarrow [0, 1]$, está dada por:

$$\frac{1}{|C|} \sum_{c \in C} \int_0^1 g(P_c, S_c) dx$$

donde C es el conjunto de clases que puede predecir el modelo y $g(P_c, R_c)$ representa la gráfica formada por la precisión y la sensibilidad de la clase c .

2.3.7. Redes neuronales artificiales

Las redes neuronales artificiales son modelos computacionales inspirados en el funcionamiento del cerebro humano, específicamente en la forma en que las neuronas interactúan en la red neuronal biológica. Estas redes son una parte fundamental del campo de la inteligencia artificial y el aprendizaje profundo.

2.3.7.1. Conceptos básicos

Red Neuronal

Definición 2.3.24. *Modelo representado mediante capas, donde cada capa se compone de un número dado de nodos (neuronas), que son representaciones de números almacenados. El modelo comienza con la capa de entrada y avanza hasta la capa de salida. Los valores almacenados en la capa n se utilizan para calcular los valores almacenados en los nodos de la capa número $n + 1$, a través de un proceso de entrenamiento que permite la actualización de los valores almacenados en la red mediante un proceso de optimización recursivo.*

Estos modelos están compuestos por grandes cantidades de capas, de ahí proviene el nombre de aprendizaje profundo con el que a menudo se denominan a estos modelos.

Este sistema es análogo a las redes biológicas de neuronas, donde una neurona transmite una señal eléctrica (activación) mediante un proceso llamado sinapsis, que permite propagar la señal eléctrica a las neuronas cercanas.

Para información detallada sobre neuronas biológicas, véase [69].

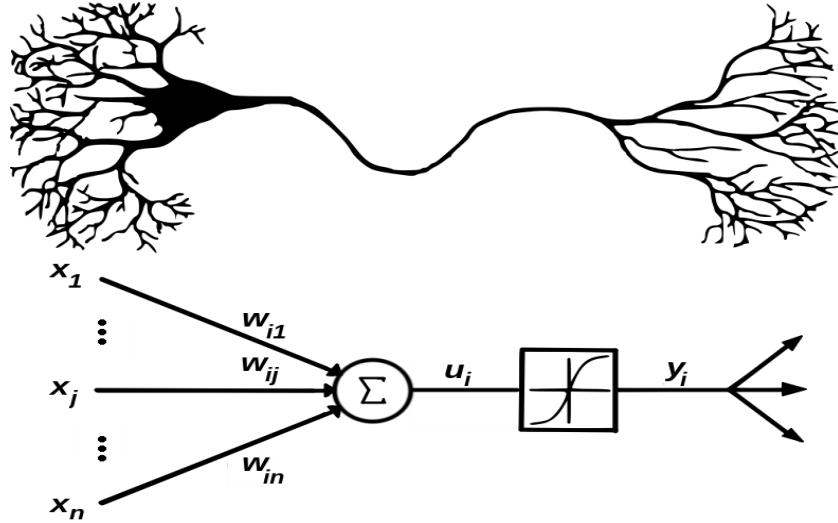


Figura 2.17: Neurona biológica vs artificial.

Imagen modificada de [78], se presenta una ilustración que compara una neurona biológica en la parte superior con la representación de una neurona artificial en la parte inferior. Esto resalta las similitudes en cuanto a su morfología.

Neurona

Definición 2.3.25. *Es la unidad básica de procesamiento que modela de manera simplificada las neuronas biológicas presentes en el cerebro humano. Cada neurona artificial realiza operaciones matemáticas en las entradas que recibe, produce una salida y contribuye al proceso de aprendizaje de la red.*

Una neurona artificial se conforma por entradas, pesos, sesgos, función de activación y salidas, es representada gráficamente como un nodo.

Capa

Definición 2.3.26. *Las redes neuronales están organizadas en capas, las cuales se definen como un conjunto de neuronas o unidades de procesamiento organizadas de manera conjunta en una estructura específica. Estas capas son los bloques fundamentales que componen una red neuronal y son responsables de realizar transformaciones en las entradas de la red.*

Las capas fundamentales para una red neuronal son:

- **Capa de Entrada:** Esta capa recibe las entradas del modelo, que pueden ser características, píxeles de una imagen, u otro tipo de datos. Cada neurona en esta capa representa una característica específica.
- **Capas Ocultas:** Son capas intermedias entre la capa de entrada y la capa de salida. Cada neurona en estas capas realiza operaciones matemáticas en sus entradas y contribuye a la representación y transformación de los datos. La presencia de múltiples capas ocultas hace que la red sea “profunda”.
- **Capa de Salida:** Esta capa produce las predicciones o resultados finales del modelo. La cantidad de neuronas en esta capa depende de la naturaleza del problema.

Conexión

Definición 2.3.27. *Se define como la relación entre las neuronas en diferentes capas de la red, que permiten la propagación de la información dentro de la red.*

La estructura de conexión es una parte esencial del diseño de una red neuronal y afecta su capacidad para aprender patrones complejos en los datos de entrada.

Peso

Definición 2.3.28. *Son los parámetros ajustables que la red neuronal utiliza para aprender a partir de los datos de entrada durante el proceso de entrenamiento. Cada conexión entre dos neuronas tiene asociado un peso que indica la fuerza y la dirección de la influencia de la salida de una neurona en la entrada de la siguiente.*

Sesgo

Definición 2.3.29. *Es un término que se refiere a un parámetro adicional asociado a cada neurona, aparte de los pesos. El sesgo proporciona a la red neuronal cierta flexibilidad y la capacidad de aprender patrones incluso cuando todas las entradas son cero.*

Función de activación

Definición 2.3.30. *Es una función no lineal que determina la salida de una neurona en función de su entrada ponderada. Estas funciones permiten a la red aproximar funciones que abstraen relaciones complejas en las entradas de la red.*

Función de pérdida

Definición 2.3.31. *Las funciones de pérdida o costo miden la discrepancia entre las predicciones del modelo y las salidas reales (o etiquetas) en el conjunto de datos de entrenamiento. Cuanto menor sea el valor de la función de pérdida, mejor será la calidad de las predicciones del modelo. Estas funciones son utilizadas durante cada iteración del entrenamiento para calcular la pérdida y ajustar los parámetros del modelo mediante algoritmos de optimización.*

Parámetro

Definición 2.3.32. *Se refiere a las variables internas ajustables que el modelo utiliza para realizar predicciones o inferencias a partir de los datos de entrada. Estos parámetros son los elementos que la red neuronal o el modelo de machine learning aprenden durante el proceso de entrenamiento.*

Hiperparámetro

Definición 2.3.33. *Son parámetros externos a un modelo de aprendizaje automático que no se aprenden durante el proceso de entrenamiento, sino que se establecen antes de iniciar el entrenamiento. Estos afectan el comportamiento y rendimiento del modelo; es decir, los hiperparámetros se eligen de antemano y generalmente se ajustan mediante prueba y error, búsqueda en cuadrícula u otras técnicas avanzadas.*

Formalmente, si tenemos k valores almacenados en las neuronas de la n -ésima capa de una red neuronal, a cada neurona en esta capa se le asigna un peso para cada nodo en la $(n + 1)$ -ésima capa. Así, $w_{a,b}^n \in \mathbb{R}$ representa el peso de la a -ésima neurona en la n -ésima capa ligado a la conexión con la b -ésima neurona de la $(n + 1)$ -ésima capa. De esta manera, podemos representar los pesos entre dos capas de una red neuronal de forma matricial mediante:

$$W_n = \begin{bmatrix} w_{1,1}^n & w_{1,2}^n & \cdots & w_{1,k}^n \\ w_{2,1}^n & w_{2,2}^n & \cdots & w_{2,k}^n \\ \vdots & \vdots & \ddots & \vdots \\ w_{j,1}^n & w_{j,2}^n & \cdots & w_{j,k}^n \end{bmatrix}$$

donde W_n es la matriz de pesos que conecta la capa n -ésima con la $(n + 1)$ -ésima capa.

Agregando el sesgo definido por una constante $b \in \mathbb{R}^j$ y la función de activación $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ utilizada para convertir el valor calculado por los pesos y sesgos en uno nuevo que se almacena en las neuronas la red.

Tenemos que para calcular el estado de la capa n -ésima, use tiene la expresión recursiva:

$$A_n = \sigma(W_n A_{n-1} + b_n) = \sigma \left(\begin{bmatrix} w_{1,1}^n & w_{1,2}^n & \cdots & w_{1,k}^n \\ w_{2,1}^n & w_{2,2}^n & \cdots & w_{2,k}^n \\ \vdots & \vdots & \ddots & \vdots \\ w_{j,1}^n & w_{j,2}^n & \cdots & w_{j,k}^n \end{bmatrix} \begin{bmatrix} a_1^{n-1} \\ a_2^{n-1} \\ \vdots \\ a_k^{n-1} \end{bmatrix} + \begin{bmatrix} b_1^n \\ b_2^n \\ \vdots \\ b_j^n \end{bmatrix} \right)$$

donde se ocupa la notación matricial

$$\sigma \left(\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,k} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{j,1} & a_{j,2} & \cdots & a_{j,k} \end{bmatrix} \right) = \begin{bmatrix} \sigma(a_{1,1}) & \sigma(a_{1,2}) & \cdots & \sigma(a_{1,k}) \\ \sigma(a_{2,1}) & \sigma(a_{2,2}) & \cdots & \sigma(a_{2,k}) \\ \vdots & \vdots & \ddots & \vdots \\ \sigma(a_{j,1}) & \sigma(a_{j,2}) & \cdots & \sigma(a_{j,k}) \end{bmatrix}.$$

Así, la función final calculada por una red de profundidad N está dada por la composición: $F(x) = \sigma(W_N \sigma(\dots \sigma(W_2 \sigma(W_1 x + b_1) + b_2) \dots) + b_N)$.

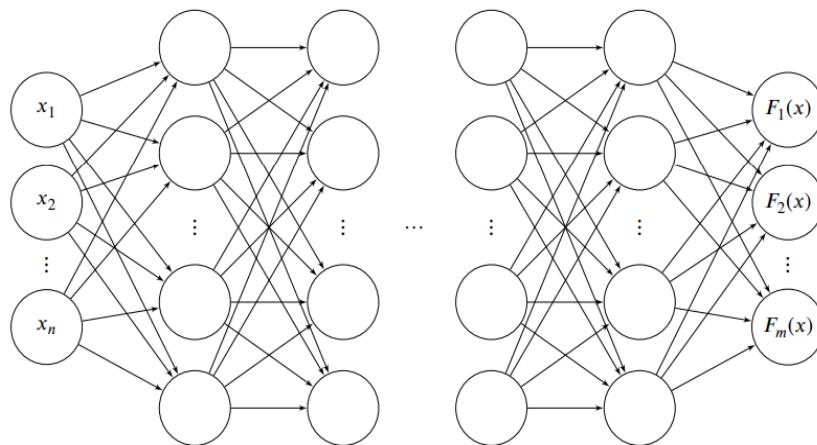


Figura 2.18: **Red neuronal artificial.**

Imagen tomada de [79] muestra la representación de una arquitectura básica de una red neuronal artificial.

2.3.7.2. Funciones de activación

La función de activación $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ es un concepto fundamental para las redes neuronales, ya que les proporciona la flexibilidad para aproximar funciones complejas que abstraen las relaciones de interés inherentes a los datos y su etiquetado. La elección de cuál utilizar depende de las características del problema a resolver, la experiencia y el nivel teórico del experto. En esta sección, se enlistan algunas de las funciones de activación más utilizadas en el contexto de redes neuronales.

Identidad

$$\sigma(x) = x$$

Escalón binario

$$\sigma(x) = \begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } x \geq 0 \end{cases}$$

Logística, sigmoide o escalón suave

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Tangente hiperbólica

$$\sigma(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Softplus

$$\sigma(x) = \ln(1 + e^x)$$

Gaussiana

$$\sigma(x) = e^{-x}$$

GELU (Unidad de error gaussiano lineal)

$$\sigma(x) = \frac{1}{2}x \left(1 + \frac{2}{\sqrt{\pi}} \int_0^{\frac{x}{\sqrt{2}}} e^{-t^2} dt \right)$$

ReLU (Unidad linealmente rectificada)

$$\sigma(x) = \begin{cases} 0 & \text{si } x \leq 0 \\ x & \text{si } x > 0 \end{cases}$$

ELU (Unidad exponencial lineal)

$$\sigma(x) = \begin{cases} \alpha(e^x - 1) & \text{si } x \leq 0 \\ x & \text{si } x > 0 \end{cases} \quad \text{donde } \alpha > 0$$

SELU (Unidad exponencial lineal escalada)

$$\sigma(x) = \begin{cases} \lambda\alpha(e^x - 1) & \text{si } x < 0 \\ \lambda x & \text{si } x \geq 0 \end{cases} \quad \text{donde } \alpha \approx 1.67326, \lambda \approx 1.0507$$

Leaky-ReLU (Unidad linealmente rectificada modificada)

$$\sigma(x) = \begin{cases} 0.01x & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases}$$

PReLU (Unidad linealmente rectificada parametrizada)

$$\sigma(x) = \begin{cases} \alpha x & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases} \quad \text{donde } \alpha > 0$$

SiLU (Unidad lineal sigmoide)

$$\sigma(x) = \frac{x}{1 + e^{-x}}$$

Softmax (máximo suave)

$$\sigma(\bar{x}) = \frac{e^{x_k}}{\sum_n e^{x_n}} \quad \text{donde } |\bar{x}| = N$$

Máxima salida

$$\sigma(\bar{x}) = \max_n \{x_n\}$$

Para información detallada sobre funciones de activación, véase [63].

2.3.7.3. Funciones de pérdida o costo

La función de pérdida, también conocida como función objetivo o función de costo, es una medida que cuantifica cuán bien un modelo de aprendizaje automático realiza en términos de la diferencia entre las predicciones del modelo y los valores reales de los datos de entrenamiento.

Formalmente, suponiendo que se desea aproximar una función $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ y una red $M(x|\theta)$ define una aproximación mediante una función $F_i : \mathbb{R}^n \rightarrow \mathbb{R}^m$.

Para cuantificar la calidad de la predicción dada por el algoritmo, se establece una función de costo que mide la discrepancia entre dos funciones. Dado un conjunto de validación o prueba con N datos $(x_i, y_i) \in \mathbb{X} \times \mathbb{Y}$, que son descritos por la relación $f(x_i) = y_i$, estos pueden ser proporcionados a la red para obtener predicciones $F_i(x_i) = \hat{y}_i$, de manera que la función de costo $C : \mathbb{X} \times M(\mathbb{X}|\theta) \rightarrow \mathbb{R}$ se define de tal forma que se cumpla que:

$$C(\hat{y}_i, y_i) \approx 1 \text{ cuando } \hat{y}_i \approx y_i$$

$$C(\hat{y}_i, y_i) \approx 0 \text{ cuando } \hat{y}_i \not\approx y_i$$

Al igual que con las funciones de activación, existen distintas funciones de costo y la elección de cuál utilizar depende de las características del problema a resolver, la experiencia y el nivel teórico del experto. En esta sección, se enlistan algunas de las funciones de costo más utilizadas en el contexto de redes neuronales, así como algunas de sus características básicas.

Sean $\vec{y} = \{y_1 \dots y_n\}$ es el conjunto de etiquetas de prueba y $\hat{\vec{y}} = \{\hat{y}_1, \dots, \hat{y}_n\}$ es el conjunto de predicciones hechas por el modelo para los datos de prueba, se definen las siguientes funciones de costo:

Raíz cuadrada media (RMSE)

Se entiende como residuos dados por la diferencia entre el valor predicho y el valor real obtenido.

Características

- Penaliza los valores que son muy grandes.
- No es fácil de interpretar.
- Funciona bien para optimizar regresiones en general.

$$C(\vec{y}, \hat{\vec{y}}) = \sqrt{\sum_{i=1}^n \frac{(\hat{y}_i - y_i)^2}{n}}$$

Para información detallada sobre RMSE, véase [65].

Error absoluto medio (MAE)

Es una medida de precisión y se calcula como la suma media de los valores absolutos de los errores, tabaja bajo la idea general de que es mejor tener pocos errores grandes que muchos errores pequeños.

Características

- Más difícil de derivar y de que converja.
- Penaliza menos los valores grandes.
- Es fácil de interpretar.

$$C(\vec{y}, \hat{\vec{y}}) = \sum_{i=1}^n \frac{|\hat{y}_i - y_i|}{n}$$

Para información detallada sobre MAE, véase [65].

Error absoluto medio escalado MASE

Es una medida de precisión similar al MAE pero esta contempla un factor de escala.

Características

- Difícil de derivar y de que converja.
- Escala univariante.
- Simétrica.
- Es fácil de interpretar.

$$C(\vec{y}, \hat{\vec{y}}) = \frac{\sum_{i=1}^n |\hat{y}_i - y_i|}{\frac{n}{n-1} \sum_{i=2}^n |\hat{y}_i - y_{i-1}|}$$

Para información detallada sobre MASE, véase [66].

Entropía cruzada categórica Categorical (Cross-Entropy)

La entropía cruzada categórica, es una medida de precisión para variables categóricas.

Características

- Difícil de derivar y de que converja.
- Escala univariante.
- Simétrica.
- Es fácil de interpretar.

$$C(\vec{y}, \hat{\vec{y}}) = -\frac{1}{n} \sum_{k=1}^n \sum_{i=1}^c y_i \log(\hat{y}_i)$$

Donde:

- i es la clase.
- c el número de clases.
- y_i la clase real.
- \hat{y}_i la clase predicha.

Para información detallada sobre entropía cruzada, véase [68].

Entropía cruzada binaria (Binary Cross-Entropy)

La entropía cruzada binaria, es una medida de precisión para variables binarias.

Características

- Difícil de derivar y de que converja.
- Escala univariante.
- Simétrica.
- Es fácil de interpretar.

$$C(\vec{y}, \hat{\vec{y}}) = -\frac{1}{n} \sum_{k=1}^n \sum_{i=1}^c [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)]$$

Donde:

- n número de ejemplos.
- c el número de clases.
- y_i la clase a predecir.
- p_i la probabilidad predicha para la clase i .

Para información detallada sobre entropía binaria , véase [70].

2.3.7.4. Optimizadores

Dada la interpretación como aproximadores de funciones, una red neuronal $M(x|\theta)$ debe ajustar sus predicciones de manera que matemáticamente garantice la mejora en la convergencia a la función de interés a modelar. Esto se logra a través de algoritmos que buscan minimizar la función de costo utilizada en el entrenamiento del modelo.

Estos algoritmos reciben el nombre de optimizadores y están basados en conceptos de cálculo multivariable. La idea general es la siguiente:

Dada una función diferenciable $f : \mathbb{R}^n \rightarrow \mathbb{R}$ definimos el gradiente $\nabla f : \mathbb{R}^n \rightarrow \mathbb{R}$ evaluado en un punto $\mathbf{x} \in \mathbb{R}^n$ mediante $\nabla f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_1}(\mathbf{x}), \dots, \frac{\partial f}{\partial x_n}(\mathbf{x}) \right)$, el cual indica la dirección del ascenso más pronunciado descrito por la función, dado que en el punto \mathbf{x} la función es continua y diferenciable puede obtenerse una linearización, por lo tanto, el vector $-\nabla f(\mathbf{x})$ indica la dirección del descenso más pronunciado descrito por la función.

Para obtener el valor mínimo de la función, se debe comenzar en un punto inicial $x_0 \in \mathbb{R}^n$, calcular el valor $-\nabla f(x_0)$, y luego proceder a calcular un nuevo punto $x_1 = x_0 - \lambda \nabla f(x_0)$, donde $\lambda \geq 0$ se denomina tasa de aprendizaje.

Este proceso continúa iterativamente hasta que nos acercamos a una región cercana a nuestro mínimo deseado, proporcionando así los parámetros adecuados para el modelo.

Los optimizadores ofrecen distintas estrategias de aproximación a los mínimos, todos basados en esta idea general. En esta sección, enlistamos algunos de los más populares. La elección de cuál utilizar depende del problema y la naturaleza de la función a minimizar.

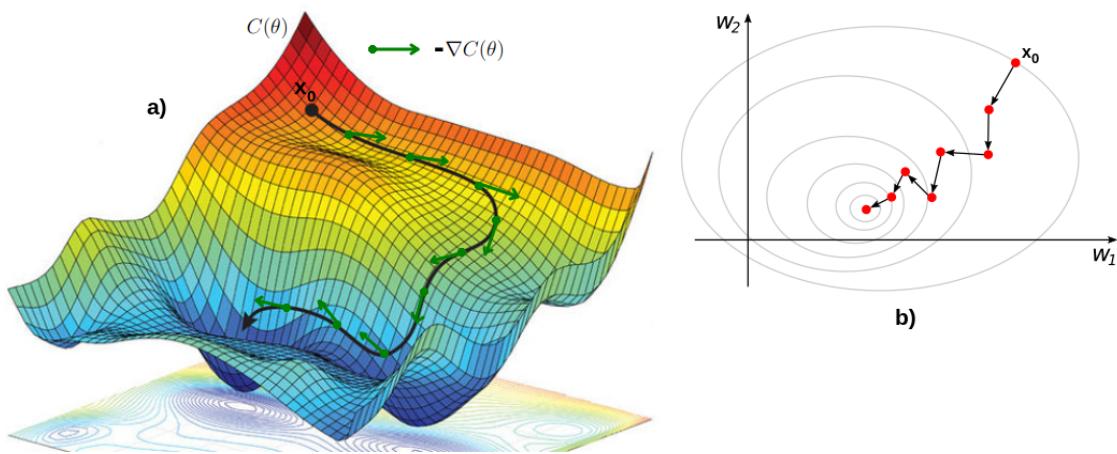


Figura 2.19: Descenso de gradiente.

Representación gráfica de la ejecución del algoritmo de descenso de gradiente. a) Ilustra una optimización sobre una superficie. b) Exhibe la ejecución del algoritmo en un gráfico de las curvas de nivel de la superficie.

Descenso estocástico de gradiente (SGD)

El término “estocástico” se refiere a que este algoritmo utiliza una muestra aleatoria de ejemplos de entrenamiento en cada iteración para calcular la dirección del descenso del gradiente.

Dada una función de costo $C(\theta)$, donde θ representa los parámetros del modelo, SGD actualiza los parámetros mediante la regla:

$$\theta_{t+1} = \theta_t - \alpha \nabla C(\theta_t)$$

Aquí, α es el parámetro de tasa de aprendizaje, que controla el tamaño del paso que se toma en la dirección opuesta al gradiente. El gradiente $\nabla C(\theta_t)$ indica la dirección y magnitud del mayor incremento en la función de costo en el punto θ_t . Al restar este gradiente ponderado por la tasa de aprendizaje, se realiza un descenso en la dirección que minimiza la función de costo.

En cada iteración de SGD, se selecciona un subconjunto aleatorio de muestras del conjunto de entrenamiento en lugar de usar todo el conjunto de entrenamiento completo (descenso de gradiente clásico). Esta aproximación estocástica hace que el algoritmo sea computacionalmente más eficiente y más adecuado para conjuntos de datos grandes.

El proceso de actualizar los parámetros mediante SGD se repite hasta que se cumpla un criterio de parada, como un número máximo de iteraciones o cuando la mejora en la función de costo es menor que un umbral predefinido.

Si bien SGD es efectivo, también presenta algunos desafíos. Debido a la naturaleza estocástica de la selección de lotes, el algoritmo puede converger más lentamente y puede quedar atrapado en óptimos locales. Para abordar estos problemas, han surgido variantes de SGD, como Momentum, Adam y RMSprop, que ajustan la dirección y el tamaño de los pasos de actualización para mejorar el rendimiento del algoritmo.

Para obtener información detallada sobre SGD, véase [71].

Descenso estocástico de gradiente con impulso (Momentum)

Es una variante del SGD que mejora la convergencia y suaviza el proceso de actualización de los parámetros durante el entrenamiento de redes neuronales.

El objetivo principal es minimizar una función de costo ajustando los parámetros del modelo. A diferencia del SGD estándar, este algoritmo utiliza un promedio acumulado de los gradientes calculados en la iteración previa para guiar las actualizaciones de los parámetros, lo que le permite tener en cuenta la historia de los gradientes.

La actualización de los parámetros en el SGD con Momentum se realiza mediante las siguientes fórmulas:

$$v_{t+1} = \mu v_t - \alpha \nabla C(\theta_t)$$
$$\theta_{t+1} = \theta_t + v_{t+1}$$

Aquí, α es el parámetro de tasa de aprendizaje, que determina el tamaño del paso en la dirección opuesta al gradiente, y μ es el coeficiente de momentum, que controla la contribución de los gradientes pasados en la actualización. El término $\nabla C(\theta_t)$ representa el gradiente de la función de costo en el punto θ_t .

El algoritmo comienza inicializando v_0 como un vector de ceros. En cada iteración, se calcula el promedio ponderado de los gradientes anteriores mediante μv_t , lo que introduce un impulso que mantiene la dirección de la actualización. Luego, se calcula el gradiente actual $\nabla C(\theta_t)$ y se realiza la actualización de los parámetros θ_{t+1} sumando el promedio acumulado v_{t+1} .

El uso de momentum en el algoritmo ayuda a suavizar las oscilaciones y el ruido inherentes al descenso del gradiente, lo que puede acelerar la convergencia. Además, el momentum también puede ayudar a escapar de óptimos locales y superar regiones de gradientes planos.

Es importante ajustar adecuadamente la tasa de aprendizaje α y el coeficiente de momentum μ para obtener un buen rendimiento. Un valor de μ cercano a 1 conserva en mayor medida la información de los gradientes pasados, mientras que un valor bajo de μ reduce esta influencia.

Para obtener información detallada sobre este método, véase [72].

Algoritmo de gradiente adaptativo (AdaGrad)

Se caracteriza por adaptar la tasa de aprendizaje de forma automática para cada parámetro del modelo. Este algoritmo ajusta la tasa de aprendizaje según la magnitud de los gradientes anteriores para acelerar la convergencia en dimensiones con gradientes escasos.

Se utiliza un conjunto de gradientes acumulados para adaptar la tasa de aprendizaje en cada paso de actualización de los parámetros. La actualización de los parámetros se realiza utilizando las siguientes fórmulas:

$$\begin{aligned} g_t &= \nabla C(\theta_t) \\ G_t &= G_{t-1} + g_t^2 \\ \theta_{t+1} &= \theta_t - \frac{\alpha}{\sqrt{G_t + \varepsilon}} * g_t \end{aligned}$$

Aquí, g_t es el gradiente de la función de costo C en el paso t , G_t es una matriz diagonal que contiene la suma acumulada de los gradientes al cuadrado hasta el paso t , θ_t representa los parámetros del modelo en el paso t , α es la tasa de aprendizaje inicial y ε es una pequeña constante para evitar divisiones por cero.

La idea principal es que las dimensiones con gradientes grandes suelen requerir pasos de aprendizaje más pequeños, mientras que las dimensiones con gradientes pequeños pueden permitirse pasos de aprendizaje más grandes.

El algoritmo logra esto adaptando la tasa de aprendizaje para cada dimensión en función de la historia acumulada de los gradientes.

Una de las ventajas es que no requiere ajustes manuales de la tasa de aprendizaje, ya que esta se adapta automáticamente durante el entrenamiento. Sin embargo, una limitación es que la acumulación de los gradientes al cuadrado en la matriz G_t puede hacer que la tasa de aprendizaje se vuelva demasiado pequeña a medida que aumentan las iteraciones, lo que puede frenar el proceso de aprendizaje.

Es un algoritmo popular en el entrenamiento de redes neuronales, especialmente en problemas donde los datos son escasos o donde las características tienen diferentes escalas. Proporciona una adaptación eficiente de la tasa de aprendizaje y puede ayudar a mejorar la convergencia y el rendimiento del modelo.

Para información detallada sobre el método, véase [73].

Propagación de raíz cuadrática media (RMSProp)

Este algoritmo ajusta la tasa de aprendizaje según la magnitud de los gradientes recientes para mejorar la convergencia en problemas con gradientes escasos o variables.

Se utiliza la media de los cuadrados de los gradientes para adaptar la tasa de aprendizaje en cada paso de actualización de los parámetros. La actualización de los parámetros se realiza utilizando las siguientes fórmulas:

$$\begin{aligned} g_t &= \nabla C(\theta_t) \\ E[g^2]_t &= \beta E[g^2]_{t-1} + (1 - \beta)g_t^2 \\ \theta_{t+1} &= \theta_t - \frac{\alpha}{\sqrt{E[g^2]_t + \varepsilon}} * g_t \end{aligned}$$

Aquí, g_t es el gradiente de la función de costo C en el paso t , $E[g^2]_t$ es una media de los cuadrados de los gradientes hasta el paso t , θ_t representa los parámetros del modelo en el paso t , α es la tasa de aprendizaje inicial, β es el factor de decaimiento y ε es una pequeña constante para evitar divisiones por cero.

La idea principal es ajustar la tasa de aprendizaje para cada parámetro según la magnitud de los gradientes recientes. Al utilizar una media de los cuadrados de los gradientes, da mayor importancia a los gradientes más recientes y reduce la influencia de los gradientes pasados. Esto permite que la tasa de aprendizaje se adapte de forma más sensible a las variaciones en el paisaje de la función de costo.

Proporciona una adaptación eficiente de la tasa de aprendizaje y puede mejorar el rendimiento del modelo en problemas con gradientes variables o escasos.

Para información detallada sobre el método, véase [74].

Optimización de momento adaptativo (Adam)

Este algoritmo combina las ventajas del algoritmo RMSProp y el Momentum. Adam adapta la tasa de aprendizaje de forma automática para cada parámetro del modelo y mantiene una estimación de los momentos de primer y segundo orden de los gradientes.

Se utilizan dos momentos para adaptar la tasa de aprendizaje en cada paso de actualización de los parámetros. Los momentos de primer y segundo orden se calculan utilizando las siguientes fórmulas:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)g_t^2$$

Donde g_t es el gradiente de la función de costo J en el paso t , m_t es el momento de primer orden y v_t es el momento de segundo orden.

Para corregir el sesgo de los momentos en las primeras iteraciones, se utilizan los siguientes pasos de corrección de sesgo:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

Luego, los parámetros se actualizan utilizando la siguiente fórmula:

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\hat{v}_t} + \varepsilon} \hat{m}_t$$

Aquí, θ_t representa los parámetros del modelo en el paso t , α es la tasa de aprendizaje inicial, β_1 y β_2 son factores de decaimiento, y ε es una pequeña constante para evitar divisiones por cero.

Combina el uso de momentos de primer y segundo orden para adaptar la tasa de aprendizaje de forma eficiente. Los momentos de primer orden ayudan a estabilizar la dirección de las actualizaciones, mientras que los momentos de segundo orden escalan la tasa de aprendizaje de acuerdo con la magnitud de los gradientes.

Su capacidad para adaptar la tasa de aprendizaje y mantener estimaciones de los momentos de primer y segundo orden lo convierte en un algoritmo robusto para una variedad de problemas.

Para información detallada sobre le metodo, véase [74].

AdaDelta

Es un algoritmo que adapta la tasa de aprendizaje de forma automática para cada parámetro del modelo. A diferencia de otros algoritmos, no utiliza una tasa de aprendizaje fija, sino que ajusta la tasa de aprendizaje según el historial de actualizaciones anteriores.

Se utiliza una media de los cuadrados de las actualizaciones anteriores para adaptar la tasa de aprendizaje en cada paso de actualización de los parámetros. La actualización de los parámetros se realiza utilizando las siguientes fórmulas:

$$g_t = \nabla C(\theta_t)$$

$$E[g^2]_t = \rho E[g^2]_{t-1} + (1 - \rho)g_t^2$$

$$\Delta\theta_t = -\frac{\sqrt{\Delta\theta_{t-1} + \epsilon}}{\sqrt{E[g^2]_t + \epsilon}} * g_t$$

$$\begin{aligned}\theta_{t+1} &= \theta_t + \Delta\theta_t \\ E[\Delta\theta^2]_t &= \rho E[\Delta\theta^2]_{t-1} + (1 - \rho)\Delta\theta_t^2\end{aligned}$$

Aquí, g_t es el gradiente de la función de costo C en el paso t , $E[g^2]_t$ es la media de los cuadrados de los gradientes hasta el paso t , θ_t representa los parámetros del modelo en el paso t , $\Delta\theta_t$ es la actualización de los parámetros en el paso t , ε es una pequeña constante para evitar divisiones por cero y ρ es un factor de decaimiento.

La idea principal es que la tasa de aprendizaje se adapte según el historial de actualizaciones anteriores. En lugar de utilizar una tasa de aprendizaje fija, utiliza la relación entre las actualizaciones recientes y los gradientes pasados para ajustar la tasa de aprendizaje de forma adaptativa.

Una de las ventajas es que elimina la necesidad de ajustar manualmente la tasa de aprendizaje, ya que esta se adapta automáticamente durante el entrenamiento. Además, puede ayudar a superar los problemas de convergencia lenta o explosiva que pueden ocurrir con tasas de aprendizaje fijas.

Para información detallada sobre el método, véase [77].

2.3.7.5. Retropropagación

Al trabajar con aproximaciones de funciones, no solo se requiere que las predicciones del modelo sean óptimas. Si la función de costo muestra una diferencia grande entre las predicciones y el comportamiento deseado de la red, se necesita hacer un reajuste a los parámetros θ del modelo $M(x|\theta)$. Esto se logra en el proceso de entrenamiento mediante el método conocido como **retropropagación**, que es una técnica de optimización que permite que la red ajuste sus parámetros en función del error cometido en las predicciones, con el objetivo de mejorar el rendimiento del modelo.

El principal desafío de aplicar el descenso de gradiente a las redes neuronales es calcular las derivadas parciales de la función de costo con respecto a cada peso y sesgo individual, es decir, $\frac{\partial C}{\partial w_{i,j}}$ y $\frac{\partial C}{\partial b_j}$.

Aquí es donde entra la retropropagación. Este algoritmo nos ayuda a calcular estos valores para la última capa de conexiones y, con estos resultados, avanza de manera inductiva hacia atrás a través de la red, calculando las derivadas parciales de cada capa hasta llegar a la primera capa de la red. De ahí su nombre.

Por cuestiones de simplicidad, se considera la función de costo sobre un solo dato etiquetado x .

Dado que solo se sabe cómo calcular ∇C_x para un punto x , se utiliza la relación

$$\nabla C = \nabla \left(\sum_{i=1}^N C_{x_i} \right) = \sum_{i=1}^N \nabla C_{x_i}.$$

lo que permite llevar a cabo este proceso para cada punto y sumar los valores del gradiente. Esto es importante porque utilizar la retropropagación en un conjunto de entrenamiento grande para cada iteración del entrenamiento se convierte en un enfoque computacionalmente costoso. En cambio, si se escogen algunos elementos

del conjunto de entrenamiento para calcular el gradiente y se actualiza la red, se obtienen resultados satisfactorios a través de un proceso recursivo.

Sea $z_j^l = \sum_k w_{j,k}^l a_k^{l-1} + b_j^l$ donde $a_j^l = \sigma(z_j^l)$ y $A^l = \sigma(Z^l)$ representan los valores enviados por la capa $(n - 1)$ -ésima antes de aplicar la función de activación y $Z^l := \sum_k z_k^l e_k$ es un vector con entradas correspondientes a los valores z_j^l y e_k son vectores de la base canónica.

Considerando $\delta_j^l = \frac{\partial C}{\partial z_j^l}$ y $\Delta^l = \sum_k \delta_k^l e_k$, estos valores son útiles para propagar el algoritmo hacia atrás a través de la red y están directamente relacionados con $\frac{\partial C}{\partial w_{i,j}}$ y $\frac{\partial C}{\partial b_j}$ mediante la regla de la cadena :

$$\frac{\partial C}{\partial w_{i,j}^l} = \frac{\partial C}{\partial z_j^l} \frac{\partial z_j^l}{\partial w_{i,j}^l} = \delta_j^l a_i^{l-1} \text{ y } \frac{\partial C}{\partial b_j^l} = \frac{\partial C}{\partial z_j^l} \frac{\partial z_j^l}{\partial b_j^l} = \delta_j^l$$

Dado que a_i^{l-1} es un valor disponible para cualquier nodo de la red, si se calcula el valor de δ_j^l , se resuelve el problema de calcular el gradiente.

El primer paso es calcular este valor para la última capa de la red, es decir δ_j^L , para una red con L capas. Dado que, nuevamente por la regla de la cadena se tiene

$$\delta_j^L = \frac{\partial C}{\partial a_j^L} \frac{\partial a_j^L}{\partial z_j^L} = \frac{\partial C}{\partial a_j^L} \sigma'(z_j^L)$$

observando que $F(x) = A^L = (a_1^L, \dots, a_k^L)$ y $f(x) = (y_1, \dots, y_k)$ tenemos

$$\delta_j^L = (a_j^L - y_j) \sigma'(z_j^L)$$

Lo cual se calcula fácilmente mediante una computadora. Expresando este resultado como vector se tiene

$$\Delta^L = \nabla_{A^L} \odot \sigma'(Z^L).$$

Donde $\nabla_{A^L} = \left(\frac{\partial C}{\partial a_1^L}, \dots, \frac{\partial C}{\partial a_k^L} \right)$ es el gradiente de C tomado con respecto a los elementos de A^L y \odot es el Producto de Hadamard, definido como:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,k} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{j,1} & a_{j,2} & \cdots & a_{j,k} \end{bmatrix} \odot \begin{bmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,k} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ b_{j,1} & b_{j,2} & \cdots & b_{j,k} \end{bmatrix} = \begin{bmatrix} a_{1,1}b_{1,1} & a_{1,2}b_{1,2} & \cdots & a_{1,k}b_{1,k} \\ a_{2,1}b_{2,1} & a_{2,2}b_{2,2} & \cdots & a_{2,k}b_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{j,1}b_{j,1} & a_{j,2}b_{j,2} & \cdots & a_{j,k}b_{j,k} \end{bmatrix}.$$

Ahora solo resta propagar este procedimiento hacia atrás en la red para obtener δ_j^{L-1} , esto se logra aplicando la regla de la cadena

$$\delta_j^{L-1} = \frac{\partial C}{\partial z_j^{L-1}} = \nabla_{Z^L} C \cdot \frac{\partial Z^L}{\partial z_j^{L-1}} = \sum_i^k \frac{\partial C}{\partial z_i^L} \frac{\partial z_i^L}{\partial z_j^{L-1}} = \sum_i^k \delta_i^L \frac{\partial z_i^L}{\partial z_j^{L-1}}.$$

Si nos concentramos en el término $\frac{\partial z_i^L}{\partial z_j^{L-1}}$ tenemos

$$\begin{aligned}\frac{\partial z_i^L}{\partial z_j^{L-1}} &= \frac{\partial}{\partial z_j^{L-1}} \left(\sum_k w_{i,k}^L a_k^{L-1} + b_i^L \right) = \frac{\partial}{\partial z_j^{L-1}} \left(\sum_k w_{i,k}^L \sigma(z_k^{L-1}) + b_i^L \right) = \\ &\frac{\partial}{\partial z_j^{L-1}} (w_{i,j}^L \sigma(z_j^{L-1})) = w_{i,j}^L \sigma'(z_j^{L-1})\end{aligned}$$

Lo cual, nuevamente, se calcula fácilmente por una computadora. Por lo tanto

$$\delta_j^{L-1} = \sum_i \delta_i^L w_{i,j}^L \sigma'(z_j^{L-1})$$

Esta fórmula nos indica cómo calcular cualquier δ_j^l , conociendo Δ^{l+1} . Dado que sabemos cómo calcular Δ^L en la última capa de la red, tenemos completo el algoritmo de retropropagación.

2.3.7.6. Teoremas de aproximación.

Por mera formalidad matemática, enunciaremos y proporcionaremos un esbozo de la demostración de los siguientes teoremas, donde el primero nos garantiza que, dada una función de activación que cumple ciertas propiedades, puede ser utilizada por una red neuronal para aproximar funciones continuas. El segundo permite extender el resultado a funciones de activación que son continuas.

Teorema 2.3.1.

Sea f una función discriminatoria continua. Entonces, una red neuronal con f como función de activación es un逼近ador universal.

Demostración.

Sea n un número natural. Decimos que una función de activación $f: \mathbb{R} \rightarrow \mathbb{R}$ es n -discriminatoria si la única medida signada de Borel μ tal que

$$\int f(y \cdot x + \theta) d\mu(x) = 0 \quad \text{para todo } y \in \mathbb{R}^n \text{ y } \theta \in \mathbb{R}$$

es la medida cero.

Así, decimos que f es discriminatoria si es n -discriminatoria para cualquier $n \in \mathbb{N}$.

Sabiendo que dado un espacio topológico Ω y una función $f: \mathbb{R} \rightarrow \mathbb{R}$, definimos a una red neuronal con función de activación f como un逼近ador universal en Ω si se cumple que $\Sigma_n(f)$ es denso en $C(\Omega)$.

Donde $C(\Omega) = \{f: \Omega \rightarrow \mathbb{R} | f \in C^1\}$, es el conjunto de funciones continuas de Ω a \mathbb{R} y $\Sigma_n(f) = \text{Gen}\{f(y \cdot x + \theta) | y \in \mathbb{R}^n, \theta \in \mathbb{R}\}$, donde $y \cdot x$ representa el producto punto estándar en \mathbb{R}^n . El conjunto $\Sigma_n(f)$ consta de todas las funciones que pueden ser calculadas por una red neuronal con una sola capa oculta y función de activación f .

Buscando la contradicción, supongamos que $\Sigma_n(f)$ no es denso en $C(I_n)$ donde $I_n = [0, 1]^n = \{x = (x_1, \dots, x_n) \in \mathbb{R}^n | x_i \in [0, 1] \text{ para cualquier } i = 1, \dots, n\}$. Por definición de densidad, se sigue que $\Sigma_n(f) \neq C(I_n)$.

Luego, apelando a el teorema de Hahn-Banach que dice que dado un espacio vectorial normado $(V, \|\cdot\|)$ y dos conjuntos $A, B \subseteq V$ no vacíos, cerrados, disjuntos y convexos, tal que uno de ellos es compacto. Entonces, existe un funcional lineal continuo $f \neq 0$, algún $\alpha \in \mathbb{R}$ y un $\varepsilon > 0$ tal que $f(x) \leq \alpha - \varepsilon$ para cualquier $x \in A$ y $f(y) \geq \alpha + \varepsilon$ para cualquier $y \in B$.

Esto nos garantiza que dado un espacio vectorial normado real $(V(\mathbb{R}), \|\cdot\|)$ y $U \subseteq V(\mathbb{R})$ un subespacio lineal tal que $U \neq V$. Entonces, existe una aplicación lineal continua $f: V \rightarrow \mathbb{R}$ con $f(x) = 0$ para cualquier $x \in U$, y $f \not\equiv 0$. (*)

Dado que $I_n \subset \mathbb{R}$ cumple con las hipótesis anteriores, se puede concluir que existe algún funcional lineal continuo $F: C(I_n) \rightarrow \mathbb{R}$ tal que $F \neq 0$ pero $F(g) = 0$ para cualquier $g \in \Sigma_n(f)$.

Usando el Teorema de Representación de Riesz que nos dice, que dado Ω un subconjunto de \mathbb{R}^n y $F: C(\Omega) \rightarrow \mathbb{R}$ un funcional lineal en el espacio de funciones reales continuas con dominio en Ω . Entonces, existe una medida Borel con signo μ en Ω tal que para cualquier $f \in C(\Omega)$, tenemos que

$$F(f) = \int_{\Omega} f(x) d\mu(x).$$

garantizamos que existe alguna medida de Borel μ tal que

$$F(g) = \int_{I_n} g(x) d\mu(x) \text{ para todo } g \in C(I_n). \quad (*)$$

Sin embargo, dado que para cualquier y y θ la función $f(y \cdot x + \theta)$ es un elemento de $\Sigma_n(f)$, esto significa que para todo $y \in \mathbb{R}^n$, $y \theta \in \mathbb{R}$ tenemos $\int f(y \cdot x + \theta) d\mu(x) = 0$, lo que significa que $\mu = 0$ (ya que f es discriminatorio) y, por lo tanto, $F(g) = 0$ para cualquier $g \in C(I_n)$.

Lo cual contradice claramente la afirmación (*) por lo que se debe tener que $\overline{\Sigma_n(f)} = C(I_n)$, que por definición garantiza la densidad, con esto concluye la prueba. ■

Teorema 2.3.2.

Aproximación universal

Una red neuronal artificial con una función de activación $f: \mathbb{R} \rightarrow \mathbb{R}$ continua, es un aproximador universal.

Demostración.

Tomando una función $f: \mathbb{R} \rightarrow \mathbb{R}$, tal que $f \in C^1$ por el teorema anterior basta demostrar que si $\Sigma_1(f)$ es denso en $C([0, 1])$, entonces $\Sigma_n(f)$ es denso en $C([0, 1]^n)$.

Para ello ocupemos el hecho de que el espacio generado por el conjunto $A = \{g(a \cdot x) \mid a \in \mathbb{R}^n, g \in C([0, 1])\}$ es denso en $C([0, 1]^n)$.

Esto significa que dada una función $h \in C([0, 1]^n)$, existe una función en A tal que la distancia entre estas dos es mínima.

Formalmente sea $h \in C([0, 1]^n)$ y $\varepsilon > 0$, existe una función $g_k \in C([0, 1])$ tal que

$$\left| h(x) - \sum_{k=1}^N g_k(a_k \cdot x) \right| < \frac{\varepsilon}{2}$$

Para cada función $g_k(a_k \cdot x)$ como se tiene la hipótesis de que $\Sigma_1(f)$ es denso en $C([0, 1])$, se concluye que para toda g_k , existe una suma de funciones tal que cumple con

$$\left| g_k(a_k \cdot x) - \sum_{i=1}^{N_k} f(y_{k,i} \cdot x + \theta_{k,i}) \right| < \frac{\varepsilon}{2k}.$$

Aplicando la desigualdad del triángulo $|x + y| \leq |x| + |y|$ tenemos

$$\left| h(x) - \sum_{k=1}^N \sum_{i=1}^{N_k} f(y_{k,i} \cdot x + \theta_{k,i}) \right| < \left| h(x) - \sum_{k=1}^N g_k(a_k \cdot x) \right| + \frac{k(\varepsilon/2k)}{2k} < \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

Esto demuestra que podemos aproximarnos arbitrariamente a cualquier función en $C([0, 1]^n)$ usando funciones de $\Sigma_n(f)$, con lo que concluye la demostración. ■

Para más detalles sobre los conceptos de análisis funcional usados en este teorema, véase [60].

Para más detalles sobre las matemáticas detrás de las redes neuronales, véase [79].

2.3.7.7. Transferencia de conocimiento

Son técnicas que permiten utilizar el conocimiento de redes neuronales previamente entrenadas en un tarea específica para mejorar el rendimiento en una tarea relacionada o diferente.

Este enfoque es especialmente útil cuando existen problemas como conjuntos de datos limitados o cuando ya se ha entrenado un modelo en una tarea similar la que se quiere realizar.

Las técnicas más usadas son:

- **Transferencia de Características:** Consiste en la reutilización de las representaciones aprendidas por un modelo en una tarea específica para otra tarea relacionada.
- **Transferencia de Modelo:** Implica reutilizar un modelo preentrenado en una tarea similar para inicializar los pesos de un modelo en una nueva tarea. Este enfoque puede acelerar el proceso de entrenamiento y mejorar el rendimiento.
- **Transferencia de Aprendizaje:** Se refiere a la transferencia de conocimiento desde una tarea fuente a una tarea objetivo. El modelo se entrena inicialmente en una tarea fuente y luego se adapta o ajusta a la tarea objetivo. Este enfoque es particularmente útil cuando las tareas comparten similitudes en la estructura de los datos o en los patrones subyacentes.
- **Transferencia de Conjunto de Datos** Implica transferir el conocimiento al intercambiar o combinar conjuntos de datos entre tareas relacionadas. Esto puede ser útil cuando se tiene acceso a datos más ricos o más abundantes en una tarea y se utilizan para mejorar el rendimiento en otra tarea.

