

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO

MAESTRÍA Y DOCTORADO EN CIENCIAS E INGENIERÍA



“Modelo basado en *deep learning* para el diagnóstico de Tuberculosis pulmonar utilizando radiografías de tórax y perfiles clínicos”

TESIS

Que para obtener el grado de maestría en ingeniería presenta:

MIGUEL ANGEL GUERRERO CHEVANNIER

Directora

DRA. DORA LUZ FLORES GUTIÉRREZ

Codirectora

DRA. RAQUEL MUÑIZ SALAZAR

ENSENADA, BAJA CALIFORNIA, MÉXICO

2022

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO

MAESTRÍA Y DOCTORADO EN CIENCIAS E INGENIERÍA

“Modelo basado en *deep learning* para el diagnóstico de Tuberculosis pulmonar utilizando radiografías de tórax y perfiles clínicos”

TESIS

Que para obtener el grado de maestría en ingeniería presenta:

MIGUEL ANGEL GUERRERO CHEVANNIER

Aprobada por:

Dora Luz Flores Gutiérrez
Directora de tesis

Raquel Muniz Salazar
Codirectora de tesis

Eunice Vargas Viveros
Miembro del comité

David Cervantes Vasquez
Miembro del comité

Rafael Laniado Laborin
Miembro del comité

Ensenada Baja California, México. agosto 2022

A mi familia

Agradecimientos

Esta tesis no sería posible sin el apoyo incondicional que me brindaron mis padres, Mónica Elizabeth Chevannier Campos y Miguel Angel Guerrero Núñez, quienes me han inspirado a siempre buscar la superación personal, me han motivado a aprovechar cada oportunidad que se ha presentado y han servido de guía para ser la persona que soy el día de hoy. Mi madre, que en cada etapa de mi desarrollo me ha apoyado y motivado de sobremanera para culminar cada uno de mis proyectos. Mi padre, que ha provisto incansablemente mis necesidades y me ha brindado todas las herramientas necesarias para mi formación personal y profesional. Ambos, unos seres excepcionales a los cuales les debo cada uno de mis logros. A mi hermana, Giovana Guerrero Chevannier por ser un ejemplo de perseverancia. A mis amigos, quienes me han acompañado día a día, que han estado en los buenos momentos para celebrar y en los malos momentos brindando apoyo emocional. A mi novia Valeria, quien me acompañó en cada etapa de este proyecto y me ayudó a superar las dificultades que se han presentado.

No puedo agradecerle lo suficiente a mi directora de tesis, la Dra. Dora Luz Flores Gutiérrez, por darme la oportunidad de ser su alumno, por guiarme en cada paso de este proyecto y durante toda mi etapa universitaria. Gracias a mi codirectora de tesis, la Dra. Raquel Muñiz Salazar, por su guía, por motivarme a buscar la excelencia y un agradecimiento en especial por su paciencia. Ambas son un ejemplo para seguir y me siento muy honrado de haber tenido la oportunidad de trabajar al lado de personas tan increíbles.

Le agradezco a mi comité de tesis, la Dra. Eunice Vargas, el Dr. David Cervantes y el Dr. Rafael Laniado por su guía y consejo durante el desarrollo de esta investigación.

Finalmente le agradezco a la UABC por darme las herramientas y conocimientos que me han permitido desarrollarme profesionalmente y gracias a ello he logrado llegar hasta aquí.

Resumen

La tuberculosis (TB) es una enfermedad infecciosa causada por el complejo *Mycobacterium tuberculosis*, ésta usualmente se presenta en los pulmones, aunque puede afectar cualquier órgano del cuerpo, antes de la pandemia de COVID-19 (2020) la TB ocupaba el primer puesto en el mundo en causa de mortalidad por un único agente infeccioso. Uno de los pilares en la lucha contra la TB es un diagnóstico oportuno, se estima que un individuo con la enfermedad activa no diagnosticada puede contagiar de cinco a 15 personas por año, quienes a su vez contagiarán otros individuos [1]. Recientemente la Inteligencia Artificial ha sido utilizada para el desarrollo de sistemas de Diagnóstico Asistido por Computadora (CAD) de diversas enfermedades pulmonares mediante el análisis de imágenes biomédicas con valores de sensibilidad y especificidad que han logrado superar al ojo humano entrenado. Este trabajo de investigación tuvo como objetivo generar un modelo capaz de clasificar imágenes de radiografías de tórax en tres clases: pacientes con radiografía de tórax normal (normal), pacientes con tuberculosis pulmonar confirmada bacteriológicamente y radiografía anormal (TB positivo) y pacientes con neumonía causada por un agente distinto a TB (neumonía positiva). Para la construcción del conjunto de datos se obtuvieron datos de tres fuentes distintas; dos conjuntos de datos públicos para los grupos normal y neumonía positiva, mientras que para el grupo de pacientes TB positivo se llevó a cabo un estudio clínico en la Clínica de TB de Tijuana a lo largo de un año. Para la construcción del modelo se hizo uso de distintas redes neuronales convolucionales preentrenadas, obteniendo el mejor resultado con la red DenseNet121. Finalmente se obtuvieron valores de precisión de