De manera general, para aplicar la mayoría de las técnicas de transferencia de conocimiento, es necesario contar con un modelo preentrenado $M_1(x|\theta) \approx f$, el cual exhibe un rendimiento óptimo al desempeñar la tarea, caracterizada por una precisa aproximación a la función f .

Geometricamente tenemos:

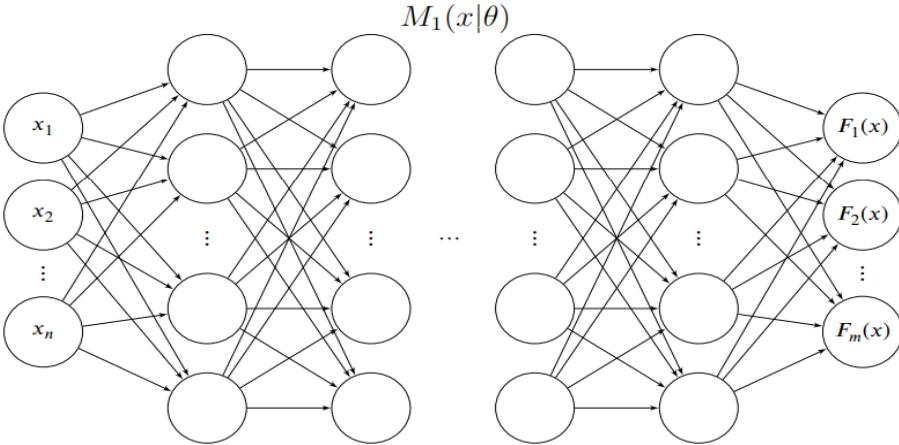


Figura 2.20: **Red neuronal preentrenada.**

Representación gráfica da una red neuronal artificial preentrenada $M_1(x|\theta)$

Si la red preentrenada cuenta con N capas, el siguiente paso consiste en sustituir la primera y la N -ésima capa, es decir, la de entrada y la de salida, por aquellas que se adecuen a las estructuras de los nuevos datos $\{(x'_1, y'_1), \dots, (x'_n, y'_n)\} \subseteq \mathbb{X} \times \mathbb{Y}$ que vamos a utilizar para reentrenar la red.

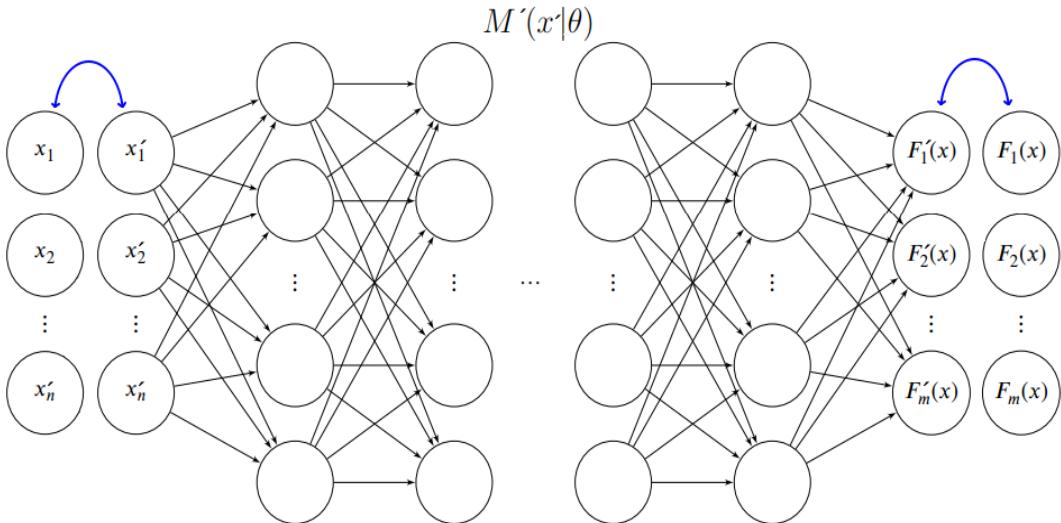


Figura 2.21: **Red neuronal adecuada.**

Representación gráfica de una red neuronal artificial preentrenada $M_1(x|\theta)$ a la que se le cambió la capa de salida y la de entrada para generar la nueva red $M'(x'|\theta)$.

Finalmente, se elige la técnica de transferencia de conocimiento a utilizar, lo que se traduce en seleccionar los parámetros $\theta' \subseteq \theta$ que se ocuparán para inicializar el modelo y definir cuáles serán reajustados en el reentrenamiento y cuáles se mantendrán fijos o congelados, esto es, hacemos un reentrenamiento para el modelo $M'(x'|\theta')$ que dará paso al modelo final con transferencia de conocimiento $M_{\text{tf}}(x|\theta)$.

2.3.7.8. Arquitecturas de redes neuronales

Las redes neuronales permiten definir una cantidad enorme de configuraciones que se establecen según la disposición dada por las relaciones entre capas y conexiones que componen la red. La elección de esta configuración es crucial para el rendimiento de la red en una tarea específica.

Estas configuraciones se denominan arquitecturas, ya que cada una se representa mediante una gráfica particular formada por nodos y vértices que representan las neuronas, las capas y conexiones presentes en el modelo, lo que induce una arquitectura de red.

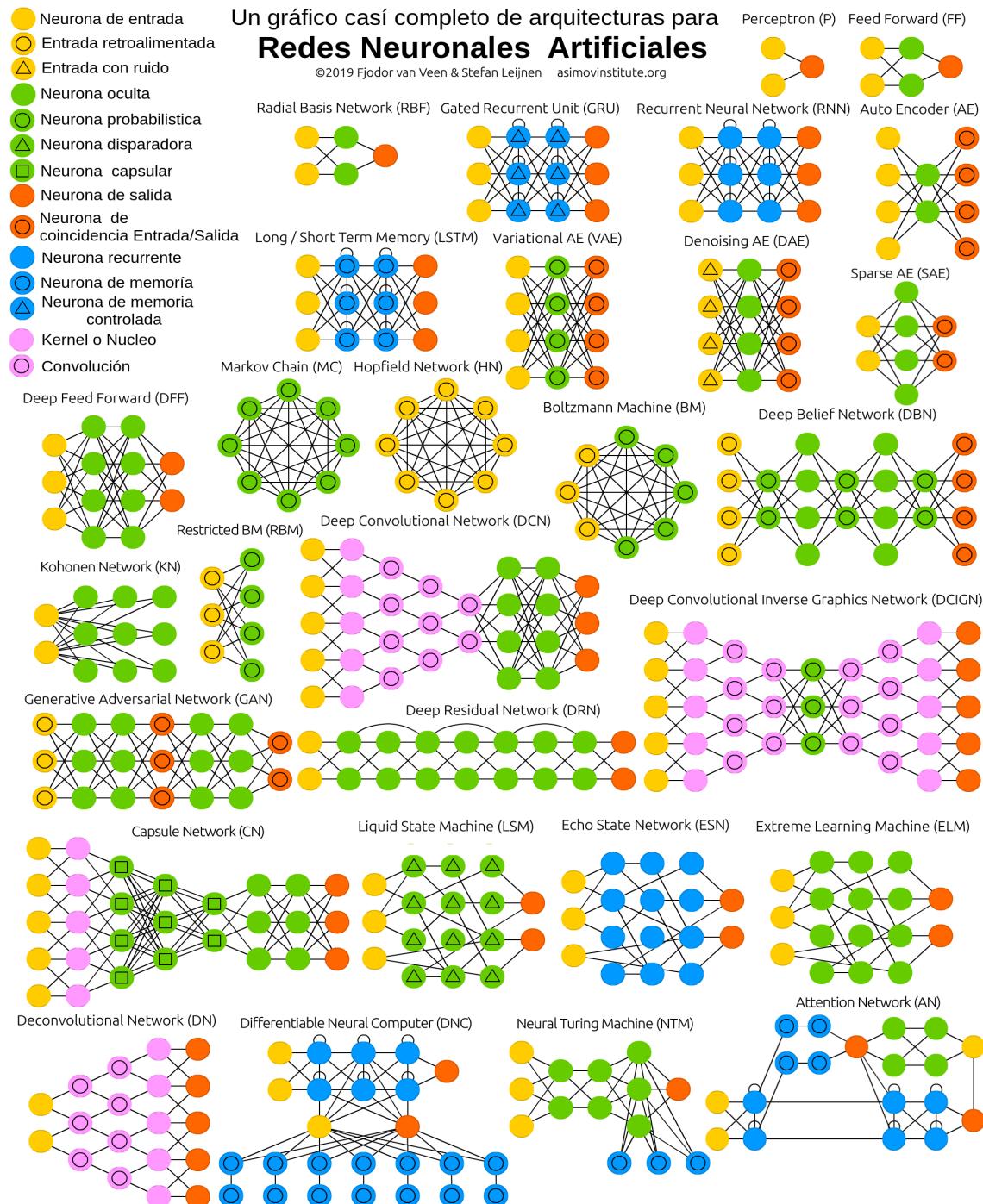


Figura 2.22: **Arquitecturas de redes neuronales.**
 Imagen modificada de [82], que muestra de forma gráfica algunas de las arquitecturas más comunes y sus aspectos relevantes.

Capítulo 3

Estado del arte

En este capítulo se exploran las técnicas e investigaciones previas al presente trabajo, que están directamente relacionadas o presentan un antecedente importante para el desarrollo de los objetivos propuestos en el presente trabajo.

3.1. Aprendizaje profundo para detección y clasificación de objetos

En esta sección se hace referencia a algunas de las herramientas de aprendizaje profundo en el ámbito de visión computacional desarrolladas para la detección de objetos y clasificación de imágenes, cuyo impacto ha sido relevante en el campo.

3.1.1. YOLO

YOLO: *You Only Look Once*, por su nombre en inglés que se traduce como solo necesitas mirar una vez al español, es una familia de modelos de visión artificial basados en convoluciones que se utilizan en diversas tareas, como la detección, segmentación y seguimiento de objetos, entre otros.

Ultralytics, ofrece implementaciones de estos modelos en cuatro tamaños de acuerdo a la profundidad: pequeña (s), mediana (m), grande (l) y extra grande (x), cada una de las cuales proporciona tasas de precisión progresivamente más altas. Cada variante también requiere un tiempo diferente para ser entrenada.

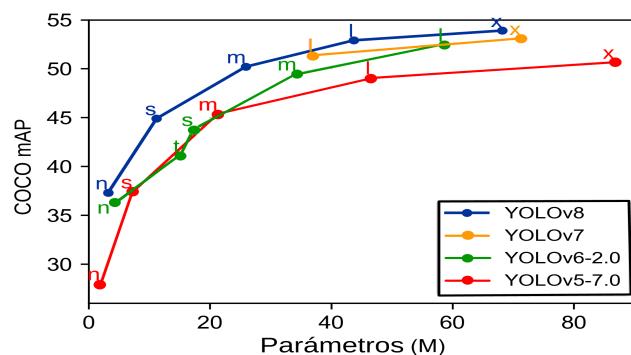


Figura 3.1: Tamaños de YOLO.

Imagen modificada de [51], que muestra la evaluación del mAP en el conjunto de entrenamiento COCO objetos para distintas versiones y tamaños de la implementación de YOLO por ultralytics.

Arquitectura de YOLO

Esta arquitectura hace uso de la generación de características a partir de imágenes de entrada. Posteriormente, estas características se introducen en un sistema de predicción que crea cuadros alrededor de los objetos y anticipa sus clases. La red YOLO se compone de tres partes principales:

Columna vertebral: Se trata de una red neuronal convolucional que extrae características de la imagen a diferentes niveles de granularidad, es decir, a distintos niveles de detalle.

Cuello: Son una serie de capas que pueden mezclar y combinar las características extraídas de las imágenes para pasárlas a una predicción.

Cabeza: Accede a las características del cuello y lleva a cabo la regresión, lo cual conduce a la predicción de los cuadros delimitadores y la clase del objeto presente en la imagen.

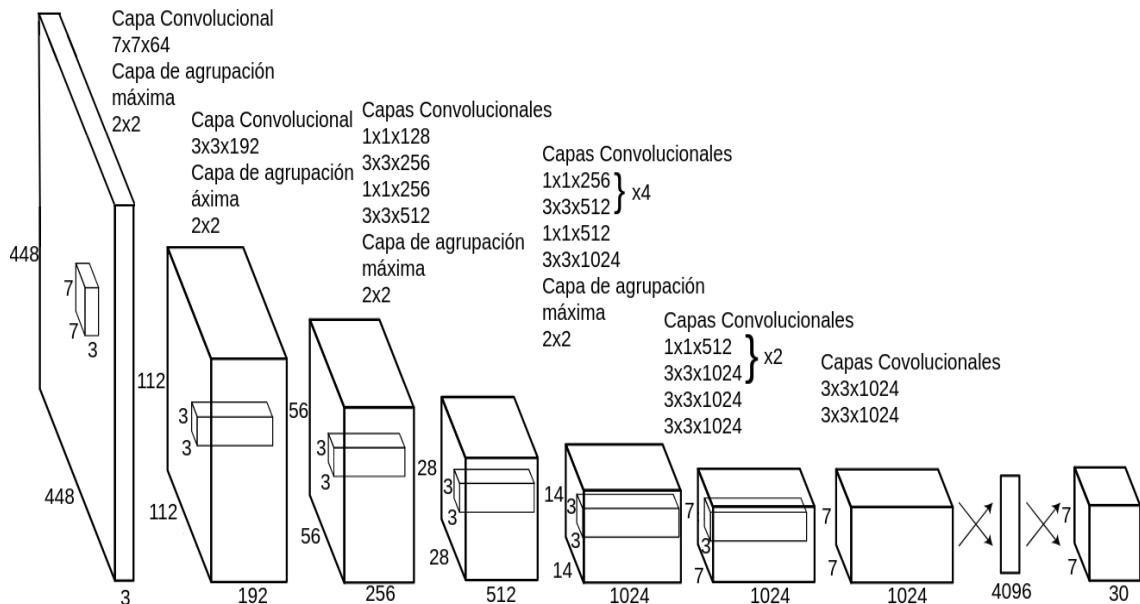


Figura 3.2: Arquitectura de YOLO.

Imagen modificada de [13], donde se muestra una representación gráfica de las arquitectura YOLO y las transformaciones internas de una imagen en la red.

YOLOv8 incluye funciones de aumento de datos que aplica transformaciones geométricas a los datos para una mayor variedad, su columna vertebral esta definida por una red DenseNet [29].

Detección de objetos de YOLO

Para llevar a cabo la detección de objetos, el algoritmo comienza dividiendo la imagen en una cuadrícula de $n \times n$ subregiones.

Para cada subdivisión, se generan N posibles cuadros delimitadores con el objetivo de detectar objetos. Estos cuadros se ponderan según la probabilidad de encerrar un objeto. De esta manera, se calculan un total de $n \times n \times N$ cuadros delimitadores, los cuales son posteriormente discriminados mediante un umbral mínimo de probabilidad.

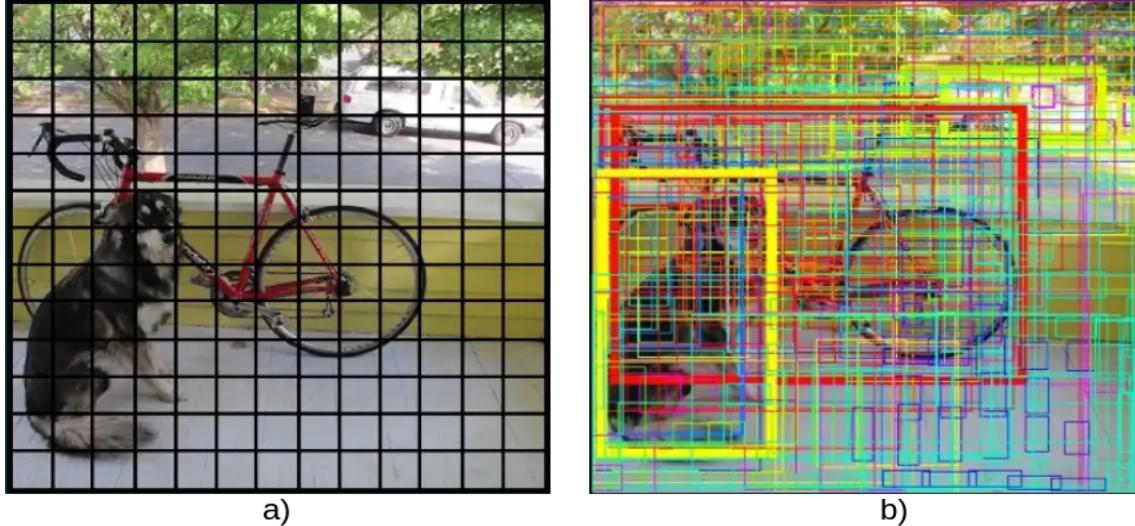


Figura 3.3: **Algoritmo de detección.**

Imagen tomada de [13], donde se ilustra el algoritmo de detección para YOLO. a) Muestra la división de la imagen original en subregiones cuadradas. b) Muestra la generación de cuadros delimitadores que han superado el umbral establecido.

Los cuadros que no fueron descartados por el umbral pasan por un segundo proceso denominado supresión de no máximos, en el cual, a partir de los cuadros delimitadores con la ponderación más alta, se realiza la unión que elimina los traslapes existentes entre ellos para mejorar la detección.

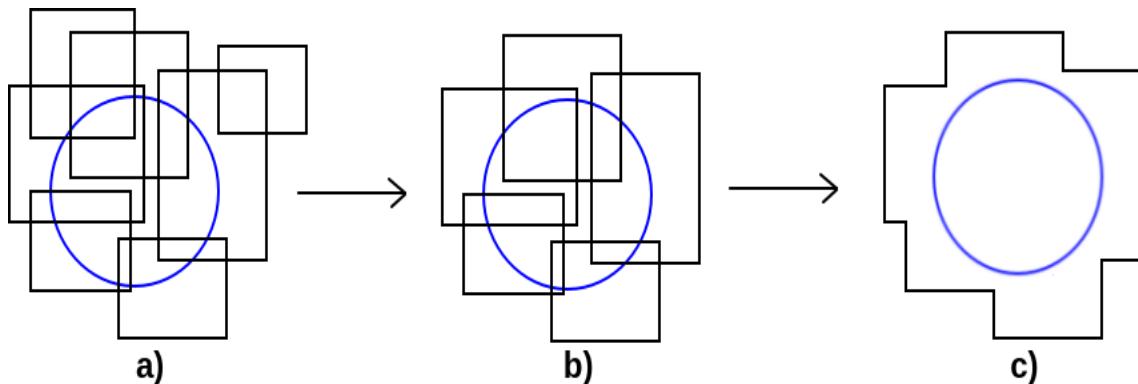


Figura 3.4: **supresión de no máximos.**

Imagen ilustrativa del proceso de supresión de no máximos para la detección del objeto representado por el círculo azul, utilizada por YOLO para mejorar la detección de objetos. a) Muestra los recuadros delimitadores inicialmente propuestos por el modelo. b) Muestra los recuadros delimitadores después de eliminar los menos probables. c) Muestra la imagen de la unión de los traslapes de los recuadros más probables.

Finalmente, se obtienen los cuadros delimitadores con una probabilidad más alta de contener un objeto que posteriormente pasan a ser clasificados.

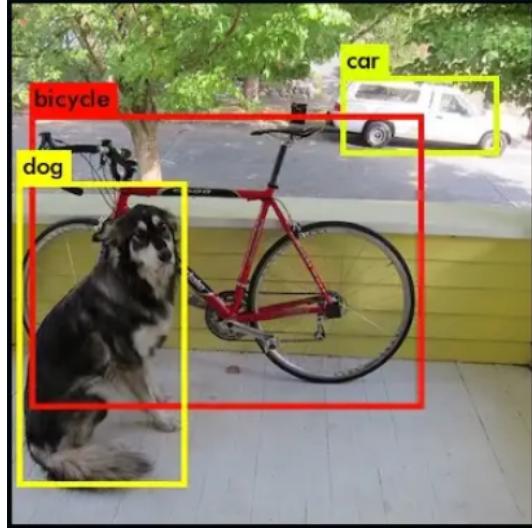


Figura 3.5: Predicción final de YOLO.

Imagen tomada de [13], donde se muestra la detección y clasificación lograda por YOLO. En la imagen, se observa la correcta detección y clasificación de un perro, una bicicleta y un auto.

El algoritmo de detección se puede resumir gráficamente mediante el siguiente esquema.

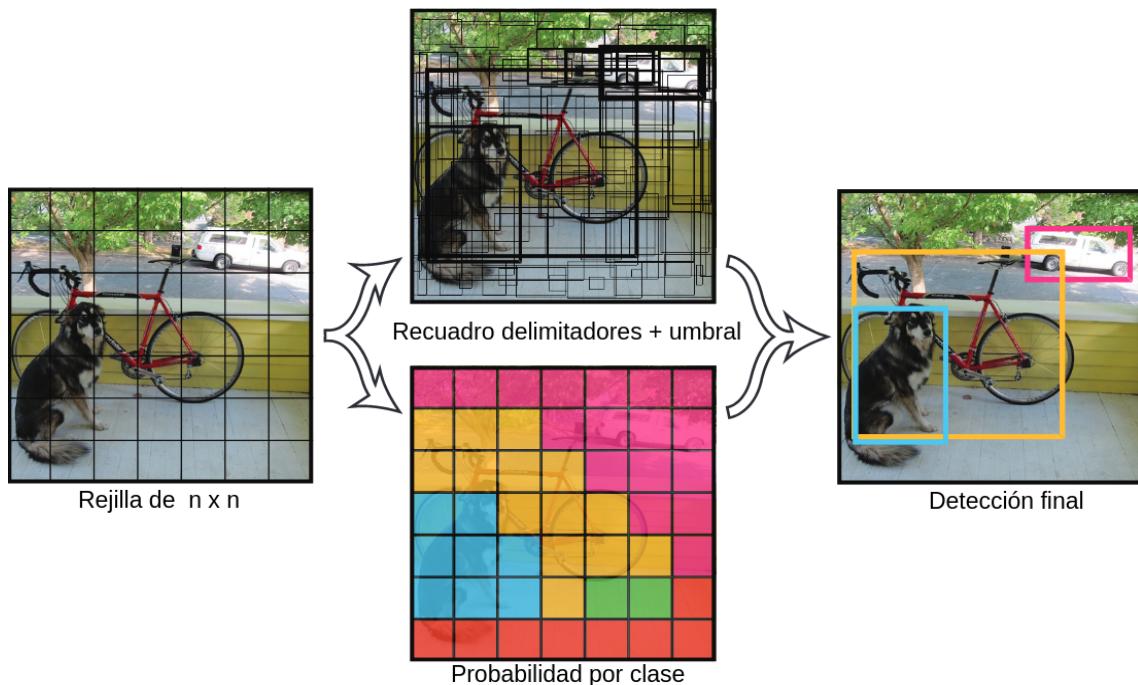


Figura 3.6: Detección YOLO.

Imagen tomada de [13], donde se muestra un resumen gráfico del algoritmo de detección que ejecuta YOLO.

Para artículo e información completa sobre YOLO, véase [12] , [37] y [13]

3.1.2. Transformador de visión (Vision transformer)

Transformador clásico (transformer)

El modelo de aprendizaje profundo en el que se basan los transformadores de visión tuvo sus orígenes en el procesamiento de lenguaje natural, específicamente para traducciones automáticas y se basa bajo el concepto de atención. Posteriormente, fue adaptado y aplicado en la detección de objetos, clasificación de imágenes y tareas relacionadas debido a sus resultados destacados en estos contextos. A continuación, se presenta una breve descripción de su funcionamiento:

En este modelo, las entradas son secuencias de palabras, cada palabra se transforma mediante un proceso de incrustación de vectores. Esta técnica implica convertir cada palabra en un vector compuesto por números reales. Estos vectores pertenecen a un espacio vectorial multidimensional donde la proximidad entre vectores refleja la relación semántica entre las palabras que representan, y viceversa.

A cada palabra representada en este espacio vectorial se le agrega un vector que codifica la posición de la palabra en la secuencia.

Cada vector entra en un codificador, que es una capa que transforma dichas entradas en representaciones aprendidas por el modelo, de tal manera que le permite tener más en cuenta el contexto de las palabras sin perder la información posicional de estas; este proceso recibe el nombre de **atención**.

El modelo cuenta con un decodificador que es una capa diseñada para prestar atención únicamente a las palabras de posiciones anteriores a la que recibe como entrada. Así, la predicción del modelo depende de las palabras que le preceden en la secuencia. El vector de entrada para el decodificador también se agrega a la información de la posición de la palabra en la secuencia de la misma manera que lo hace el codificador.

El decodificador produce como salidas palabras que pasan por una capa denominada Softmax, la cual tiene la capacidad de asignar probabilidades. Estas probabilidades se utilizan para predecir la próxima palabra en la secuencia. En otras palabras, el modelo realiza predicciones de palabras basándose en el contexto de las palabras que la preceden.

Formalmente, el modelo $T(x|\theta)$ se caracteriza por tener la capacidad de aprender una función incrustadora $i : W \hookrightarrow \mathbb{R}^M$, donde W es un conjunto de palabras. Si las palabras $w_1, w_2 \in W$ tienen una relación semántica denotada por $w_1 \sim w_2$, entonces la distancia entre sus representaciones vectoriales es más pequeña que con otras palabras. Es decir, si $w_1, w_2, w_3 \in W$ cumplen $w_1 \sim w_2$ y $w_1 \not\sim w_3$, entonces para $i(w_1) = x_1, i(w_2) = x_2, i(w_3) = x_3 \in \mathbb{R}^M$, se tiene que $|x_1 - x_2| \leq |x_1 - x_3|$.

También tiene la capacidad de aprender una función codificadora $C : \mathbb{R}^M \rightarrow \mathbb{R}^{M+p}$ y su respectiva función decodificadora $D : \mathbb{R}^{M+p} \rightarrow \mathbb{R}^{M+p'}$, de tal manera que para una palabra $w \in W$, se tiene que $D(i(w)) \in \mathbb{R}^{M+p}$ representa la información referente al contexto y posición de la palabra.

Para artículo sobre transformador, véase [10]

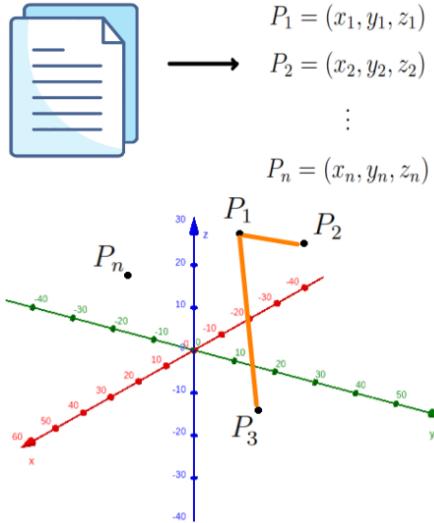


Figura 3.7: Transformador.

Imagen representativa del proceso llevado a cabo por el transformador, ilustra la transformación de n palabras en un espacio tridimensional donde se muestra la distancia entre dos palabras.

Transformador de visión

Se trata del modelo adaptado de un transformador para tareas de visión por computadora. La forma en que procede es muy similar al transformador original; solo es necesario adaptar los conceptos utilizados para palabras ahora a imágenes.

Dado que el transformador original tiene como entrada un vector unidimensional que representa una secuencia de palabras, en este caso se requiere adaptar el proceso para ejecutarse en matrices, que es la estructura que representa una imagen.

Se comienza dividiendo la imagen en parches (*patches*), lo cual implica tomar subimágenes de tamaño fijo. Este enfoque emula la separación de secuencias de texto en palabras y se realiza de la siguiente manera:

Suponiendo que la imagen cuenta con $n \times m$ píxeles y una cantidad c de canales, se subdivide la imagen en una cantidad $N = \frac{n \times m}{p^2}$ de submatrices de dimensión $p \times p \times c$.

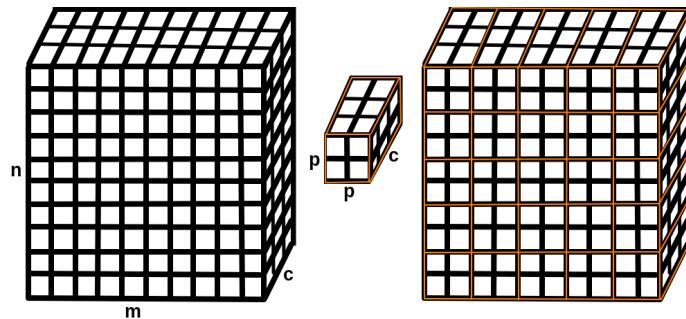


Figura 3.8: División en parches.

Ilustración que representa la división de una imagen con dimensiones de $10 \times 10 \times 3$ píxeles. Se realiza una subdivisión en $\frac{10 \cdot 10}{2^2} = 25$ submatrices de dimensiones $2 \times 2 \times 3$.

Dada la subdivisión, cada parche pasa por un proceso denominado aplanado, donde se transforma de una submatriz de dimensiones $p \times p \times c$ a un vector unidimensional de longitud $1 \times p \cdot p \cdot c$.

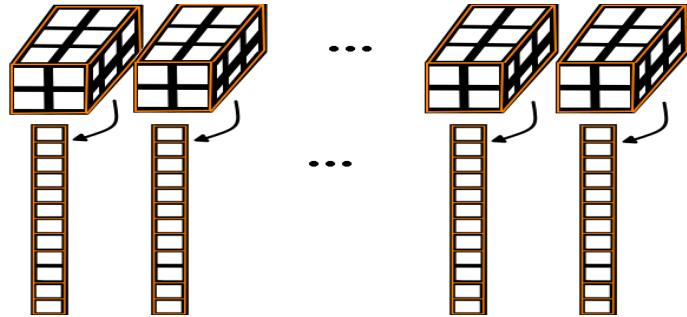


Figura 3.9: Aplanado.

Este esquema que proporciona una representación visual clara de cómo se transforma cada parche de la imagen en un vector unidimensional mediante la concatenación de píxeles, lo que facilitaría la comprensión del proceso de aplanado.

Cada uno de estos vectores unidimensionales se proyecta linealmente u ortogonalmente, lo que permite representar cada vector en un espacio de dimensión menor. Esto genera una secuencia de vectores que pueden ser utilizados para aprender la incrustación, similar a la descrita en el transformador para lenguaje natural. Posteriormente, estos vectores pasan por las capas codificadora y decodificadora, respectivamente. En este contexto, el vector posición que se agrega en estas capas se refiere a la posición del parche con respecto a la imagen.

Formalmente, dado un espacio vectorial normado $(V, \|\cdot\|)$ y dos vectores $\vec{u}, \vec{w} \in V$, definimos la proyección ortogonal de \vec{w} sobre \vec{u} como:

$$\text{proj}_{\vec{u}}(\vec{w}) = \frac{\vec{u} \cdot \vec{w}}{\|\vec{u}\|^2} \cdot \vec{u}.$$

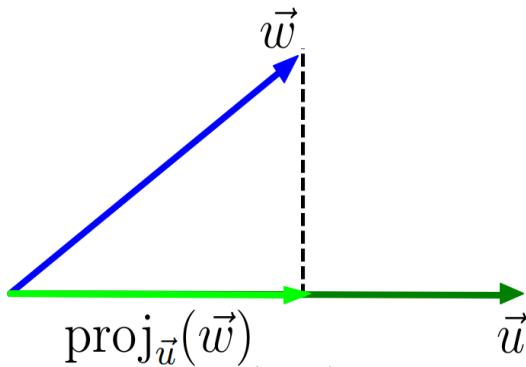


Figura 3.10: Proyección ortogonal.

Esquema que representa la proyección del vector \vec{w} sobre el vector \vec{u} . Nótese que geométricamente esto se entiende como la sombra que proyecta el vector \vec{w} sobre \vec{u} .

Finalmente, se presenta un esquema gráfico de las capas mencionadas del transformador de visión.

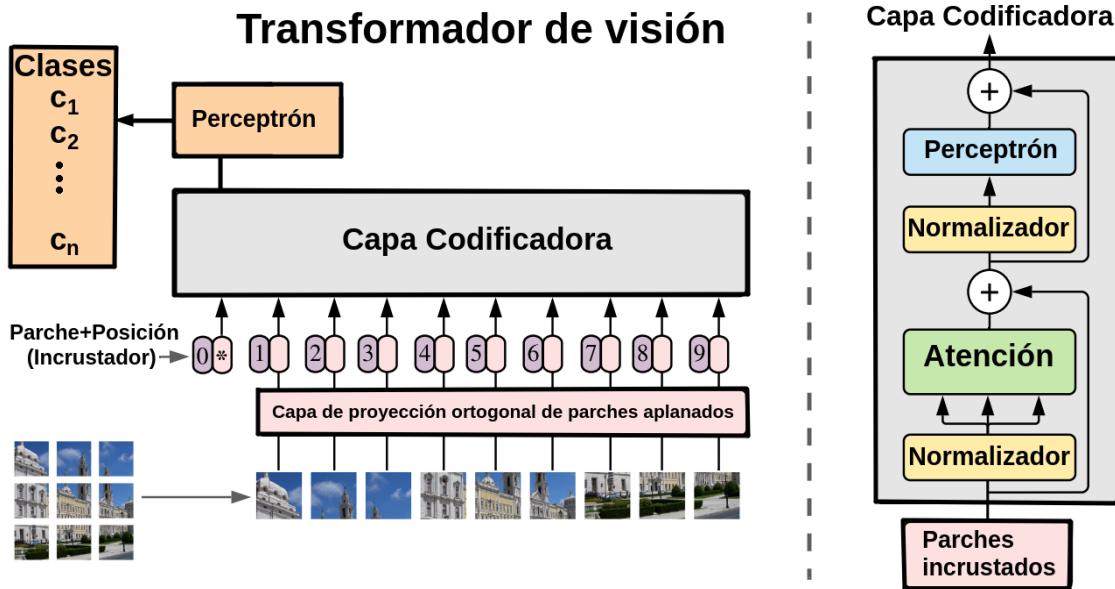


Figura 3.11: Transformador de visión.

Imagen modificada de [11], donde se ilustran las capas anteriormente mencionadas de la arquitectura del Transformer de visión.

Para información detallada sobre el transformador de visión, véase [11]

3.1.3. Redes neuronales convolucionales

Conforman una de las principales herramientas para la visión computacional. Su nombre proviene del hecho de que cuentan con capas que realizan una operación matemática llamada **convolución**. De manera general, permiten extraer características de una imagen.

Cada píxel que conforma la imagen sirve de entrada a una neurona específica de la red. De esta forma, la red puede aprender características y relaciones de interés entre píxeles de la imagen.

El problema esencial de una red neuronal no convolucional surge al analizar imágenes cuando se necesita clasificar dos objetos de la misma clase pero que no están centrados de la misma manera dentro de la imagen. A la hora de predecir, la red tendrá un mal desempeño en el intento de comparar las características de la imagen con las que ha aprendido, puesto que los píxeles presentarán una distribución distinta a la que la red aprendió.

Este problema se resuelve mediante la implementación de la convolución en redes neuronales, ya que brinda la capacidad de extraer características de una forma parecida a como lo hacen los humanos. Es decir, en el ejemplo dado, lo que caracteriza al número 9 no es la información, relación ni la posición que se tiene en términos de píxeles, sino características morfológicas claras que lo distinguen de otros números. Se conforma por un círculo y una línea que se unen de una forma particular, esta característica es inherente al número e independiente del tamaño o la posición de este con respecto a la imagen. La convolución permite a una red neuronal aprender estas características y compararlas de una forma similar a la que lo haría un humano:

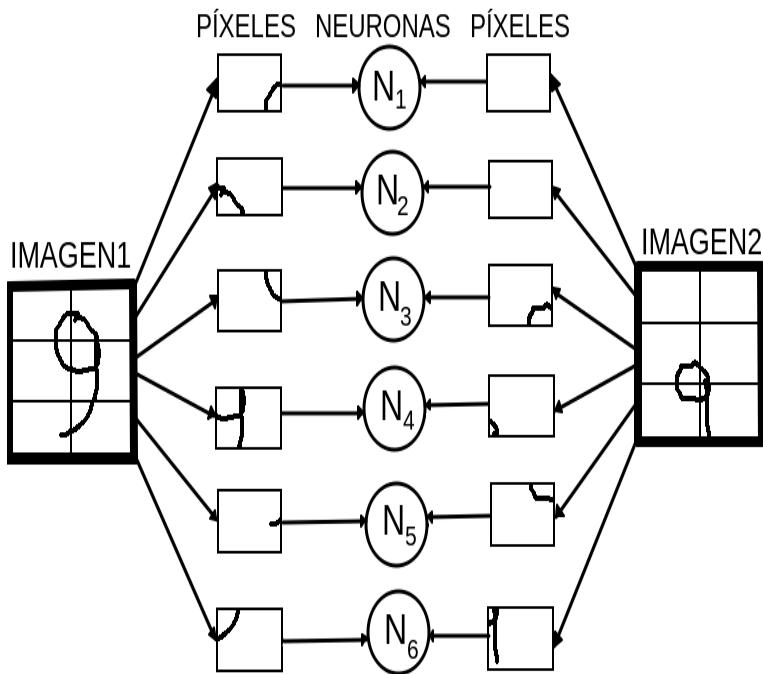


Figura 3.12: Red neuronal clásica comparando imágenes.

Imagen que representa la imposibilidad de una red clásica para entender una imagen de una misma clase debido a la variación en la posición del objeto.

Supongamos que se quiere distinguir entre un número 9 y un número 8. Para un humano, dada su experiencia, es evidente notar que mientras el 9 se caracteriza por un círculo y una línea unidos de una forma particular, el número 8 está caracterizado por dos círculos que se unen de una manera específica. Así, para clasificar si la imagen de un número dado es un 8 o un 9, basta con inspeccionar si cumple las características de un 8 o las de un 9 para asignar una probabilidad a cada posibilidad.

Las redes neuronales convolucionales se caracterizan por dos tipos especiales de capas ocultas: la capa de **convolución** y la capa de **agrupación**. Por lo tanto, se entienden como una forma de red neuronal clásica que cuenta con capas especiales que le permiten al modelo aprender características morfológicas de la imagen.

Las funciones de estas capas se definieron tomando como inspiración el comportamiento de las neuronas en la corteza cerebral visual. Estas neuronas permiten al cerebro humano extraer características de las imágenes percibidas por los ojos, esto a través de señales eléctricas que llegan a estas neuronas mediante el nervio óptico, el cual se genera a partir de los estímulos oculares. Los científicos descubrieron que la corteza cerebral encargada del procesamiento visual de imágenes se compone de dos tipos distintos de neuronas: las **simples** y las **complejas**.

Las neuronas **simples** se encargan de procesar las señales eléctricas provenientes de los estímulos de una pequeña región del campo visual. Hay muchas neuronas de este tipo que permiten cubrir todo el campo visual. Dada la limitada región del campo visual a la que tienen acceso, estas se activan mediante impulsos eléctricos que se disparan cuando hay presencia de ejes o líneas orientadas de una forma específica.

Las neuronas **complejas** procesan grupos de señales que provienen de varias neuronas simples, por lo que tienen acceso a más información proveniente del campo

visual. Estas neuronas se activan mediante impulsos eléctricos cuando provienen de estímulos visuales más complejos, como la orientación de líneas en movimiento en direcciones específicas, independientemente de su ubicación exacta dentro del campo visual.

Dentro de la corteza cerebral encargada del procesamiento visual, existen capas intercaladas de estos dos tipos de neuronas, formando complejas redes neuronales biológicas que brindan al humano la capacidad de análisis visual con la que cuenta.

Para información detallada sobre estas neuronas, véase [8].

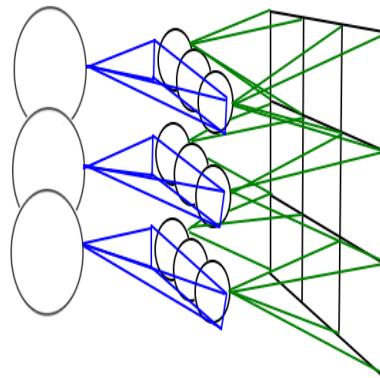


Figura 3.13: Neuronas simples y complejas.

En la imagen, se muestra una capa de neuronas simples, representadas por círculos más pequeños, seguida de una capa de neuronas complejas, representadas por círculos más grandes. El campo de visión completo se subdivide en regiones cuadriculares que representan el acceso a los estímulos para las neuronas simples. A su vez, se muestran los estímulos de múltiples neuronas simples a los que accede una neurona compleja.

¿Cómo se puede emular el funcionamiento de estas neuronas en una red neuronal artificial?

Para lograrlo se requiere del concepto de convolución. Para detectar ejes al igual que las células simples, es necesario analizar una subregión de píxeles de la imagen para determinar si hay un cambio drástico en el valor de los píxeles en esta región y, así, saber si hay un eje presente. Con esta idea, podemos tomar subregiones de la imagen a través de una submatriz que se denomina núcleo, filtro o ***kernel***. Cada casilla de esta submatriz tiene asignado un número fijo, estos valores numéricos determinan el tipo de filtro que describe esta submatriz.

El proceso de convolución consiste en traslapar la submatriz sobre la imagen y multiplicar cada valor numérico del filtro por el valor del píxel sobre el que está posado. Luego, se procede a sumar el resultado de todas las multiplicaciones resultantes entre el filtro y los valores de píxeles en la imagen para obtener un nuevo resultado que se coloca en una nueva matriz.

Este proceso se repite iterativamente moviendo el filtro sobre la imagen un determinado número de píxeles, lo cual se denomina zancada o ***stride***, que se ejecuta de izquierda a derecha y de arriba hacia abajo. De esta forma, según el tamaño y el tipo de filtro que se use, se determina una nueva imagen. Esta nueva imagen presenta transformaciones que permiten detectar ejes de diversas maneras.

0	0	0
0	2	0
0	0	0

139	138	135	122	140	135
138	122	137	255	0	138
135	138	135	138	140	255
255	0	140	122	0	0
135	143	122	138	0	135
122	255	143	135	140	122

0	0	0			
139	138	135	122	140	135
0	2	0			
138	122	137	255	0	138
0	0	0			
135	138	135	138	140	255
255	0	140	122	0	0
135	143	122	138	0	135
122	255	143	135	140	122

$0 \times 139 = 0$
 $0 \times 138 = 0$
 $0 \times 135 = 0$
 $0 \times 138 = 0$
 $2 \times 122 = 244$
 $0 \times 137 = 0$
 $0 \times 135 = 0$
 $0 \times 138 = 0$
 $0 \times 135 = 0$

244

Figura 3.14: Convolución como neurona simple.

La imagen muestra una primera iteración del proceso de convolución que emula una neurona simple. Se utiliza un filtro de 3×3 , resaltado en azul, que se superpone a una región de la imagen original.

Para emular el funcionamiento de una neurona **compleja**, utilizaremos otro tipo de operación que nos ayuda a agrupar la información procesada por la capa de convolución y a depender menos de la posición y tamaño de los objetos presentes en la imagen. Esta es la capa de **agrupación**, la cual tiene dos objetivos: reducir la imagen y resaltar las características de interés.

Existen distintas operaciones que se pueden ejecutar en la capa de agrupación, pero uno de los más usados es el de **agrupación máxima**. En este método, a la matriz resultante de la capa de convolución se le sobreponen una matriz de una dimensión menor seleccionada, y de los píxeles de la imagen sobre los que se superpone, se selecciona el valor más grande. El resto se descarta y se lo asigna a una nueva matriz, de igual manera que en la convolución.

Se mueve la submatriz a través del parámetro de zancada o *stride*, y el proceso se lleva a cabo en toda la matriz. De esta forma, obtenemos una matriz reducida que codifica las características relevantes de acuerdo al filtro usado y descartando el resto.

El **relleno** o *padding* es una técnica que permite la aplicación de filtros a matrices, permitiendo modificar las dimensiones de estas mismas para poder hacer cálculos matriciales consistentes. Se logra agregando marcos de valores constantes a las dimensiones de las matrices.

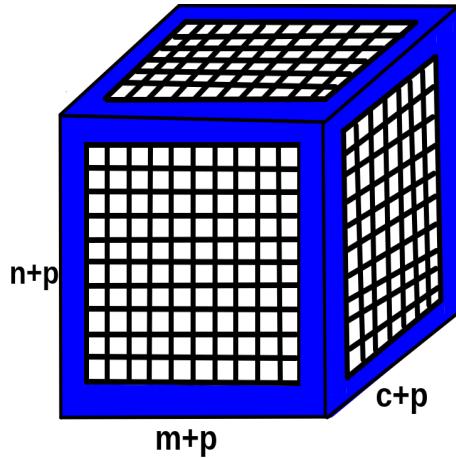


Figura 3.15: **Relleno.**

Imagen que ilustra el relleno de p pixeles en cada dimensión de una matriz.

Al igual que en las redes neuronales biológicas, las redes neuronales convolucionales permiten extraer características de interés en las imágenes a través de una iteración de capas convolucionales, de agrupación y de capas de procesamiento, como en las redes neuronales clásicas. Dicho esto, a pesar de que hay una gran variedad de filtros para elegir con respecto a la convolución y la agrupación, no hay porque preocuparse por cuáles elegir, ya que las redes neuronales convolucionales durante el entrenamiento tienen la capacidad de aprender los filtros que mejor se adapten a las características de interés que tenemos sobre los datos.

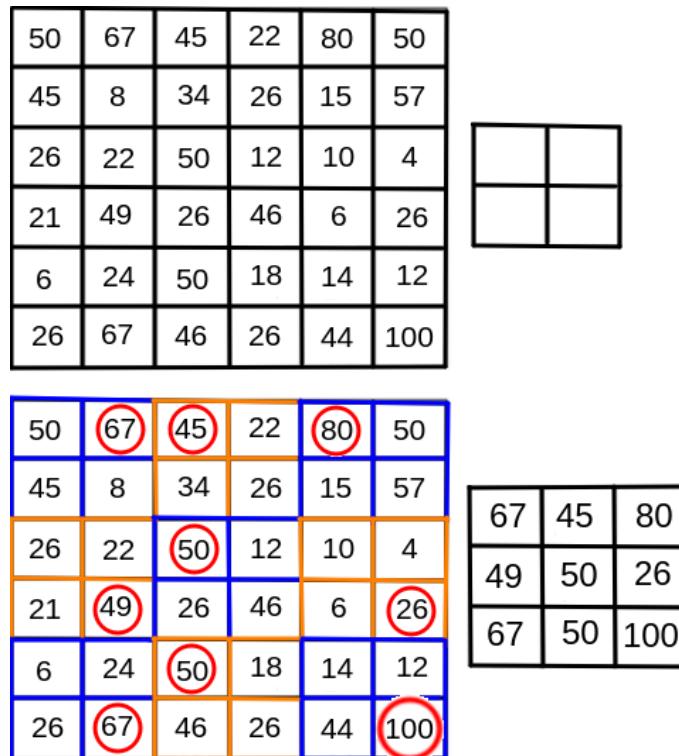


Figura 3.16: **Agrupación como neurona compleja.**

La imagen muestra una matriz resultante de una convolución a la que se le aplica una agrupación máxima a través de un filtro de 2×2 . La alternancia de colores muestra el recorrido del filtro, resaltando con un círculo rojo el valor máximo que conserva la operación. Finalmente, se muestra la matriz reducida resultante que contiene estos valores máximos.

Formalmente la convolución es un opeador en el espacio de funciones continuas $* : C(\mathbb{R}) \times C(\mathbb{R}) \rightarrow C(\mathbb{R})$ que asigna dos funciones a una tercera que representa la magnitud en la que se superponen y una versión trasladada e invertida de la segunda mediante:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau$$

Esta definición puede adecuarse a espacios discretos mediante:

$$f[m] * g[m] = \sum_n f[n]g[m - n]$$

De esta manera la convolución puede ser extendida a matrices, así dadas dos matrices $A, B \in M_n(\mathbb{R})$ la convolución $A * B$ está definida por:

$$(A * B)_{ij} = \sum_{k=1}^p \sum_{l=1}^q A_{i-k+r, j-l+s} \cdot B_{kl}$$

De forma general, dada una matriz $A_{n \times m \times c}$, al aplicarle algún relleno de ancho p , resulta en la matriz $A_{n+p \times m+p \times c+p}$ y dado un filtro $F_{t \times t \times t}$ con salto s la convolución produce una matriz de dimensiones definidas por:

$$A_{n+p \times m+p \times c+p} * F_{t \times t \times t} = B_{\lfloor \frac{n+p-f}{s} + 1 \rfloor \times \lfloor \frac{m+p-f}{s} + 1 \rfloor \times \lfloor \frac{c+p-f}{s} + 1 \rfloor}$$

Donde $\lfloor \cdot \rfloor : \mathbb{R} \rightarrow \mathbb{Z}$ es la función **piso**, definida por $\lfloor x \rfloor = \max\{m \in \mathbb{Z} \mid m \leq x\}$.

Para más información sobre redes convolucionales, véase [9]

3.1.4. RetinaNet

Es una arquitectura de red neuronal profunda diseñada para la detección de objetos en tiempo real.

Está basada en ResNet con una implementación de anclajes (*anchors*) que permiten predecir la clase y la ubicación de los objetos presentes en una imagen.

Esta arquitectura se compone de cuatro módulos principales que le otorgan su funcionalidad:

Red neuronal convolucional base: Se utiliza ResNet para generar un mapa de características de la imagen analizada.

Generador de anclas: Este funciona para crear cuadrículas de tamaño y escala predefinidos sobre la imagen original, las cuales se denominan anclas. Estas anclas son puntos de referencia utilizados en la predicción de la ubicación de los objetos presentes en la imagen.

Red clasificadora: Es la parte de la arquitectura que permite predecir la probabilidad de que cada ancla se asocie a una clase específica a través de una función softmax.

Red de regresión: Es la parte de la arquitectura encargada de predecir el tamaño y la ubicación de cada objeto asociado a un ancla, lo anterior a través de una regresión lineal que proporciona las coordenadas.

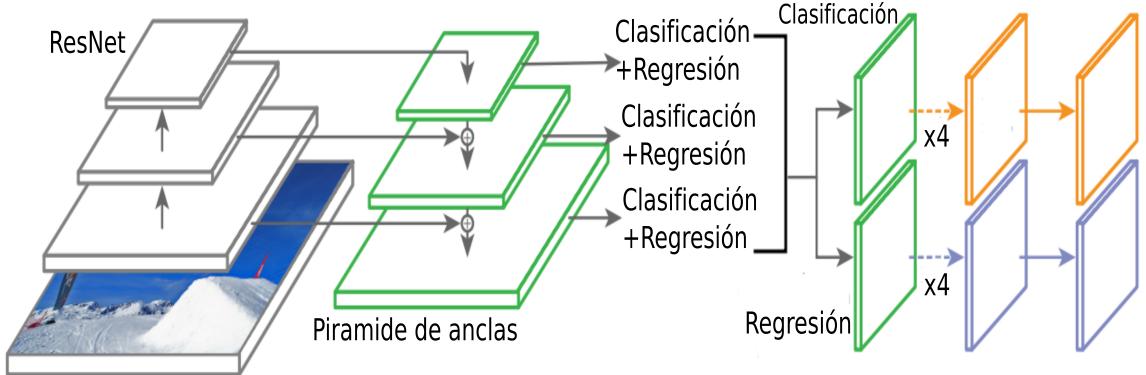


Figura 3.17: **Arquitectura de RetinaNet.**

Imagen modificada de [83], donde se muestra una representación gráfica de la arquitectura RetinaNet. Iniciando desde la derecha, se muestra la red base ResNet, seguida por la pirámide que define las anclas y, finalmente, las capas de clasificación y regresión lineal.

El desempeño de RetinaNet se compara de acuerdo a la profundidad utilizada para ResNet, como se muestra en las siguientes gráficas.

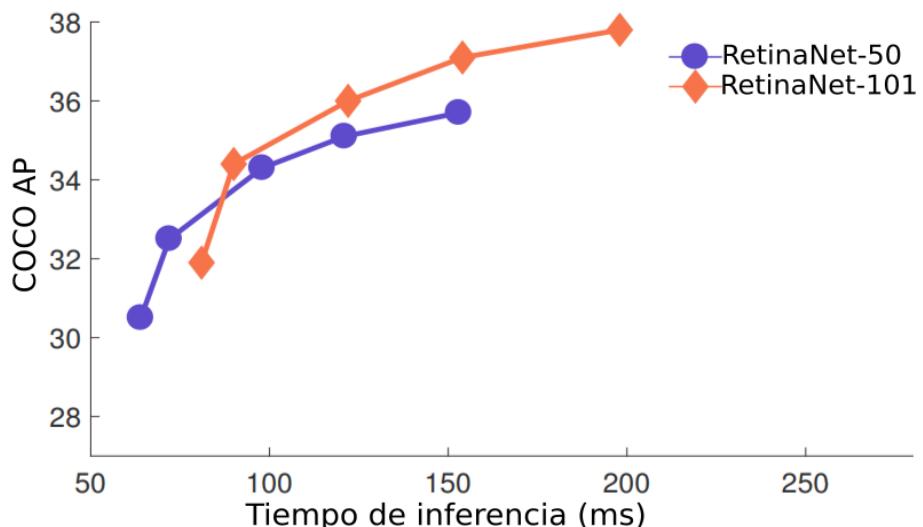


Figura 3.18: **Desempeño de RetinaNet.**

Imagen modificada de [83], donde se muestra la gráfica comparativa entre la exactitud y el tiempo de predicción. En naranja se muestra usando una ResNet de profundidad 101 y en azul una de profundidad 50.

Para información detallada sobre RetinaNet, véase [83].

Capítulo 4

Antecedentes

En este capítulo se presentan investigaciones que documentan resultados y esfuerzos por resolver tareas relacionadas con los objetivos de la presente investigación. En general, se abordan investigaciones que se centran en la detección y clasificación de COVID19 haciendo uso de imágenes médicas y técnicas de aprendizaje profundo.

4.1. Aprendizaje profundo en detección COVID-19

En esta sección se presentan algunas de las investigaciones y aportaciones científicas respecto al uso de técnicas de aprendizaje profundo para la tarea de detección de COVID-19.

4.1.1. Detección de COVID-19 en imágenes médicas mediante técnicas de aprendizaje profundo

En esta investigación reportada en [20] por Dandi Yang, et al., se utilizaron cuatro modelos de redes neuronales convolucionales preentrenados para aplicar transferencia de conocimiento en la búsqueda de mejoras en la clasificación a través de tomografías computarizadas y radiografías de rayos X para la detección de COVID-19. Los modelos comparados son:

VGG-16: Es una arquitectura de red neuronal convolucional desarrollada en 2014. Consiste en 16 capas, donde se apilan 13 capas convolucionales con filtros y capas de agrupación máxima. Entre estas capas, se aplica la función de activación ReLU. Luego, hay tres capas totalmente conectadas que contienen la mayoría de los parámetros de la red. Finalmente, se usa una función softmax para producir las probabilidades para cada clasificación. Esta red está entrenada para clasificar imágenes en 1000 categorías de objetos y tiene un tamaño de entrada de imagen de 224 por 224 píxeles con 3 canales de color.

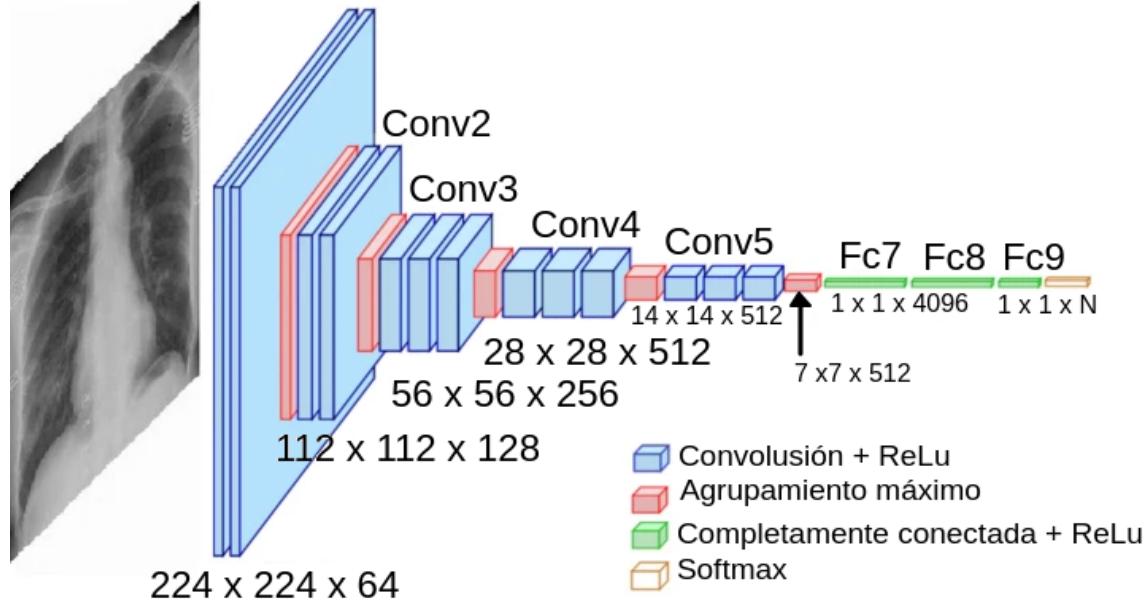


Figura 4.1: Arquitectura VGG16.

Imagen modificada de: [31] donde se muestra el flujo de una imagen de rayos X dentro de la red VGG16, se muestran las agrupaciones de sus capas importantes y se menciona de que tipo son.

Para artículo sobre VGG-16, véase [31].

DenseNet121: Es una red neuronal convolucional que tiene 120 capas de convolución y 4 de agrupamiento por promedio (*average pool*), cada capa está conectada directamente con todas las demás capas subsecuentes lo que permite que las capas más profundas utilicen características extraídas con anterioridad, tiene un tamaño de entrada de imagen de 224 por 224 pixeles con 3 canales de color.

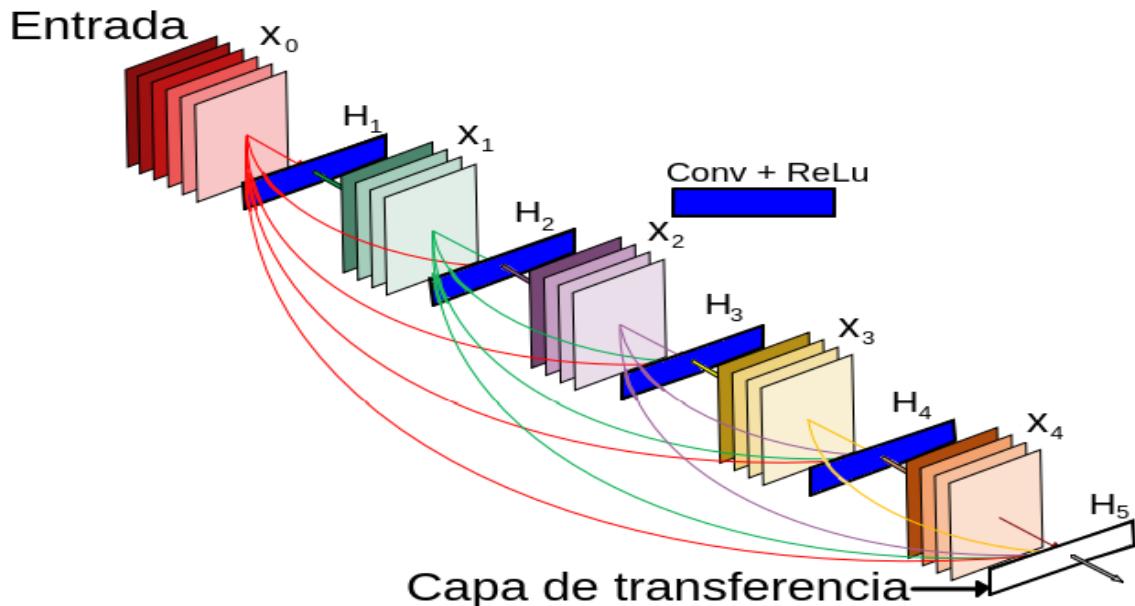


Figura 4.2: Arquitectura DenseNet.

Imagen modificada de [29] que ilustra un bloque básico de 5 capas de la arquitectura DenseNet, donde cada capa tiene acceso a las características aprendidas de la capa que le antecede.

Para artículo sobre DenseNet, véase [29]

ResNet-50: Cuenta con 50 capas (48 capas convolucionales, una capa de agrupamiento máximo y una capa de agrupamiento promedio). Tiene un tamaño de entrada de imagen de 224 por 224 píxeles con 3 canales de color y es residual, lo que significa que sus capas tienen conexiones con algunas pero no todas sus capas subsecuentes. Consta de una capa de entrada, 4 etapas subsiguientes y una capa de salida. Recibe información de etapas anteriores, ejecuta un paso de la red y proporciona la salida.

ResNet-152 : Es una mejora mediante la incorporación de más capas a **ResNet-50** logrando un total de 150 capas.

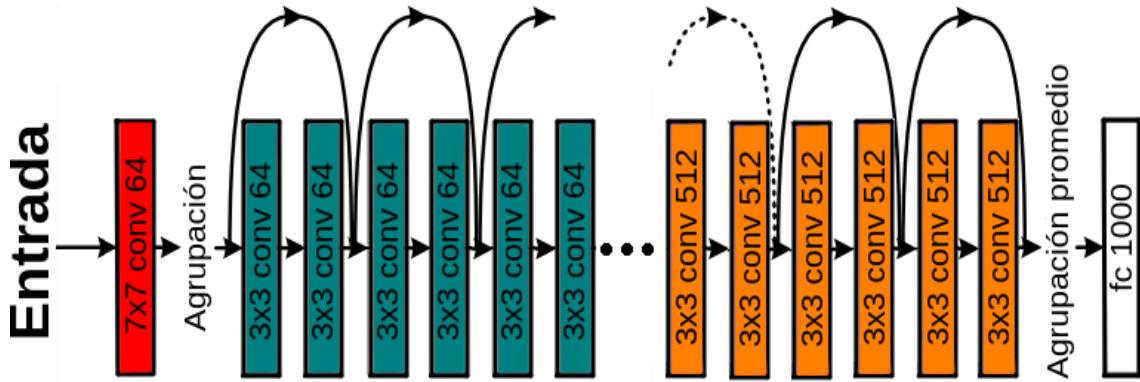


Figura 4.3: **ResNet 50**

Imagen donde se muestra una represnetación trunca de la arquitectura de ResNet.

Para artículo sobre ResNet, véase [30].

En esta investigación se reporta que tanto ResNet como DenseNet obtuvieron una puntuación F1 por encima del 96 % al clasificar a individuos sanos y enfermos de COVID-19 utilizando tomografías computarizadas. Se aplicaron técnicas de transferencia de conocimiento para abordar el problema de la insuficiencia de datos de entrenamiento y para optimizar el tiempo necesario para entrenar los modelos.

VGG-16 se utilizó para la clasificación entre pacientes sanos y enfermos por COVID-19 haciendo uso de imágenes de rayos X. Logró una alta precisión del 99 %, mejorando en la detección de COVID-19 y neumonía.

También se utilizó Fast.AI, una librería gratuita y de código abierto construida sobre PyTorch que proporciona componentes de alto nivel para el aprendizaje profundo. Estos componentes permiten a los usuarios obtener resultados de vanguardia rápidamente, incluso si no son expertos en aprendizaje profundo.

De esta investigación se concluye que, al menos para la detección de COVID-19, la red VGG-16 tiene un mejor desempeño que el mostrado por las redes probadas en tomografías computarizadas, lo que brinda resultados positivos sobre el uso de aprendizaje profundo en imágenes de rayos X.

Un resumen de los desempeños reportados se muestra en las siguientes tablas:

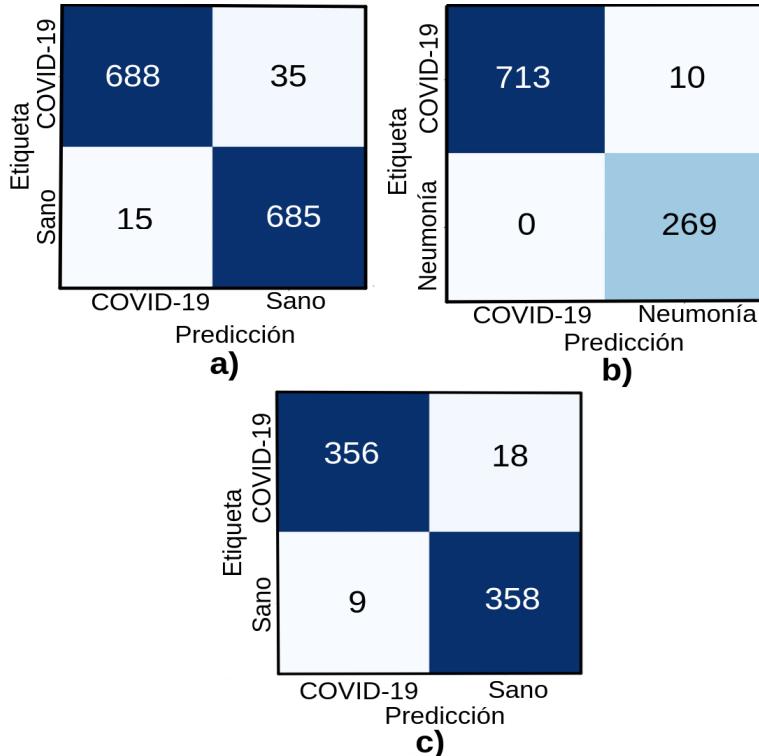


Figura 4.4: **Matrices de confusión.**

Imagen que muestra las matrices de confusión para los conjuntos de validación reportadas: a) Matriz asociada a VGG-16 para la clasificación de sanos y enfermos de COVID-19. b) Matriz asociada a VGG-16 para la clasificación de enfermos por COVID-19 y enfermos por neumonía. c) Matriz de confusión asociada a ResNet para la clasificación de sanos y enfermos de COVID-19 haciendo uso de Fast.AI.

Para información detallada sobre la investigación, véase [20].

4.1.2. Aprendizaje profundo y aprendizaje automático

En esta investigación, reportada en [17] por Wang D. et al., se presenta un método para el diagnóstico automático de COVID-19 utilizando imágenes de rayos X de tórax. Se emplean 5 redes neuronales preentrenadas: VGG16, InceptionV3, ResNet50, DenseNet121 y Xception, junto con un algoritmo de selección de características para extraer las características significativas de las imágenes de pacientes enfermos de COVID-19. Además, se prueban distintos modelos clásicos de aprendizaje automático como clasificadores.

Los autores proponen una metodología que se divide en 3 etapas:

Preprocesamiento de los datos: Las imágenes se establecen en un tamaño de 224×224 píxeles, se normaliza la intensidad de los píxeles en un rango de $[-1, 1]$, y finalmente se aplica un aumento de datos mediante rotaciones y zoom aleatorios de 30° y 20 % respectivamente.

Se procede a utilizar los 5 modelos convolucionales preentrenados en la clasificación de imágenes de objetos cotidianos para extraer las características de las imágenes de pacientes enfermos de COVID-19. Posteriormente, estas imágenes de pacientes son evaluadas por distintos clasificadores de aprendizaje automático (árboles de

decisión, bosque aleatorio, impulso adaptativo (*Adaptive Boosting*), empaquetado (*bagging*) y máquinas de vectores de soporte).

Finalmente, se evalúan las combinaciones de estos métodos para realizar una comparativa.

Algunas de las arquitecturas utilizadas en esta metodología ya han sido expuestas en algún punto del desarrollo de este trabajo, con la excepción de:

Xception: Es una arquitectura de red neuronal convolucional con 71 capas de profundidad que puede clasificar imágenes en 1000 categorías. El tamaño de la entrada de imagen de la red es de 299×299 píxeles con 3 canales de color. Involucra convoluciones profundas separables en profundidad y fue desarrollada por investigadores de Google.

Convolución profunda separable

En una convolución convencional, se aplican filtros en todas las dimensiones de los datos de entrada, lo que implica un alto costo computacional, especialmente cuando la entrada tiene múltiples canales.

En una convolución profunda separable, en cambio, se dividen las operaciones de convolución en dos etapas:

Convolución profunda: Primero se aplica una convolución por canal (o profundidad) de la entrada, pero de forma independiente para cada canal. Esto significa que cada canal se convoluciona con un conjunto de filtros específicos para ese canal, pero las operaciones de convolución se realizan por separado en cada canal.

Convolución espacial: Después de la convolución profunda, se aplica una convolución de filtro de dimensión 1×1 (conocida como convolución espacial) para combinar las características aprendidas de manera independiente en cada canal.

Esto reduce significativamente el número de operaciones requeridas en comparación con una convolución tradicional.

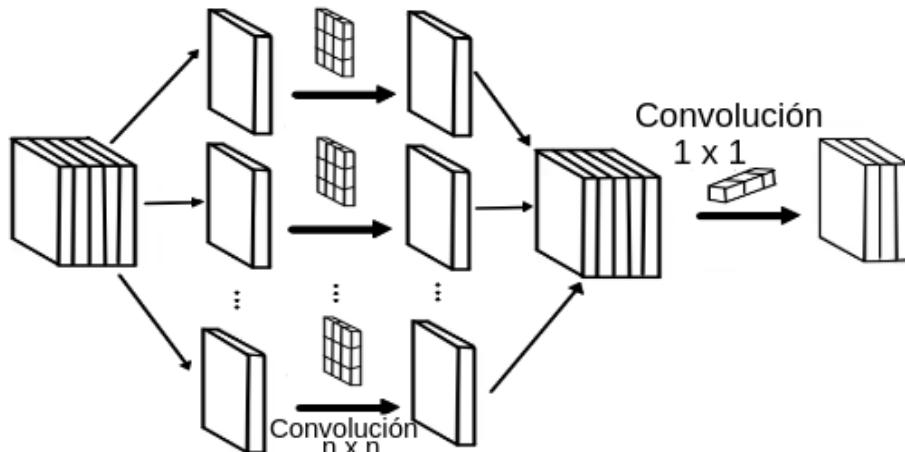


Figura 4.5: **Convolución profunda separable.**

Imagen que muestra un esquema de la aplicación de una convolución profunda separable.

Convolución en Xception

Es una convolución profunda separable con una ligera adaptación, de igual forma se dividen las operaciones de convolución en dos etapas:

Convolución de profundidad: Aplica filtros 1D de tamaño 1×1 a cada canal de entrada.

Convolución espacial: Aplica filtros 2D de tamaño variable a los canales de salida de la convolución de profundidad.

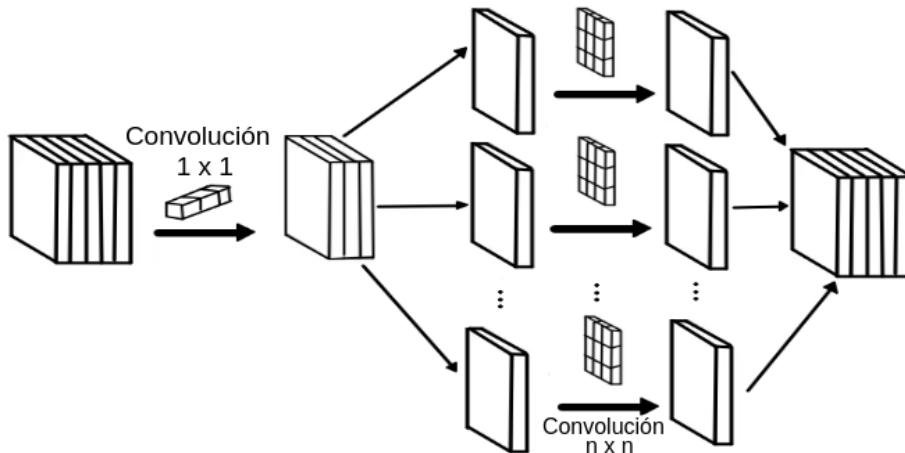


Figura 4.6: **Convolución en Xception.**

Imagen que muestra un esquema de la aplicación de la convolución adaptada en Xception.

Para información sobre Xception, véase [32] y [16]

InceptionV3 : Es una red neuronal convolucional con 48 capas de profundidad. La red preentrenada puede clasificar imágenes en 1000 categorías de objetos. El tamaño de la entrada de imagen de la red es de 299×299 , fue desarrollada por Google, trabaja fundamentalmente con la factorización de convoluciones.

Así una convolución con un filtro de tamaño $n \times n$ puede ser descompuesta en dos convoluciones una convolución de tamaño $n \times 1$ seguida de una convolución de tamaño $1 \times n$.

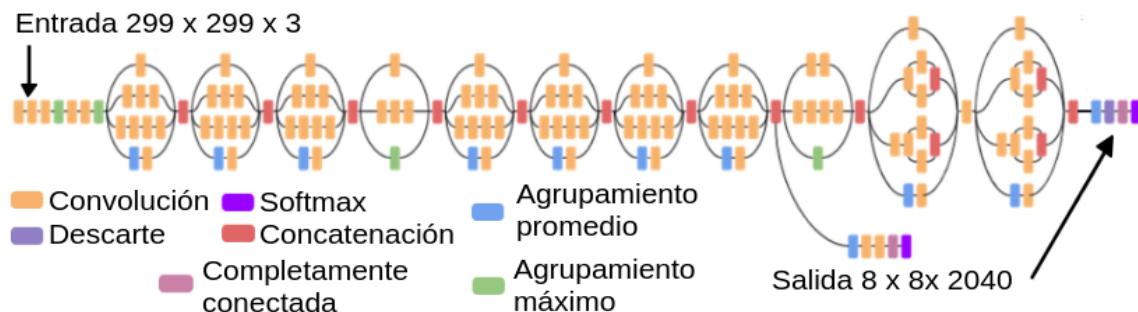


Figura 4.7: **Arquitectura InceptionV3.**

Imagen modificada de [36], donde se muestran las capas del modelo y su tipo.

Para información detallada sobre InceptionV3, véase [15] y [36]

Esta investigación presenta varios resultados interesantes, de los cuales se resaltan los siguientes:

Distintas métricas de los modelos de aprendizaje profundo después de realizar transferencia de conocimiento basada en sus preentrenamientos dados para diversos objetos, de donde concluyen que Xception es el mejor en esta prueba.

Modelo	Sensibilidad	Especificidad	Presición	Exactitud	F1	AUC
VGG16	91.73	98.97	98.70	95.64	93.81	95.34
Inception	92.46	98.76	98.45	95.72	95.36	95.75
ResNet50	86.29	95.65	94.59	92.73	91.85	92.47
DenseNet50	91.24	98.35	97.91	95.08	94.46	94.75
Xception	94.16	99.17	98.97	96.75	96.38	95.54

Tabla 4.1: **Evaluación de preentrenamiento.**

Tabla recreada de [17], donde se muestra un resumen de las métricas evaluadas para medir la eficacia de la transferencia de conocimiento.

Respecto a las métricas reportadas al combinar las arquitecturas de aprendizaje profundo con cada uno de los clasificadores de aprendizaje automático propuestos, se concluye lo siguiente:

Para el modelo VGG16, el mejor desempeño se alcanza con el método de empaquetamiento, con un puntaje F1 de 98.17.

Para el modelo InceptionV3, el mejor desempeño se obtiene con el clasificador de máquina de soporte vectorial, con un puntaje F1 de 98.91.

En el caso de ResNet50, se logra el mejor desempeño con el clasificador de bosques aleatorios, obteniendo un puntaje F1 de 95.69.

DenseNet121 muestra un desempeño óptimo con la máquina de soporte vectorial, con un puntaje F1 de 97.78.

Finalmente, Xception muestra el rendimiento óptimo con la máquina de soporte vectorial, obteniendo un puntaje F1 de 88.24.

De esta investigación se concluye que a través del uso de técnicas combinadas de aprendizaje profundo y aprendizaje automático, se puede obtener un buen desempeño en el diagnóstico de COVID-19, incluso con la desventaja de contar con un conjunto de radiografías de rayos X de baja calidad y cantidad.

Para obtener información detallada de la investigación, véase [17].

4.1.3. IKONOS

En la investigación reportada en [14], por Juliana C. Gomes et al., se propone un sistema inteligente de apoyo al diagnóstico por imágenes de rayos X.

Se propone el desarrollo de IKONOS, una aplicación de escritorio para apoyar y optimizar el diagnóstico de COVID19 a través de imágenes de rayos X de tórax que sea una herramienta de fácil mantenimiento y escalabilidad, utilizando algoritmos

de baja complejidad computacional para la minimización de costos.

El conjunto de datos consiste en 170 imágenes de rayos X de tórax de pacientes con COVID19, neumonía viral y neumonía bacteriana.

Se extraen características de textura de las imágenes utilizando los momentos de Haralick [24] y Zernike [25], posteriormente se entrenaron y evaluaron diferentes clasificadores que incluyen K-vecinos cercanos, máquina de soporte vectorial, bosques aleatorios y redes neuronales.

Esta investigación permite concluir que, aun con una poca cantidad de datos, las redes neuronales profundas tienen la capacidad de diagnosticar COVID19, ya que muestran una precisión del 94 %, que es superior a la de los otros modelos analizados para los cuales la precisión está en el rango del 88 % al 92 %, lo cual nos da un referente de la viabilidad del uso de redes neuronales para la identificación y detección de COVID19.

Para información detallada de la investigación, véase [14].

4.2. Detección y localización de COVID-19 usando imágenes médicas

Hasta ahora se han presentado algunas investigaciones sobre el uso de aprendizaje profundo para la detección de objetos y el diagnóstico de COVID19. En esta sección, se presentan investigaciones donde se combinan estas ideas para dar paso a modelos de aprendizaje profundo que no solo diagnostican la enfermedad, sino que también localizan las regiones afectadas en las imágenes médicas, lo cual es uno de los objetivos del presente trabajo.

4.2.1. Red de discriminación y localización de lesiones COVID19

En la investigación documentada en [7] por Xia Ma et al., se propone una red con un módulo de atención de campo multirreceptivo para diagnosticar COVID-19 en imágenes de tomografía computarizada. Este módulo de atención incluye tres partes: un módulo de convolución piramidal (PCM), un bloque de atención espacial de campo multirreceptivo (SAB) y un bloque de atención de canal de campo multirreceptivo (CAB). El PCM puede mejorar la capacidad de diagnóstico de la red para lesiones de diferentes tamaños y formas. El papel de SAB y CAB es enfocar las características extraídas de la red en el área de la lesión para mejorar la capacidad de discriminación y localización de COVID-19. Se verificó la efectividad del método propuesto en dos conjuntos de datos, uno propio de los autores y otro disponible públicamente.

El modelo propuesto utiliza VGG16 como base. A diferencia de VGG16, se reemplaza la capa de convolución antes de cada capa de agrupación con cada módulo de atención de campo multirreceptor propuesto y se evalúa su desempeño.

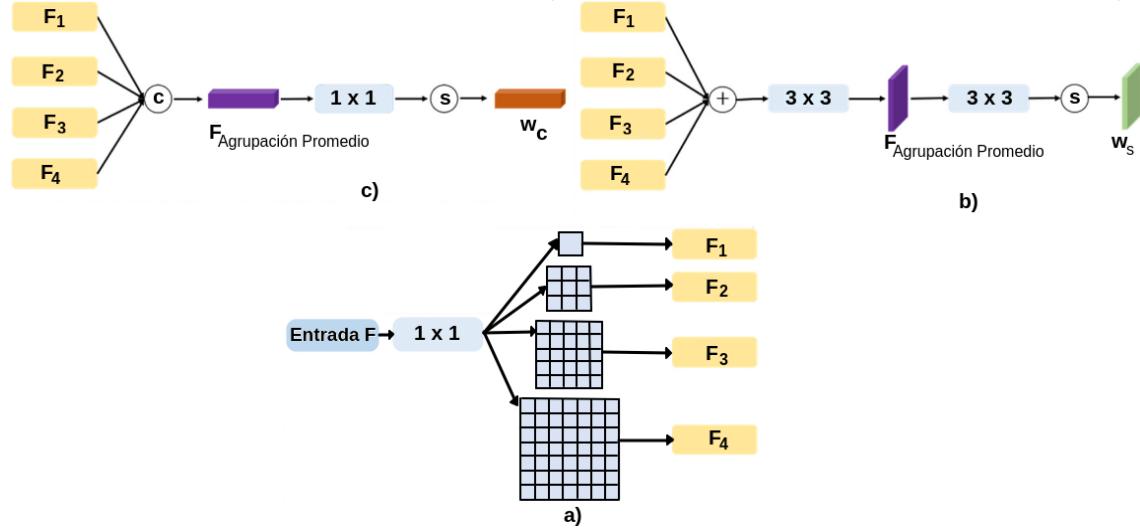


Figura 4.8: **Modulos propuestos.**

Imagen modificada de [7], que ilustra los nodulos propuestos en los modelos. a) Muestra la estructura básica del módulo PCM. b) Muestra la estructura básica del módulo SAB. c) Muestra la estructura básica de CAB.

El modelo se entrenó en un conjunto de datos de 200 imágenes de TC de tórax, de las cuales 100 eran de pacientes con COVID19 y 100 eran de pacientes sin COVID19.

El modelo mostró una mejora acorde a la implementación de cada módulo propuesto, la cual se resume en la siguiente tabla.

Método	Precisión	Especificidad	Sensibilidad
Vgg16	93.02	94.32	91.25
VGG16+SAB	94.76	96.35	93.32
VGG16+CAB	96.02	96.83	95.13
VGG16+SAB+CAB	97.12	96.89	97.21

Tabla 4.2: **Desempeño de las implementaciones propuestas.**

Tabla recreada de [7], donde se muestra un resumen de las métricas del desempeño de las implementaciones de los módulos SAB y CAB.

De esta investigación se rescata una interesante propuesta para mejorar la atención de las regiones pequeñas de interés en tomografía computarizada, así como un precedente en la capacidad de las técnicas de aprendizaje profundo en la localización de COVID-19.

Para obtener información detallada de la investigación, véase [7].

4.2.2. Aprendizaje profundo para la identificación y localización de COVID19

La investigación documentada en [5], por Karem Daiane Marcomini et al., es de especial relevancia ya que utiliza la base de datos pública *SIIM-FISABIO-RSNA* [21],

la cual contiene imágenes de rayos X anotadas con daño pulmonar causado por COVID-19, y esta base de datos juega un papel fundamental en el presente trabajo. En esta investigación se propone un modelo de aprendizaje que puede discernir entre dos clases de las cuatro propuestas en la base de datos, así como las áreas de opacidad pulmonar referentes a estas clases. Dado que las clases están desequilibradas (el 69,2 % de las imágenes son COVID19 positivas), se aplicó un aumento de datos a las imágenes de la categoría sanos.

El conjunto de datos se dividió en conjuntos de entrenamiento y prueba con una proporción de 90:10, y para la clasificación se aplicó una validación cruzada de 5 veces al conjunto de entrenamiento. Para realizar la clasificación, se utilizó la arquitectura **EfficientNetB4** [4], mientras que YOLOv5 [13] se empleó para la tarea de detección.

Los datos se reorganizaron en dos categorías de imágenes del conjunto de datos: apariencia positiva para COVID19 y negativa para neumonía. Se agruparon las apariencias típicas e indeterminadas en la clase positiva de COVID19, y se eliminaron las imágenes relacionadas con otros tipos de neumonía (etiquetadas como atípicas), ya que es probable que las apariencias típicas e indeterminadas correspondan a una infección por COVID19.

Al clasificador basado en la arquitectura EfficientNetB4 se le reemplazó por una operación de agrupación promedio global, seguida de una normalización por lotes y una capa densa con una neurona que utiliza la función de activación sigmoide. El entrenamiento se realizó con 30 épocas en lotes de 16 imágenes por paso utilizando un optimizador Adam con una tasa de aprendizaje de 0.0001. Se utilizó el aumento de datos durante el entrenamiento y todas las imágenes se redimensionaron a 380x380 píxeles y se aplicó un procesamiento, en el que se normaliza el histograma de intensidades.

Para la localización se utilizó la red YOLOv5 con una resolución de entrada de 512×512 , preentrenada en Common Objects in Context (COCO) [3]. Se entrenó durante 50 épocas con un tamaño de lote de 8.

Posterior a la salida de los detectores de objetos, se eliminaron los cuadros de limitadores superpuestos obtenidos durante la etapa de predicción, mediante el algoritmo de supresión no máxima (NMS) utilizando un valor umbral de 0,5.

Las métricas para el desempeño del detector de opacidades se resumen en la siguiente tabla.

Conjunto	VP	FP	FN	mAP
Diagnosticado COVID19	495	169	254	59.51
Predicho COVID19	478	226	271	53.57

Tabla 4.3: **Evaluación de detector.**

Tabla recreada de [5], donde se muestran las métricas de evaluación para el modelo de detección.

El modelo utilizado para la clasificación presentó métricas que reflejan un desempeño satisfactorio en la tarea para la que fue entrenado, las cuales se resumen en la siguiente tabla.

Fold	VP	FN	VN	FP	Exactitud	Precisión	Recall	F1	AUROC
0	373	24	92	70	0.832	0.842	0.940	0.888	0.883
1	348	49	105	57	0.810	0.859	0.877	0.868	0.856
2	364	33	95	67	0.821	0.845	0.917	0.879	0.862
3	367	30	103	59	0.841	0.862	0.924	0.892	0.893
4	360	37	106	56	0.834	0.865	0.907	0.886	0.887
μ	362.4	34.6	100.2	61.8	0.828	0.855	0.913	0.883	0.876
σ	8.4	8.4	5.6	5.6	0.011	0.009	0.012	0.008	0.015

Tabla 4.4: **Evaluación de clasificación.**

Tabla recreada de [5], donde se condensan las métricas relacionadas con la clasificación dada por el modelo.

De esta investigación se puede concluir que si bien el modelo presenta un desempeño muy bueno en la clasificación de imágenes , su desempeño en la localización es muy malo al tratarse de un diagnóstico médico ya que solo se logra un índice de mAP del 59.51 % del 100 % posible.

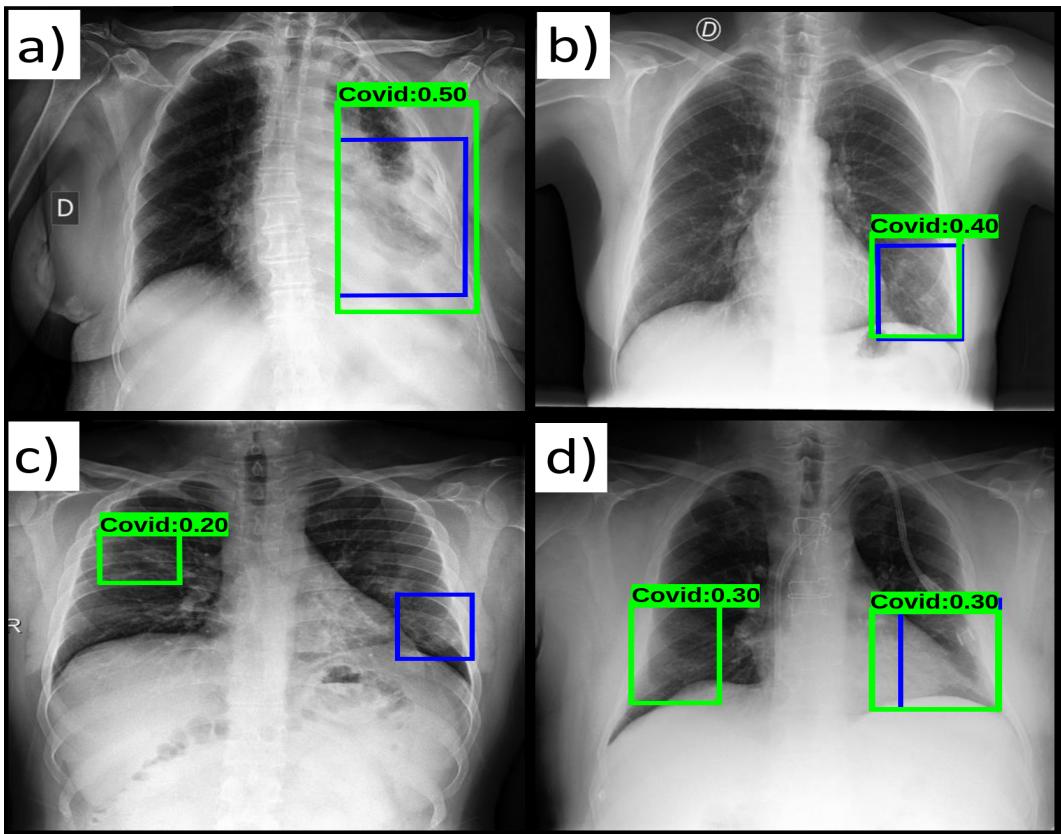


Figura 4.9: **Predicciones del modelo.**

Imagen recortada de [5], donde a)-c) muestran las predicciones (verde) vs las anotaciones de los expertos(azul).

Esto marca un antecedente directo en la búsqueda de la resolución para el problema que busca resuelven en la presente investigación y nos proporciona una visión clara del desafío que representa.

Para artículo completo de la investigación, véase [5].

4.3. Conclusiones sobre los antecedentes

La necesidad de diagnóstico del virus SARS-CoV-2, principalmente realizado mediante pruebas de PCR que detectan material genético asociado al virus, ha impulsado la exploración de nuevas técnicas de diagnóstico y detección de enfermedades. En este contexto, el desarrollo y refinamiento de métodos de aprendizaje profundo basados en el análisis de imágenes médicas se ha vuelto crucial.

El avance en la detección y diagnóstico de enfermedades a través de imágenes médicas, como las radiografías de rayos X, con su relativa accesibilidad en costos y tiempos, es de gran interés y valor para médicos y científicos.

Hasta ahora, diversas investigaciones han utilizado herramientas y técnicas de aprendizaje profundo en tareas cruciales, como la clasificación de pacientes con COVID19 utilizando imágenes de rayos X o tomografías computarizadas, así como la detección de lesiones causadas por esta enfermedad. Si bien estas investigaciones han arrojado resultados positivos, ninguna ha abordado completamente el problema de identificación y localización de daños causados por COVID19, con métricas satisfactorias para la precisión conjunta de localización y clasificación.

Se cuenta con una investigación previa que utiliza la base de datos aquí utilizada con el objetivo de identificar y localizar daños causados por COVID19, pero presentó un desempeño deficiente en la métrica de precisión conjunta de localización y clasificación (mAP). Esto indica la necesidad y motiva a mejorar esta métrica en el diagnóstico de enfermedades.

Para abordar este desafío, se propuso un problema en la plataforma *Kaggle* por la *Society for Imaging Informatics in Medicine*. El equipo ganador logró una mAP del 63.5 %, mientras que otros 1,304 participantes obtuvieron métricas inferiores. Esto destaca la dificultad de mejorar la métrica mAP con los datos propuestos.

En resumen, existe un respaldo positivo para emplear técnicas de aprendizaje profundo en el diagnóstico de enfermedades a través de imágenes médicas, pero mejorar la métrica mAP sigue siendo un desafío importante en este campo. La competencia en *Kaggle* proporciona un precedente sobre la dificultad y la importancia de mejorar estas métricas para lograr diagnósticos más precisos.

Capítulo 5

Materiales

En este capítulo se detallan los materiales esenciales empleados en el desarrollo de la presente investigación.

El propósito es documentar los elementos necesarios para facilitar la replicación o mejora en los resultados presentados.

En las siguientes secciones, se proporcionan todos los detalles relevantes sobre la computadora y los conjuntos de datos utilizados.

5.0.1. Computadora usada

Para manejar los múltiples y grandes volúmenes de imágenes y sus anotaciones, que superan el peso de 1TB (terabyte), equivalente a 1000 GB, así como para su exploración, visualización y preparación, se requiere una computadora con una gran capacidad de almacenamiento interno. Además, debido a la necesidad de entrenar modelos de aprendizaje profundo, es necesario que esta misma cuente con una buena capacidad de cómputo. Para estos fines, se utilizó una computadora **Dell Inc. Precision Tower 7910**, modificada con las siguientes especificaciones:

- **Memoria:** 128 Gb.
- **Procesador:** Intel® Xenon® E5-2630 v3 × 32
- **GPU:** NVIDIA GeForce RTX™ 3090 Ti capacidad 24 GB
- **Capacidad en disco:** 3.9 TB
- **Sistema operativo:** Ubuntu 23.10 64-bit
- **Kernel:** Linux 6.5.0-25-generic

5.0.2. Conjuntos de datos utilizados

5.0.2.1. Presencia de opacidades

Se requiere el uso auxiliar del conjunto de datos público proporcionado por Kaggle en el desafío **RSNA Pneumonia Detection Challenge** [23], el cual se puede revisar a través del enlace: <https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/data>.

Este conjunto contiene 32,777 imágenes de radiografías de tórax de 1024×1024 píxeles proporcionadas por tres instituciones distintas:

Medical Imaging Data Resource Center (MIDRC)

RSNA International Covid-19 Open Radiology Database (RICORD)

Banco de Imagen Médica de la Comunidad Valenciana (BIMCV-COVID-19 Dataset)

Las anotaciones fueron realizadas por un grupo internacional de radiólogos voluntarios, quienes se encargaron de etiquetar la presencia y ubicación de neumonía mediante la delimitación de recuadros en las imágenes.

Para información detallada del conjunto de datos, véase [23].

5.0.2.2. Análisis exploratorio

En esta sección se presenta un análisis exploratorio del conjunto de datos que contiene imágenes de radiografías con presencia de opacidades generadas por neumonía, así como contraejemplos sin presencia de neumonía.

El conjunto de datos cuenta con 14,863 imágenes anotadas con opacidades causadas por neumonía y también incluye imágenes de pacientes sanos.

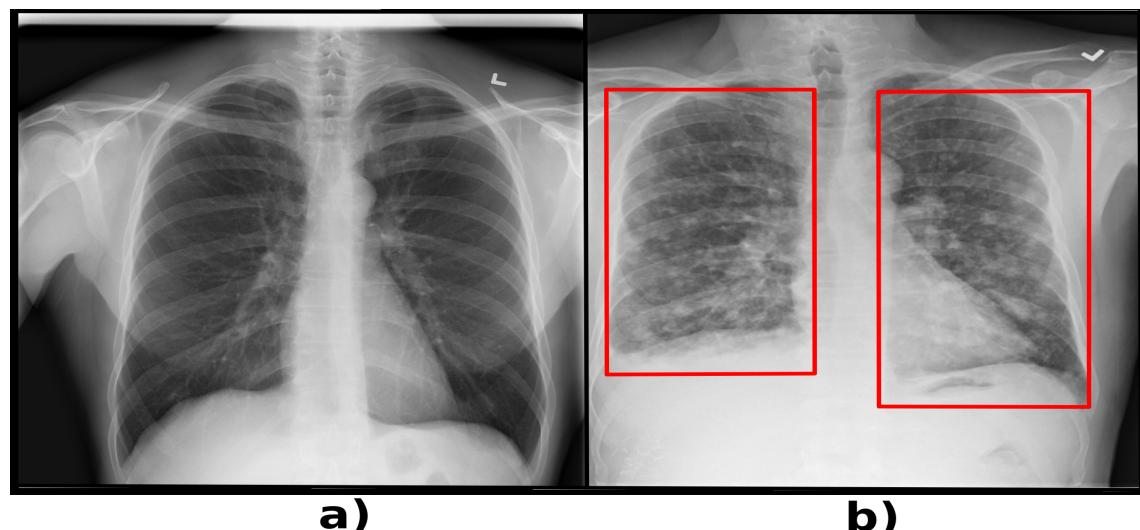


Figura 5.1: **Anotaciones neumonía.**

Imágenes tomadas del conjunto *RSNA Pneumonia* donde: a) muestra un ejemplo de una imagen sin opacidades por neumonía, y b) una imagen con anotaciones para opacidades causadas por neumonía.