94.4% en el modelo final utilizando datos clínicos del paciente además las imágenes de radiografías de tórax.

Abstract

Tuberculosis (TB) is a highly infectious disease caused by the *Mycobacterium tuberculosis* complex, usually this disease develops in the lungs, although it can affect any organ of the body, before the COVID-19 pandemic, TB ranked first in the world in cause of death by a single infectious agent. One of the pillars in the fight against TB is an early and accurate diagnosis, it is estimated that an individual with undiagnosed active disease can infect five to 15 people per year, who in turn will infect other individuals (World Health Organization 2021). Artificial Intelligence has recently been used for the development of Computer Aided Diagnosis (CAD) systems for various lung diseases through the analysis of biomedical images with sensitivity and specificity values that have managed to outperform the human eye. This research work aimed to create a model capable of classifying chest X-ray images into three classes: patients with normal chest X-ray (normal), patients with bacteriologically confirmed pulmonary tuberculosis and abnormal x-ray (TB positive), and patients with pneumonia caused by an agent other than TB (pneumonia positive) For the construction of the data set, data were obtained from three different sources; two public datasets for the groups of normal patients and positive pneumonia patients, while for the TB positive group a clinical study was carried out at the Tijuana TB Clinic for a year. For the construction of the model, different pretrained convolutional neural networks were tested, obtaining the best result with the DenseNet121 network. Finally, accuracy values of 94.4% were obtained in the final model using clinical data of the patient in addition to chest X-ray images.

Índice

Agradecimientos	iv
Resumen	vi
Abstract	viii
Índice	1
Índice de tablas	3
Índice de figuras	4
Lista de abreviaciones	5
Estructura de la tesis	6
1. Introducción	8
1.1. Antecedentes	15
1.2. Justificación	18
1.3. Hipótesis	19
1.4. Objetivo general	20
1.5. Objetivos específicos	20
2. Metodología	21
2.1 Declaración de ética y manejo de los datos	21
2.2 Criterios de inclusión y exclusión	22
2.3 Recolección de información	23
2.4 Captura de las imágenes	23
2.5 Conjunto de datos	26
2.6 Tratamiento de las imágenes	28
2.7 Preprocesamiento de los datos	30
2.8 Estructura general del modelo	32
2.9 Construcción de los modelos	34
2.10 Optimización de hiperparámetros	39
2.11 10-Fold Cross Validation	39
2.12 Métricas de evaluación	40
3. Resultados y Discusiones	45
3.1 Colecta de información y construcción del conjunto de datos	45
3.2 Comparación entre arquitecturas de red	49

3.3	Construcción del modelo clasificador	50
3.4	Resultados del clasificador con dos clases	53
3.5	Clasificador con tres clases	55
3.6	Comparación entre modelos	57
4.	Conclusión y trabajo futuro	59
5.	Referencias	61
6.	Apéndices	65

Índice de tablas

Tabla 1. Comparación de valores de sensibilidad, especificidad, farmacoresistencia y tiempo de pruebas de diagnóstico para la TB pulmonar.	10
Tabla 2. Tabla comparativa de diferentes bases de datos de imágenes de radiografías de tórax positivas a tuberculosis.	14
Tabla 3. Resultados obtenidos con distintas redes y conjuntos de datos en la clasificación de radiografías de tórax para la detección de TB pulmonar	17
Tabla 4. Criterios de inclusión y exclusión definidos para el estudio clínico.	22
Tabla 5. Conjuntos de datos utilizados para la conformación del conjunto de datos utilizado en este estudio.	27
Tabla 6. Número de elementos por clase utilizados para los conjuntos de entrenamiento, validación y prueba.	49
Tabla 7. Rendimientos alcanzados en clasificación binaria con distintas arquitecturas de CNN.	50
Tabla 8. Métricas alcanzadas por el modelo de clasificación binaria.	55
Tabla 9. Métricas alcanzadas por el modelo de clasificación multiclase.	57
Tabla 10. Comparación de métricas de desempeño de cada modelo.	58