Del total de imágenes, el 40.4 % tiene anotaciones sobre la presencia de opacidades por neumonía, mientras que el 59.6 % no las contiene, lo cual representa un balanceo de datos aceptable.

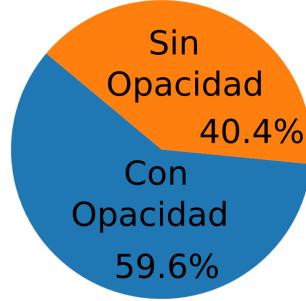


Figura 5.2: Distribución por clases.

Gráfica de pastel que ilustra el porcentaje de elementos en cada clase definida para el conjunto *RSNA Pneumonia*.

Las imágenes con presencia de opacidades causadas por neumonía cuentan con entre 1 y 4 opacidades presentes por imagen.

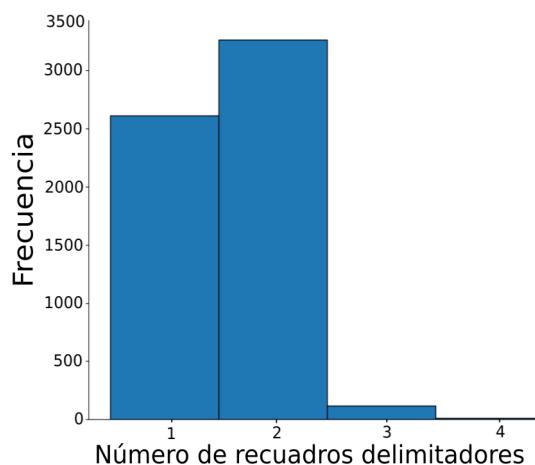


Figura 5.3: Recuadros delimitadores por imagen.

Histograma que ilustra el número de recuadros delimitadores de daños por neumonía presentes por imagen del conjunto *RSNA Pneumonia*.

La mayoría de los recuadros delimitadores en las anotaciones de opacidades por neumonía tienen un área entre el 2% y el 10% de la imagen.

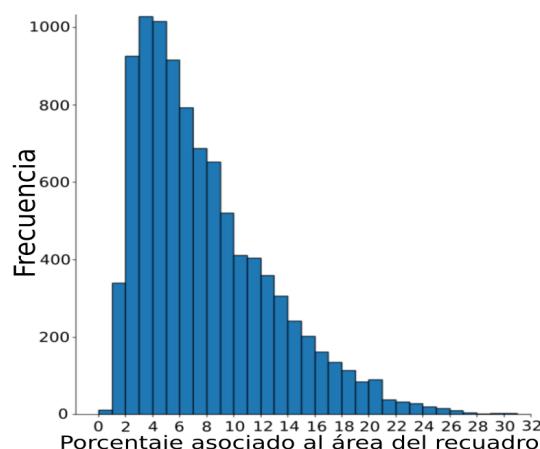


Figura 5.4: Área de recuadros delimitadores.

Histograma que ilustra el porcentaje del área total de la imagen que representan los recuadros delimitadores en el conjunto *RSNA Pneumonia*.

Visualizando los histogramas para distintas imágenes con y sin opacidades causadas por neumonía, se observa que no hay un patrón consistente en estos.

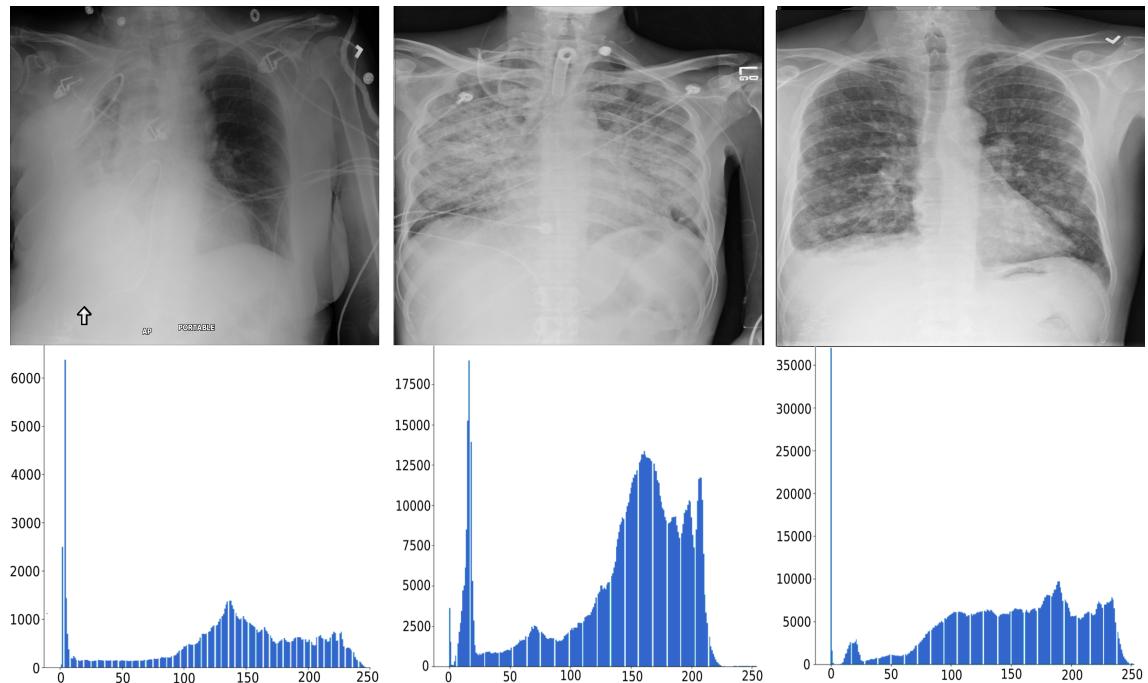


Figura 5.5: Histograma de imágenes con opacidades.

Imagen que ilustra, de arriba hacia abajo, imágenes del conjunto *RSNA Pneumonia* que tienen presencia de opacidades por neumonía con sus respectivos histogramas, donde el eje *x* corresponde a los valores de píxel y el eje *y* a la frecuencia de valores.

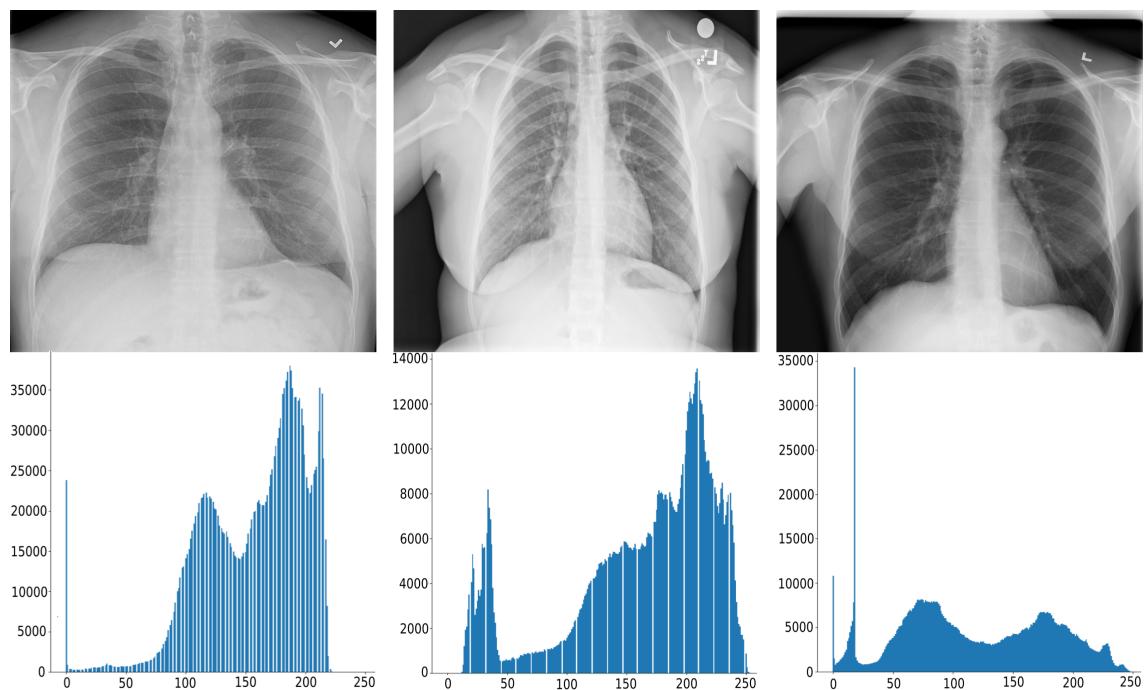


Figura 5.6: Histograma de imágenes sin opacidades.

Imagen que ilustra, de arriba hacia abajo, imágenes del conjunto *RSNA Pneumonia* sin presencia de opacidades por neumonía con sus respectivos histogramas, donde el eje *x* corresponde a los valores de píxel y el eje *y* a la frecuencia de valores.

Así se concluye con el análisis exploratorio del conjunto, haciendo énfasis en las características más notorias y representativas.

5.0.2.3. Opacidades Variadas

Se requiere el uso auxiliar del conjunto de datos público de imágenes de rayos X de tórax **NIH Chest X-ray14** [26].

Está conformado por 112,120 imágenes de rayos X de tórax de 1024×1024 píxeles, de las cuales 1000 cuentan con recuadros delimitadores, mostrando 15 clases distintas:

- 1 Sin hallazgos.
- 2 Cardiomegalia.
- 3 Edema pulmonar.
- 4 Neumonía.
- 5 Nódulos pulmonares.
- 6 Atelectasia.
- 7 Derrame pleural.
- 8 Fibrosis pulmonar.
- 9 Fractura costal.
- 10 Neumotórax.
- 11 Masa pulmonar.
- 12 Hilios pulmonares prominentes.
- 13 Calcificación pulmonar.
- 14 Infiltrado intersticial.
- 15 Líneas de Kerley B.

Para información detallada del conjunto de datos, véase [26].

5.0.2.4. Análisis exploratorio

En esta sección se presenta un análisis exploratorio del conjunto de datos que contiene imágenes de radiografías con anotaciones sobre distintos hallazgos médico.

El conjunto de datos cuenta con 112,120 imágenes, de las cuales 61,345 cuentan con anotaciones de 9 de las 15 clases distintas de hallazgos médicos.



Figura 5.7: Distribución de datos.

Gráfica de pastel que ilustra el porcentaje de datos con presencia de recuadros delimitadores y sin presencia de recuadros.

Dentro de las 9 clases que cuentan con recuadros delimitadores encontramos:

Sin hallazgos: Imágenes sin ningún tipo de los 14 hallazgos médicos.



Figura 5.8: **Sin hallazgos.**
Ejemplo de imagen que no muestra hallazgos médicos.

Atelectasia: Colapso de un pulmón o parte de un pulmón.

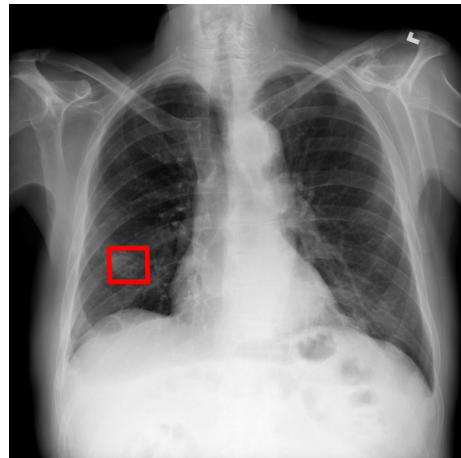


Figura 5.9: **Atelectasia.**
Ejemplo de imagen que muestra atelectasia y su respectiva anotación.

Derrame pleural (Efusión): Acumulación de líquido entre la pleura y la pared torácica.

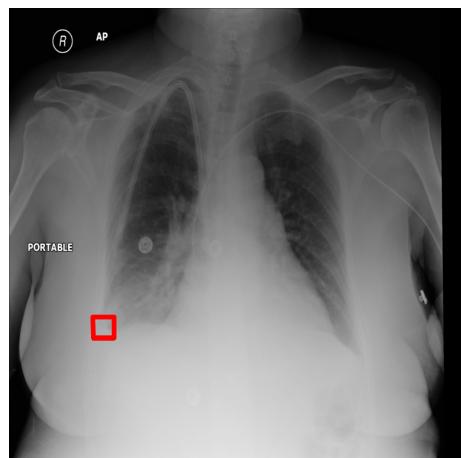


Figura 5.10: **Efusión.**
Ejemplo de imagen que muestra efusión y su respectiva anotación.

Cardiomegalia: Agrandamiento del corazón.

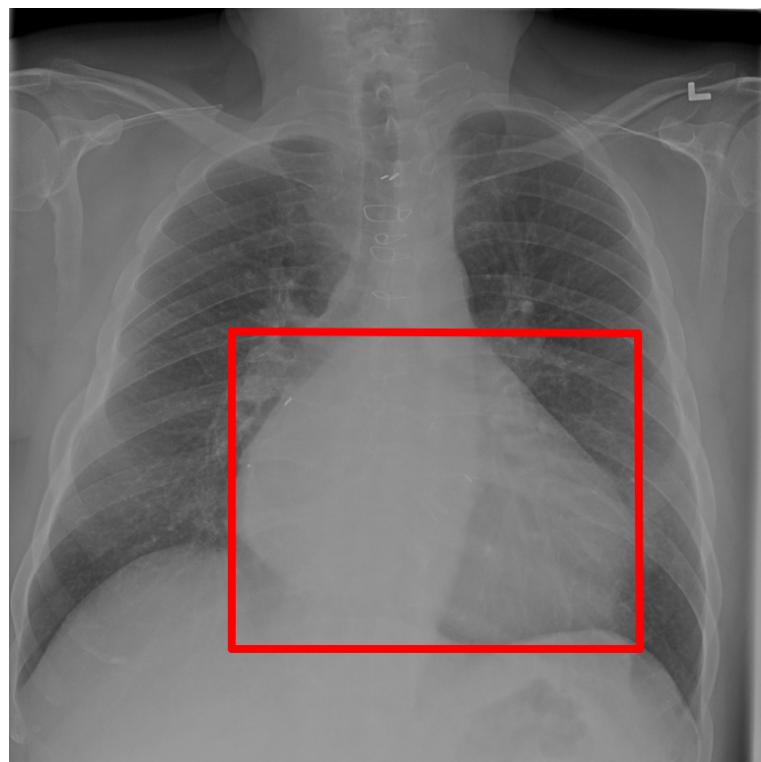


Figura 5.11: **Cardiomegalia.**

Ejemplo de imagen que muestra cardiomegalia y su respectiva anotación.

Infiltración: Inflamación del tejido pulmonar.

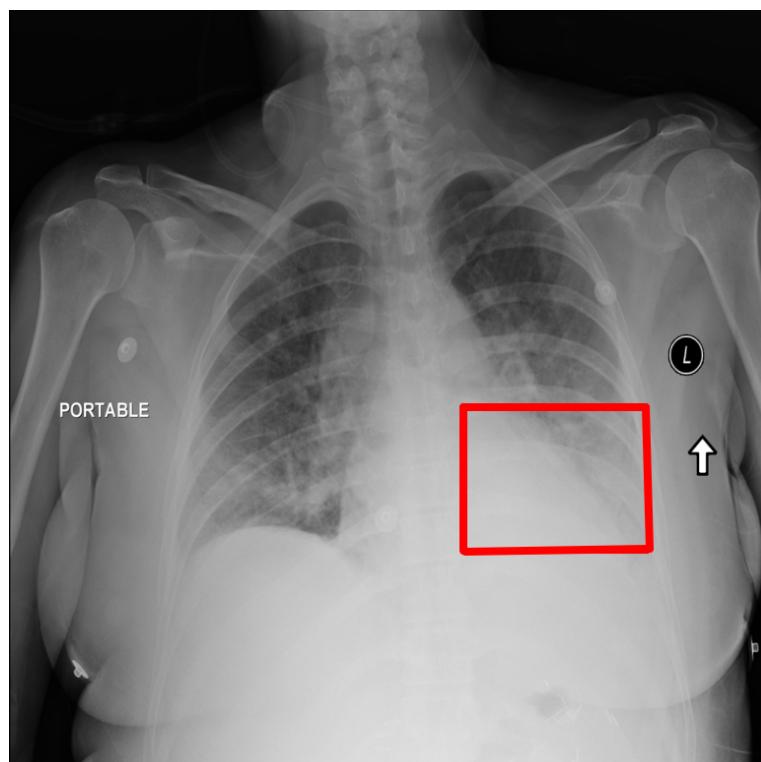


Figura 5.12: **Infiltración.**

Ejemplo de imagen que muestra infiltración y su respectiva anotación.

Neumonía: Condición médica caracterizada por inflamación en los pulmones, donde el oxígeno en los alvéolos es sustituido por fluido.

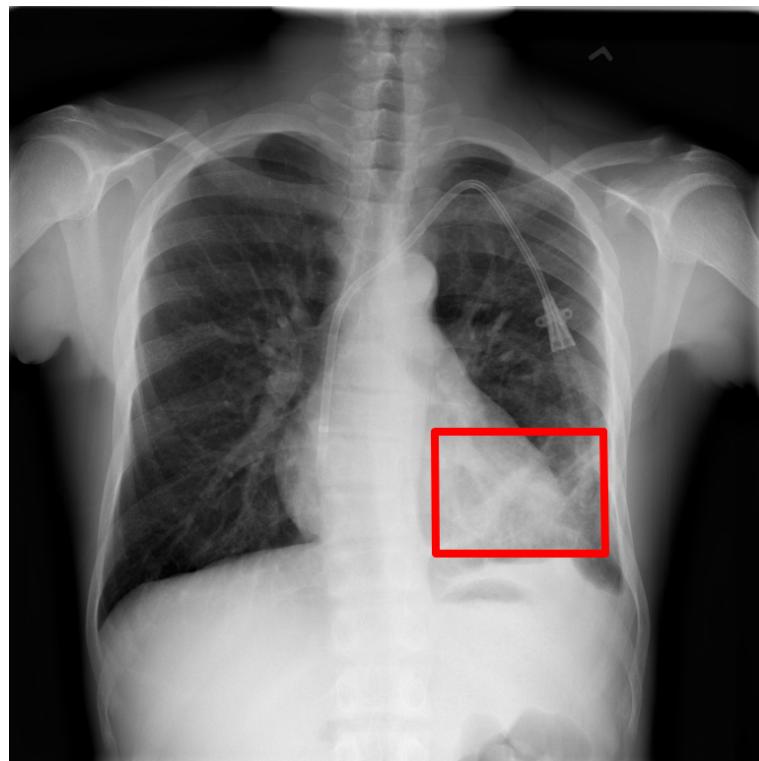


Figura 5.13: Neumonía.

Ejemplo de imagen que muestra neumonía y su respectiva anotación.

Neumotórax: Presencia de aire entre los pulmones y la pared torácica.

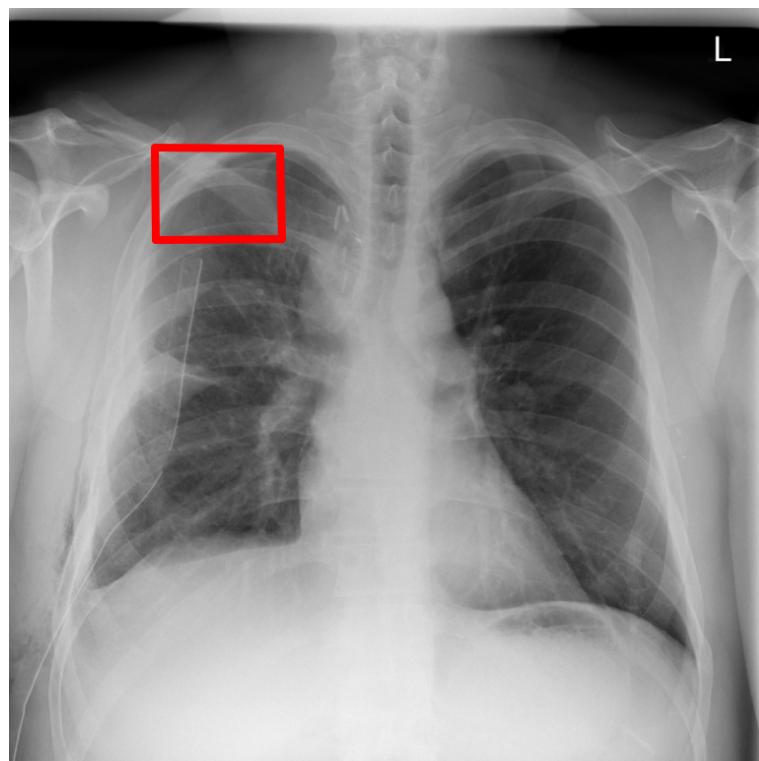


Figura 5.14: Neumotórax.

Ejemplo de imagen que muestra neumotórax y su respectiva anotación.

Masa pulmonar: Masa anormal en el pulmón.

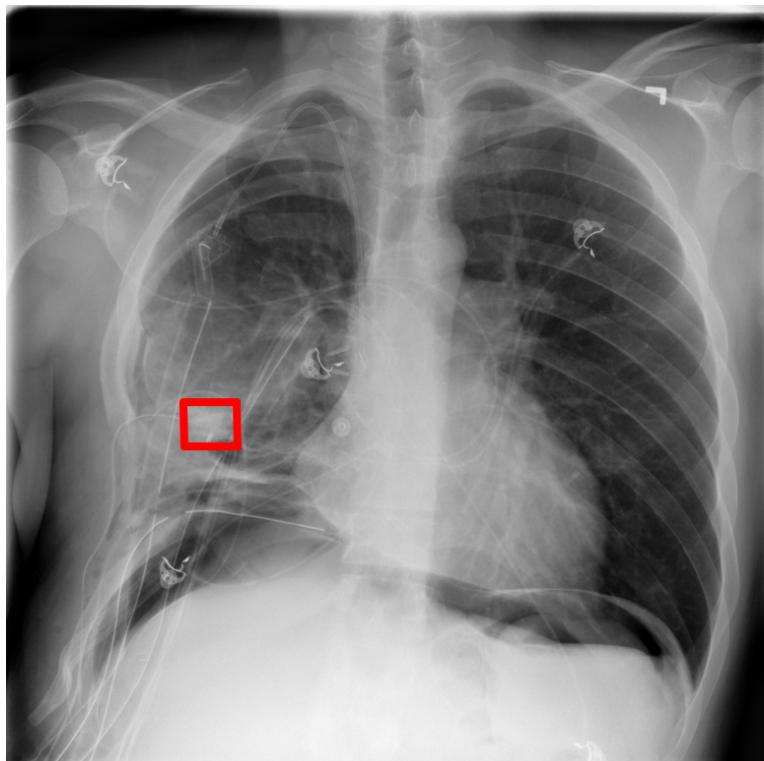


Figura 5.15: **Masa pulmonar.**

Ejemplo de imagen que muestra una masa pulmonar y su respectiva anotación.

Nódulo: Masas sólidas en los pulmones.

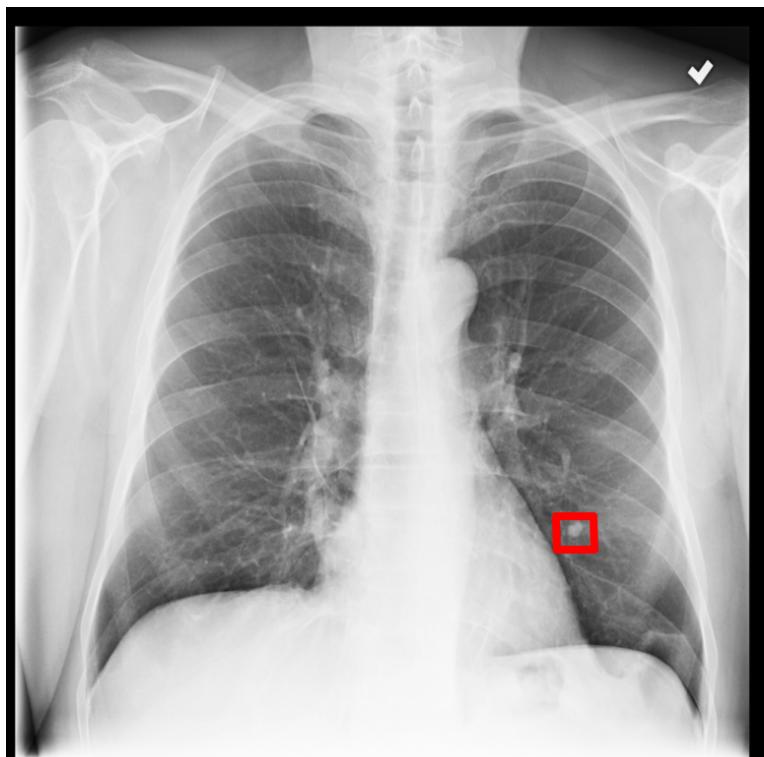


Figura 5.16: **Nódulo.**

Ejemplo de imagen que muestra un nódulo pulmonar y su respectiva anotación.

Se puede visualizar la distribución de estas clases, donde se muestra un desbalance de clases por parte de las imágenes sin hallazgos con respecto al resto.

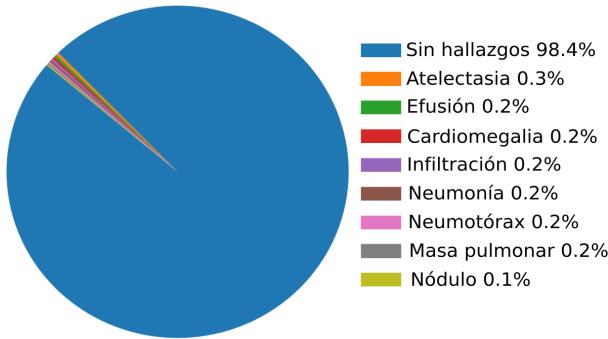


Figura 5.17: **Distribución de datos por clase desbalanceada.**

Gráfica de pastel que ilustra el porcentaje de clases etiquetadas.

Dado que en promedio hay 122 elementos por clase, reducimos aleatoriamente el subconjunto sin hallazgos a 122 elementos para balancear las clases.

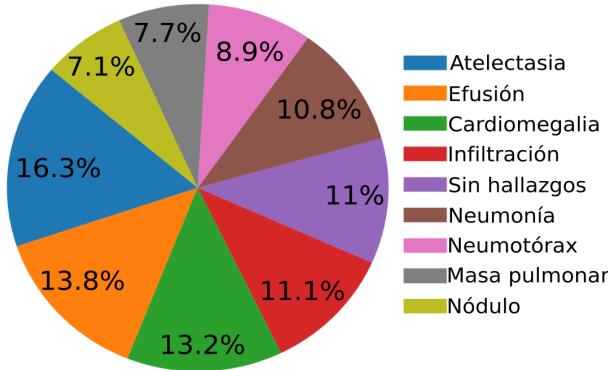


Figura 5.18: **Distribución de datos por clase valanceada.**

Gráfica de pastel que ilustra el porcentaje de clases etiquetadas.

Todas las imágenes cuentan con un solo recuadro delimitador, como se puede visualizar en el siguiente histograma.

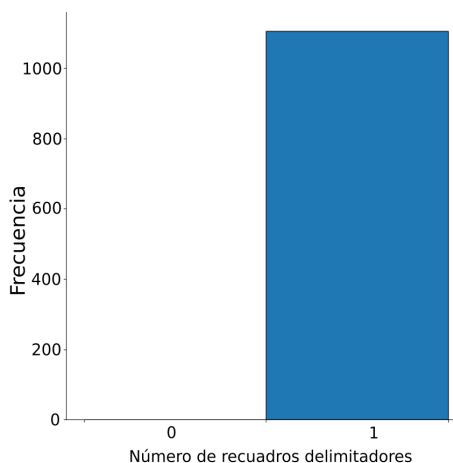


Figura 5.19: **Recuadros delimitadores por imagen.**

Histograma que ilustra el número de recuadros delimitadores presentes por imagen del conjunto.

La mayoría de los recuadros delimitadores en las anotaciones de opacidades tienen un área entre el 1% y el 8% de la imagen, lo cual es útil para entrenar a la red en la localización de daños pequeños.

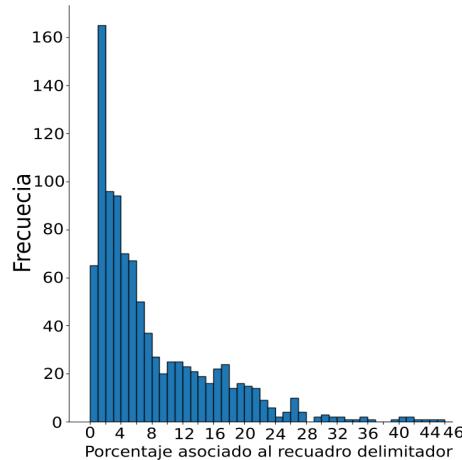


Figura 5.20: **Área de recuadros delimitadores.**

Histograma que ilustra el porcentaje del área total de la imagen que representan los recuadros delimitadores en el conjunto.

Se procede a analizar los histogramas de valores por píxeles de algunas imágenes pertenecientes a cada clase con la finalidad de observar si presentan algún patrón. Comenzando por las imágenes sin hallazgos, debido a la similitud de las tomas, el histograma muestra una tendencia hacia los tonos blancos.

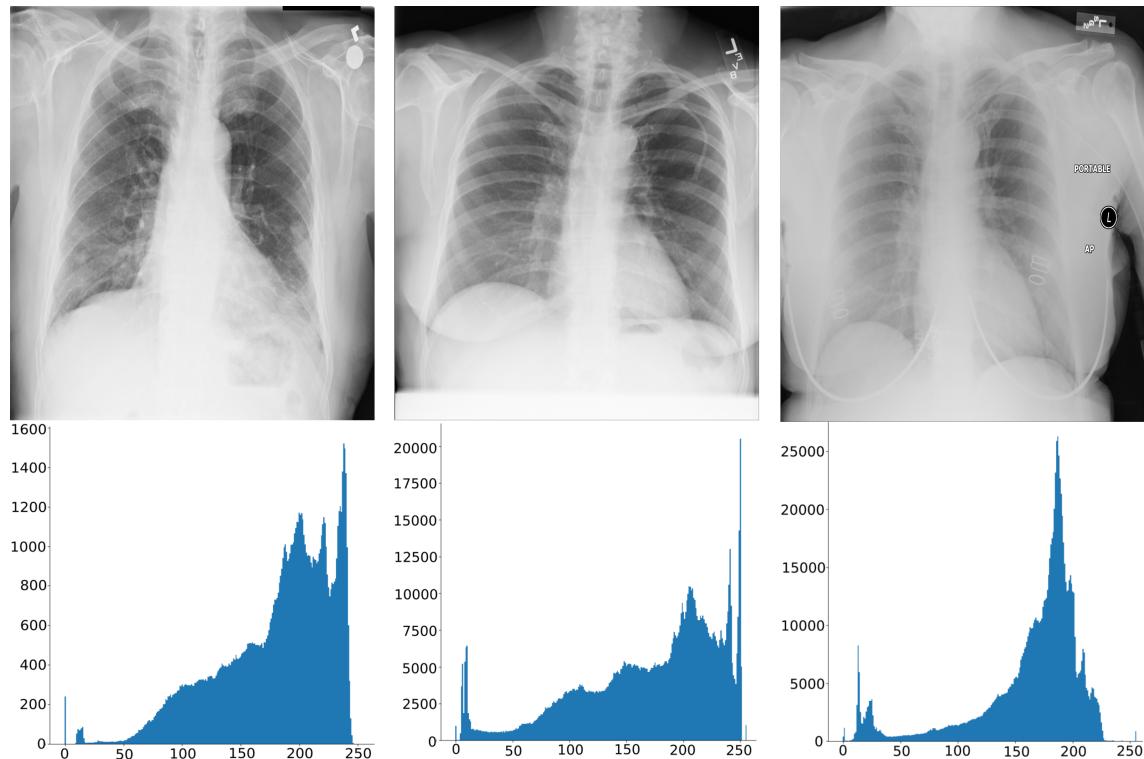


Figura 5.21: **Histograma de imágenes sin hallazgos.**

Imagen que ilustra, de arriba hacia abajo, imágenes del conjunto sin hallazgos médicos y sus respectivos histogramas, donde el eje x corresponde a los valores de píxel y el eje y a la frecuencia de valores.

Graficando los histogramas para imágenes con cardiomegalia, no se observa una tendencia dada la variabilidad de las imágenes.

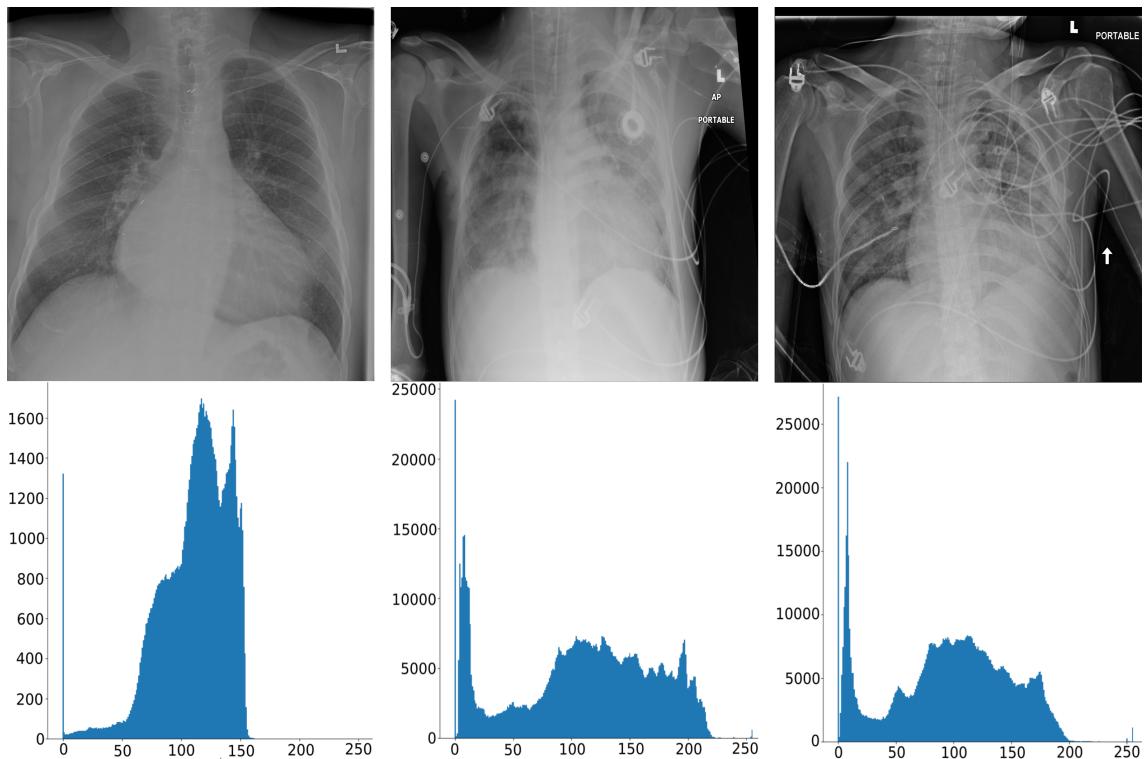


Figura 5.22: Histograma de imágenes con cardiomegalia.

Imagen que ilustra, de arriba hacia abajo, imágenes del conjunto con cardiomegalia y sus respectivos histogramas, donde el eje x corresponde a los valores de píxel y el eje y a la frecuencia de valores.

Graficando los histogramas para imágenes con atelectasia, no se observa una tendencia dada la variabilidad de las imágenes.

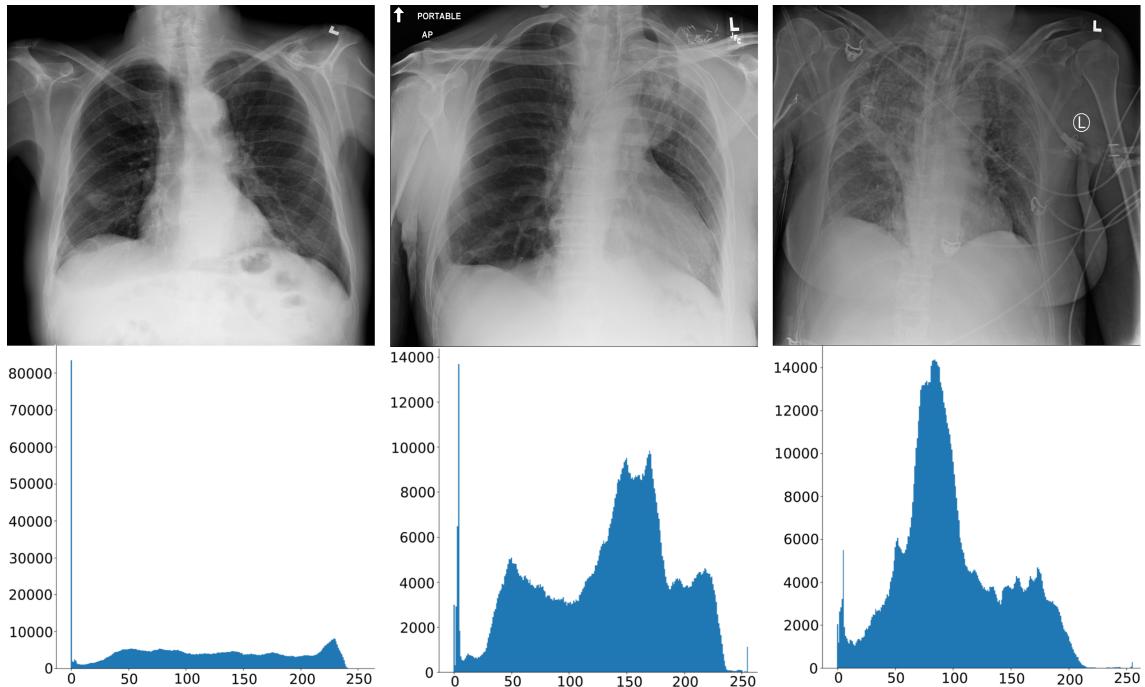


Figura 5.23: Histograma de imágenes con atelectasia.

Imagen que ilustra, de arriba hacia abajo, imágenes del conjunto con atelectasia y sus respectivos histogramas, donde el eje x corresponde a los valores de píxel y el eje y a la frecuencia de valores.

Graficando los histogramas para imágenes con efusión, no se observa una tendencia dada la variabilidad de las imágenes.

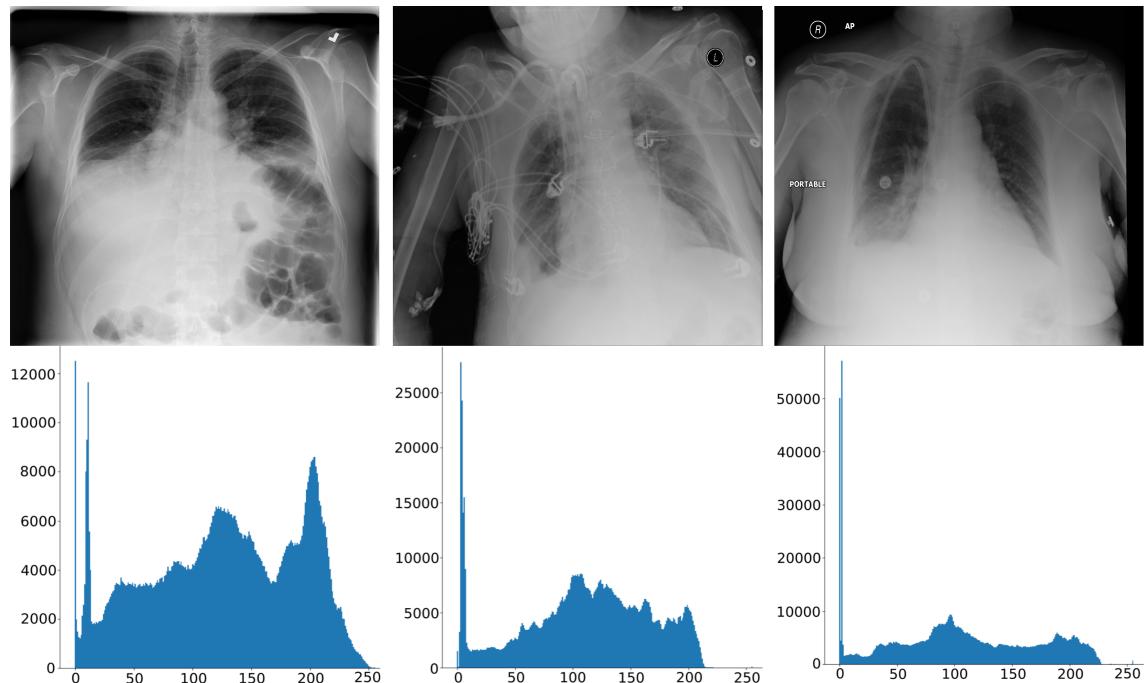


Figura 5.24: **Histograma de imágenes con efusión.**

Imagen que ilustra, de arriba hacia abajo, imágenes del conjunto con efusión y sus respectivos histogramas, donde el eje x corresponde a los valores de píxel y el eje y a la frecuencia de valores.

Graficando los histogramas para imágenes con infiltración, no se observa una tendencia dada la variabilidad de las imágenes.

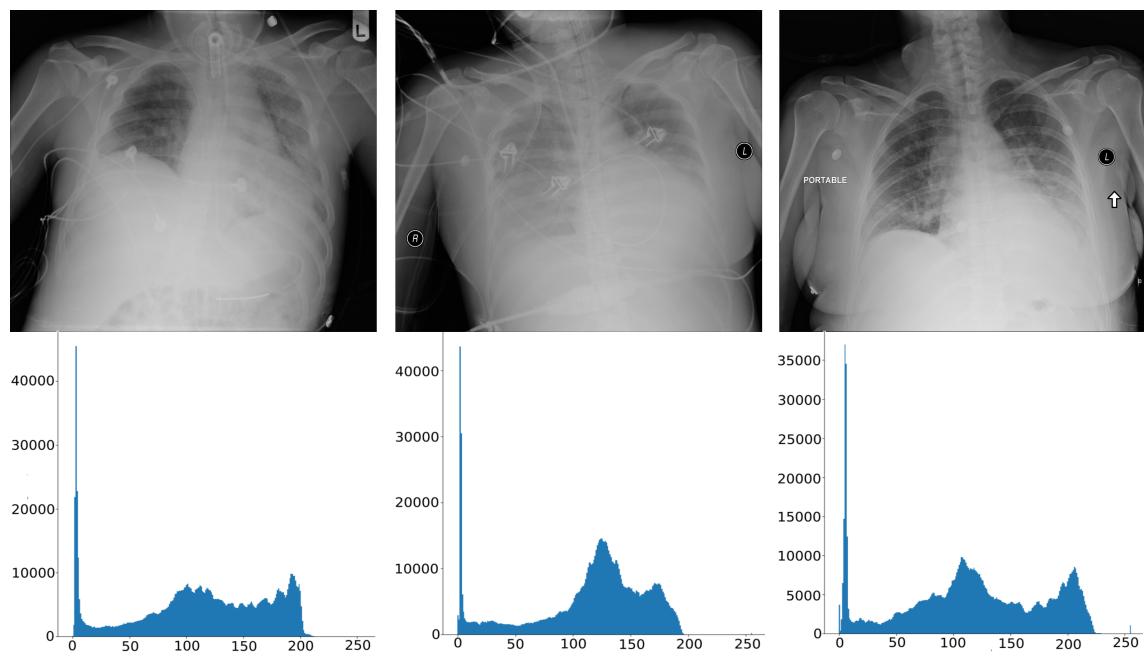


Figura 5.25: **Histograma de imágenes con infiltración.**

Imagen que ilustra, de arriba hacia abajo, imágenes del conjunto con infiltración y sus respectivos histogramas, donde el eje x corresponde a los valores de píxel y el eje y a la frecuencia de valores.

Graficando los histogramas para imágenes con neumonía, no se observa una tendencia dada la variabilidad de las imágenes.

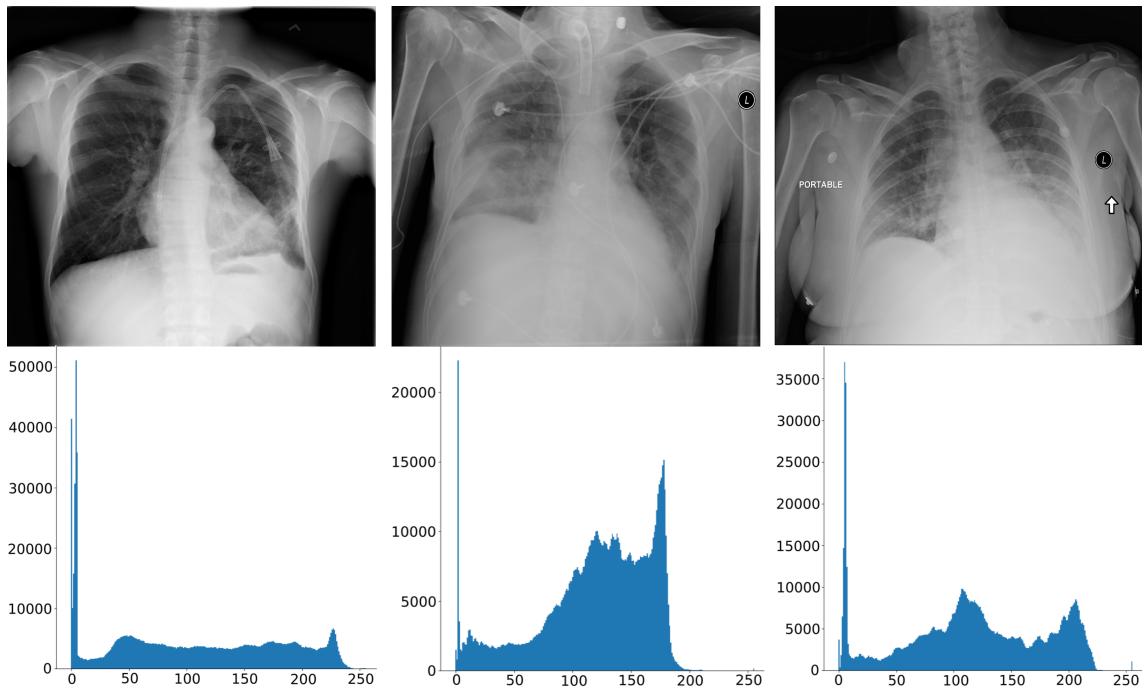


Figura 5.26: **Histograma de imágenes con neumonía.**

Imagen que ilustra, de arriba hacia abajo, imágenes del conjunto con neumonía y sus respectivos histogramas, donde el eje x corresponde a los valores de píxel y el eje y a la frecuencia de valores.

Graficando los histogramas para imágenes con neumotórax, no se observa una tendencia dada la variabilidad de las imágenes.

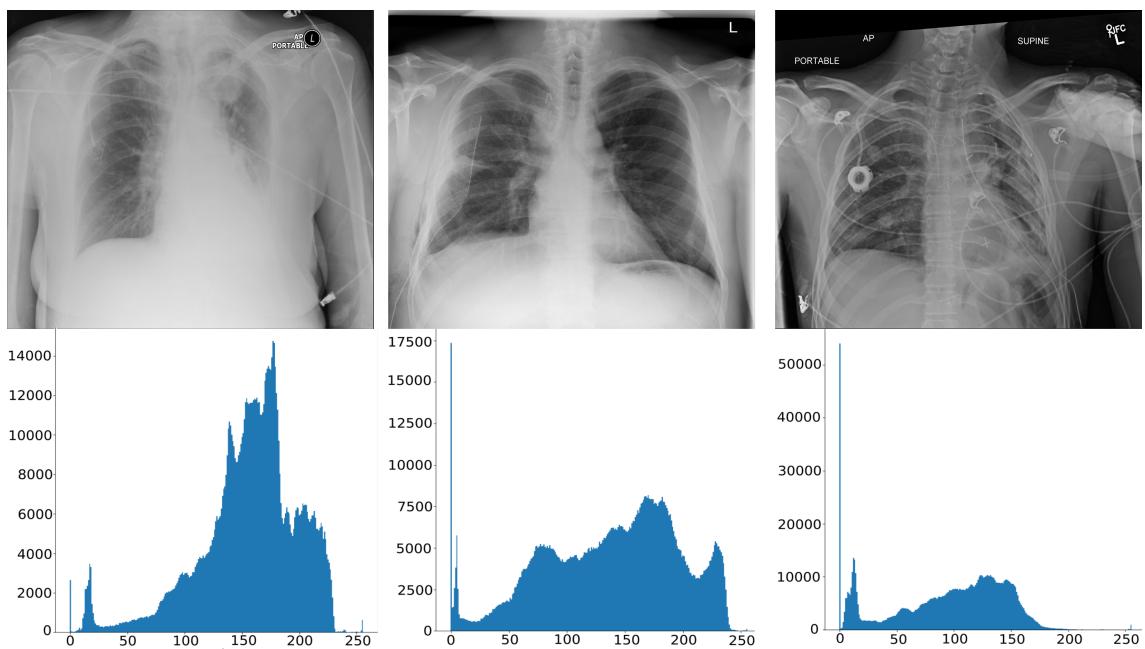


Figura 5.27: **Histograma de imágenes con neumotórax.**

Imagen que ilustra, de arriba hacia abajo, imágenes del conjunto con neumotórax y sus respectivos histogramas, donde el eje x corresponde a los valores de píxel y el eje y a la frecuencia de valores.

Graficando los histogramas para imágenes con masa pulmonar , no se observa una tendencia dada la variabilidad de las imágenes.

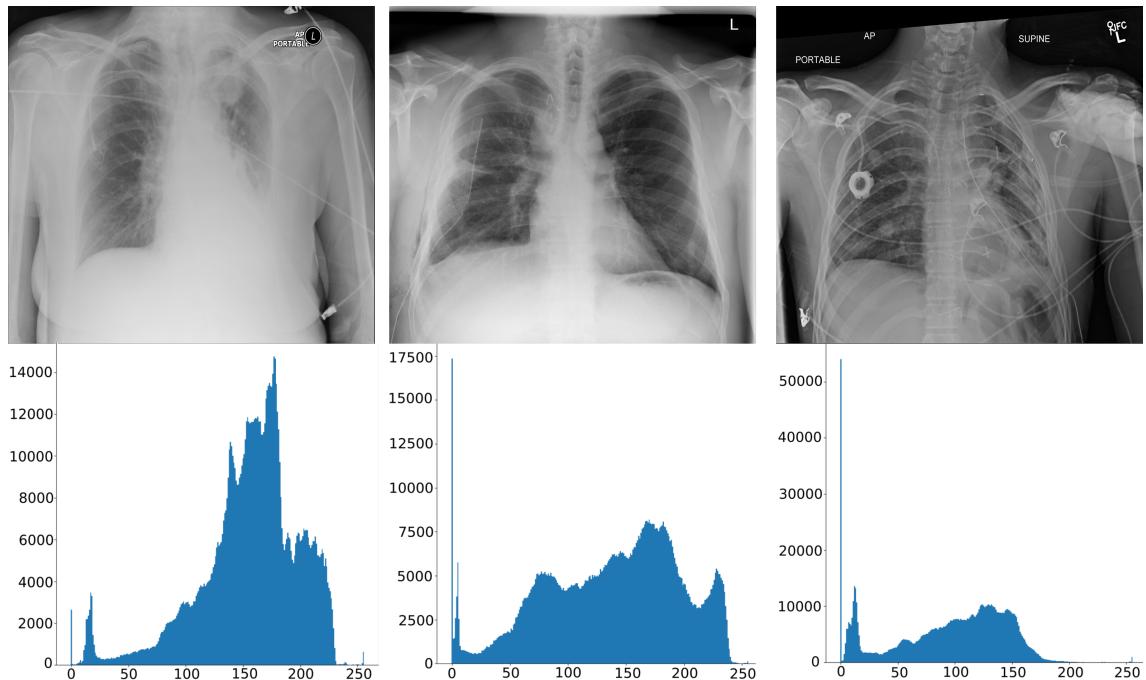


Figura 5.28: Histograma de imágenes con masa pulmonar.

Imagen que ilustra, de arriba hacia abajo, imágenes del conjunto con masa pulmonar y sus respectivos histogramas, donde el eje x corresponde a los valores de píxel y el eje y a la frecuencia de valores.

Graficando los histogramas para imágenes con masa pulmonar , no se observa una tendencia dada la variabilidad de las imágenes.

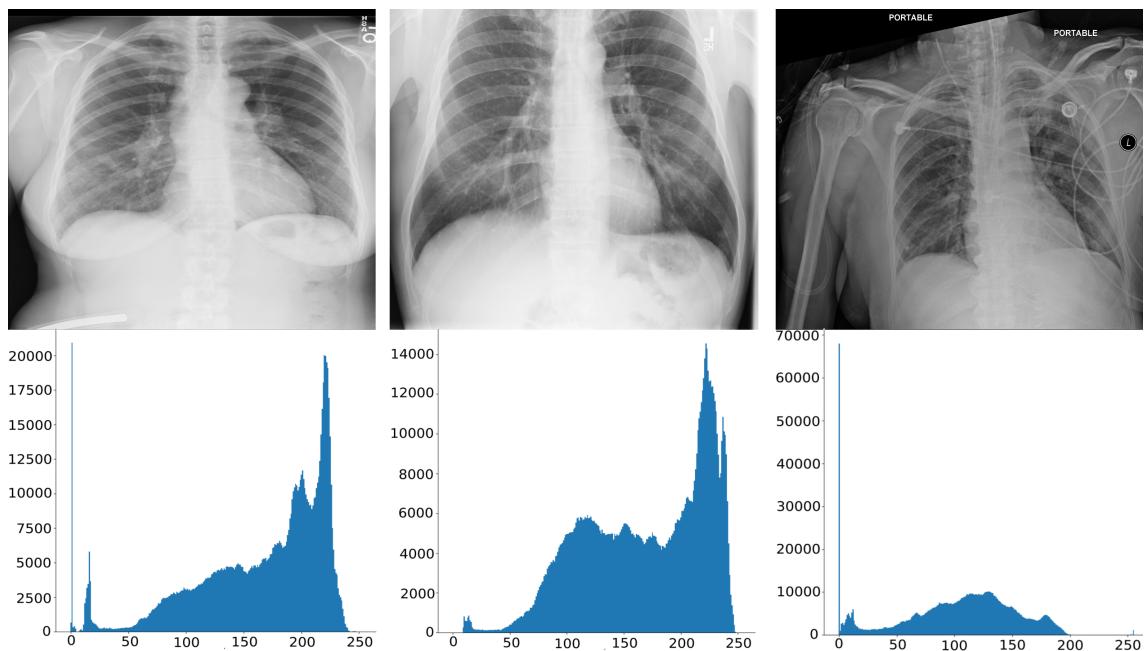


Figura 5.29: Histograma de imágenes con nódulo.

Imagen que ilustra, de arriba hacia abajo, imágenes del conjunto con nódulo y sus respectivos histogramas, donde el eje x corresponde a los valores de píxel y el eje y a la frecuencia de valores.

5.0.2.5. COVID19

El conjunto principal de datos que se utiliza en esta investigación es el ***SIIM-FISABIO-RSNA COVID-19 Kaggle Challenge*** [22], disponible en la competencia de Kaggle llamada *SIIM-FISABIO-RSNA COVID-19 Detection* a través del siguiente enlace:

[https://www.kaggle.com/c/siim-covid19-detection/data.](https://www.kaggle.com/c/siim-covid19-detection/data)

Este conjunto de datos es esencial para el desarrollo de esta investigación y consta de 24,313 imágenes de radiografías de tórax de diferentes tamaños con anotaciones sobre daños causados por COVID19, proporcionadas por tres instituciones:

Medical Imaging Data Resource Center (MIDRC)

RSNA International Covid-19 Open Radiology Database (RICORD)

Banco de Imagen Médica de la Comunidad Valenciana (BIMCV-COVID-19 Dataset)

Los datos fueron recopilados por el sistema sanitario de Valencia, España, que se divide en departamentos de salud ubicados en diferentes provincias. Un mapa interactivo de las regiones donde se tomaron las radiografías está disponible en el siguiente enlace:

<https://maigva.github.io/maps/HealthDepartCOVID19.html> [21].

El conjunto de datos se creó utilizando dos bases de datos públicas: **MIDRC-RICORD** [27] y **BIMCV** [21]. Estas bases de datos proporcionaron radiografías de pacientes positivos y negativos para COVID19. Luego, se eliminaron las imágenes que no tenían etiquetas DICOM asociadas, y se seleccionaron solo imágenes frontales. Además, se llevó a cabo un proceso de anonimización para proteger los datos personales de los pacientes.

El proceso de anotación fue realizado por 22 radiólogos provenientes de América del Norte, América del Sur y Europa. De estos, 9 tenían especialización en tórax y 13 no. Además, 19 de ellos estaban ejerciendo su profesión, mientras que 3 eran residentes de nivel senior en radiología.

Los anotadores tuvieron acceso a MD.ai, una herramienta de software de anotación proporcionada por el Centro AIMI para investigadores de Stanford, que ayuda a seleccionar y crear conjuntos de datos de imágenes médicas. Se puede consultar su página web aquí: <https://www.md.ai/>

Contaron con acceso a capacitaciones para el uso del software mediante conferencias web, así como materiales de referencia y casos de ejemplo para cada categoría anotada.

25 ejemplos fueron anotados por un radiólogo experto con 15 años de experiencia en radiología y 10 en radiología de tórax. Estas anotaciones se tomaron como referencia para evaluar el desempeño de cada anotador, requiriendo que estos alcanzaran un porcentaje de coincidencia del 60 % o más con el experto.

Los anotadores clasificaron las imágenes de acuerdo a los estudios en 4 categorías, siguiendo los siguientes estándares:

Clasificación de la radiografía	Hallazgos presentes
Apariencia típica	Opacidades bilaterales que muestran fibrosis o volumen pulmonar reducido, centrales y periféricas difusas.
Apariencia indeterminada	Ausencia de hallazgos típicos: Opacidades en la zona superior del pulmón opacidades unilaterales multifocales y opacidades centrales con preservación periférica relativa.
Apariencia atípica	Ausencia de hallazgos típicos o indeterminados: Neumotórax sin características de neumonía, masas o nódulos, neumonía lobar, cicatrización o fibrosis.
Negativo para neumonía	Sin opacidades pulmonares.

Tabla 5.1: **Criterios de anotación.**

Tabla recreada de [22], donde se muestra el esquema y criterios utilizados para la anotación de datos.

Se anotaron recuadros delimitadores en las opacidades del espacio aéreo pulmonar para los exámenes clasificados. Se consideraron derrames pleurales, masas/nódulos o neumotórax. Para los casos negativos de neumonía, no se colocaron cuadros delimitadores.

Para las opacidades que estaban muy cercanas o adyacentes, los anotadores colocaron un solo cuadro que cubría el área, con el objetivo de reducir la variabilidad.

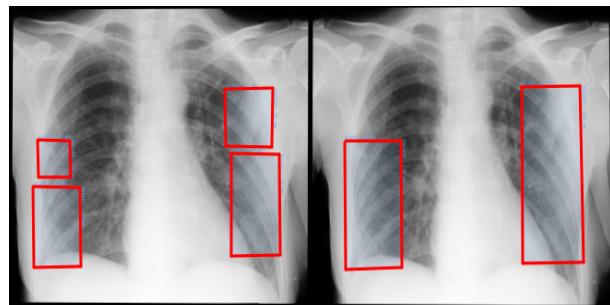


Figura 5.30: **Etiquetado de imágenes.**

Imagen modificada de [22], donde a la izquierda se muestra una imagen con anotaciones de opacidades cercanas, y a la derecha se presenta la anotación resultante del criterio para el conjunto de datos.

Por último, se presentan los siguientes estándares de interpretación para imágenes de rayos X:

Se proponen las siguientes clasificaciones de severidad en enfermedades pulmonares, de acuerdo al número de lesiones encontradas:

Se puede notar que las categorías coinciden con el siguiente lenguaje de informe de diagnóstico, que no está limitado para COVID-19.

Para obtener información detallada sobre los datos utilizados, consulte [21].

Clasificación de severidad.	Criterio de evaluación
Leve	Opacidades:1-2 zonas
Moderado	Opacidades:3-4 zonas
Severo	Opacidades:+4 zonas

Tabla 5.2: **Severidades.**

Tabla recreada de [28], donde se muestra una clasificación de severidad.

Clasificación radiográfica	Hallazgos en imagen	Informe sugerido
Apariencia típica	Opacidades periféricas bilaterales multifocales ,Opacidades con morfología redondeada con distribución predominante en pulmón inferior.	Los hallazgos típicos de neumonía por COVID19 están presentes.Sin embargo, estos pueden superponerse con otras infecciones y otras lesiones pulmonares agudas.
Apariencia indeterminada	Ausencia de hallazgos típicos de covid y distribución predominante unilateral, central o pulmonar superior.	Hallazgos indeterminados para la neumonía por COVID19 y que pueden ocurrir con una variedad de infecciones y condiciones no infecciosas.
Apariencia atípica	Neumotórax o derrame pleural, edema pulmonar, consolidación lobar, nódulo o masa pulmonar solitario diminutos y difusos.	Hallazgos atípicos o informados con poca frecuencia para neumonía por COVID19 puede tener diagnosticos alternativos.
Negativo para neumonía	Sin opacidades pulmonares.	Sin hallazgos de neumonía. Sin embargo, los hallazgos de la radiografía de tórax pueden estar ausentes al principio del curso de la neumonia por COVID19.

Tabla 5.3: **Lenguaje de informe.**

Tabla recreada de [28], donde se muestra el lenguaje de informe propuesto para los hallazgos de radiografías de rayos X.

Para detalles sobre la anotación y curación del conjunto de datos, consulte [22].

5.0.2.6. Análisis exploratorio

En esta sección se presenta un análisis exploratorio del conjunto de datos que contiene imágenes de radiografías con anotaciones sobre distintas lesiones pulmonares causadas por la enfermedad COVID19.

El conjunto de datos cuenta con 6,334 imágenes, las cuales tienen anotaciones de 3 clases distintas de daño pulmonar, así como el recuadro delimitador que señala la localización del daño en la imagen.

Negativo para neumonía: Imagenes sin daños causados por COVID19.



Figura 5.31: **Negativo para neumonía.**

Ejemplo de imagen que muestra una persona sin neumonía por COVID19.

Aparaciencia típica: Imagenes con daños típicos del COVID19.

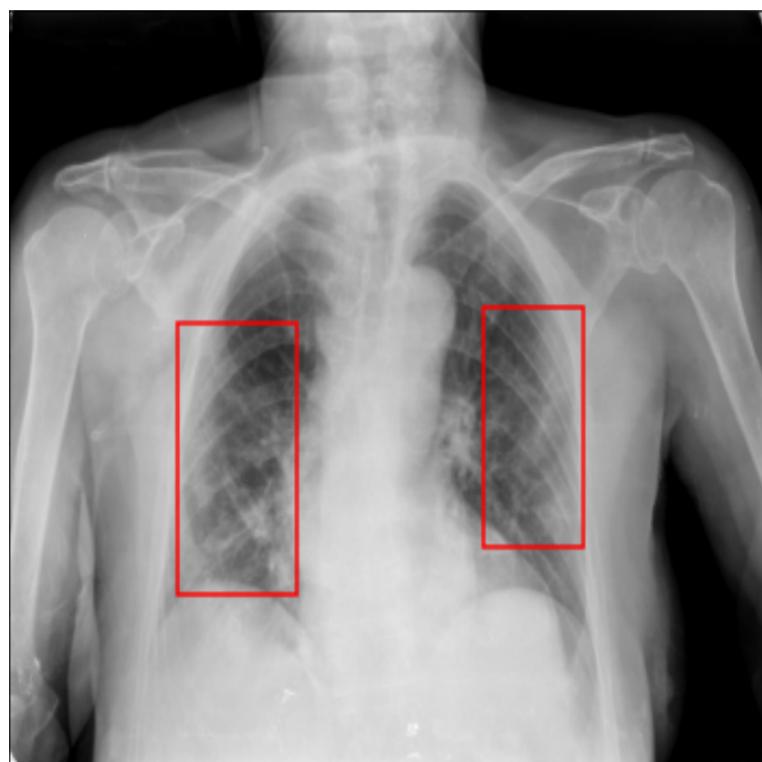


Figura 5.32: **Aparaciencia típica.**

Ejemplo de imagen que muestra daños típicos de COVID19.

Aparaciencia indeterminada: Imagenes con daños pulmonares ajenos a CO-VID19.

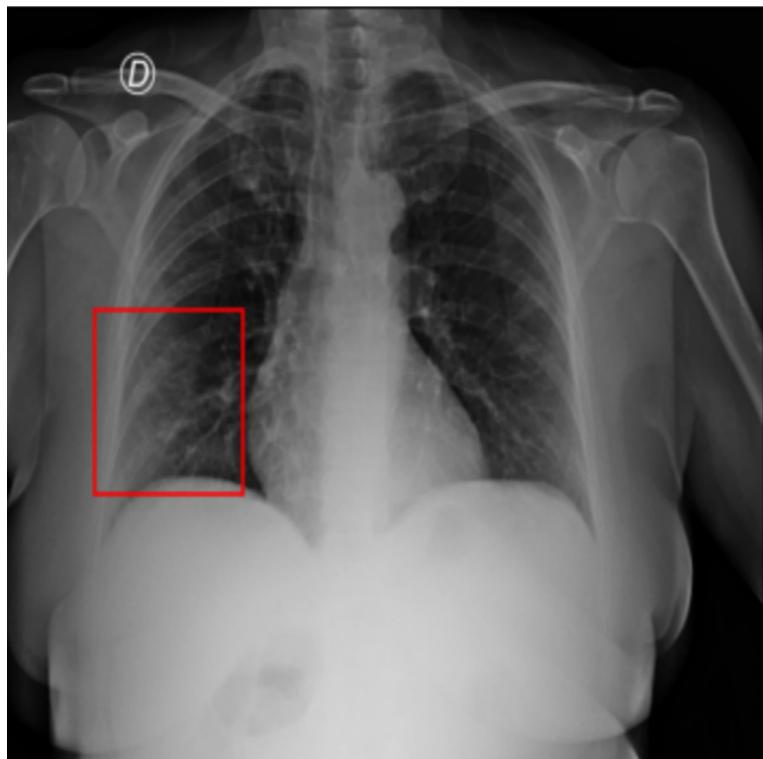


Figura 5.33: **Aparaciencia indeterminada.**
Ejemplo de imagen que muestra daños ajenos a COVID19.

Aparaciencia atípica: Imagenes con daños pulmonares pasientes con COVID19 poco observados.

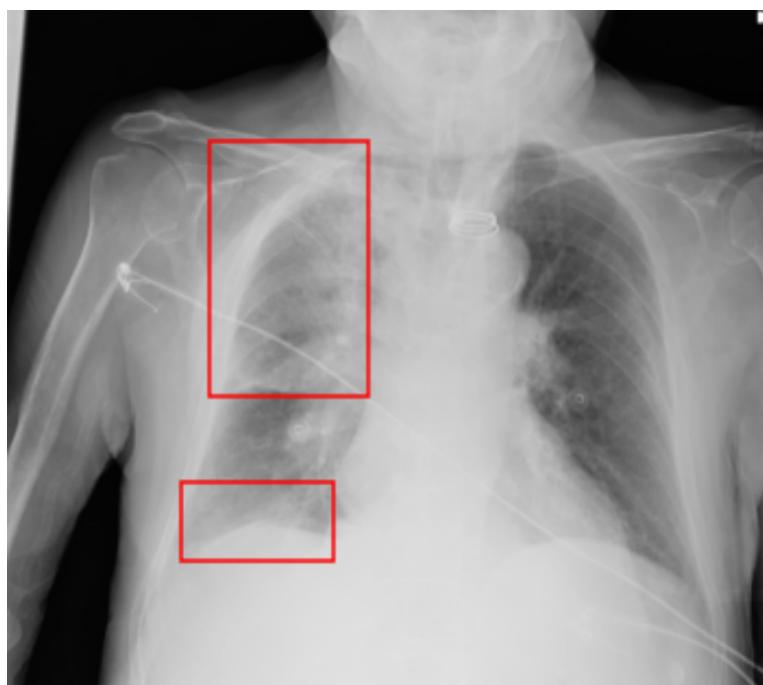


Figura 5.34: **Aparaciencia atípica.**
Ejemplo de imagen que muestra daños atípicos de COVID19.

Visualizando la distribución de clases en el conjunto, se nota una cantidad muy pequeña de casos atípicos.

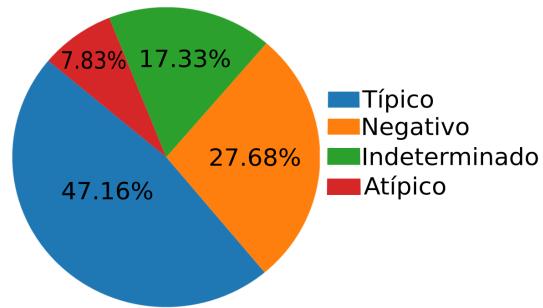


Figura 5.35: **Distribución de clases.**

Gráfica de pastel que ilustra el porcentaje de elementos en cada clase definida para el conjunto.

Se tienen imágenes tanto en monocromo1 como monocromo2, por lo que todas las imágenes deben ser transformadas a monocromo2. Se debe tener en cuenta la inversión de colores entre negros y blancos.

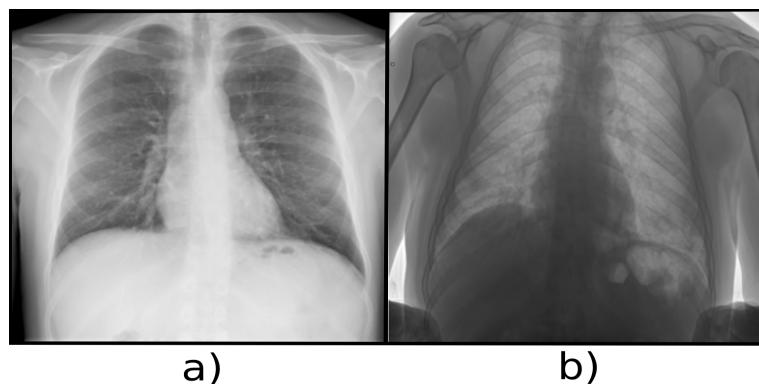


Figura 5.36: **Monocromo 2 y 1.**

Ejemplos de imágenes tomadas del conjunto. a) Muestra una imagen en monocromo2, mientras que b) muestra una imagen en monocromo1.

Las imágenes cuentan con entre 0 y 5 recuadros delimitador, como se puede visualizar en el siguiente histograma.

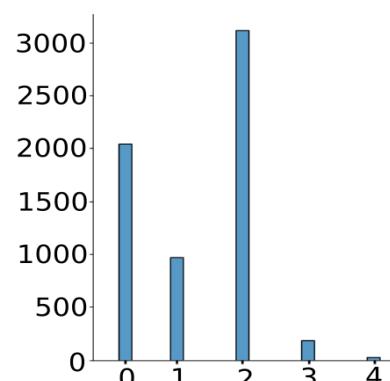


Figura 5.37: **Recuadros delimitadores por imagen.**

Histograma que ilustra el número de recuadros delimitadores presentes por imagen del conjunto.

Las imágenes tienen tamaños muy variados, como se observa en la siguiente gráfica.

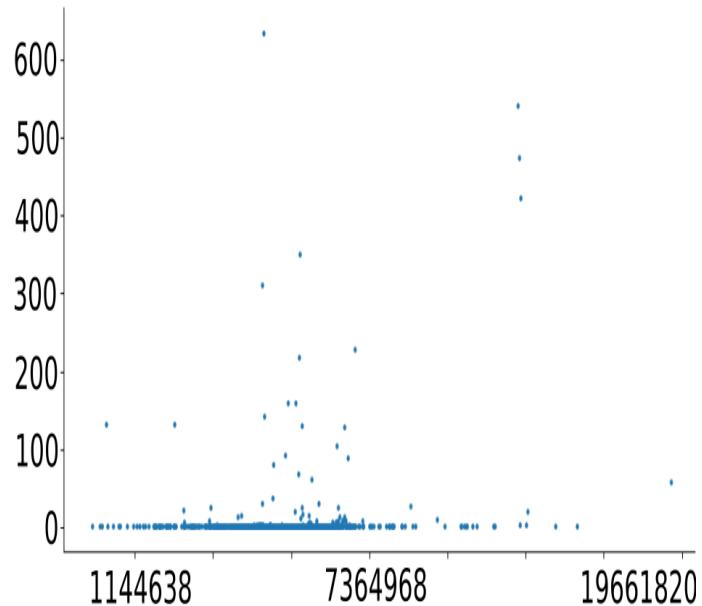


Figura 5.38: **Área de las imágenes.**

Gráfica que muestra el tamaño de cada imagen en el conjunto, donde el eje x representa el área de la imagen en píxeles y el eje y muestra la frecuencia de imágenes de ese tamaño.

La mayoría de los recuadros delimitadores en las anotaciones de daño pulmonar tienen un área entre el 1% y el 15% con respecto al área de la imagen.

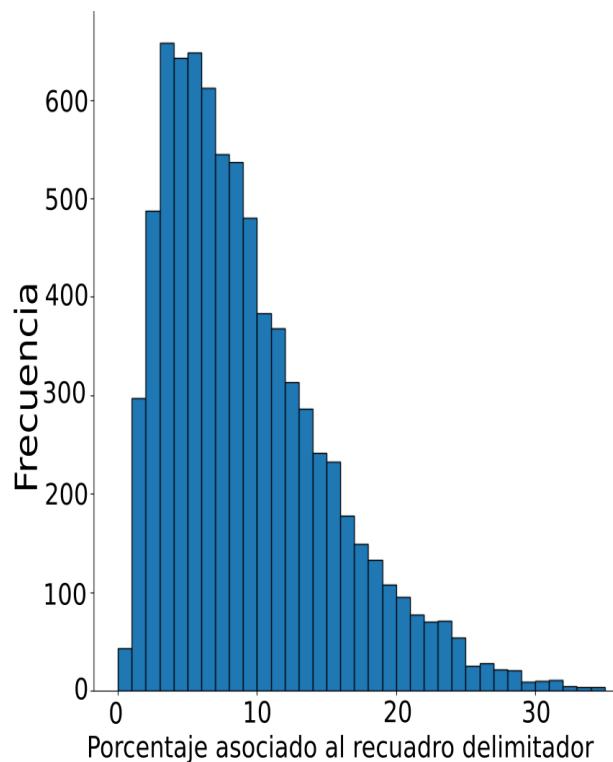


Figura 5.39: **Área de recuadros delimitadores.**

Histograma que ilustra el porcentaje del área total de la imagen que representan los recuadros delimitadores en el conjunto.

Estimando la cantidad de imágenes sin recuadros delimitadores por clase.

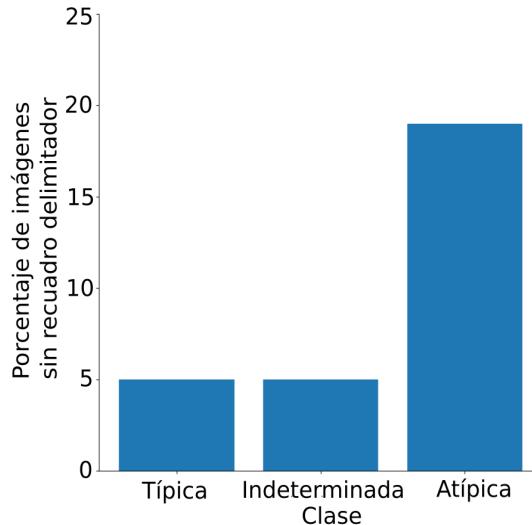


Figura 5.40: **Imágenes sin recuadros delimitadores.**

Gráfica que ilustra el porcentaje de imágenes sin recuadros delimitadores para cada tipo de daño pulmonar en el conjunto.

Se procede a analizar los histogramas de valores por píxeles de algunas imágenes pertenecientes a cada clase con la finalidad de observar si presentan algún patrón. Comenzando por las imágenes sin daños, donde no se presenta ningún patrón debido a la alta variabilidad de los datos.

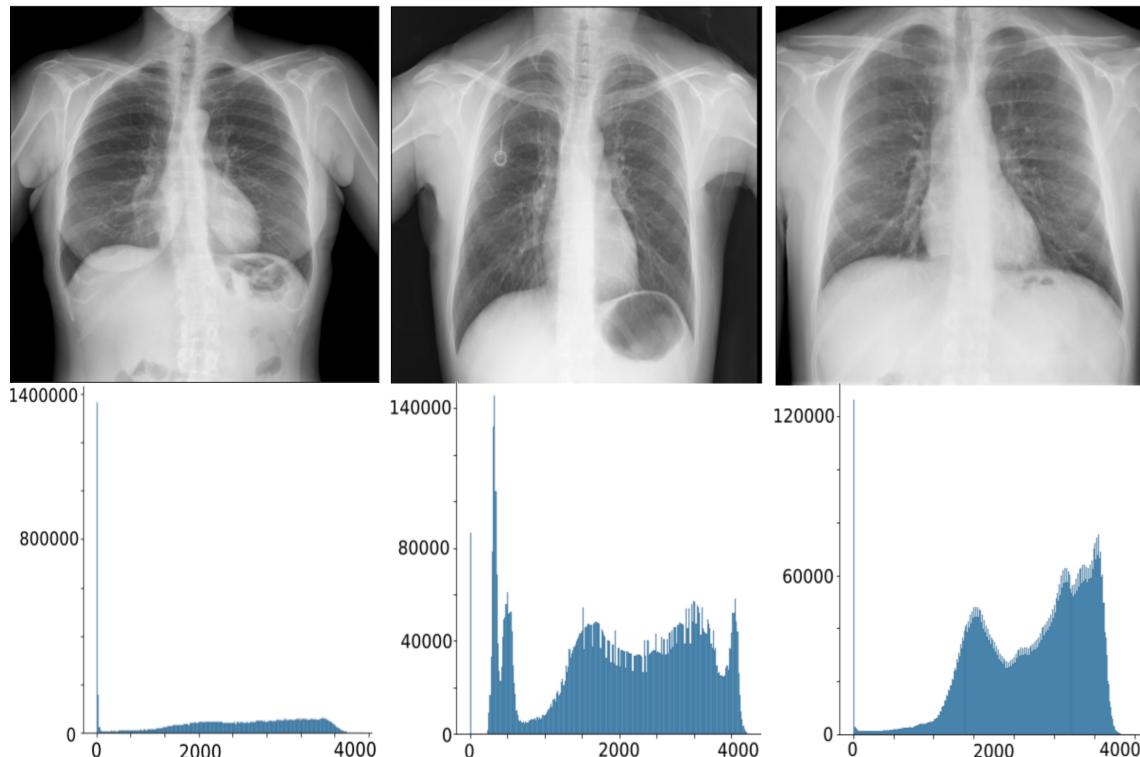


Figura 5.41: **Histograma de imágenes sin daños.**

Imagen que ilustra, de arriba hacia abajo, imágenes del conjunto sin daños pulmonares y sus respectivos histogramas, donde el eje x corresponde a los valores de píxel y el eje y a la frecuencia de valores.

Graficando los histogramas para imágenes con daños típicos, no se observa una tendencia dada la alta variabilidad de las imágenes.

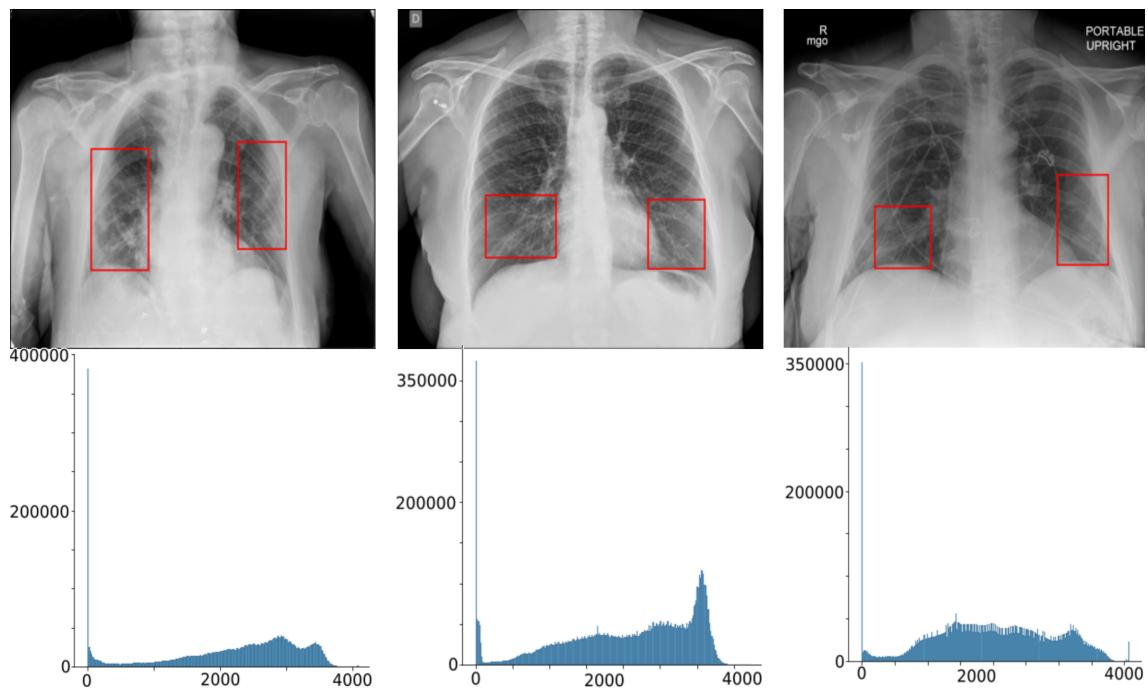


Figura 5.42: Histograma de imágenes con daños típicos.

Imagen que ilustra, de arriba hacia abajo, imágenes del conjunto con daños típicos y sus respectivos histogramas, donde el eje x corresponde a los valores de píxel y el eje y a la frecuencia de valores.

Graficando los histogramas para imágenes con daños indeterminados, no se observa una tendencia dada la alta variabilidad de las imágenes.

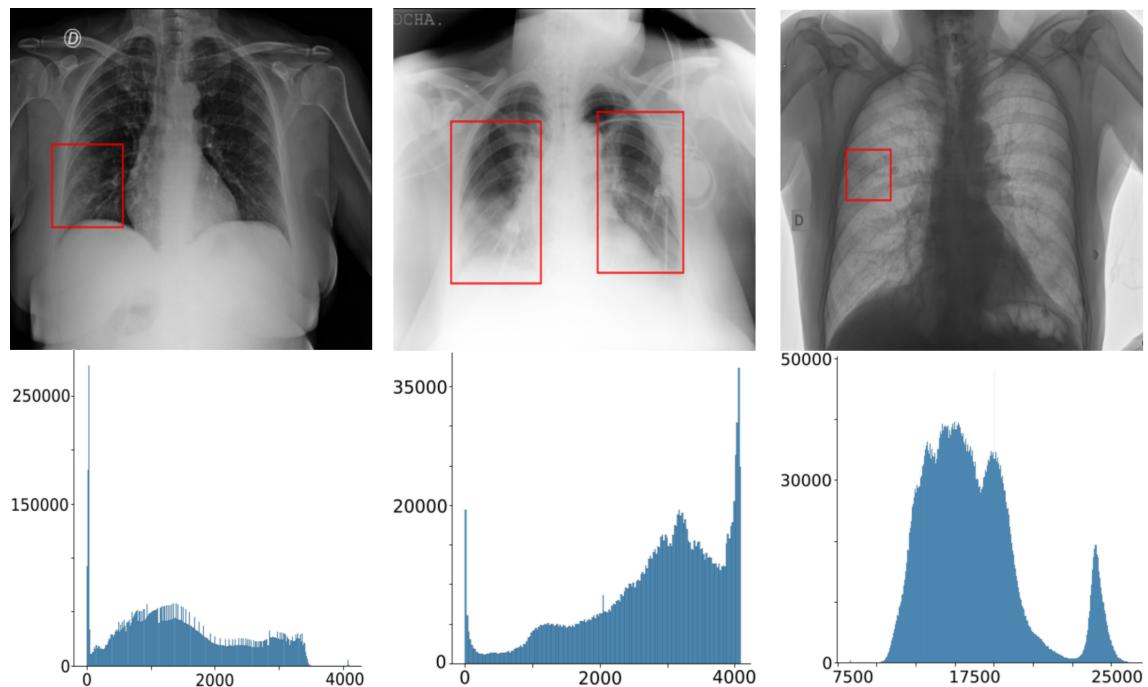


Figura 5.43: Histograma de imágenes con daños indeterminados.

Imagen que ilustra, de arriba hacia abajo, imágenes del conjunto con daños indeterminados y sus respectivos histogramas, donde el eje x corresponde a los valores de píxel y el eje y a la frecuencia de valores.

Graficando los histogramas para imágenes con daños indeterminados, no se observa una tendencia dada la alta variabilidad de las imágenes.

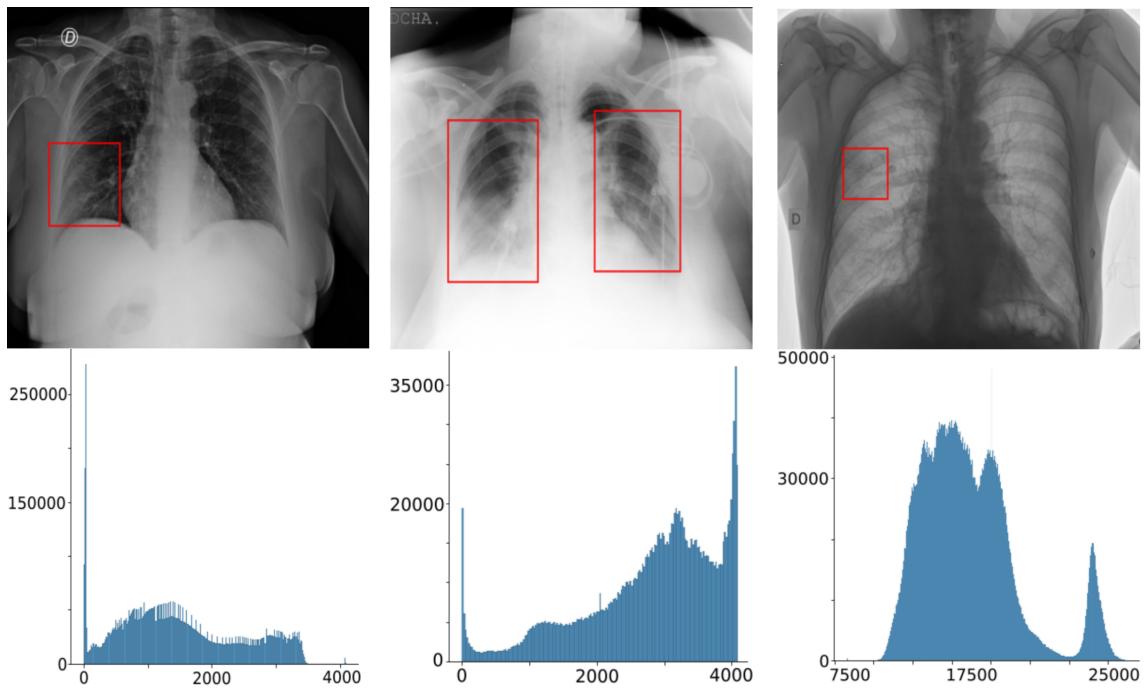


Figura 5.44: Histograma de imágenes con daños indeterminados.

Imagen que ilustra, de arriba hacia abajo, imágenes del conjunto con daños indeterminados y sus respectivos histogramas, donde el eje *x* corresponde a los valores de píxel y el eje *y* a la frecuencia de valores.

Graficando los histogramas para imágenes con daños atípicos, no se observa una tendencia dada la alta variabilidad de las imágenes.

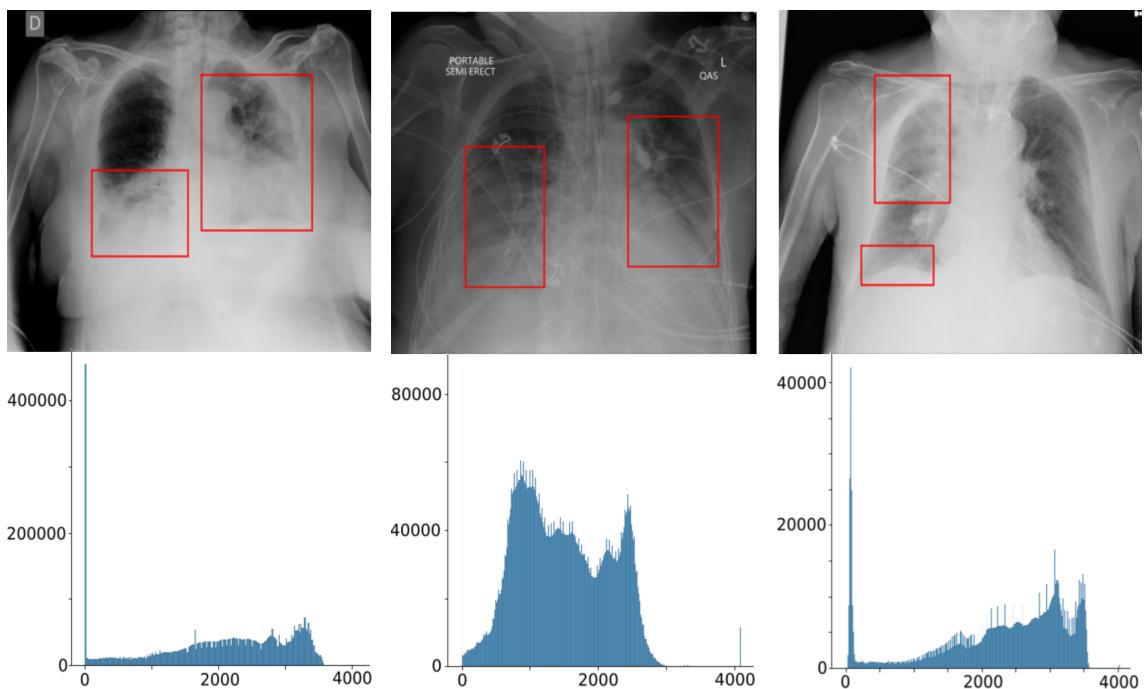


Figura 5.45: Histograma de imágenes con daños atípicos.

Imagen que ilustra, de arriba hacia abajo, imágenes del conjunto con daños atípicos y sus respectivos histogramas, donde el eje *x* corresponde a los valores de píxel y el eje *y* a la frecuencia de valores.

Este conjunto de datos presenta menos homogeneidad, es muy probable que esto se deba a la situación pandémica en la que se definió el conjunto.

Capítulo 6

Métodos

En esta sección se presentan las acciones y propuestas planteadas para alcanzar el objetivo principal de la presente investigación, que consiste en mejorar la métrica mAP para modelos de aprendizaje profundo en la tarea de localizar y detectar opacidades causadas por COVID19.

Esto proporciona una visión general de la investigación y su rigor.

Para este trabajo, se comienza con dos redes neuronales:

Una implementación de RetinaNet como está descrita en el artículo original [83], proporcionada en el repositorio público de GitHub disponible en el enlace

<https://github.com/yhenon/pytorch-retinanet>

Y la implementación de YOLO [13], en la versión 8 proporcionada por Ultralytics [51].

Ambas pre-entrenadas en el conjunto de datos COCO Objects [3].

Se comienza con el conjunto de datos *RSNA-PNEUMONIA* descrito en la subsección 5.0.2.1, que permite realizar las primeras evaluaciones y decisiones sobre los modelos en la tarea de detectar la presencia o ausencia de opacidades causadas por neumonía, tarea que es más sencilla en comparación con la complejidad que representa el objetivo de este trabajo. Así se procede de la siguiente manera:

Yolov8 cuenta con implementaciones de distintos tamaños acorde a la cantidad de parámetros aprendidos por la red:

Tamaño	mAP	Parámetros
Nano	37.3	3.2 millones
Small	44.9	11.2 millones
Medium	50.2	25.9 millones
Large	52.9	43.7 millones
Xtra large	53.9	257.8 millones

Tabla 6.1: **Tamaños de Yolov8.**

Tabla recreada de [51], donde se muestran los tamaños disponibles de la arquitectura, el mAP obtenido en su preentrenamiento y el número de parámetros en la red.

Debido a que la red más pequeña requiere menos tiempo para el proceso de entrenamiento, se utiliza como modelo inicial. Primero, con los parámetros predefinidos de la red, los cuales se pueden consultar en el siguiente enlace:

<https://docs.ultralytics.com/es/modes/train/#augmentation-settings-and-hyperparameters>

Se determina el número de épocas optimas para el aprendizaje de la red, esto se logra entrenando durante algunas épocas preestablecidas e implementando una función que durante el proceso de entrenamiento guarda el modelo en el punto de mejor desempeño y también al no mostrar una mejora en las metricas despues de cierto número de épocas definidas (*patience*).

Despues se procede a explorar el espacio de parámetros, variando las opciones predefinidas del modelo para el aumeto de datos con el fin de determinar cuál ofrece el mejor resultado.

Una vez determinado esto, se procede a determinar el tipo de optimizador que brinda un mejor desempeño.

Se procede a probar la implementación de distintas tecnicas de aprendizaje automatico como el uso de una tasa de aprendizaje cosenoidal, *Drop out* y la modificación del paraméetro que da peso a la granulariad de la clásificación del modelo.

Al comparar los rendimientos ofrecidos por cada uno de estos entrenamientos se obtine el de mejor desempeño, con la experiencia obtenida del problema y dada la cantidad de tiempo que requiere el entrenamiento de la arquitectura RetinaNet, se realiza un entrenamiento con el mismo conjunto de datos y una elección de parámetros que ofrece RetinaNet para poder comparar los desempeños entre estas dos redes.

Hasta este punto se ha tratado de explorar las condiciones óptimas respecto a los paramétrios. Ahora, se explora la mejora de las métricas y desempeños a traves de los datos.

El proceso descrito anteriormente se ejecuta con el conjunto de datos original, salvo una redimensión de 640×640 píxeles. Dada la naturaleza de características finas respecto a las opacidades, se explora un entrenamiento con imágenes en su tamaño original de 1024×1024 para evaluar si esto trae alguna mejora en el aprendizaje y la calidad de las predicciones para YOLO.

Una vez determinado esto, se explora el entrenamiento con distintas anotaciones para el caso de imágenes sin presencia de opacidades, mediante las siguientes opciones, recordando que los conjuntos de datos originales tienen recuadros delimitadores que encierran los objetos de interés y los contraejemplos no muestran anotación alguna.