Índice de figuras

Figura 1. Dispositivo construido para la toma de fotografías a la radiografía de tórax	25
Figura 2. Comparación de la fotografía de una radiografía antes y después del procesamiento.	29
Figura 3. Elementos generados con aumentación de datos.	31
Figura 4. Metodología general para el entrenamiento y validación de un modelo clasificador.	33
Figura 5. Diagrama de flujo de la red para el modelo de clasificación binaria.	35
Figura 6. Diagrama de flujo de la red para el modelo de clasificación multiclase.	36
Figura 7. Diagrama de flujo de la red para el modelo de clasificación multiclase con datos numéricos.	38
Figura 8. Ilustración del funcionamiento del 10-fold cross validation.	40
Figura 9. Matriz de confusión para una clasificación binaria.	41
Figura 10. Matriz de confusión para una clasificación de tres clases.	42
Figura 11. Proceso de selección y discriminación de imágenes recolectadas durante el estudio clínico.	47
Figura 12. Distribución balanceada de los datos para 2 clases; normal y TB.	48
Figura 13. Distribución balanceada de los datos para las 3 clases; normal, TB y neumonía	49
Figura 14. Distribución final de los datos para cada clase; normal, tuberculosis y neumonía.	49
Figura 15. Diagrama del clasificador multiclase.	53
Figura 16. Precisión y pérdida durante el entrenamiento del clasificador binario.	55
Figura 17. Precisión y pérdida durante la validación del clasificador binario.	55
Figura 18. Matriz de desempeño para clasificador binario.	56
Figura 19. Precisión y pérdida en durante el entrenamiento del clasificador multiclase.	57
Figura 20. Precisión y pérdida en durante la validación del clasificador multiclase.	57
Figura 21. Matriz de desempeño para clasificador multiclase.	58

Lista de abreviaciones

MTB. - *Mycobacterium tuberculosis*

TB. - Tuberculosis

IA. – Inteligencia Artificial

DL. – Deep Learning

OMS. – Organización Mundial de la Salud

CNN. – Convolutional Neural Network

CAD. – Diagnostico Asistido por Computadora

CXR. – Radiografía de Tórax

AUROC. – Area Under the Receiver Operating Characteristic

Estructura de la tesis

En el Capítulo 1: Introducción, se ofrece un preámbulo a los temas relacionados con este trabajo de tesis. Primeramente, se plantea el problema que representa la TB en salud pública a nivel mundial. Seguido de esto se abordan los distintos métodos de diagnóstico para la TB pulmonar, sus ventajas y limitaciones. Se plantea la mejora que representan los sistemas de Diagnostico Asistido por Computadora (CAD) en técnicas de diagnóstico como la radiografía. Posteriormente se establece como el aprendizaje profundo (*Deep Learning*) influye en el desarrollo de los sistemas CAD. Finalmente se establece la justificación, la hipótesis y objetivos generales y específicos del presente trabajo de investigación.

En el Capítulo 2: Metodología, se describen los métodos que se siguieron para el desarrollo de este trabajo, desde los planteamientos de ética para el manejo de datos hasta el desarrollo del modelo clasificador de radiografías.

En el Capítulo 3: Resultados, se muestra el producto de cada paso planteado en el capítulo 2. Se presenta un resumen y análisis estadístico de la población que conforma el conjunto de datos valores de evaluación del entrenamiento y validación de cada modelo generado y finalmente métricas de rendimiento de la prueba de cada modelo.

En el Capítulo 4: Discusión, se presenta un análisis de los resultados obtenidos en el capítulo previo y las implicaciones que sugiere cada uno de ellos. Se comparan los resultados de cada modelo generado y se discute las diferencias entre la implementación de datos clínicos o uso únicamente de imágenes.

En el Capítulo 5: Conclusión y trabajo futuro, se presenta la resolución final del trabajo de investigación, el cumplimiento de los objetivos planteados y se realiza un análisis del trabajo a futuro en esta línea de investigación.

1. Introducción

La tuberculosis (TB) es una enfermedad infecciosa causada por el complejo *Mycobacterium tuberculosis*, aunque es capaz de infectar cualquier órgano del cuerpo, se presenta usualmente en los pulmones, esto en un 80% de los casos. La TB representa uno de los mayores problemas de salud pública en el mundo, antes de la pandemia de COVID-19, la TB se posicionaba como la primera causa de muerte por un único agente infeccioso, superando incluso al VIH. Se estima que durante el año 2020 hubo 1.5 millones de defunciones por TB en el mundo, lo cual representa un incremento de 100,000 individuos en comparación al año 2019, donde se presentaron 1.4 millones de defunciones [1].

A pesar de la alta mortalidad que presenta, la TB es una enfermedad curable y prevenible con un diagnóstico oportuno y un tratamiento correcto, cerca de un 85% de los casos de los individuos que desarrollan la enfermedad pueden ser curados con éxito siguiendo un régimen estricto de fármacos antituberculosis durante seis meses [1].

El diagnóstico de la TB se realiza utilizando diferentes técnicas como la microscopia, cultivo microbiológico, técnicas moleculares e imagenología. Existen distintos métodos de diagnóstico avalados y recomendados por la Organización Mundial de la Salud para el diagnóstico de la TB pulmonar, estos pueden categorizarse por la tecnología que se emplea para el desarrollo de la prueba; microscopía, microbiología, molecular e imagenología. Actualmente la OMS recomienda métodos moleculares rápidos para el diagnóstico inicial de TB, por otro lado ya no se recomienda el uso de la microscopia como método de diagnóstico para la TB pulmonar [2].

El cultivo es una prueba microbiológica que era considerada como el estándar de oro para el diagnóstico de la TB pulmonar, esta consiste en depositar una muestra en un medio líquido durante un máximo de seis semanas para detectar la presencia de bacterias, esta prueba tiene la ventaja de mostrar la farmacorresistencia a todos los fármacos en la muestra, sin embargo, el tiempo de resultados se relaciona directamente con la carga bacteriana de la muestra, teniendo un tiempo de resultados de siete a 30 días [3].

En 2013, la OMS recomendó pruebas moleculares rápidas para TB y resistencia a la rifampicina como la prueba de diagnóstico inicial de TB para reemplazar la técnica de microscopía, reforzando esta recomendación en 2020 [4]. El *Xpert MTB/RIF* es la prueba molecular recomendada por la OMS con una sensibilidad del 90% en comparación con la microscopía de frotis con una sensibilidad promedio del 50%, estas pruebas tienen la ventaja de mostrar farmacorresistencia a la rifampicina y un tiempo promedio de resultados de dos horas. La desventaja principal de estas pruebas es el uso de cartuchos para su funcionamiento. [4].

Entre las pruebas de diagnóstico por imagenología se encuentra resonancia magnética por emisión de positrones , la cual resulta una opción no viable por su elevado costo y la radiografía de tórax, la cual es ha mantenido como técnica de tamizaje de la TB pulmonar por su costo relativamente bajo en comparación con otras pruebas, alta disponibilidad en centros de atención médica y un tiempo de resultados de aproximadamente 15 minutos, lo cual resulta bajo en comparación con otras pruebas [5]. La sensibilidad de una prueba se refiere a la capacidad de clasificar correctamente como enfermo a un individuo que presenta la enfermedad, mientras que la especificidad es la capacidad de la prueba de

clasificar como sano a un individuo que no presenta la enfermedad. En TB pulmonar, la radiografía de tórax tiene una sensibilidad reportada de 92%, mientras que la especificidad se calcula en un 63%, esto resulta en una sensibilidad alta y una especificidad baja en comparación de otras pruebas de diagnóstico (Tabla 1). Esto conlleva a una alta tasa de detección de anomalías en la prueba, con una baja diferenciación entre patologías que muestran patrones radiológicos similares.

Tabla 1. Comparación de valores de sensibilidad, especificidad, farmacoresistencia y tiempo de pruebas de diagnóstico para la TB pulmonar.

Prueba	Sensibilidad	Especificidad	Susceptibilidad a farmacoresistencia	Tiempo	Referencia
Radiografía de tórax	92%	63%	Sin susceptibilidad	15 minutos	[6], [7]
GeneXpert MTB/RIF	91%	100%	Rifampicina, Isoniacida, Fluoroquinolonas, inyectables de segunda línea y Etionamida	2 horas	[6], [4]
Line Probe Assay	99%	98%	Rifampicina, Isoniacida, Fluoroquinolonas e inyectables	24 horas	[6]
Cultivo líquido MGIT	100%	94%	Todos	7–30 días	[6], [3]
Microscopía de frotis	60%	96%	Sin susceptibilidad	1 día	[6], [8]