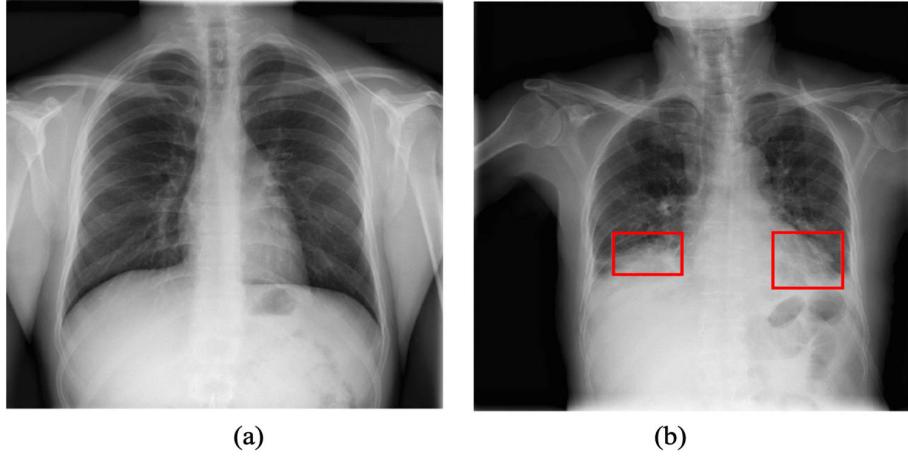


Figura 6.1: Anotaciones originales en los conjuntos.

Imagen modificada de [46]. a) Muestra una imagen de una persona sana, sin presencia de opacidades. b) Imagen que muestra anotaciones referentes a la presencia de opacidades en los pulmones.

Ahora se propone una nueva anotación para imágenes sin presencia de opacidades, que funciona como control. Se etiqueta un recuadro delimitador que abarca todo el contorno de la imagen completa. Esto no debería mostrar cambios en el desempeño del modelo, ya que estos, al no contar con anotaciones, extraen características de la imagen total.

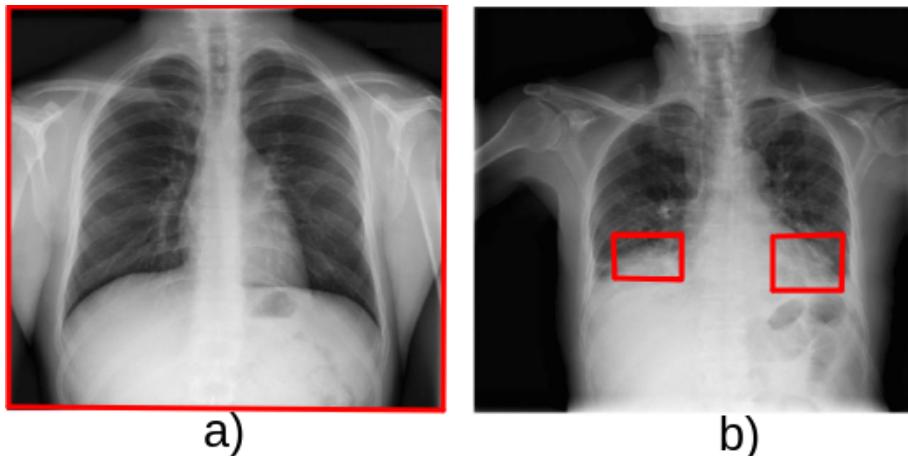


Figura 6.2: Anotaciones propuestas.

Imagen modificada de [46]. a) Muestra una imagen de una persona sana, con la anotación propuesta. b) Imagen que muestra anotaciones referentes a la presencia de opacidades en los pulmones.

Por otro lado, se propone un etiquetado en imágenes sin presencia de opacidades que centra la atención en las regiones de los pulmones. De esta forma, se espera que reduciendo la región de interés, los modelos presenten una mejora en el desempeño de la tarea.

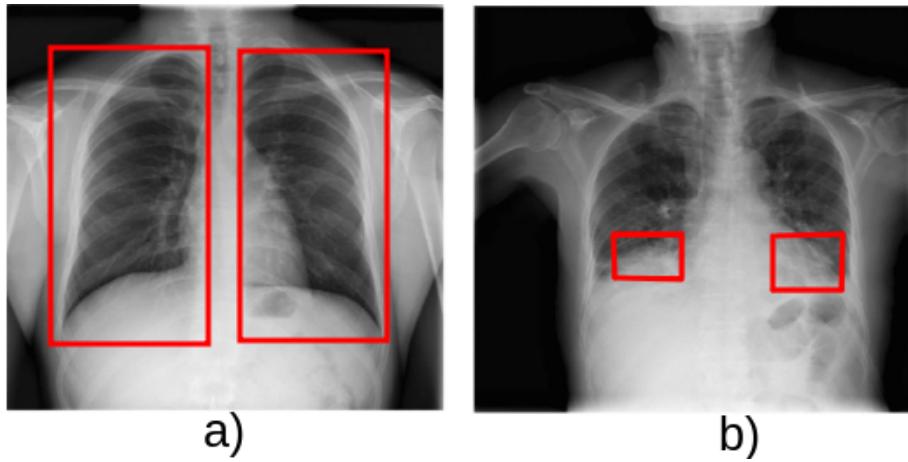


Figura 6.3: **Anotaciones propuestas.**

Imagen modificada de [46]. a) Muestra una imagen de una persona sana, con la anotación propuesta. b) Imagen que muestra anotaciones referentes a la presencia de opacidades en los pulmones.

Se propone explorar la modificación de la arquitectura al incluir capas que permitan la aplicación de atención a regiones pequeñas de la imagen, similar a lo explorado en el precedente [7] para tomografías computarizadas, lo cual mostró una mejora significativa en el rendimiento del modelo.

Posteriormente se procede a indagar qué profundidad muestra el mejor desempeño para YOLO, entrenando cada uno de los tamaños disponibles que se comparan para determinar el óptimo, así se procede nuevamente a extrapolar lo aprendido a RetinaNet para comparar los resultados entre modelos.

En este punto, el objetivo es obtener el mejor modelo en la tarea de detectar opacidades por neumonía, para luego proceder con la transferencia de conocimiento modular que busca mejorar la métrica mAP en nuestra tarea final que es la localización e identificación de daño pulmonar causado por COVID19. El proceso se lleva a cabo de la siguiente manera:

Con el modelo óptimo y las mejoras en los datos para la localización de opacidades en imágenes por neumonía, se realiza un reentrenamiento utilizando los parámetros aprendidos por el modelo, pero esta vez sobre el conjunto NIH-Chest-X-ray14, descrito en la subsección 5.0.2.3, aplicando el proceso de entrenamiento con mejor desempeño, con el objetivo de ampliar la capacidad de localización a opacidades generadas por distintas condiciones, lo que brinda flexibilidad de predicción al modelo.

Posteriormente, se realiza una última transferencia de conocimiento con el conjunto SIIM-FISABIO-RSNA COVID19, descrito en la subsección 5.0.2.5, utilizando el modelo anteriormente preentrenado. Se aplica la misma rutina de entrenamiento con el fin de mejorar el desempeño en la localización y detección de los 4 tipos de ejemplos presentes en este conjunto.

Este procedimiento se repite para la arquitectura RetinaNet, con el fin de comparar la mejora en ambos modelos.

Dependiendo de si esta metodología brinda una mejora significativa de la métrica

mAP en este punto de la investigación, se puede concluir con resultados positivos sobre los objetivos planteados o concluir con resultados negativos.

De esta forma, se cuenta con un marco de trabajo flexible y robusto que explora gran parte de las posibles mejoras en cuanto a parámetros, arquitectura y datos. Si no se muestra una mejora significativa en la métrica mAP de los modelos, se concluye la investigación con resultados negativos y se marca un precedente para la implementación de nuevas mejoras, desarrollos o aplicaciones, así como una propuesta para la evaluación en el análisis de datos de otra índole o de mayor calidad.

De esta manera, se puede resumir la metodología aquí presentada de la siguiente manera:

Elección de parámetros: Se toma el modelo de YOLO más pequeño para determinar las épocas necesarias y los parámetros que muestren un mejor desempeño en la detección de opacidades provocadas por neumonía en el conjunto RSNA-PNEUMONIA.

Mejora con respecto a los datos: Realizar entrenamientos con distintos tamaños de imagen y distintos tipos de anotaciones para los pacientes sanos utilizando YOLO para evaluar la mejora de resultados.

Comparación de modelos: Con la experiencia para YOLO, se genera un entrenamiento para RetinaNet y se comparan los resultados respecto a YOLO.

Modificación de modelos: Implementar capas de atención y evaluar la mejora en el desempeño de los modelos.

Elección de profundidad: Se entranan y evalúan los modelos de distinto tamaño para YOLO para determinar cuál es el que brinda mejores resultados.

Comparar modelos: Con la experiencia reentrenar RetinaNet para evaluar si presenta una mejora y comparar el desempeño de los modelos.

Realizar transferencia de conocimiento modular: Teniendo una red y anotaciones que muestren un buen desempeño en la detección de opacidades, aplicar transferencia de conocimiento en el conjunto NIH CHEST-X-Ray14 para extender la detección a más tipos de opacidades. Luego, proceder a aplicar otra transferencia de conocimiento en el conjunto SIIM-FISABIO-RSNA COVID19 para la detección e identificación de COVID19.

Evaluación: Evaluar el desempeño final utilizando la métrica mAP como referencia para ambos modelos.

Capítulo 7

Discusión principal

En esta sección se plasman y discuten en detalle los resultados obtenidos al aplicar la metodología propuesta en el capítulo anterior.

7.0.1. Exploración de parámetros

Se procede a buscar los parámetros mediante distintos entrenamientos para evaluar aquellos que muestren un mejor rendimiento del modelo.

Se comienza buscando los parámetros para el modelo Yolov8 que ofrecen el mejor desempeño para el modelo. Todos los entrenamientos aquí plasmados se llevaron a cabo usando el conjunto *RSNA PNEUMONIA*, descrito en la subsección 5.0.2.1, con las imágenes reescaladas a 640×640 píxeles y con sus anotaciones originales, donde se cuentan con recuadros delimitadores para la localización del daño causado por neumonía, y los contraejemplos no cuentan con anotaciones.

Se utiliza la versión *nano* del modelo para determinar los parámetros.

Para el entrenamiento, el modelo utiliza una combinación ponderada de dos funciones de costo: para la clasificación utiliza la Entropía cruzada y para la localización el Error cuadrático medio (MSE), las cuales combina en una función de pérdida total dada por:

$$\text{Total} = \lambda \text{Entropía cruzada} + (1 - \lambda) \text{MSE}$$

En el proceso de evaluación para el modelo se utiliza la métrica mAP50 y mAP50-95.

Nota importante: Dado que el conjunto de datos solo contiene anotaciones para las imágenes con opacidades generadas por neumonía, para las imágenes sin presencia de opacidades, el modelo las interpreta como imágenes de fondo. Por lo tanto, al hacer predicciones para imágenes sin opacidades, el modelo las clasifica a veces como imágenes de fondo y otras veces como listas vacías sin recuadros delimitadores con la etiqueta de sin detecciones.

Para la evaluación de la clasificación mediante las matrices de confusión, se consideró cualquiera de las dos predicciones como acertadas para imágenes sin presencia de opacidades. Este ajuste en el conteo para la matriz se aplicó en todos los casos donde las imágenes sin presencia de objetos de interés no cuentan con ningún recuadro delimitador.

Es importante resaltar que lo anterior no afecta el rendimiento del modelo, solo la interpretación asociada a los resultados.

7.0.1.1. Entrenamiento 1

Se procede a entrenar el modelo utilizando los parámetros por default haciendo explícitas las siguientes elecciones:

Parámetro	Valor
epoch	350
patience	100
optimizer	'auto'
augment	False

Tabla 7.1: **Parámetros 1.**

Tabla que muestra los parámetros utilizados en el entrenamiento 1.

Esto entrena al modelo con el uso de un optimizador determinado automáticamente por el modelo y sin ningún tipo de aumento de datos, lo que ofrece el entrenamiento más básico posible.

El modelo se entrenó durante 350 épocas, donde el mejor rendimiento se obtuvo en la época 13. Después de las 100 épocas de paciencia establecida, no se presentó una mejora, por lo tanto, se registraron los parámetros del modelo en la época 13 y en la 113.

El desempeño de las métricas durante el entrenamiento se ilustra mediante los siguientes gráficos, donde se observa el comportamiento de la función de costo, en la precisión y sensibilidad durante el entrenamiento.

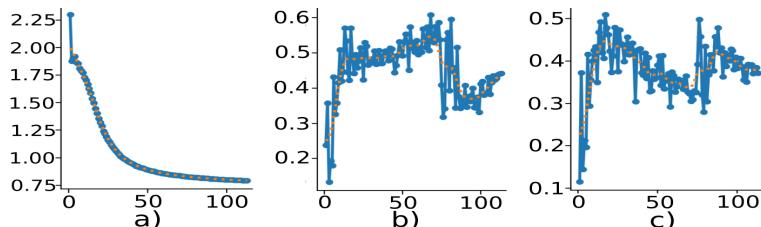


Figura 7.1: **Métricas de entrenamiento 1.**

Gráficas que muestran en el eje x el número de época y en el eje y el valor tomado por: a) La función de costo total. b) Precisión del modelo. c) Sensibilidad del modelo.

Se visualiza el desempeño de las métricas de evaluación donde se muestra el comportamiento de la función de costo y mAP durante el entrenamiento.

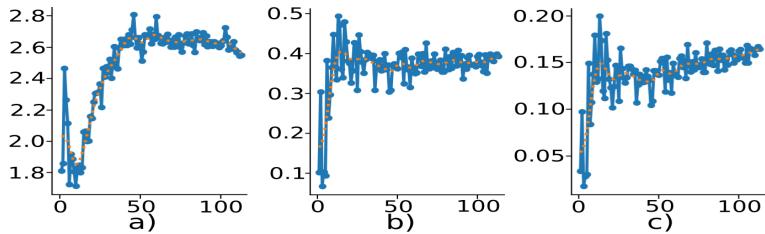


Figura 7.2: **Métricas de evaluación 1.**

Gráficas que muestran en el eje x el número de época y en el eje y el valor tomado por: a) La función de costo total. b) mAP50. c) mAP50-95.

Los valores obtenidos para cada una de las métricas en la época de mejor desempeño, que fue la 13, se resumen en la siguiente tabla:

Métrica	Valor
Precisión	0.57
Sensibilidad	0.467
mAP	0.493

Tabla 7.2: **Métricas 1.**

Tabla que muestra las métricas en la época 13.

Las matrices de confusión obtenidas para el modelo son las siguientes.

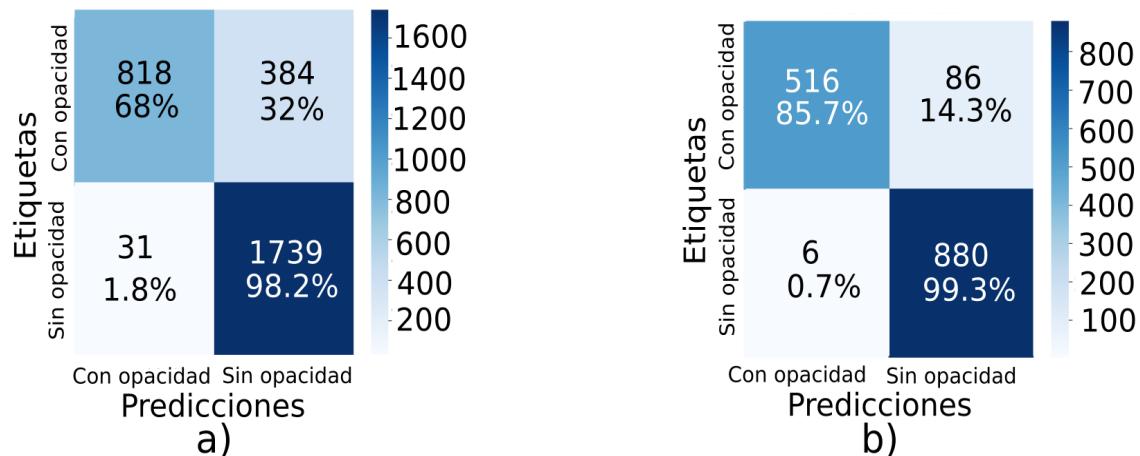


Figura 7.3: **Matrices de confusión 1.**

Matrices de confusión para la clasificación dada por el modelo, presentando tanto los conteos como el porcentaje que representan para: a) El conjunto de validación. y b) El conjunto de prueba.

Se procede a realizar una visualización de las predicciones hechas por el modelo en la época de mejor desempeño.

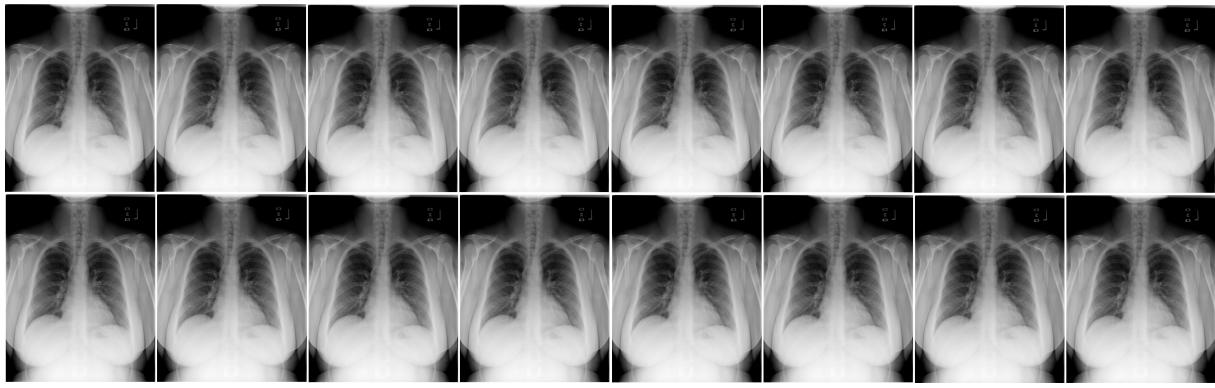


Figura 7.4: Comparación de predicciones 1.

Se presentan distintas imágenes con predicciones hechas por el modelo (rojo) junto con sus respectivas antaciones manuales (verde).

7.0.1.2. Entrenamiento 2

Se realiza un nuevo entrenamiento definiendo los siguientes parámetros.

Parámetro	Valor
epoch	350
patience	100
optimizer	'auto'
augment	True
auto_augment	'randaugment'

Tabla 7.3: Parámetros 2.

Tabla que muestra los parámetros utilizados en el entrenamiento 2.

Esto entrena al modelo con el uso de un aumento de datos aleatorio.

El modelo se entrenó durante 350 épocas, donde el mejor rendimiento se obtuvo en la época 82. Después de las 100 épocas de paciencia establecida, no se presentó una mejora, por lo tanto, se registraron los parámetros del modelo en la época 82 y en la 182.

El desempeño de las métricas durante el entrenamiento se ilustra mediante los siguientes gráficos, donde se observa el comportamiento de la función de costo, la precisión y sensibilidad durante el entrenamiento.

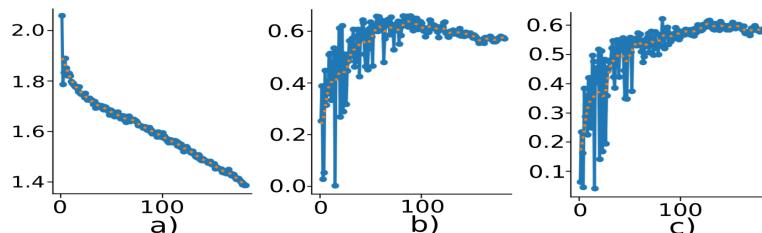


Figura 7.5: Métricas de entrenamiento 2.

Gráficas que muestran en el eje x el número de época y en el eje y el valor tomado por: a) La función de costo total. b) Precisión del modelo. c) Sensibilidad del modelo.

Se visualiza el desempeño de las métricas de evaluación donde se muestra el comportamiento de la función de costo y mAP durante el entrenamiento.

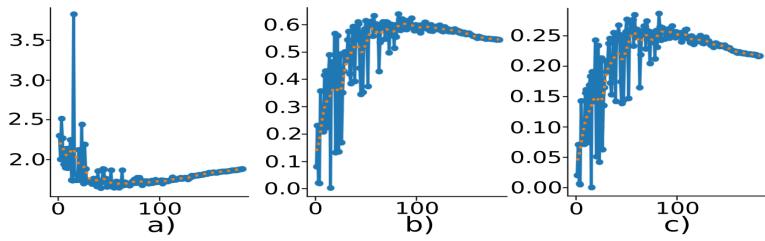


Figura 7.6: **Métricas de evaluación 2.**

Gráficas que muestran en el eje x el número de época y en el eje y el valor tomado por: a) La función de costo total. b) mAP50. c) mAP50-95.

Los valores obtenidos para cada una de las métricas en la época de mejor desempeño, que fue la 82, se resumen en la siguiente tabla:

Métrica	Valor
Precisión	0.644
Sensibilidad	0.606
mAP	0.627

Tabla 7.4: **Métricas 2.**

Tabla que muestra las métricas en la época 82.

Las matrices de confusión obtenidas para el modelo son las siguientes.

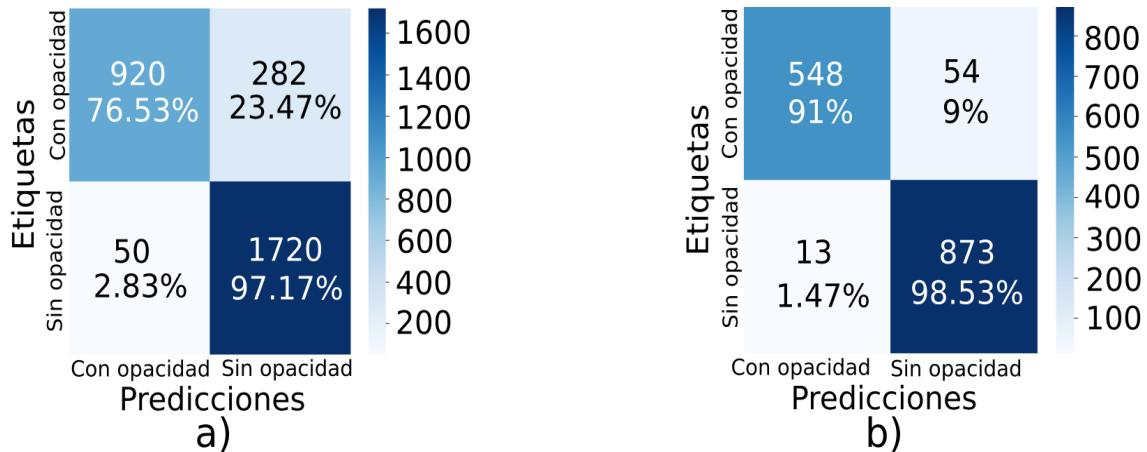


Figura 7.7: **Matrices de confusión 2.**

Matrices de confusión para la clasificación dada por el modelo, presentando tanto los conteos como el porcentaje que representan para: a) El conjunto de validación. y b) El conjunto de prueba.

Se procede a realizar una visualización de las predicciones hechas por el modelo en la época de mejor desempeño.

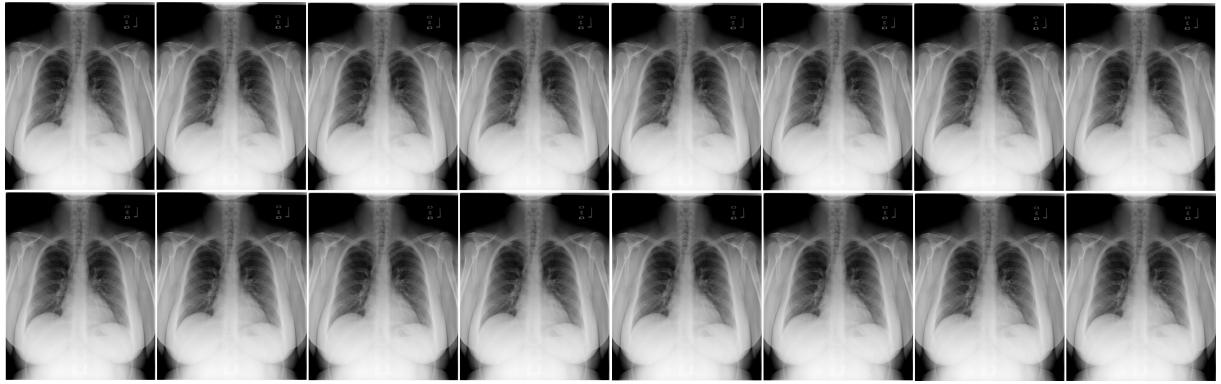


Figura 7.8: Comparación de predicciones 2.

Se presentan distintas imágenes con predicciones hechas por el modelo (rojo) junto con sus respectivas antaciones manuales (verde).

7.0.1.3. Entrenamiento 3

Se realiza un nuevo entrenamiento definiendo los siguientes parámetros.

Parámetro	Valor
epoch	350
patience	100
optimizer	'auto'
augment	True
auto_augment	'autoaugment'

Tabla 7.5: Parámetros 3.

Tabla que muestra los parámetros utilizados en el entrenamiento 3.

Esto entrena al modelo con el uso de un aumento de datos determinado automáticamente por el modelo.

El modelo se entrenó durante 350 épocas, donde el mejor rendimiento se obtuvo en la época 82. Después de las 100 épocas de paciencia establecida, no se presentó una mejora, por lo tanto, se registraron los parámetros del modelo en la época 82 y en la 182.

El desempeño de las métricas durante el entrenamiento se ilustra mediante los siguientes gráficos, donde se observa el comportamiento de la función de costo, la precisión y sensibilidad durante el entrenamiento.

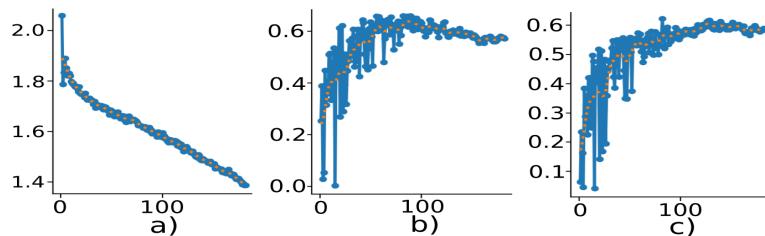


Figura 7.9: Métricas de entrenamiento 3.

Gráficas que muestran en el eje x el número de época y en el eje y el valor tomado por: a) La función de costo total. b) Precisión del modelo. c) Sensibilidad del modelo.

Se visualiza el desempeño de las métricas de evaluación donde se muestra el comportamiento de la función de costo y mAP durante el entrenamiento.

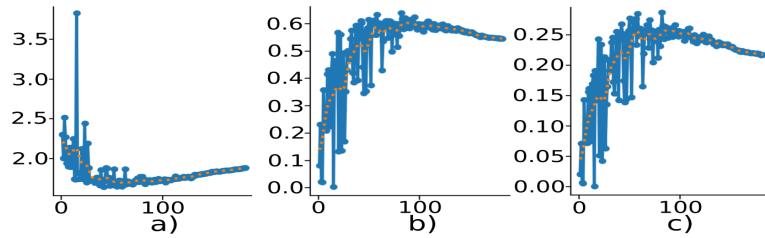


Figura 7.10: **Métricas de evaluación 3.**

Gráficas que muestran en el eje x el número de época y en el eje y el valor tomado por: a) La función de costo total. b) mAP50. c) mAP50-95.

Los valores obtenidos para cada una de las métricas en la época de mejor desempeño, que fue la 82, se resumen en la siguiente tabla:

Métrica	Valor
Precisión	0.644
Sensibilidad	0.606
mAP	0.627

Tabla 7.6: **Métricas 3.**

Tabla que muestra las métricas en la época 82.

Las matrices de confusión obtenidas para el modelo son las siguientes.

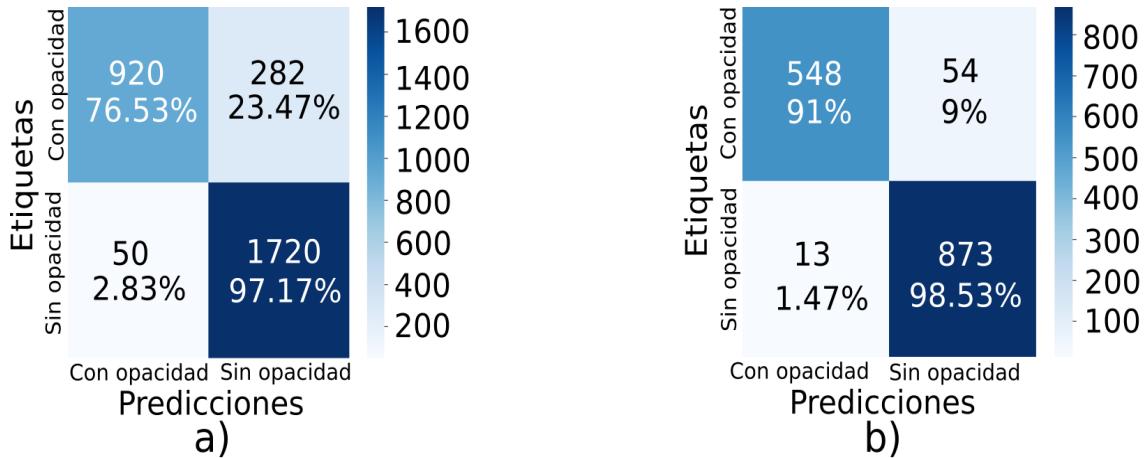


Figura 7.11: **Matrices de confusión 3.**

Matrices de confusión para la clasificación dada por el modelo, presentando tanto los conteos como el porcentaje que representan para: a) El conjunto de validación. y b) El conjunto de prueba.

Se procede a realizar una visualización de las predicciones hechas por el modelo en la época de mejor desempeño.

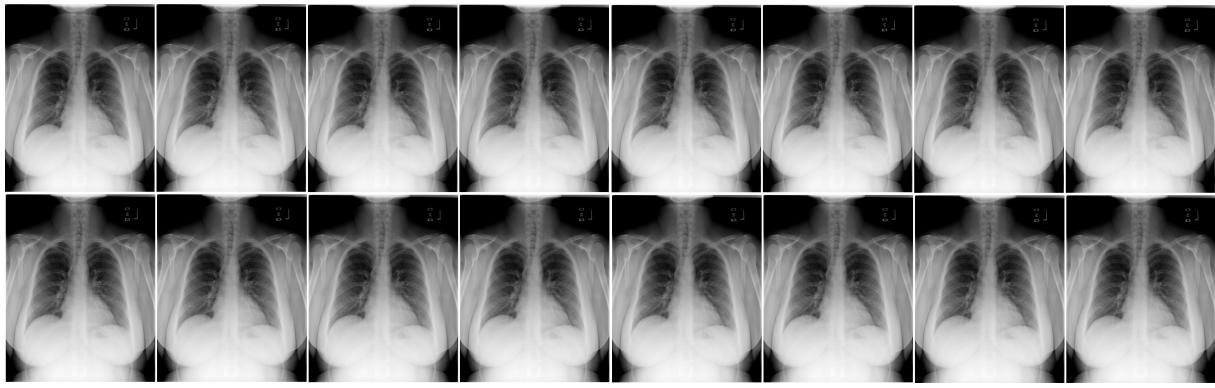


Figura 7.12: Comparación de predicciones 3.

Se presentan distintas imágenes con predicciones hechas por el modelo (rojo) junto con sus respectivas antaciones manuales (verde).

7.0.1.4. Entrenamiento 4

Se realiza un nuevo entrenamiento definiendo los siguientes parámetros.

Parámetro	Valor
epoch	350
patience	100
optimizer	'auto'
augment	True
auto_augment	'augmix'

Tabla 7.7: Parámetros 4.

Tabla que muestra los parámetros utilizados en el entrenamiento 4.

Esto entrena al modelo con el uso de un aumento de datos que combina dos o más imágenes para crear una nueva imagen de entrenamiento.

El modelo se entrenó durante 350 épocas, donde el mejor rendimiento se obtuvo en la época 82. Después de las 100 épocas de paciencia establecida, no se presentó una mejora, por lo tanto, se registraron los parámetros del modelo en la época 82 y en la 182.

El desempeño de las métricas durante el entrenamiento se ilustra mediante los siguientes gráficos, donde se observa el comportamiento de la función de costo, la precisión y sensibilidad durante el entrenamiento.

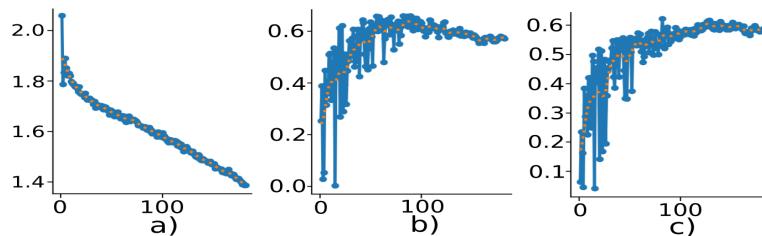


Figura 7.13: Métricas de entrenamiento 4.

Gráficas que muestran en el eje x el número de época y en el eje y el valor tomado por: a) La función de costo total. b) Precisión del modelo. c) Sensibilidad del modelo.

Se visualiza el desempeño de las métricas de evaluación donde se muestra el comportamiento de la función de costo y mAP durante el entrenamiento.

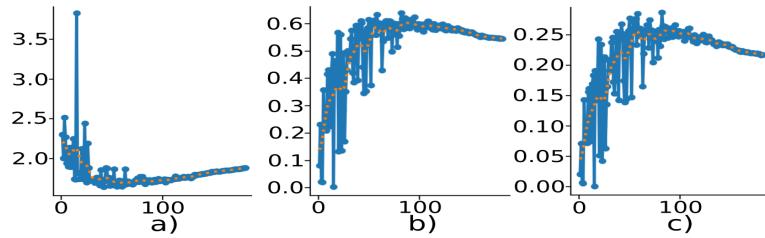


Figura 7.14: Métricas de evaluación 4.

Gráficas que muestran en el eje x el número de época y en el eje y el valor tomado por: a) La función de costo total. b) mAP50. c) mAP50-95.

Los valores obtenidos para cada una de las métricas en la época de mejor desempeño, que fue la 82, se resumen en la siguiente tabla:

Métrica	Valor
Precisión	0.644
Sensibilidad	0.606
mAP	0.627

Tabla 7.8: Métricas 4.

Tabla que muestra las métricas en la época 82.

Las matrices de confusión obtenidas para el modelo son las siguientes.

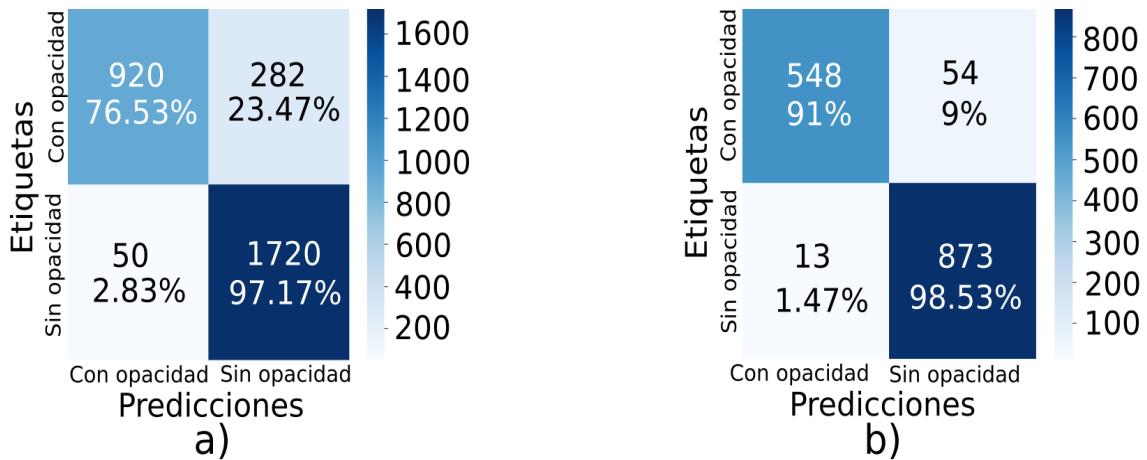


Figura 7.15: Matrices de confusión 4.

Matrices de confusión para la clasificación dada por el modelo, presentando tanto los conteos como el porcentaje que representan para: a) El conjunto de validación. y b) El conjunto de prueba.

Se procede a realizar una visualización de las predicciones hechas por el modelo en la época de mejor desempeño.

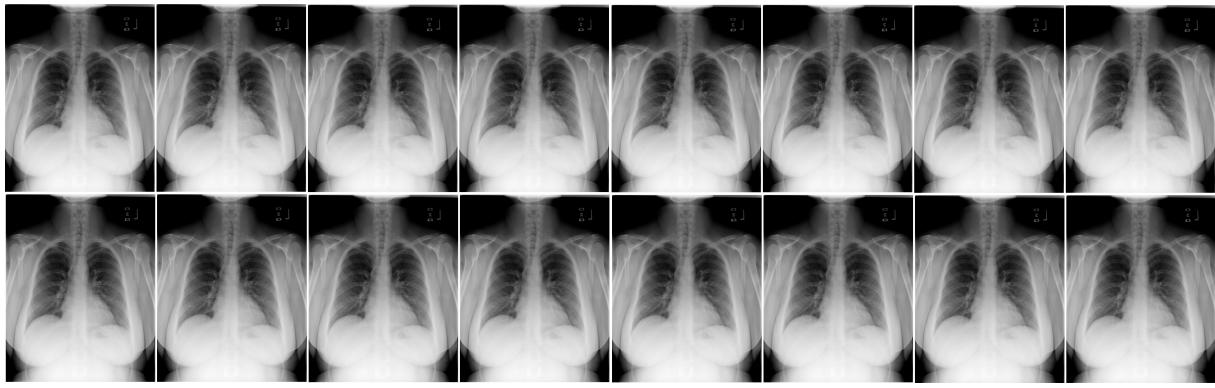


Figura 7.16: Comparación de predicciones 4.

Se presentan distintas imágenes con predicciones hechas por el modelo (rojo) junto con sus respectivas antaciones manuales (verde).

7.0.1.5. Entrenamiento 5

Se realiza un nuevo entrenamiento definiendo los siguientes parámetros.

Parámetro	Valor
epoch	350
patience	100
optimizer	'auto'
augment	True
auto_augment	None
hsv_h	0.5
hsv_s	0.5
hsv_v	0.5
degrees	35.5
translate	0.3
scale	0.6
shear	10.5
perspective	0.0
flipud	0.0
fliplr	0.0
mosaic	1.0
mixup	0.5
copy_paste	0.5
erasing	0.0
crop_fraction	0.5

Tabla 7.9: Parámetros 5.

Tabla que muestra los parámetros utilizados en el entrenamiento 5.

Esto entrena al modelo con el uso de un aumento de datos definido completamente a nuestra elección con los valores presentados en la tabla.

El modelo se entrenó durante 350 épocas, donde el mejor rendimiento se obtuvo en la época 138. Después de las 100 épocas de paciencia establecida, no se presentó una mejora, por lo tanto, se registraron los parámetros del modelo en la época 138 y en la 238.

El desempeño de las métricas durante el entrenamiento se ilustra mediante los siguientes gráficos, donde se observa el comportamiento de la función de costo, la precisión y sensibilidad durante el entrenamiento.

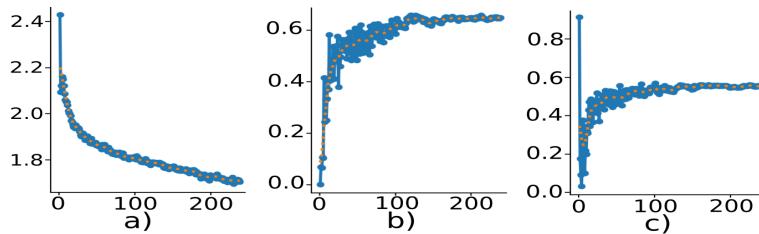


Figura 7.17: Métricas de entrenamiento 5.

Gráficas que muestran en el eje x el número de época y en el eje y el valor tomado por: a) La función de costo total. b) Precisión del modelo. c) Sensibilidad del modelo.

Se visualiza el desempeño de las métricas de evaluación donde se muestra el comportamiento de la función de costo y mAP durante el entrenamiento.

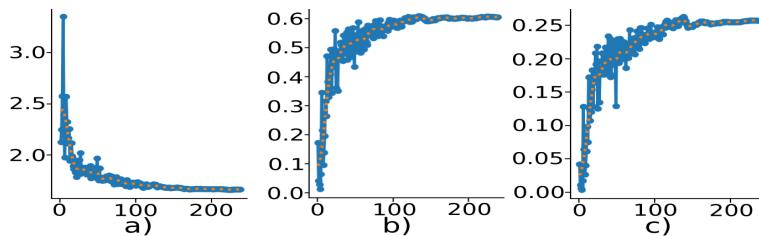


Figura 7.18: Métricas de evaluación 5.

Gráficas que muestran en el eje x el número de época y en el eje y el valor tomado por: a) La función de costo total. b) mAP50. c) mAP50-95.

Los valores obtenidos para cada una de las métricas en la época de mejor desempeño, que fue la 138, se resumen en la siguiente tabla:

Métrica	Valor
Precisión	0.565
Sensibilidad	0.553
mAP	0.568

Tabla 7.10: Métricas 5.

Tabla que muestra las métricas en la época 138.

Las matrices de confusión obtenidas para el modelo son las siguientes.

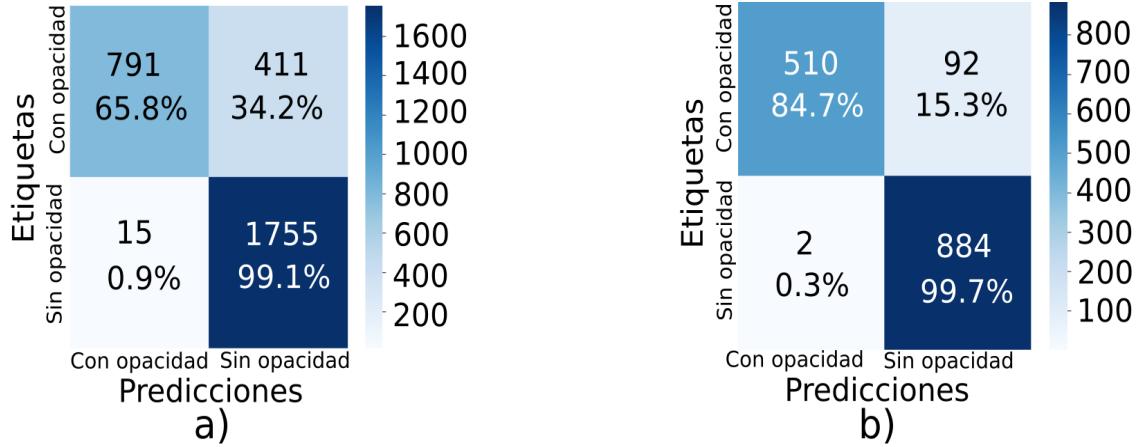


Figura 7.19: **Matrices de confusión 5.**

Matrices de confusión para la clasificación dada por el modelo, presentando tanto los conteos como el porcentaje que representan para: a) El conjunto de validación. y b) El conjunto de prueba.

Se procede a realizar una visualización de las predicciones hechas por el modelo en la época de mejor desempeño.

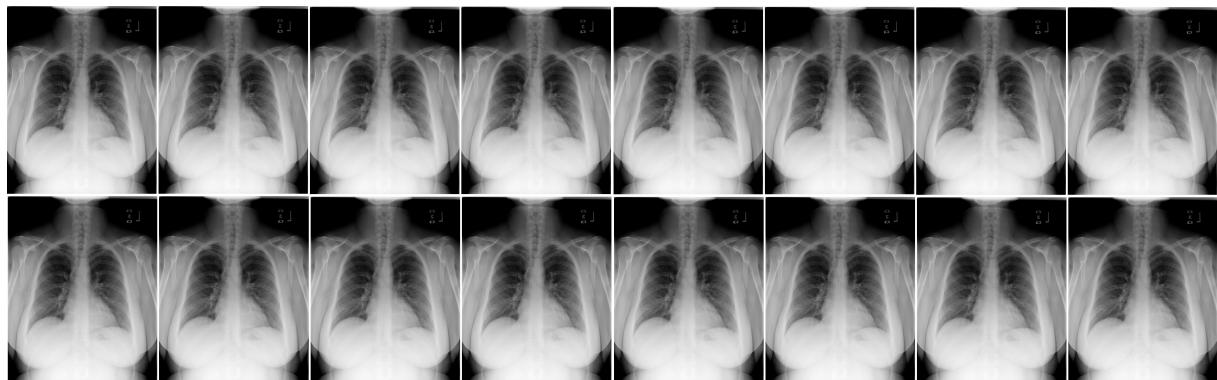


Figura 7.20: **Comparación de predicciones 5.**

Se presentan distintas imágenes con predicciones hechas por el modelo (rojo) junto con sus respectivas antaciones manuales (verde).

7.0.1.6. Entrenamiento 6

Se realiza un nuevo entrenamiento usando los parámetros utilizados en el entrenamiento 3, pero seleccionando el optimizador.

Parámetro	Valor
optimizer	'Adam'

Tabla 7.11: **Parámetros 6.**

Tabla que muestra los parámetros utilizados en el entrenamiento 6.

Dado que los netrenamientos 2,3 y 4 mostraron rendimientos identicos con el optimizador SGD seleccionado automaticamente se procede a medir el desempeño del entrenamiento 3 ahora con el optimizador Adam.

El modelo se entrenó durante 350 épocas, donde el mejor rendimiento se obtuvo en la época 131. Después de las 100 épocas de paciencia establecida, no se presentó una mejora, por lo tanto, se registraron los parámetros del modelo en la época 131 y en la 231.

El desempeño de las métricas durante el entrenamiento se ilustra mediante los siguientes gráficos, donde se observa el comportamiento de la función de costo, la precisión y sensibilidad durante el entrenamiento.

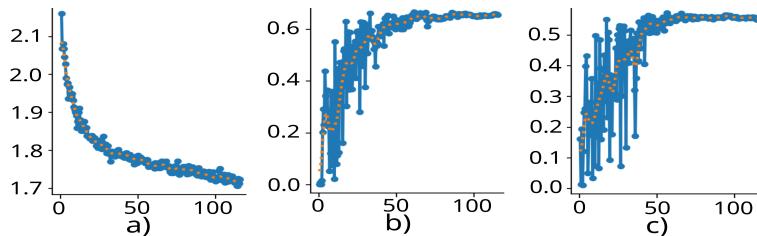


Figura 7.21: Métricas de entrenamiento 6.

Gráficas que muestran en el eje x el número de época y en el eje y el valor tomado por: a) La función de costo total. b) Precisión del modelo. c) Sensibilidad del modelo.

Se visualiza el desempeño de las métricas de evaluación donde se muestra el comportamiento de la función de costo y mAP durante el entrenamiento.

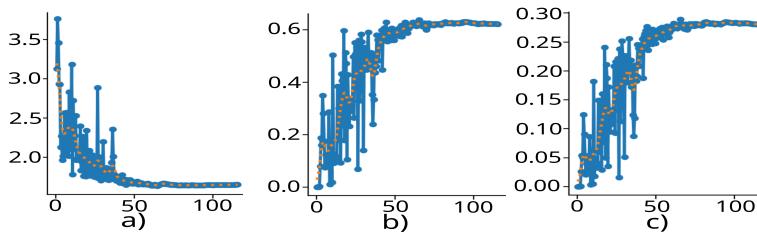


Figura 7.22: Métricas de evaluación 6.

Gráficas que muestran en el eje x el número de época y en el eje y el valor tomado por: a) La función de costo total. b) mAP50. c) mAP50-95.

Los valores obtenidos para cada una de las métricas en la época de mejor desempeño, que fue la 131, se resumen en la siguiente tabla:

Métrica	Valor
Precisión	0.635
Sensibilidad	0.575
mAP	0.623

Tabla 7.12: Métricas 6.

Tabla que muestra las métricas en la época 131.

Las matrices de confusión obtenidas para el modelo son las siguientes.