La radiografía de tórax presenta la ventaja de ser capaz de detectar TB pulmonar incluso antes de que se presenten los síntomas característicos, esta es una etapa de la TB activa denominada subclínica, donde la carga bacteriana es relativamente baja. La detección de la TB activa en la etapa subclínica juega un papel importante en la reducción de la transmisión, así como la morbilidad y mortalidad causada por la TB. La radiografía de

tórax se ha establecido como una prueba de tamizaje, esto quiere decir que se utiliza en individuos que en un principio son considerados sanos, para detectar a estos que no lo son. En los últimos años se han realizado mejoras importantes en la portabilidad de los equipos de rayos X, existen equipos aprobados por la FDA con un peso menor a 2 kg, lo cual los vuelve accesibles a zonas remotas donde no existen otras pruebas de diagnóstico [9].

En la práctica, la radiografía de tórax es examinada e interpretada por un radiólogo, por lo que el proceso resulta subjetivo a la experiencia del personal médico, existen distintas patologías que muestran patrones radiológicos similares a los de la TB, lo cual podría llevar a un diagnóstico erróneo de la enfermedad. Uno de los problemas del uso de la radiografía de tórax es la escasez de radiólogos capacitados disponibles en áreas de bajos recursos, esto en conjunto con la ausencia de otras técnicas de diagnóstico juega un papel importante en la prevalencia y dispersión de la enfermedad [10], [11], [12].

La inteligencia artificial es una rama de las ciencias computacionales que tiene como objetivo imitar comportamientos humanos inteligentes. Uno de los requisitos básicos para que se generen estos comportamientos inteligentes es el aprendizaje, de este fundamento surge el *machine learning* (ML) o *aprendizaje de máquina*, técnica que se desarrolló inicialmente para el análisis y reconocimiento de patrones en conjuntos de datos médicos [13]. El aprendizaje generalmente comienza con el sistema de algoritmo que computa las características de la imagen que se consideran más importantes para realizar la predicción o el diagnóstico de interés. El algoritmo de aprendizaje identifica la mejor combinación de estas características de imágenes para clasificar la imagen o calcular alguna métrica para la región de interés dada [14].

El *deep learning* (DL) o aprendizaje profundo es una rama del *machine learning*, que es un término general que describe los algoritmos de aprendizaje. El algoritmo que sustenta todos los métodos de DL es la red neuronal, en este caso, construida con muchas capas ocultas. Estas redes se pueden construir de distintas maneras con diferentes tipos de capas y la construcción general de una red se conoce como su arquitectura. En la década de 1980, las redes que utilizan capas convolucionales se introdujeron por primera vez para el análisis de imágenes, y la idea se formalizó en los años siguientes [15]. Estas capas convolucionales ahora forman la base de todas las tareas y análisis de imágenes de aprendizaje profundo, casi sin excepción. Las capas convolucionales usan neuronas que se conectan solo a un pequeño campo receptivo de la capa anterior. Estas neuronas se aplican a diferentes regiones de la capa anterior, operando como una ventana deslizante sobre todas las regiones y detectando efectivamente el mismo patrón local en cada ubicación. De esta forma, se conserva la información espacial y se comparten los pesos aprendidos [5]. La disponibilidad de grandes conjuntos de datos de imágenes biomédicas ha despertado el interés de la aplicación de ML para el desarrollo de sistemas de Diagnóstico Asistido por Computadora (DAC), se han desarrollado numerosos sistemas DAC basados en técnicas de DL, este ha demostrado tener buenos resultados para la clasificación y predicción de enfermedades en imágenes biomédicas de ultrasonido, rayos x, tomografías, entre otras [16], [17].

En marzo de 2021, la OMS recomendó el uso de sistemas DAC en la interpretación de radiografías en lugar de interpretación humana para tamizaje de TB pulmonar [18], estos sistemas reducen las limitaciones que tiene la radiografía de tórax como técnica de diagnóstico por sí sola. Los sistemas DAC tiene el objetivo de identificar signos

radiológicos anormales en una etapa temprana que un profesional clínico encuentre difícil de identificar, estos han sido utilizados con éxito para la detección de patologías como cáncer de mama, cáncer pulmonar, cáncer de colon, cáncer de próstata, metástasis al hueso, enfermedad de las arterias coronarias, defectos cardíacos congénitos, detección patológica del cerebro, enfermedad de Alzheimer, retinopatía diabética, entre otros [19]–[22].