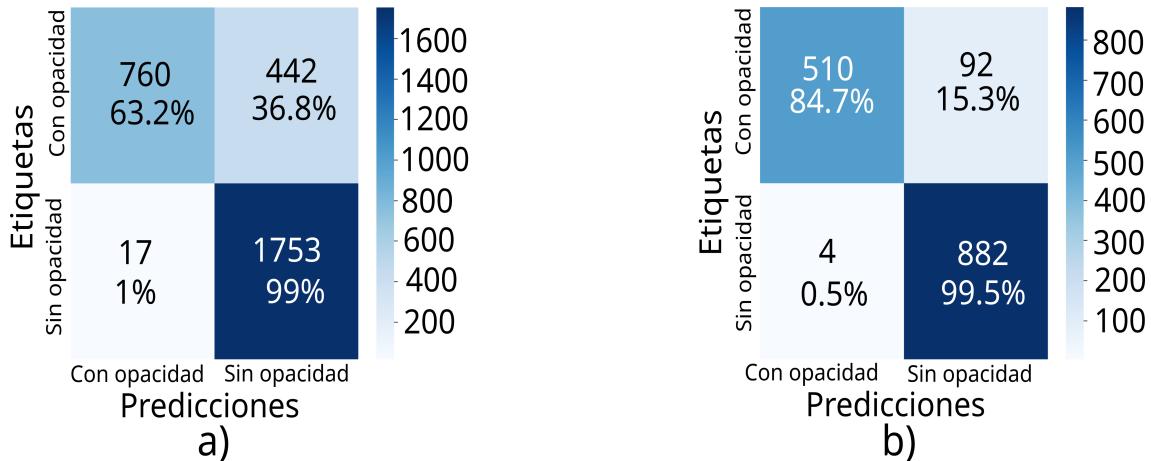


Figura 7.23: **Matrices de confusión 6.**

Matrices de confusión para la clasificación dada por el modelo, presentando tanto los conteos como el porcentaje que representan para: a) El conjunto de validación. y b) El conjunto de prueba.

Se procede a realizar una visualización de las predicciones hechas por el modelo en la época de mejor desempeño.

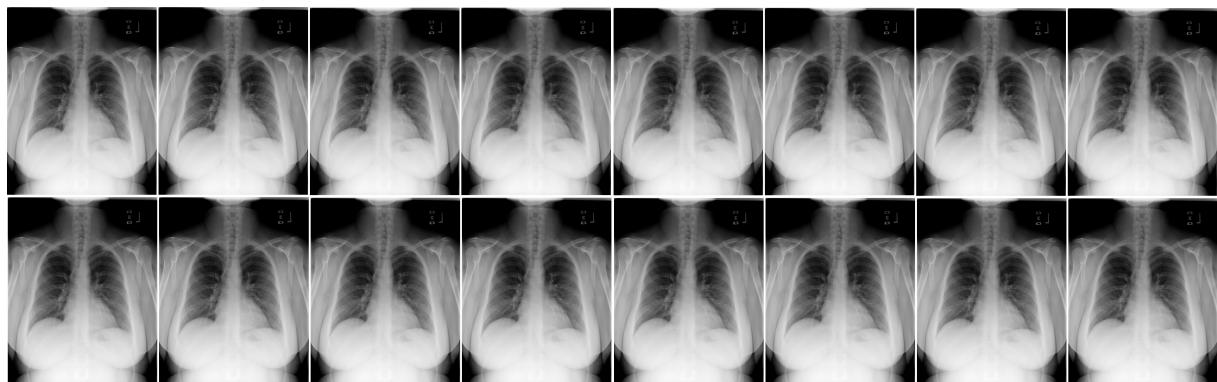


Figura 7.24: **Comparación de predicciones 6.**

Se presentan distintas imágenes con predicciones hechas por el modelo (rojo) junto con sus respectivas antaciones manuales (verde).

7.0.1.7. Entrenamiento 7

Se realiza un nuevo entrenamiento usando los parámetros utilizados en el entrenamiento 3, pero seleccionando el optimizador.

Parámetro	Valor
optimizer	'Adamax'

Tabla 7.13: **Parámetros 7.**

Tabla que muestra los parámetros utilizados en el entrenamiento 7.

Dado que los netrenamientos 2,3 y 4 mostraron rendimientos identicos con el optimizador SGD seleccionado automaticamente se procede a medir el desempeño del entrenamiento 3 ahora con el optimizador Adamax.

El modelo se entrenó durante 350 épocas, donde el mejor rendimiento se obtuvo en la época 122. Después de las 100 épocas de paciencia establecida, no se presentó una mejora, por lo tanto, se registraron los parámetros del modelo en la época 122 y en la 222.

El desempeño de las métricas durante el entrenamiento se ilustra mediante los siguientes gráficos, donde se observa el comportamiento de la función de costo, la precisión y sensibilidad durante el entrenamiento.

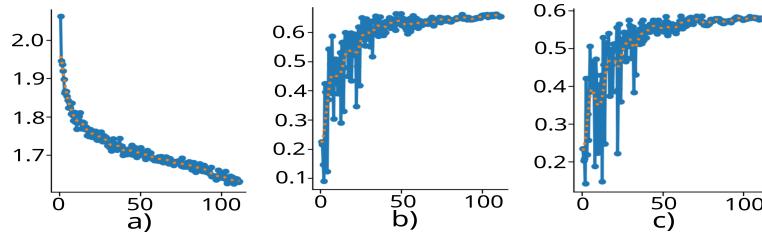


Figura 7.25: Métricas de entrenamiento 7.

Gráficas que muestran en el eje x el número de época y en el eje y el valor tomado por: a) La función de costo total. b) Precisión del modelo. c) Sensibilidad del modelo.

Se visualiza el desempeño de las métricas de evaluación donde se muestra el comportamiento de la función de costo y mAP durante el entrenamiento.

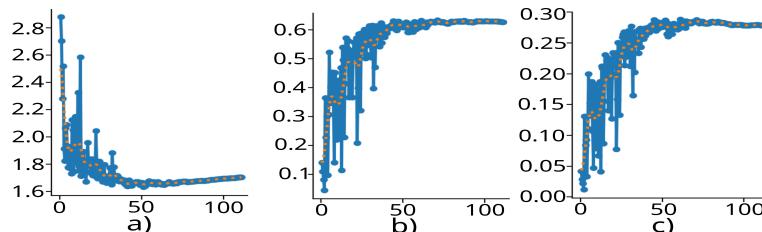


Figura 7.26: Métricas de evaluación 7.

Gráficas que muestran en el eje x el número de época y en el eje y el valor tomado por: a) La función de costo total. b) mAP50. c) mAP50-95.

Los valores obtenidos para cada una de las métricas en la época de mejor desempeño, que fue la 122, se resumen en la siguiente tabla:

Métrica	Valor
Precisión	0.643
Sensibilidad	0.559
mAP	0.62

Tabla 7.14: Métricas 7.

Tabla que muestra las métricas en la época 122.

Las matrices de confusión obtenidas para el modelo son las siguientes.

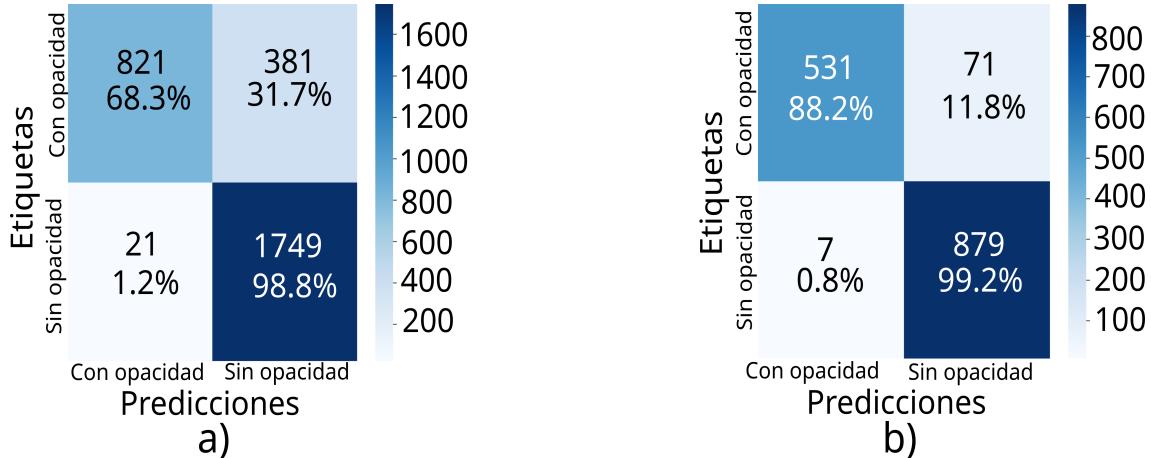


Figura 7.27: **Matrices de confusión 7.**

Matrices de confusión para la clasificación dada por el modelo, presentando tanto los conteos como el porcentaje que representan para: a) El conjunto de validación. y b) El conjunto de prueba.

Se procede a realizar una visualización de las predicciones hechas por el modelo en la época de mejor desempeño.

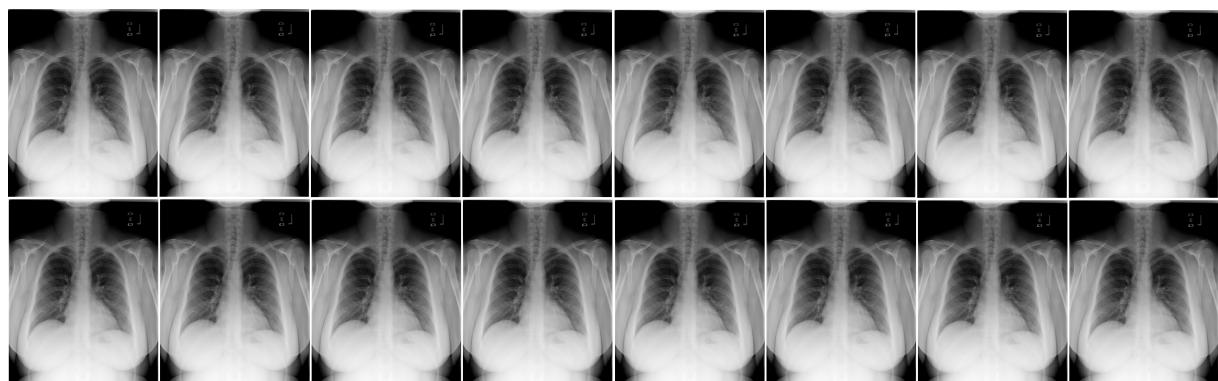


Figura 7.28: **Comparación de predicciones 7.**

Se presentan distintas imágenes con predicciones hechas por el modelo (rojo) junto con sus respectivas antaciones manuales (verde).

7.0.1.8. Entrenamiento 8

Se realiza un nuevo entrenamiento usando los parámetros utilizados en el entrenamiento 3, pero seleccionando el optimizador.

Parámetro	Valor
optimizer	'AdamW'

Tabla 7.15: **Parámetros 8.**

Tabla que muestra los parámetros utilizados en el entrenamiento 8.

Dado que los netrenamientos 2,3 y 4 mostraron rendimientos identicos con el optimizador SGD seleccionado automaticamente se procede a medir el desempeño del entrenamiento 3 ahora con el optimizador AdamW.

El modelo se entrenó durante 350 épocas, donde el mejor rendimiento se obtuvo en la época 58. Después de las 100 épocas de paciencia establecida, no se presentó una mejora, por lo tanto, se registraron los parámetros del modelo en la época 58 y en la 158.

El desempeño de las métricas durante el entrenamiento se ilustra mediante los siguientes gráficos, donde se observa el comportamiento de la función de costo, la precisión y sensibilidad durante el entrenamiento.

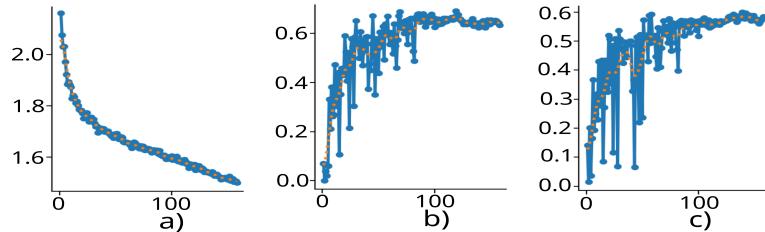


Figura 7.29: Métricas de entrenamiento 8.

Gráficas que muestran en el eje x el número de época y en el eje y el valor tomado por: a) La función de costo total. b) Precisión del modelo. c) Sensibilidad del modelo.

Se visualiza el desempeño de las métricas de evaluación donde se muestra el comportamiento de la función costo y mAP durante el entrenamiento.

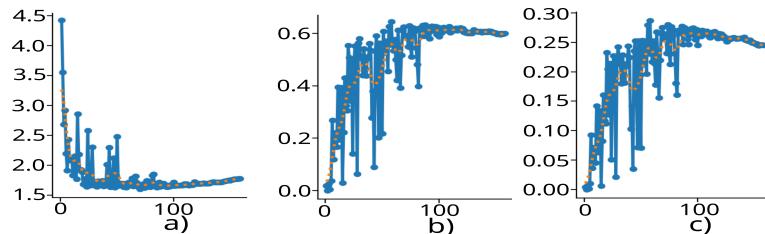


Figura 7.30: Métricas de evaluación 8.

Gráficas que muestran en el eje x el número de época y en el eje y el valor tomado por: a) La función de costo total. b) mAP50. c) mAP50-95.

Los valores obtenidos para cada una de las métricas en la época de mejor desempeño, que fue la xx, se resumen en la siguiente tabla:

Métrica	Valor
Precisión	0.647
Sensibilidad	0.583
mAP	0.63

Tabla 7.16: Métricas 8.

Tabla que muestra las métricas en la época 58.

Las matrices de confusión obtenidas para el modelo son las siguientes.

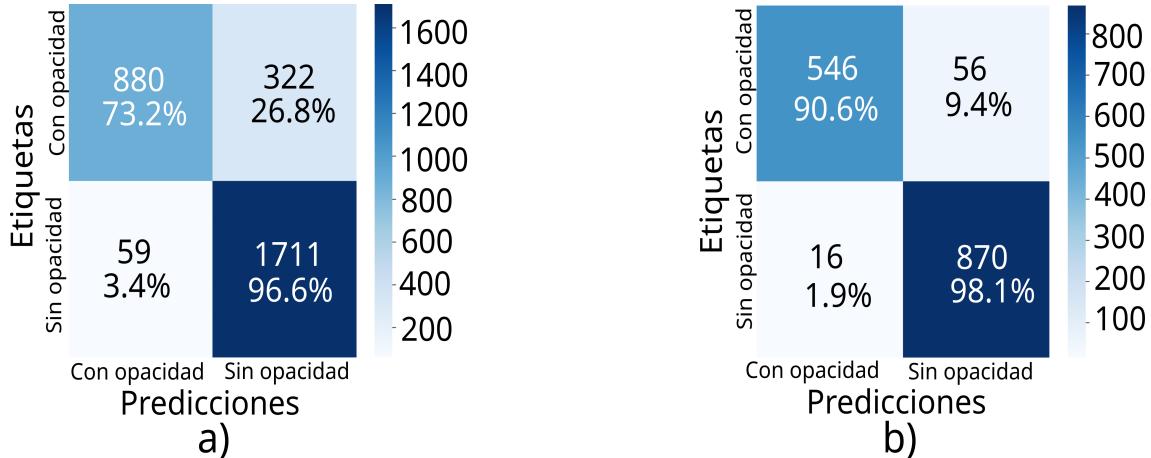


Figura 7.31: **Matrices de confusión 8.**

Matrices de confusión para la clasificación dada por el modelo, presentando tanto los conteos como el porcentaje que representan para: a) El conjunto de validación. y b) El conjunto de prueba.

Se procede a realizar una visualización de las predicciones hechas por el modelo en la época de mejor desempeño.

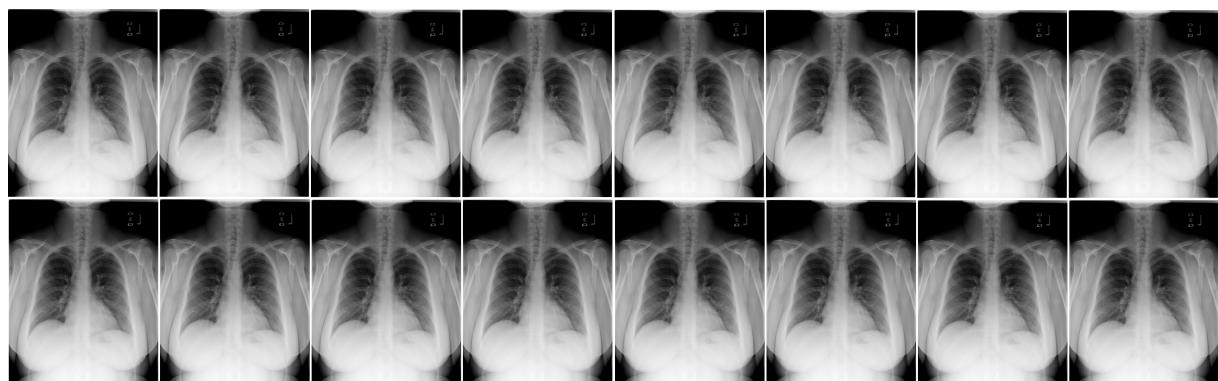


Figura 7.32: **Comparación de predicciones 8.**

Se presentan distintas imágenes con predicciones hechas por el modelo (rojo) junto con sus respectivas antaciones manuales (verde).

7.0.1.9. Entrenamiento 9

Se realiza un nuevo entrenamiento usando los parámetros utilizados en el entrenamiento 3, pero seleccionando el optimizador.

Parámetro	Valor
optimizer	'NAdam'

Tabla 7.17: **Parámetros 9.**

Tabla que muestra los parámetros utilizados en el entrenamiento 9.

Dado que los netrenamientos 2,3 y 4 mostraron rendimientos identicos con el optimizador SGD seleccionado automaticamente se procede a medir el desempeño del entrenamiento 3 ahora con el optimizador NAdam.

El modelo se entrenó durante 350 épocas, donde el mejor rendimiento se obtuvo en la época 103. Después de las 100 épocas de paciencia establecida, no se presentó una mejora, por lo tanto, se registraron los parámetros del modelo en la época 103 y en la 203.

El desempeño de las métricas durante el entrenamiento se ilustra mediante los siguientes gráficos, donde se observa el comportamiento de la función de costo, la precisión y sensibilidad durante el entrenamiento.

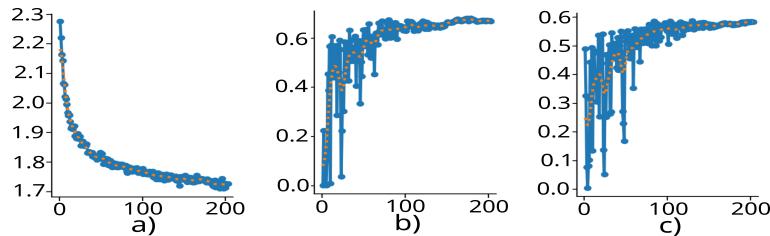


Figura 7.33: Métricas de entrenamiento 9.

Gráficas que muestran en el eje x el número de época y en el eje y el valor tomado por: a) La función de costo total. b) Precisión del modelo. c) Sensibilidad del modelo.

Se visualiza el desempeño de las métricas de evaluación donde se muestra el comportamiento de la función de costo y mAP durante el entrenamiento.

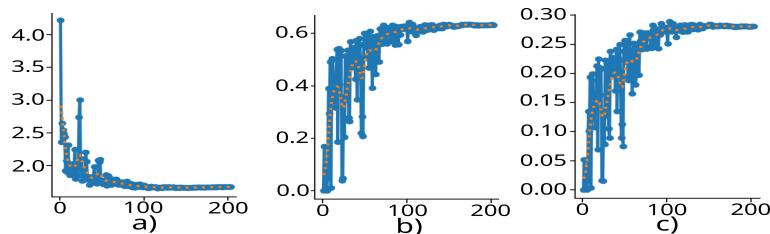


Figura 7.34: Métricas de evaluación 9.

Gráficas que muestran en el eje x el número de época y en el eje y el valor tomado por: a) La función de costo total. b) mAP50. c) mAP50-95.

Los valores obtenidos para cada una de las métricas en la época de mejor desempeño, que fue la 103, se resumen en la siguiente tabla:

Métrica	Valor
Precisión	0.636
Sensibilidad	0.575
mAP	0.619

Tabla 7.18: Métricas 9.

Tabla que muestra las métricas en la época 103.

Las matrices de confusión obtenidas para el modelo son las siguientes.

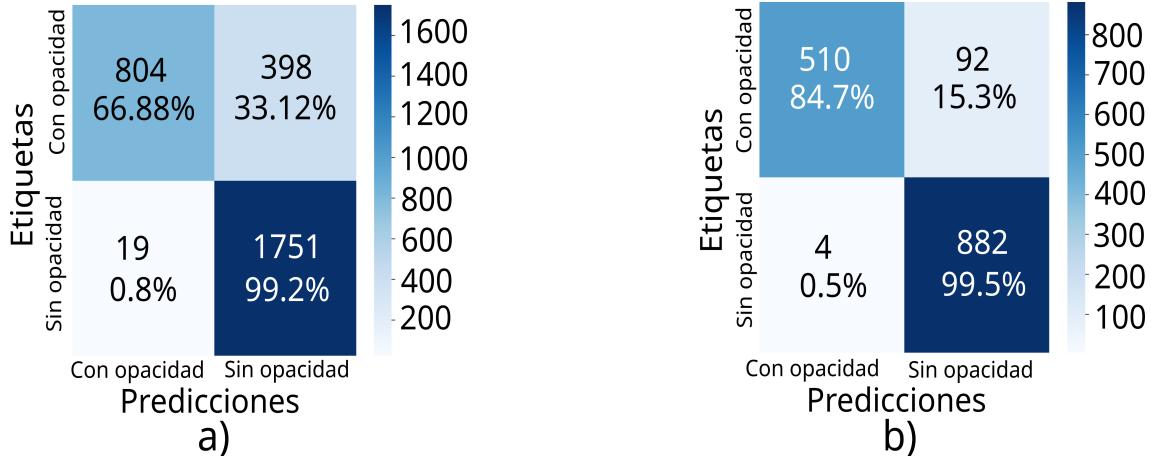


Figura 7.35: **Matrices de confusión 9.**

Matrices de confusión para la clasificación dada por el modelo, presentando tanto los conteos como el porcentaje que representan para: a) El conjunto de validación. y b) El conjunto de prueba.

Se procede a realizar una visualización de las predicciones hechas por el modelo en la época de mejor desempeño.

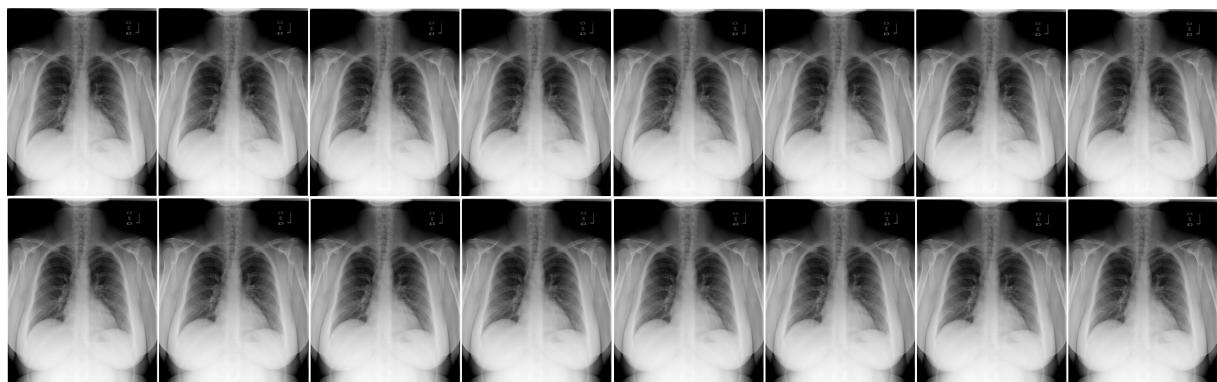


Figura 7.36: **Comparación de predicciones 9.**

Se presentan distintas imágenes con predicciones hechas por el modelo (rojo) junto con sus respectivas antaciones manuales (verde).

7.0.1.10. Entrenamiento 10

Se realiza un nuevo entrenamiento usando los parámetros utilizados en el entrenamiento x, pero seleccionando el optimizador.

Parámetro	Valor
optimizer	'RAdam'

Tabla 7.19: **Parámetros 10.**

Tabla que muestra los parámetros utilizados en el entrenamiento 10.

Dado que los netrenamientos 2,3 y 4 mostraron rendimientos identicos con el optimizador SGD seleccionado automaticamente se procede a medir el desempeño del entrenamiento 3 ahora con el optimizador RAdam.

El modelo se entrenó durante 350 épocas, donde el mejor rendimiento se obtuvo en la época 106. Después de las 100 épocas de paciencia establecida, no se presentó una mejora, por lo tanto, se registraron los parámetros del modelo en la época 106 y en la 206.

El desempeño de las métricas durante el entrenamiento se ilustra mediante los siguientes gráficos, donde se observa el comportamiento de la función de costo, la precisión y sensibilidad durante el entrenamiento.

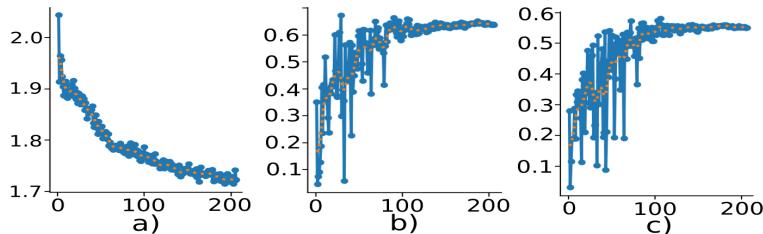


Figura 7.37: Métricas de entrenamiento 10.

Gráficas que muestran en el eje x el número de época y en el eje y el valor tomado por: a) La función de costo total. b) Precisión del modelo. c) Sensibilidad del modelo.

Se visualiza el desempeño de las métricas de evaluación donde se muestra el comportamiento de la función de costo y mAP durante el entrenamiento.

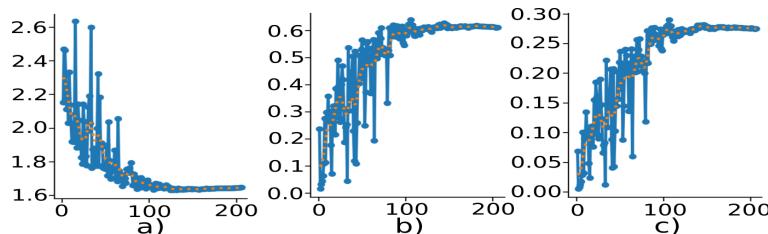


Figura 7.38: Métricas de evaluación 10.

Gráficas que muestran en el eje x el número de época y en el eje y el valor tomado por: a) La función de costo total. b) mAP50. c) mAP50-95.

Los valores obtenidos para cada una de las métricas en la época de mejor desempeño, que fue la 106, se resumen en la siguiente tabla:

Métrica	Valor
Precisión	0.635
Sensibilidad	0.576
mAP	0.625

Tabla 7.20: Métricas 10.

Tabla que muestra las métricas en la época 106.

Las matrices de confusión obtenidas para el modelo son las siguientes.

Se procede a realizar una visualización de las predicciones hechas por el modelo en la época de mejor desempeño.

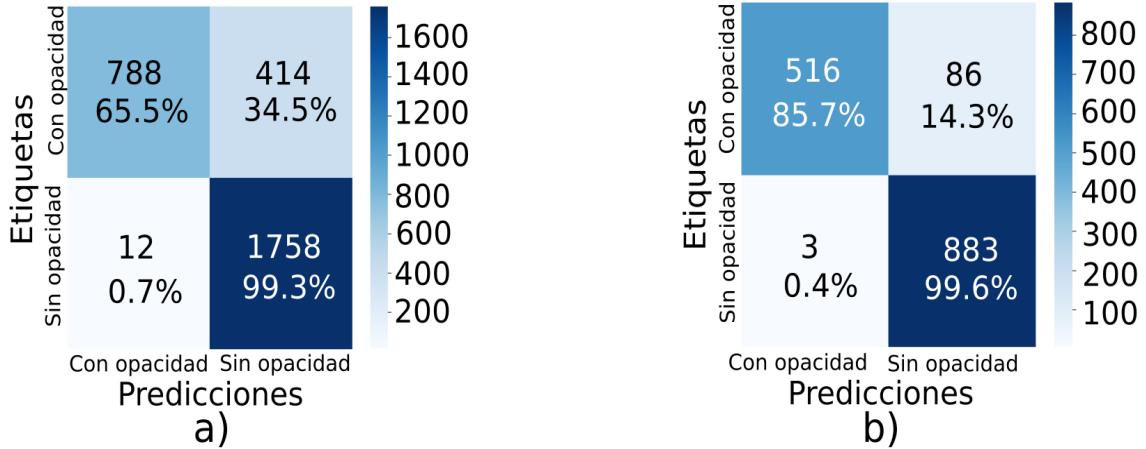


Figura 7.39: Matrices de confusión 10.

Matrices de confusión para la clasificación dada por el modelo, presentando tanto los conteos como el porcentaje que representan para: a) El conjunto de validación. y b) El conjunto de prueba.

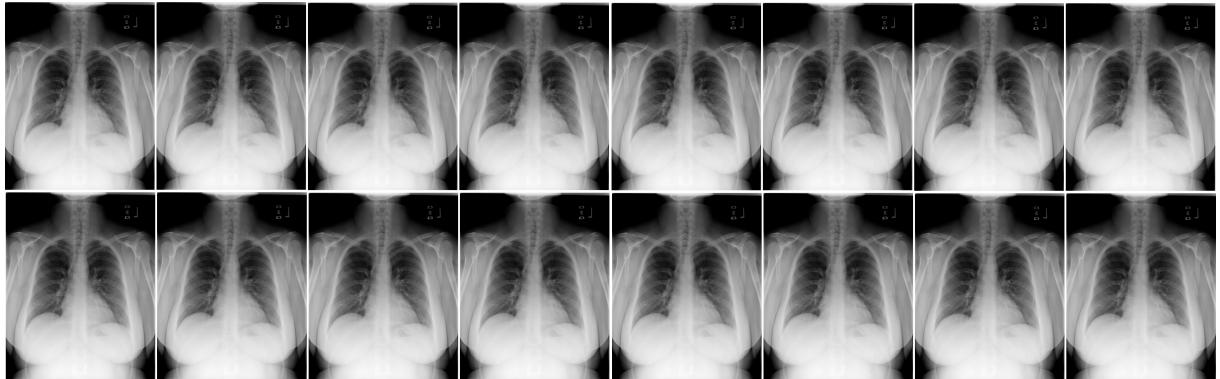


Figura 7.40: Comparación de predicciones 10.

Se presentan distintas imágenes con predicciones hechas por el modelo (rojo) junto con sus respectivas antaciones manuales (verde).

7.0.1.11. Entrenamiento 11

Se realiza un nuevo entrenamiento usando los parámetros utilizados en el entrenamiento x, pero seleccionando el optimizador.

Parámetro	Valor
optimizer	'RMSProp'

Tabla 7.21: Parámetros 11.

Tabla que muestra los parámetros utilizados en el entrenamiento 11.

Dado que los netrenamientos 2,3 y 4 mostraron rendimientos identicos con el optimizador SGD seleccionado automaticamente se procede a medir el desempeño del entrenamiento 3 ahora con el optimizador RMSProp.

El modelo se entrenó durante 350 épocas, donde el mejor rendimiento se obtuvo en la época 29. Después de las 100 épocas de paciencia establecida, no se presentó una mejora, por lo tanto, se registraron los parámetros del modelo en la época 29 y en la 129.

El desempeño de las métricas durante el entrenamiento se ilustra mediante los siguientes gráficos, donde se observa el comportamiento de la función de costo, la precisión y sensibilidad durante el entrenamiento.

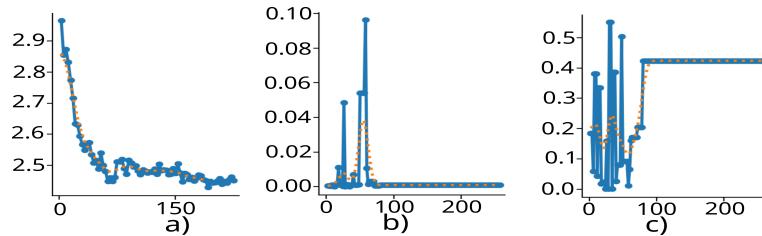


Figura 7.41: **Métricas de entrenamiento 11.**

Gráficas que muestran en el eje x el número de época y en el eje y el valor tomado por: a) La función de costo total. b) Precisión del modelo. c) Sensibilidad del modelo.

Se visualiza el desempeño de las métricas de evaluación donde se muestra el comportamiento de la función de costo y mAP durante el entrenamiento.

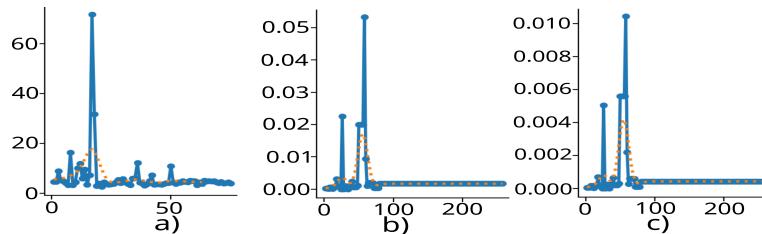


Figura 7.42: **Métricas de evaluación 11.**

Gráficas que muestran en el eje x el número de época y en el eje y el valor tomado por: a) La función de costo total. b) mAP50. c) mAP50-95.

Los valores obtenidos para cada una de las métricas en la época de mejor desempeño, que fue la 29, se resumen en la siguiente tabla:

Métrica	Valor
Precisión	0.0833
Sensibilidad	0.0329
mAP	0.0463

Tabla 7.22: **Métricas 11.**

Tabla que muestra las métricas en la época 29.

Las matrices de confusión obtenidas para el modelo son las siguientes.

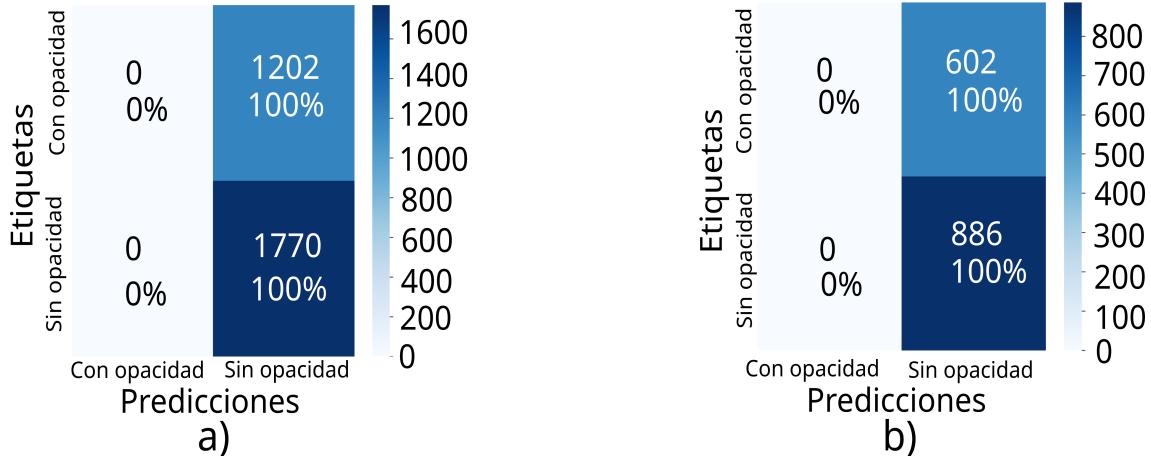


Figura 7.43: **Matrices de confusión 11.**

Matrices de confusión para la clasificación dada por el modelo, presentando tanto los conteos como el porcentaje que representan para: a) El conjunto de validación. y b) El conjunto de prueba.

Se procede a realizar una visualización de las predicciones hechas por el modelo en la época de mejor desempeño.

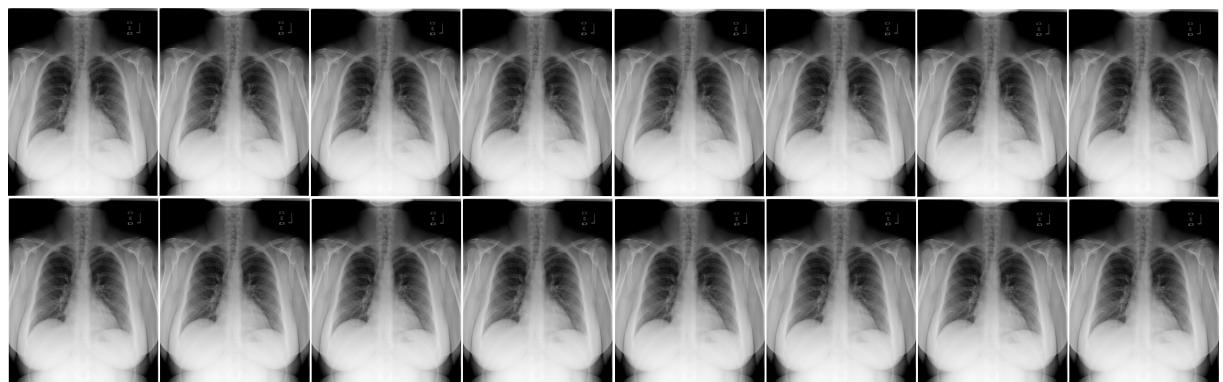


Figura 7.44: **Comparación de predicciones 11.**

Se presentan distintas imágenes con predicciones hechas por el modelo (rojo) junto con sus respectivas antaciones manuales (verde).

7.0.1.12. Entrenamiento 12

Se realiza un nuevo entrenamiento usando los parámetros utilizados en el entrenamiento 8, pero especificando la opción de clase única.

Parámetro	Valor
single_cls	True

Tabla 7.23: **Parámetros 12.**

Tabla que muestra los parámetros utilizados en el entrenamiento 12.

Esto permite que el problema se trate como una clasificación binaria en este caso tratándose solo de presencia o ausencia de opacidades .

El modelo se entrenó durante 350 épocas, donde el mejor rendimiento se obtuvo en la época 58. Después de las 100 épocas de paciencia establecida, no se presentó

una mejora, por lo tanto, se registraron los parámetros del modelo en la época 58 y en la 158.

El desempeño de las métricas durante el entrenamiento se ilustra mediante los siguientes gráficos, donde se observa el comportamiento de la función de costo, la precisión y sensibilidad durante el entrenamiento.

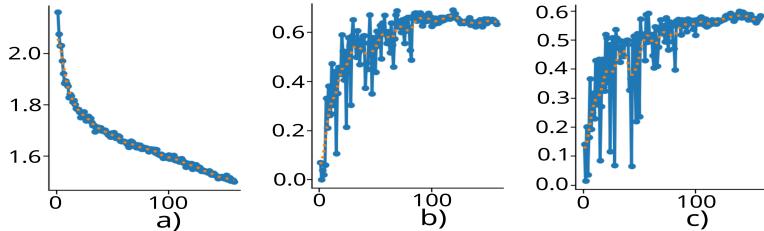


Figura 7.45: **Métricas de entrenamiento 12.**

Gráficas que muestran en el eje x el número de época y en el eje y el valor tomado por: a) La función de costo total. b) Precisión del modelo. c) Sensibilidad del modelo.

Se visualiza el desempeño de las métricas de evaluación donde se muestra el comportamiento de la función de costo y mAP durante el entrenamiento.

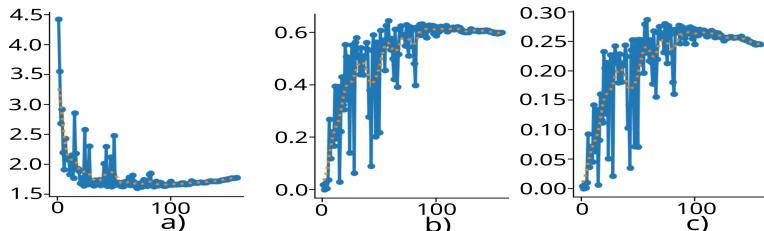


Figura 7.46: **Métricas de evaluación 12.**

Gráficas que muestran en el eje x el número de época y en el eje y el valor tomado por: a) La función de costo total. b) mAP50. c) mAP50-95.

Los valores obtenidos para cada una de las métricas en la época de mejor desempeño, que fue la 58, se resumen en la siguiente tabla:

Métrica	Valor
Precisión	0.647
Sensibilidad	0.583
mAP	0.63

Tabla 7.24: **Métricas 12.**

Tabla que muestra las métricas en la época 58.

Las matrices de confusión obtenidas para el modelo son las siguientes.

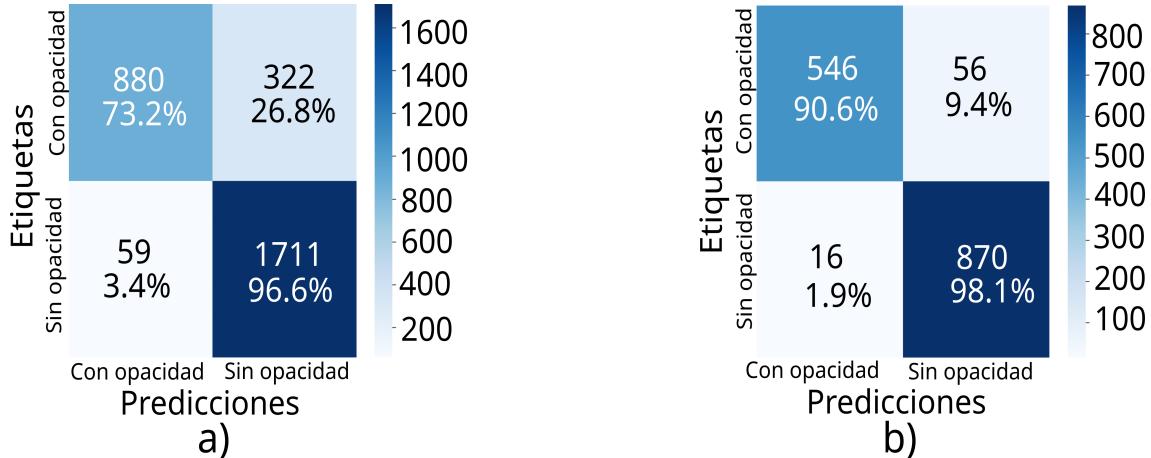


Figura 7.47: Matrices de confusión 12.

Matrices de confusión para la clasificación dada por el modelo, presentando tanto los conteos como el porcentaje que representan para: a) El conjunto de validación. y b) El conjunto de prueba.

Se procede a realizar una visualización de las predicciones hechas por el modelo en la época de mejor desempeño.

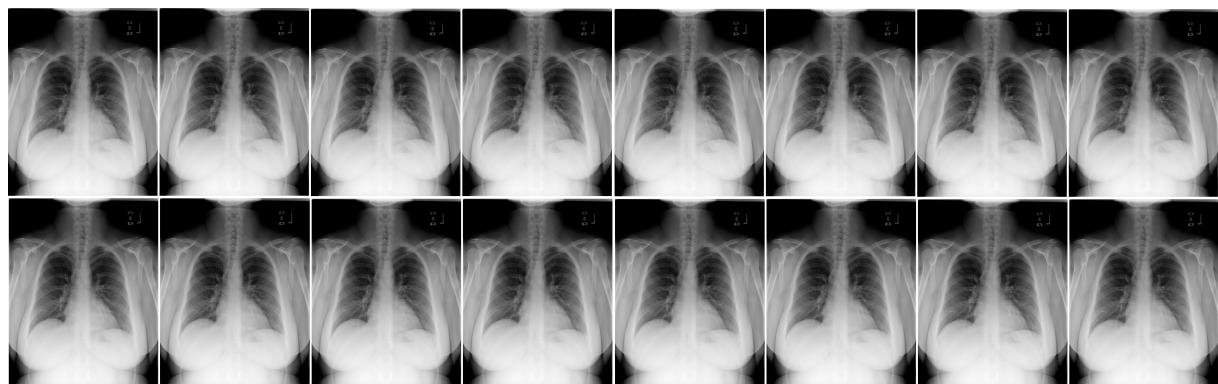


Figura 7.48: Comparación de predicciones 12.

Se presentan distintas imágenes con predicciones hechas por el modelo (rojo) junto con sus respectivas antaciones manuales (verde).

7.0.1.13. Entrenamiento 13

Se realiza un nuevo entrenamiento usando los parámetros utilizados en el entrenamiento 8, pero aplicando una taza de aprendizaje cosenoidal.

Parámetro	Valor
cos_lr	True

Tabla 7.25: Parámetros 13.

Tabla que muestra los parámetros utilizados en el entrenamiento 13.

Esto permite explorar un mejor progreso en la búsqueda de parámetros óptimos para el modelo.

El modelo se entrenó durante 350 épocas, donde el mejor rendimiento se obtuvo en la época . Después de las 100 épocas de paciencia establecida, no se presentó

una mejora, por lo tanto, se registraron los parámetros del modelo en la época 58 y en la 158.

El desempeño de las métricas durante el entrenamiento se ilustra mediante los siguientes gráficos, donde se observa el comportamiento de la función de costo, la precisión y sensibilidad durante el entrenamiento.

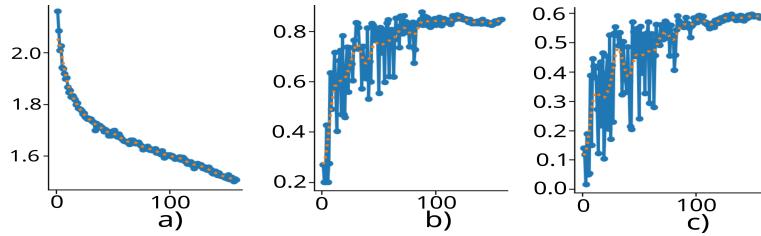


Figura 7.49: **Métricas de entrenamiento 13.**

Gráficas que muestran en el eje x el número de época y en el eje y el valor tomado por: a) La función de costo total. b) Precisión del modelo. c) Sensibilidad del modelo.

Se visualiza el desempeño de las métricas de evaluación donde se muestra el comportamiento de la función de costo y mAP durante el entrenamiento.

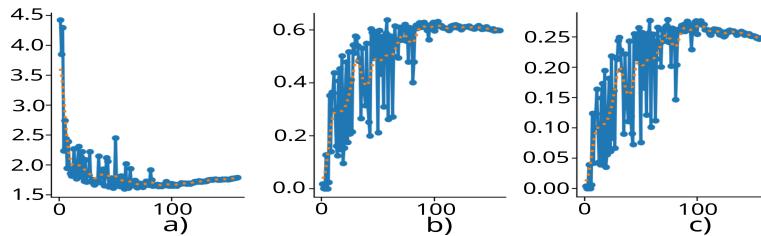


Figura 7.50: **Métricas de evaluación 13.**

Gráficas que muestran en el eje x el número de época y en el eje y el valor tomado por: a) La función de costo total. b) mAP50. c) mAP50-95.

Los valores obtenidos para cada una de las métricas en la época de mejor desempeño, que fue la 58, se resumen en la siguiente tabla:

Métrica	Valor
Precisión	0.632
Sensibilidad	0.561
mAP	0.616

Tabla 7.26: **Métricas 13.**

Tabla que muestra las métricas en la época 58.

Las matrices de confusión obtenidas para el modelo son las siguientes.

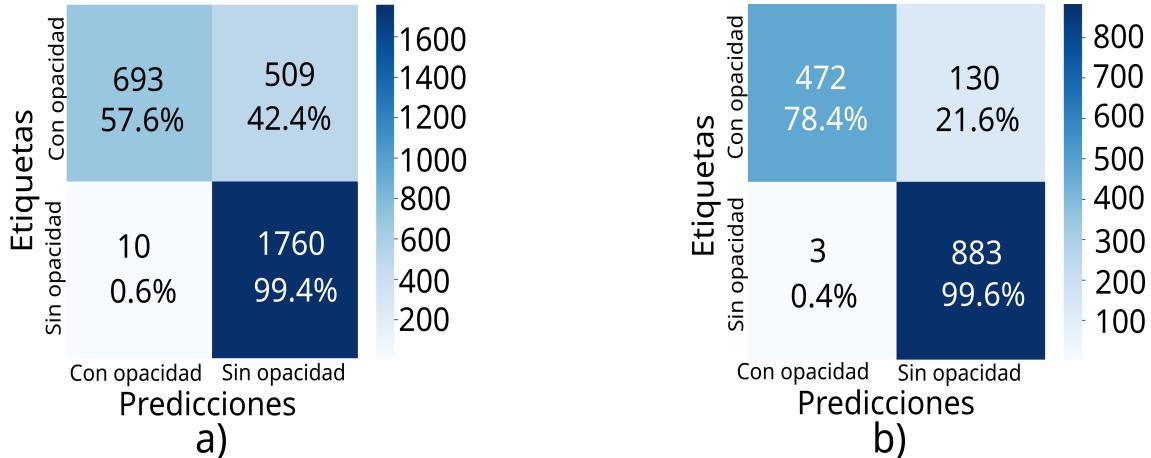


Figura 7.51: **Matrices de confusión 13.**

Matrices de confusión para la clasificación dada por el modelo, presentando tanto los conteos como el porcentaje que representan para: a) El conjunto de validación. y b) El conjunto de prueba.

Se procede a realizar una visualización de las predicciones hechas por el modelo en la época de mejor desempeño.

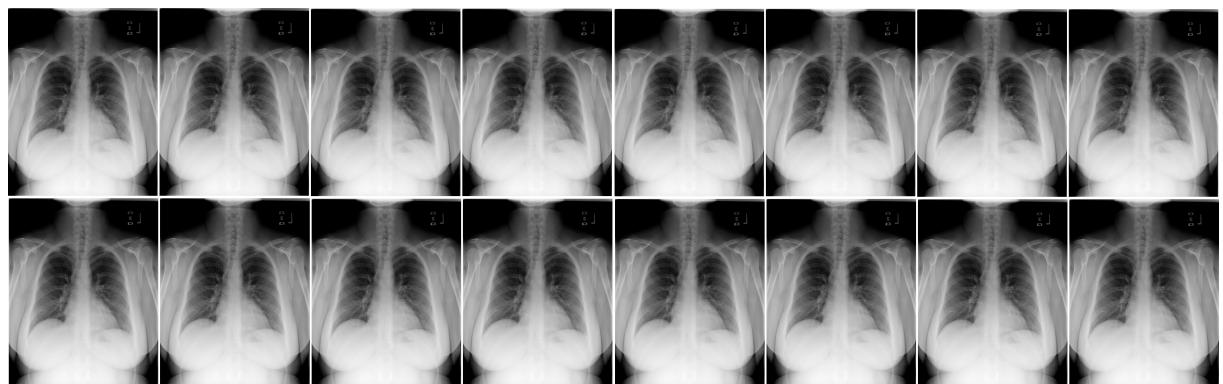


Figura 7.52: **Comparación de predicciones 13.**

Se presentan distintas imágenes con predicciones hechas por el modelo (rojo) junto con sus respectivas antaciones manuales (verde).

7.0.1.14. Entrenamiento 14

Se realiza un nuevo entrenamiento usando los parámetros utilizados en el entrenamiento 8, pero aplicando método *drop out*.

Parámetro	Valor
dropout	0.5

Tabla 7.27: **Parámetros 14.**

Tabla que muestra los parámetros utilizados en el entrenamiento 14.

Esto permite desechar con un 0.5 % de probabilidad ciertos elementos del conjunto de prueba para evitar sobre ajuste.

El modelo se entrenó durante 350 épocas, donde el mejor rendimiento se obtuvo en la época 58. Después de las 100 épocas de paciencia establecida, no se presentó

una mejora, por lo tanto, se registraron los parámetros del modelo en la época 58 y en la 158.

El desempeño de las métricas durante el entrenamiento se ilustra mediante los siguientes gráficos, donde se observa el comportamiento de la función de costo, la precisión y sensibilidad durante el entrenamiento.

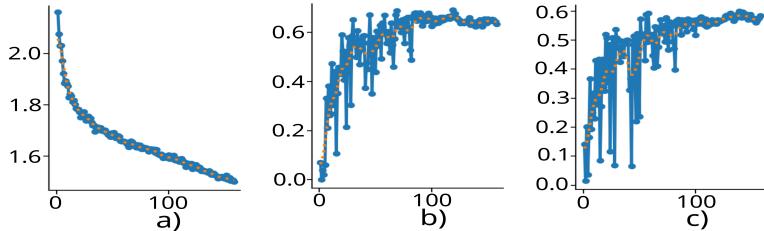


Figura 7.53: **Métricas de entrenamiento 14.**

Gráficas que muestran en el eje x el número de época y en el eje y el valor tomado por: a) La función de costo total. b) Precisión del modelo. c) Sensibilidad del modelo.

Se visualiza el desempeño de las métricas de evaluación donde se muestra el comportamiento de la función de costo y mAP durante el entrenamiento.

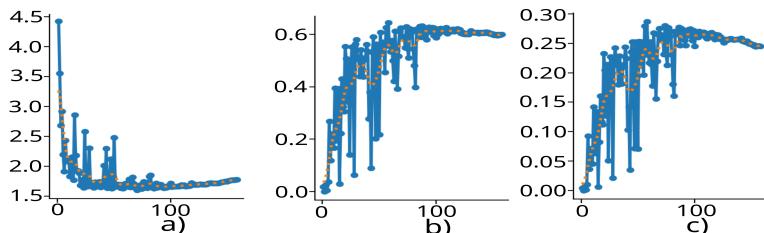


Figura 7.54: **Métricas de evaluación 14.**

Gráficas que muestran en el eje x el número de época y en el eje y el valor tomado por: a) La función de costo total. b) mAP50. c) mAP50-95.

Los valores obtenidos para cada una de las métricas en la época de mejor desempeño, que fue la 58, se resumen en la siguiente tabla:

Métrica	Valor
Precisión	0.647
Sensibilidad	0.583
mAP	0.63

Tabla 7.28: **Métricas 14.**

Tabla que muestra las métricas en la época 58.

Las matrices de confusión obtenidas para el modelo son las siguientes.

Se procede a realizar una visualización de las predicciones hechas por el modelo en la época de mejor desempeño.

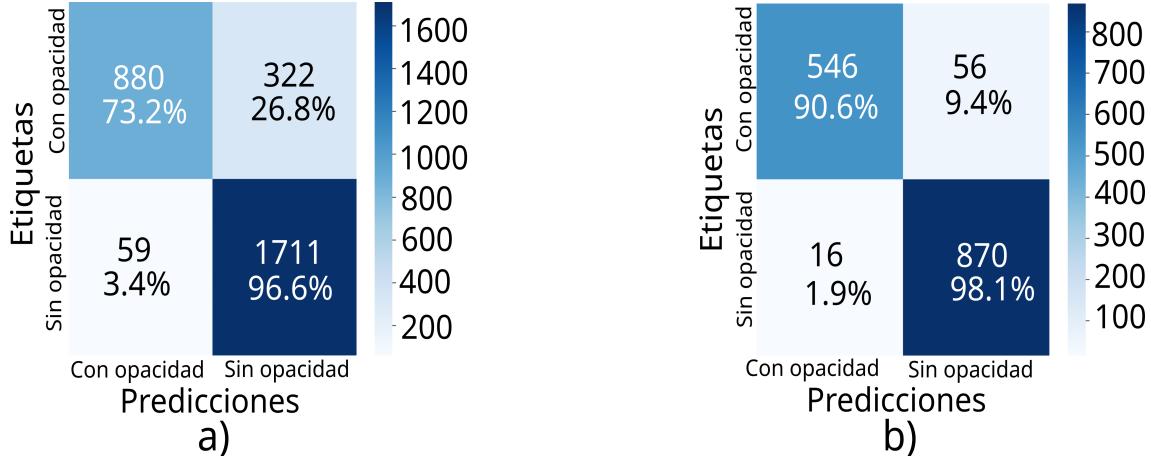


Figura 7.55: **Matrices de confusión 14.**

Matrices de confusión para la clasificación dada por el modelo, presentando tanto los conteos como el porcentaje que representan para: a) El conjunto de validación. y b) El conjunto de prueba.

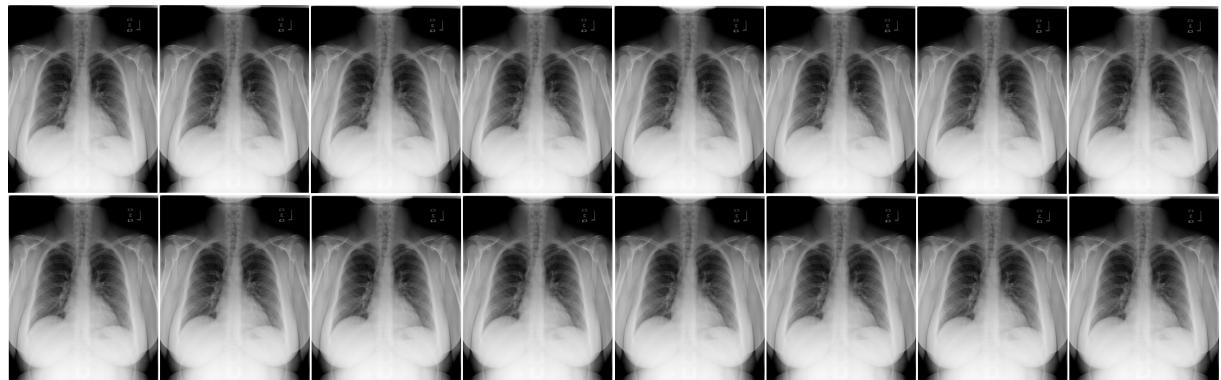


Figura 7.56: **Comparación de predicciones 14.**

Se presentan distintas imágenes con predicciones hechas por el modelo (rojo) junto con sus respectivas antaciones manuales (verde).

7.0.1.15. Entrenamiento 15

Se realiza un nuevo entrenamiento usando los parámetros utilizados en el entrenamiento 8, pero modificando el parametro dfl.

Parámetro	Valor
dfl	2.5

Tabla 7.29: **Parámetros 15.**

Tabla que muestra los parámetros utilizados en el entrenamiento 15.

Esto permite darle más peso de la pérdida focal de distribución, para probar una clasificación de grano más fino.

El modelo se entrenó durante 350 épocas, donde el mejor rendimiento se obtuvo en la época 68. Después de las 100 épocas de paciencia establecida, no se presentó una mejora, por lo tanto, se registraron los parámetros del modelo en la época 68 y en la 168.

El desempeño de las métricas durante el entrenamiento se ilustra mediante los

siguientes gráficos, donde se observa el comportamiento de la función de costo, la precisión y sensibilidad durante el entrenamiento.

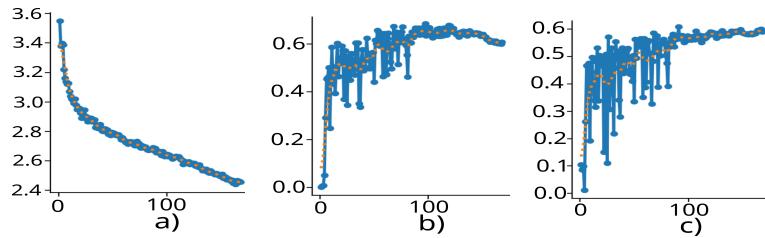


Figura 7.57: **Métricas de entrenamiento 15.**

Gráficas que muestran en el eje x el número de época y en el eje y el valor tomado por: a) La función de costo total. b) Precisión del modelo. c) Sensibilidad del modelo.

Se visualiza el desempeño de las métricas de evaluación donde se muestra el comportamiento de la función de costo y mAP durante el entrenamiento.

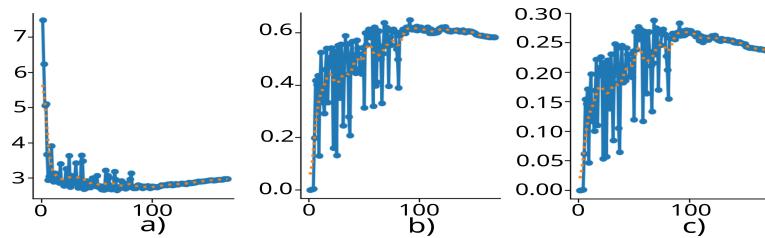


Figura 7.58: **Métricas de evaluación 15.**

Gráficas que muestran en el eje x el número de época y en el eje y el valor tomado por: a) La función de costo total. b) mAP50. c) mAP50-95.

Los valores obtenidos para cada una de las métricas en la época de mejor desempeño, que fue la 68, se resumen en la siguiente tabla:

Métrica	Valor
Precisión	0.678
Sensibilidad	0.562
mAP	0.619

Tabla 7.30: **Métricas 15.**

Tabla que muestra las métricas en la época 68.

Las matrices de confusión obtenidas para el modelo son las siguientes.

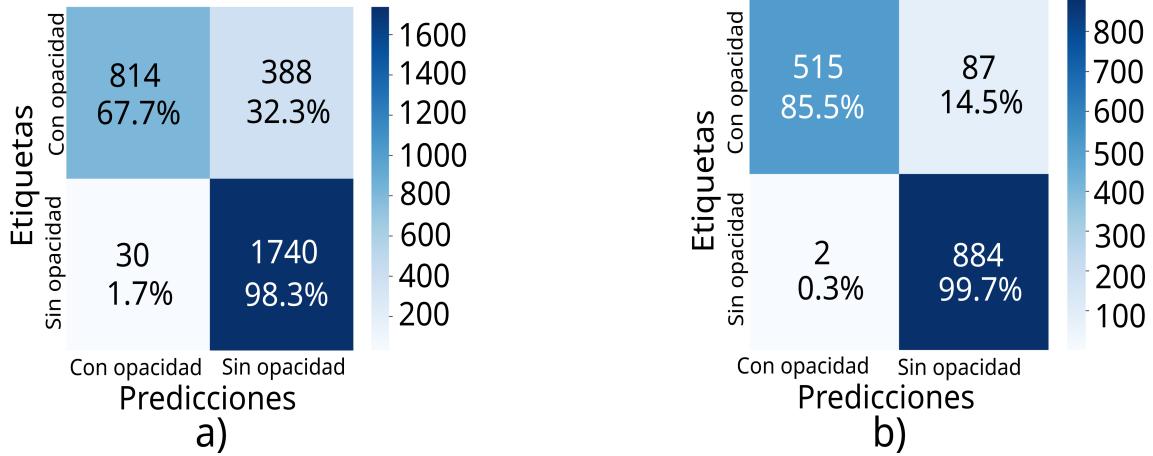


Figura 7.59: **Matrices de confusión 15.**

Matrices de confusión para la clasificación dada por el modelo, presentando tanto los conteos como el porcentaje que representan para: a) El conjunto de validación. y b) El conjunto de prueba.

Se procede a realizar una visualización de las predicciones hechas por el modelo en la época de mejor desempeño.

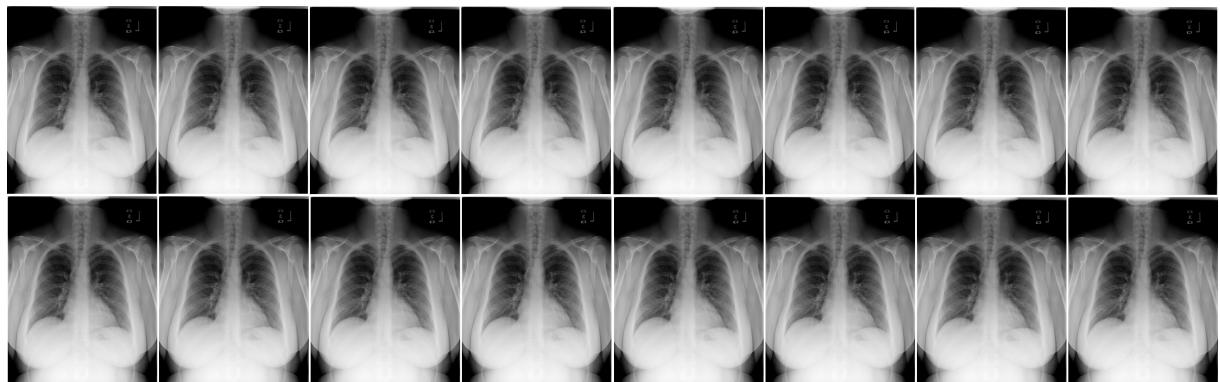


Figura 7.60: **Comparación de predicciones 15.**

Se presentan distintas imágenes con predicciones hechas por el modelo (rojo) junto con sus respectivas antaciones manuales (verde).

Capítulo 8

Conclusiones

En esta sección se discutirán las conclusiones resultados, perspectivas y comentarios finales.

(Falta agregar)

Capítulo 9

Repository

Los códigos utilizados para el desarrollo de esta investigación están contenidos en el repositorio público de GitHub, accesible mediante la siguiente liga:

<https://github.com/DeadWolfX/COVID>

En dicho repositorio se encuentra la carpeta principal denominada COVID con la siguiente estructura de subcarpetas:

- **Anotaciones:** Carpeta que contiene las distintas anotaciones utilizadas para entrenar los modelos de aprendizaje profundo.
- **Codigos_Datos:** Carpeta que contiene tres cuadernos de Jupyter Lab, cada uno con códigos y funciones utilizados para generar las anotaciones contenidas en la carpeta anteriormente descrita.
- **Datos:** Contiene archivos .txt respectivos a cada conjunto de datos utilizados, donde se resume de manera estructurada información relevante sobre cada conjunto.
- **Exploratorios:** Contiene tres cuadernos de Jupyter Lab, uno por cada conjunto de datos utilizado. Estos cuadernos contienen los códigos y funciones utilizados para realizar los respectivos análisis exploratorios de los conjuntos de datos.
- **env.yml:** Configuración de ambiente virtual de Anaconda que contiene todas las librerías necesarias para los análisis exploratorios, visualización, preprocesamiento de datos y manipulación de etiquetados.

Bibliografía y referencias

- [1] ESTEVAN. ET AL. *Examen radiográfico del tórax en las neumonías de probable causa bacteriana*, scielouy Arch. Pediatr. Urug (73) 15 - 21, 2002 ISSN 1688-1249, http://www.scielo.edu.uy/scielo.php?script=sci_arttext&pid=S1688-12492002000100004
- [2] HIJKATA. ET AL. *Dyspnoea, fever, patchy ground-glass opacities and intermittent severe epigastralgia*, BMJ Publishing Group Ltd (65) 890–890, 2010 doi: 10.1136/thx.2010.136655 <https://thorax.bmj.com/content/65/10/890>
- [3] LIN. ET AL. *Microsoft COCO: Common Objects in Context*, arXiv 2014 doi: 10.48550/ARXIV.1405.0312 <https://arxiv.org/abs/1405.0312>
- [4] TAN. ET AL. *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. Proceedings of the 36th International Conference on Machine Learning 6105–6114 , 2019. Chaudhuri, Kamalika and Salakhutdinov, Ruslan 97, Proceedings of Machine Learning Research <https://proceedings.mlr.press/v97/tan19a.html>
- [5] KAREM DAIANE MARCOMINI. ET AL. *A deep learning approach for COVID-19 screening and localization on chest x-ray images*. Proc. SPIE 12033, Medical Imaging 2022: Computer-Aided Diagnosis, 1203327 . <https://doi.org/10.1117/12.2613177>
- [6] POLAT H. ET AL. *Automatic detection and localization of COVID-19 pneumonia using axial computed tomography images and deep convolutional neural networks*. Int J Imaging Syst Technol 31(2):509-524. 2021 doi: 10.1002/ima.22558. Epub 2021 Feb 16. PMID: 33821092; PMCID: PMC8013431.<https://onlinelibrary.wiley.com/doi/10.1002/ima.22558>
- [7] XIA MA. ET AL. *COVID-19 lesion discrimination and localization network based on multi-receptive field attention module on CT images*. Optik Volume 241 (Stuttg). 2021 doi: 10.1016/j.ijleo.2021.167100. Epub 2021 May 7. PMID: 33976457; PMCID: PMC8103744.<https://www.sciencedirect.com/science/article/pii/S0030402621007762>
- [8] CARANDINI M. ET AL. *What simple and complex cells compute*. J Physiol 463-466. 2006 doi: 10.1113/jphysiol.2006.118976 Epub 2006 Sep 14. PMID: 16973710; PMCID: PMC1890437.<https://physoc.onlinelibrary.wiley.com/doi/full/10.1113/jphysiol.2006.118976>
- [9] LECUN. ET AL. *Gradient-Based Learning Applied to Document Recognition*. Proceedings of the IEEE. vol. 86, no. 11, pp. 2278-2324 doi: 8610.1109/5.726791. <https://ieeexplore.ieee.org/document/726791>

- [10] VASWANI. ET AL. *Attention Is All You Need*, arXiv 2017 doi: 10.48550/ARXIV.1706.03762 <https://arxiv.org/abs/1706.03762>
- [11] DOSOVITSKIY. ET AL. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*, arXiv 2020 doi: 10.48550/ARXIV.2010.11929 <https://arxiv.org/abs/2010.11929>
- [12] TAN. ET AL. *EfficientDet: Scalable and Efficient Object Detection*, arXiv 2019 doi: 10.48550/ARXIV.1911.09070 <https://arxiv.org/abs/1911.09070>
- [13] REDMON. ET AL. *You Only Look Once: Unified, Real-Time Object Detection*, arXiv 2015 doi: 10.48550/ARXIV.1506.02640 <https://arxiv.org/abs/1506.02640>
- [14] GOMES JC. ET AL. *IKONOS: an intelligent tool to support diagnosis of COVID-19 by texture analysis of X-ray images.*, Res. Biomed. Eng. 2022;38(1):15–28 doi: 0.1007/s42600-020-00091-7. Epub 2020 Sep 3. PMID: PMC7471577. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7471577/>
- [15] SZEGEDY. ET AL. *Rethinking the Inception Architecture for Computer Vision*, arXiv 2015 doi: 10.48550/ARXIV.1512.00567 <https://arxiv.org/abs/1512.00567>
- [16] FRANÇOIS CHOLLET. ET AL. *Xception: Deep Learning with Depthwise Separable Convolutions*, arXiv 2016 doi: 10.48550/ARXIV.1610.02357 <https://arxiv.org/abs/1610.02357>
- [17] WANG D. ET AL. 2020 *An efficient mixture of deep and machine learning models for COVID-19 diagnosis in chest X-ray images*, 15(11): e0242535. <https://doi.org/10.1371/journal.pone.0242535>
- [18] SHEN D. ET AL. *Deep Learning in Medical Image Analysis*, Annu Rev Biomed Eng. 2017 doi: 10.1146/annurev-bioeng-071516-044442. Epub 2017 Mar 9. PMID: 28301734 PMCID: PMC5479722. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5479722/>
- [19] KUCHANA. ET AL. *AI aiding in diagnosing, tracking recovery of COVID-19 using deep learning on Chest CT scans*, Multimed Tools Appl 80, 9161–9175 (2021) <https://doi.org/10.1007/s11042-020-10010-8>
- [20] DANDI YANG. ET AL. *Detection and analysis of COVID-19 in medical images using deep learning techniques*, Nature Scientific Reports 2021 <https://doi.org/10.1038/s41598-021-99015-3>
- [21] VAYÁ. ET AL. *BIMCV COVID-19+: a large annotated dataset of RX and CT images from COVID-19 patients*, arXiv 2020 doi: 10.48550/ARXIV.2006.01174 <https://arxiv.org/abs/2006.01174>
- [22] P. LAKHANI. ET AL. *The 2021 SIIM-FISABIO-RSNA Machine Learning COVID-19 Challenge: Annotation and Standard Exam Classification of COVID-19 Chest Radiographs.*, OSFPreprints 2021 doi: 10.31219/osf.io/532ek <https://doi.org/10.31219/osf.io/532ek>
- [23] RAJPURKAR. ET AL. *CheXNet: Radiologist-level pneumonia detection on chest X-rays using deep learning.*, Nature Medicine, 25(11), 1527-1532.

- [24] R. HARALICK. ET AL. *extural features for image classificatio.*, SMC, vol. 3, pp. 610–621, 1973.
- [25] SHU H. ET AL. *Moment-based approaches in imaging. Part 1, basic features.*, IEEE Eng Med Biol Mag. 2007 Sep-Oct;26(5):70-4. doi: 10.1109/emb.2007.906026. PMID: 17941325; PMCID: PMC2230630.
- [26] WANG. ET AL. *ChestX-ray8: Hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases.*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3462-3471).
- [27] TSAI. ET AL. *The RSNA International COVID-19 Open Radiology Database (RICORD)*, Radiology 299(1) E204-E213 2021 doi: 10.1148/radiol.2021203957 <https://doi.org/10.1148/radiol.2021203957>
- [28] LITMANOVICH. ET AL. *Review of Chest Radiograph Findings of COVID-19 Pneumonia and Suggested Reporting Language.*, Journal of Thoracic Imaging 35(6) 354-360, 2020 doi: 10.1097/RTI.0000000000000541 https://journals.lww.com/thoracicimaging/Fulltext/2020/11000/Review_of_Chest_Radiograph_Findings_of_COVID_19.4.aspx
- [29] HUANG. ET AL. *Densely Connected Convolutional Networks*, arXiv 2016 doi: 10.48550/ARXIV.1608.06993 <https://arxiv.org/pdf/1608.06993>
- [30] HE, KAIMING. ET AL. *Deep Residual Learning for Image Recognition*, arXiv 2015 doi: 10.48550/ARXIV.1512.03385 <https://arxiv.org/abs/1512.03385>
- [31] ROHIT THAKUR. TOWARDS DATA SCIENCE 2019, *Step by step VGG16 implementation in Keras for beginners.* <https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c>
- [32] SIK-HO TSANG. TOWARDS DATA SCIENCE 2018, *Review: Xception — With Depthwise Separable Convolution, Better Than Inception-v3 (Image Classification)* <https://towardsdatascience.com/review-xception-with-depthwise-separable-convolution-better-than-inception-v3-image-de967dd42568>
- [33] RONNEBERGER. ET AL. *U-Net: Convolutional Networks for Biomedical Image Segmentation*, arXiv 2015 doi: 10.48550/ARXIV.1505.04597 <https://arxiv.org/abs/1505.04597>
- [34] ORGANIZACIÓN MUNDIAL DE LA SALUD (OMS) <https://www.who.int/es/emergencies/diseases/novel-coronavirus-2019>
- [35] THE NATIONAL CENTER FOR BIOTECHNOLOGY INFORMATION (NCBI) <https://www.ncbi.nlm.nih.gov/books/NBK554776/>
- [36] INCEPTION V3 MODEL ARCHITECTURE <https://iq.opengenus.org/inception-v3-model-architecture/>
- [37] JACOB SOLAWETZ. ET AL. *What is YOLOv5? A Guide for Beginners.*, roboflow 2020 <https://blog.roboflow.com/yolov5-improvements-and-evaluation/>

- [38] NADIA ROJAS. ET AL. *Derrame pleural en radiografía y tomografía.*, SlideShare 2015 <https://es.slideshare.net/nadiarojasvalenzuela/derrame-pleural-54365182>
- [39] REYNOLDS JH. ET AL. *Pneumonia in the immunocompetent patient.* Radiol. 2010 Dec;83(996):998-1009. doi: 10.1259/bjr/31200593. PMID: 21088086; PMCID: PMC3473604.
- [40] NICHOLAS P. ET AL. *Atypical Pneumonia: Definition, Causes, and Imaging Features.* RadioGraphics 2021 41:3, 720-741
- [41] REBECCA DEZUBE. ET AL. *Introducción al aparato respiratorio.*, MDS, MANUAL MDS Versión para público en general 2021
- [42] NAINA MARIKAR S. ET AL. *Pharmacological inhibition of human EZH2 can influence a regenerative β-like cell capacity with in vitro insulin release in pancreatic ductal cells.*, Clin Epigenet. 2023;15(1):101. doi: 10.1186/s13148-023-01491-z <https://www.msdmanuals.com/es-mx/hogar/trastornos-del-pulm%C3%B3n-y-las-v%C3%ADas-respiratorias/biolog%C3%ADa-de-los-pulmones-y-de-las-v%C3%ADas-respiratorias/introducci%C3%B3n-al-aparato-respiratorio>
- [43] DUTHIE J. ET AL. *Anatomy and physiology of respiration.* Nursing (Lond). 1984 Jul;2(27):785-7. PMID: 6429584.
- [44] MINCHIN S. ET AL. *Understanding biochemistry: structure and function of nucleic acids.* Biochem. 2019 Oct 16;63(4):433-456. doi: 10.1042/EBC20180038. PMID: 31652314; PMCID: PMC6822018.
- [45] MACK CD. ET AL. *Effectiveness and use of reverse transcriptase polymerase chain reaction point of care testing in a large-scale COVID-19 surveillance system.* Pharmacoepidemiol Drug Saf. 2022 May;31(5):511-518. doi: 10.1002/pds.5424. Epub 2022 Mar 11. PMID: 35225407; PMCID: PMC9088538.
- [46] TÜRK F. ET AL. *Detection of Lung Opacity and Treatment Planning with Three-Channel Fusion CNN Model.* Arab J Sci Eng. 2023 Apr 14:1-13. doi: 10.1007/s13369-023-07843-4.
- [47] ZOMPATORI M. ET AL. *Diffuse ground-glass opacity of the lung. A guide to interpreting the high-resolution computed tomographic (HRCT) picture.* Radiol Med. 1994 Nov;88(5):576-81. Italian. PMID: 7824771.
- [48] KEVIN P. ET AL. *Probabilistic Machine Learning: An introduction.* MIT Press 2022; 29-101.
- [49] TOM MITCHELL. ET AL. *Machine Learning.* McGraw Hill, 1997; ISBN 0070428077.
- [50] TEDRAKE, RUSS. ET AL. *Robotic Manipulation (Perception, Planning, and Control).* Course Notes for MIT 6.421, 2023; <http://manipulation.mit.edu>.
- [51] JOCHER. ET AL. *YOLOvxx* Ultralytics. 2023; <https://github.com/ultralytics/ultralytics>.

- [52] SANTOS LÓPEZ G. ET AL. *SARS-CoV-2: basic concepts, origin and treatment advances*. Gac Med Mex. 2021;157(1):84-89. English. doi: 10.24875/GMM.M21000524. PMID: 34125824.
- [53] PFEIFFER D. ET AL. *Advanced X-ray Imaging Technology*. Recent Results Cancer Res. 2020;216:3-30. doi: 10.1007/978-3-030-42618-7_1. PMID: 32594383.
- [54] SEERAM E. ET AL. *Computed Tomography: A Technical Review*. Radiol Technol. 2018 Jan;89(3):279CT-302CT. PMID: 29298954.
- [55] FISHER AR. ET AL. *Magnetic resonance imaging techniques*. Clin Liver Dis. 2002 Feb;6(1):53-72, vi. doi: 10.1016/s1089-3261(03)00066-7. PMID: 11933596.
- [56] WELLS PN. ET AL. *Ultrasound imaging*. Phys Med Biol. 2006 Jul 7;51(13):R83-98. doi: 10.1088/0031-9155/51/13/R06. Epub 2006 Jun 20. PMID: 16790922.
- [57] LAROBINA M. ET AL. *Thirty Years of the DICOM Standard*. Tomography. 2023 Oct 6;9(5):1829-1838. doi: 10.3390/tomography9050145. PMID: 37888737; PMCID: PMC10610864.
- [58] SARIPALLE R. ET AL. *Using HL7 FHIR to achieve interoperability in patient health record*. J Biomed Inform. 2019 Jun;94:103188. doi: 10.1016/j.jbi.2019.103188. Epub 2019 May 4. PMID: 31063828.
- [59] BERND JÄHNE. ET AL. *Computer Vision and Applications A Guide for Students and Practitioners*. Academic Press 2000 Elsevier <https://doi.org/10.1016/B978-0-12-379777-3.X5000-6>
- [60] WALTER RUDIN. *Functional Analysis*. Universidad de Michigan. McGraw-Hill, 1991. ISBN 0070542368, 9780070542365.
- [61] STEPHEN H. FRIEDBERG. *Linear Algebra*. Pearson College Division (2002) ISBN-13: 978-0130084514
- [62] ARISTOMENIS S. ET AL. *Machine Learning Paradigms*. Springer Cham <https://doi.org/10.1007/978-3-319-19135-5>
- [63] SHIV RAM DUBEY. ET AL. *Activation Functions in Deep Learning: A Comprehensive Survey and Benchmark*. (2022) arXiv.
- [64] LORENZO CIAMPICONI. ET AL. *A survey and taxonomy of loss functions in machine learning*. (2023) arXiv.
- [65] CHAI. ET AL. *Root mean square error (RMSE) or mean absolute error (MAE)?— Arguments against avoiding RMSE in the literature*. (2014). Geoscientific Model Development. 7. 1247-1250. 10.5194/gmd-7-1247-2014.
- [66] PHILIP HANS FRANSES. ET AL. *A note on the Mean Absolute Scaled Error*. (2016). International Journal of Forecasting, Volume 32, Pages 20-22. <https://doi.org/10.1016/j.ijforecast.2015.03.008>

- [67] SHAH A. ET AL. *Advancement of deep learning in pneumonia/Covid-19 classification and localization: A systematic review with qualitative and quantitative analysis*. Chronic Diseases and Translational Medicine (2022) Volume 8, Issue 3. 154-171 <https://doi.org/10.1002/cdt.3.17>
- [68] ANQI MAO. ET AL. *Cross-Entropy Loss Functions: Theoretical Analysis and Applications*. (2023). arXiv.
- [69] J M S PEARCE. ET AL. *Sir Charles Scott Sherrington (1857–1952) and the synapse*. Journal of Neurology, Neurosurgery & Psychiatry 74 (2004) 44–544. doi: 10.1136/jnnp.2003.01792
- [70] RUBY. ET AL. *Binary cross entropy with deep learning technique for Image classification*. (2020). International Journal of Advanced Trends in Computer Science and Engineering. 9. 10.30534/ijatcse/2020/175942020.
- [71] LEUNG. ET AL. *SGD*. (2009). 10.1007/978-3-540-29676-8_6433.
- [72] YANLI LIU. ET AL. *An Improved Analysis of Stochastic Gradient Descent with Momentum*. (2020). arXiv.
- [73] RACHEL WARD. ET AL. *AdaGrad stepsizes: Sharp convergence over nonconvex landscapes*. (2021). arXiv.
- [74] B. SOUJANYA. ET AL. *Optimization with ADAM and RMSprop in Convolution neural Network (CNN): A Case study for Telugu Handwritten Characters*. (2020) International Journal of Emerging Trends in Engineering Research Volume 8. No. 9 <https://doi.org/10.30534/ijeter/2020/38892020>
- [75] FRACTAL FOUNDATION *What are fractals?* <https://fractalfoundation.org/resources/what-are-fractals/>
- [76] CHURRUCA M. ET AL. *COVID-19 pneumonia: A review of typical radiological characteristics*. World J Radiol. 2021 Oct 28;13(10):327-343. doi: 10.4329/wjr.v13.i10.327. PMID: 34786188; PMCID: PMC8567439.
- [77] MATTHEW D. ET AL. *ADADELTA: An Adaptive Learning Rate Method*. (2012). arXiv.
- [78] MAGIQUO *Redes neuronales o el arte de imitar al cerebro*. (2019). <https://magiquo.com/redes-neuronales-o-el-arte-de-imitar-al-cerebro-humano/>
- [79] GUILHOTO. ET AL. *An overview of artificial neural networks for mathematicians.*, Univ. Chicago 2018.
- [80] TSUNG-YI LIN. ET AL. *Focal Loss for Dense Object Detection.*, ar-Xiv 2018. doi: <https://doi.org/10.48550/arXiv.1708.02002> Focus to learn more
- [81] SALINAS-ESCUDERO. ET AL. *A survival analysis of COVID-19 in the Mexican population.*, BMC Public Health 20, 1616 (2020). <https://doi.org/10.1186/s12889-020-09721-2>
- [82] JESÚS *Gráfica completa de redes neuronales*, DATA SMARTS 2022 <https://www.datasmarts.net/grafica-completa-de-redes-neuronales/>

- [83] TSUNG-YI LIN. ET AL. *Focal Loss for Dense Object Detection.*, arXiv 2018. <https://doi.org/10.48550/arXiv.1708.02002>
- [84] EVE CURIE *Madame Curie: A Biography.*, Da Capo Press; Reissue. ISBN 0306810387
- [85] ANDREW HODGES *Alan Turing: The Enigma.*, Princeton University Press. ISBN 9780691164724
- [86] A. M. TURING *COMPUTING MACHINERY AND INTELLIGENCE.*, Mind, Volume LIX, Issue 236, October 1950, Pages 433-460 <https://doi.org/10.1093/mind/LIX.236.433>
- [87] W. ROBERT NITSKE *The Life of Wilhelm Conrad Röntgen, Discoverer of the X Ray.*, University of Arizona Press, ISBN 0816502595.
- [88] MCCARTHY *Recursive Functions of Symbolic Expressions and Their Computation by Machine.*, Communications of the ACM 1960, Volume 3, 184-195 <https://doi.org/10.1145/367177.367199>
- [89] LESTER EARNEST *Dr. McCarthy's lecture The Present State of Research on AI.*, Turing award winners website https://amturing.acm.org/award_winners/mccarthy_1118322.cf

Listado de Figuras

Figura1.1	Marie Skłodowska-Curie (1867 - 1934).	5
Figura1.2	Alan Turing (1912-1954).	6
Figura1.3	Wilhelm Conrad Röntgen (1845-1923).	7
Figura1.4	John Patrick McCarthy (1927-2011).	8
Figura2.1	Esquema del aparato respiratorio.	10
Figura2.2	Esquema de la estructura del ADN y el ARN.	11
Figura2.3	Imagen sin presencia de opacidades frente a imagen con presencia de opacidades.	12
Figura2.4	Ejemplo de opacidad causada por consolidación.	13
Figura2.5	Ejemplo de opacidad causada por derrame pleural.	13
Figura2.6	Ejemplo de opacidad causada por neumonía.	14
Figura2.7	Fotos de un píxel.	16
Figura2.8	Etiquetado numérico para colores.	17
Figura2.9	Representación gráfica de la combinación de colores.	18
Figura2.10	Estructura geométrica de una imagen dada como matriz.	19
Figura2.11	Estructura geométrica de una imagen dada como matriz.	19
Figura2.12	MONOCROMO1.	20
Figura2.13	MONOCROMO2.	20
Figura2.14	Segmentación y detección de objetos.	28
Figura2.15	Matriz de confusión para $n = 2$ clases.	29
Figura2.16	Índice Jaccard.	31
Figura2.17	Neurona biológica vs artificial.	33
Figura2.18	Red neuronal artificial.	35
Figura2.19	Descenso de gradiente.	40
Figura2.20	Red neuronal preentrenada.	50
Figura2.21	Red neuronal adecuada.	50
Figura2.22	Arquitecturas de redes neuronales.	51
Figura3.1	Tamaños de YOLO.	52
Figura3.2	Arquitectura de YOLO.	53
Figura3.3	Algoritmo de detección.	54
Figura3.4	supresión de no máximos.	54
Figura3.5	Predicción final de YOLO.	55
Figura3.6	Detección YOLO.	55
Figura3.7	Transformador.	57
Figura3.8	División en parches.	57
Figura3.9	Aplanado.	58
Figura3.10	Proyección ortogonal.	58
Figura3.11	Transformador de visión.	59
Figura3.12	Red neuronal clásica comparando imágenes.	60

Figura3.13	Neuronas simples y complejas.	61
Figura3.14	Convolución como neurona simple.	62
Figura3.15	Relleno.	63
Figura3.16	Agrupación como neurona compleja.	63
Figura3.17	Arquitectura de RetinaNet.	65
Figura3.18	Desempeño de RetinaNet.	65
Figura4.1	Arquitectura VGG16.	67
Figura4.2	Arquitectura DenseNet.	67
Figura4.3	ResNet 50	68
Figura4.4	Matrices de confusión.	69
Figura4.5	Convolución profunda separable.	70
Figura4.6	Convolución en Xception.	71
Figura4.7	Arquitctura InceptionV3.	71
Figura4.8	Modulos propuestos.	74
Figura4.9	Predicciones del modelo.	76
Figura5.1	Anotaciones neumonía.	79
Figura5.2	Distribución por clases.	80
Figura5.3	Recuadros delimitadores por imagen.	80
Figura5.4	Área de recuadros delimitadores.	80
Figura5.5	Histograma de imágenes con opacidades.	81
Figura5.6	Histograma de imágenes sin opacidades.	81
Figura5.7	Distribución de datos.	82
Figura5.8	Sin hallazgos.	83
Figura5.9	Atelectasia.	83
Figura5.10	Efusión.	83
Figura5.11	Cardiomegalia.	84
Figura5.12	Infiltración.	84
Figura5.13	Neumonía.	85
Figura5.14	Neumotórax.	85
Figura5.15	Masa pulmonar.	86
Figura5.16	Nódulo.	86
Figura5.17	Distribución de datos por clase desvalanceada.	87
Figura5.18	Distribución de datos por clase valanceada.	87
Figura5.19	Recuadros delimitadores por imagen.	87
Figura5.20	Área de recuadros delimitadores.	88
Figura5.21	Histograma de imágenes sin hallazgos.	88
Figura5.22	Histograma de imágenes con cardiomegalia.	89
Figura5.23	Histograma de imágenes con atelectasia.	89
Figura5.24	Histograma de imágenes con efusión.	90
Figura5.25	Histograma de imágenes con infiltración.	90
Figura5.26	Histograma de imágenes con neumonía.	91
Figura5.27	Histograma de imágenes con neumotórax.	91
Figura5.28	Histograma de imágenes con masa pulmonar.	92
Figura5.29	Histograma de imágenes con nódulo.	92
Figura5.30	Etiquetado de imágenes.	94
Figura5.31	Negativo para neumonía.	96
Figura5.32	Aparaciencia típica.	96
Figura5.33	Aparaciencia indeterminada.	97
Figura5.34	Aparaciencia atípica.	97
Figura5.35	Distribución de clases.	98

Figura5.36 Monocromo 2 y 1.	98
Figura5.37 Recuadros delimitadores por imagen.	98
Figura5.38 Área de las imágenes.	99
Figura5.39 Área de recuadros delimitadores.	99
Figura5.40 Imágenes sin recuadros delimitadores.	100
Figura5.41 Histograma de imágenes sin daños.	100
Figura5.42 Histograma de imágenes con daños típicos.	101
Figura5.43 Histograma de imágenes con daños indeterminados. .	101
Figura5.44 Histograma de imágenes con daños indeterminados. .	102
Figura5.45 Histograma de imágenes con daños atípicos.	102
Figura6.1 Anotaciones originales en los conjuntos.	105
Figura6.2 Anotaciones propuestas.	105
Figura6.3 Anotaciones propuestas.	106
Figura7.1 Métricas de entrenamiento 1.	109
Figura7.2 Métricas de evaluación 1.	110
Figura7.3 Matrices de confusión 1.	110
Figura7.4 Comparación de predicciones 1.	111
Figura7.5 Métricas de entrenamiento 2.	111
Figura7.6 Métricas de evaluación 2.	112
Figura7.7 Matrices de confusión 2.	112
Figura7.8 Comparación de predicciones 2.	113
Figura7.9 Métricas de entrenamiento 3.	113
Figura7.10 Métricas de evaluación 3.	114
Figura7.11 Matrices de confusión 3.	114
Figura7.12 Comparación de predicciones 3.	115
Figura7.13 Métricas de entrenamiento 4.	115
Figura7.14 Métricas de evaluación 4.	116
Figura7.15 Matrices de confusión 4.	116
Figura7.16 Comparación de predicciones 4.	117
Figura7.17 Métricas de entrenamiento 5.	118
Figura7.18 Métricas de evaluación 5.	118
Figura7.19 Matrices de confusión 5.	119
Figura7.20 Comparación de predicciones 5.	119
Figura7.21 Métricas de entrenamiento 6.	120
Figura7.22 Métricas de evaluación 6.	120
Figura7.23 Matrices de confusión 6.	121
Figura7.24 Comparación de predicciones 6.	121
Figura7.25 Métricas de entrenamiento 7.	122
Figura7.26 Métricas de evaluación 7.	122
Figura7.27 Matrices de confusión 7.	123
Figura7.28 Comparación de predicciones 7.	123
Figura7.29 Métricas de entrenamiento 8.	124
Figura7.30 Métricas de evaluación 8.	124
Figura7.31 Matrices de confusión 8.	125
Figura7.32 Comparación de predicciones 8.	125
Figura7.33 Métricas de entrenamiento 9.	126
Figura7.34 Métricas de evaluación 9.	126
Figura7.35 Matrices de confusión 9.	127
Figura7.36 Comparación de predicciones 9.	127
Figura7.37 Métricas de entrenamiento 10.	128

Figura7.38 Métricas de evaluación 10.	128
Figura7.39 Matrices de confusión 10.	129
Figura7.40 Comparación de predicciones 10.	129
Figura7.41 Métricas de entrenamiento 11.	130
Figura7.42 Métricas de evaluación 11.	130
Figura7.43 Matrices de confusión 11.	131
Figura7.44 Comparación de predicciones 11.	131
Figura7.45 Métricas de entrenamiento 12.	132
Figura7.46 Métricas de evaluación 12.	132
Figura7.47 Matrices de confusión 12.	133
Figura7.48 Comparación de predicciones 12.	133
Figura7.49 Métricas de entrenamiento 13.	134
Figura7.50 Métricas de evaluación 13.	134
Figura7.51 Matrices de confusión 13.	135
Figura7.52 Comparación de predicciones 13.	135
Figura7.53 Métricas de entrenamiento 14.	136
Figura7.54 Métricas de evaluación 14.	136
Figura7.55 Matrices de confusión 14.	137
Figura7.56 Comparación de predicciones 14.	137
Figura7.57 Métricas de entrenamiento 15.	138
Figura7.58 Métricas de evaluación 15.	138
Figura7.59 Matrices de confusión 15.	139
Figura7.60 Comparación de predicciones 15.	139

Listado de Tablas

4.1 Evaluación de preentrenamiento.	72
4.2 Desempeño de las implementaciones propuestas.	74
4.3 Evaluación de detector.	75
4.4 Evaluación de clasificación.	76
5.1 Criterios de anotación.	94
5.2 Severidades.	95
5.3 Lenguaje de informe.	95
6.1 Tamaños de Yolov8.	103
7.1 Parámetros 1.	109
7.2 Métricas 1.	110
7.3 Parámetros 2.	111
7.4 Métricas 2.	112
7.5 Parámetros 3.	113
7.6 Métricas 3.	114
7.7 Parámetros 4.	115
7.8 Métricas 4.	116
7.9 Parámetros 5.	117

7.10 Métricas 5.	118
7.11 Parámetros 6.	119
7.12 Métricas 6.	120
7.13 Parámetros 7.	121
7.14 Métricas 7.	122
7.15 Parámetros 8.	123
7.16 Métricas 8.	124
7.17 Parámetros 9.	125
7.18 Métricas 9.	126
7.19 Parámetros 10.	127
7.20 Métricas 10.	128
7.21 Parámetros 11.	129
7.22 Métricas 11.	130
7.23 Parámetros 12.	131
7.24 Métricas 12.	132
7.25 Parámetros 13.	133
7.26 Métricas 13.	134
7.27 Parámetros 14.	135
7.28 Métricas 14.	136
7.29 Parámetros 15.	137
7.30 Métricas 15.	138