La aplicación de Redes Neuronales Convolucionales (CNN) en el análisis de radiografías de tórax para la detección de patologías pulmonares ha demostrado tener buenos resultados [23]. En respuesta a la pandemia de COVID-19, se desarrollaron modelos capaces de detectar la infección de SARS-Cov-2, en una radiografía de tórax con valores de precisión mayores al 95% [24]–[26]. Otros estudios enfocados en la detección de TB en imágenes de radiografías de tórax han logrado precisiones superiores al 95% [23].

Una de las barreras en la rama de la clasificación de imágenes biomédicas resulta de la aún limitada disponibilidad de bases de datos, en adición, la mayoría de las bases de datos que se encuentran publicadas se limitan únicamente al alojamiento de imágenes de radiografías de tórax, sin incluir otra información clínica (Tabla 2).

Tabla 2. Tabla comparativa de diferentes bases de datos de imágenes de radiografías de tórax positivas a tuberculosis.

Conjunto de datos	Tamaño	Fuente	Etiquetas	Referencia
RSNA Pneumonia Challenge	5,863 CXR frontales y laterales	RSNA/Kaggle	normal y neumonía	[27]
JSRT	247 CXR frontales	Japanese Society of Radiological Technology	154 casos con nódulos (100 malignos, 54 benignos) y 93 casos normales	[28]
Montgomery	138 CXR frontales	Montgomery County Department of Health	58 casos de TB, 80 casos normales	[29]
Shenzhen	662 CXR frontales	Shenzhen No.3 People's Hospital, Guangdong Medical College, Shenzhen, China	336 casos de TB, 326 normales	[29]
KIT dataset	10,848 CXR	Korean Tuberculosis Institute bajo la Korea Association of Tuberculosis	7020 casos normales	[30]
Indiana dataset	8,121 CXR de 3,996 pacientes	Indiana Network for Patient Care con varios hospitales afiliados con la Universidad Escuela de Medicina de Indiana	patologías como cardiomegalia, edema pulmonar, opacidades y efusión pleural	[31]
RIH-CXR	17,202 CXR frontales	Hospital de Rhode Island	normales y anormales	[32]
CheXpert	224,316 CXR frontales y laterales de 95,240 pacientes	Hospital de Stanford	14 observaciones en reportes radiológicos	[33]
MIMIC-CXR	473,057 CXR de 63,478 pacientes	MIT Beth Israel Deaconess Medical Center	227,943 estudios etiquetados en 14 patologías	[34]
Chest-XRay8	108,948 CXR frontales de 30,805 pacientes	Extraído de la base de datos del sistema de comunicación de archivo de imágenes clínicas en hospitales afiliados al National Institutes of Health Clinical Center	14 patologías incluyendo atelectasia, consolidación, infiltración neumotórax, edema, enfisema, fibrosis, efusión, neumonía, engrosamiento pleural, cardiomegalia, nódulos, masas y hernias	[35]
ChestXray14	112,120 CXR frontales de 32,717 pacientes	National Institutes of Health Clinical Center		[35]
PadChest	160 868 CXR	Hospital Universitario de San Juan	etiquetas para 19 patologías	[36]

Las investigaciones recientes enfocadas en la detección de TB pulmonar por medio del análisis de radiografías de tórax con el uso de IA se enfocan en probar distintas arquitecturas de CNNs utilizando los mismos conjuntos de imágenes, la generación de nuevas bases de datos de imágenes, que además incluya datos demográficos y clínicos de los pacientes para el entrenamiento de las CNNs, podría representar una mejoría en los resultados del clasificador [37].

Este estudio propone contribuciones importantes en el área de la clasificación de imágenes radiografías de tórax de TB. La primera es la generación de una nueva base de datos que además de incluir radiografías de tórax, incluye variables demográficas y datos clínicos de los pacientes participantes, que pueda ser utilizada tanto en el estudio presente como en futuras investigaciones. Por otro lado, se contempla la utilización de la base de datos generada anteriormente para la prueba de distintas arquitecturas de CNNs.

1.1. Antecedentes

Se han empleado distintos métodos para la clasificación de imágenes de radiografías de tórax para la detección de TB pulmonar, estos implican un preprocesamiento de las imágenes y en conjunto con modelos de *deep learning*. Se han obtenido resultados superiores al 95% de precisión, lo cual supera los valores reportados que puede alcanzar el personal radiológico entrenado. Durante el año 2016, se realizó un estudio mostrado en [38] en el cual se llevó a cabo una revisión del único software para la detección de TB por medio de radiografías de tórax disponible hasta la fecha para nombrado CAD4TB,

este trabajo mostró que el software es capaz de alcanzar un desempeño de hasta 0.84 de Área Bajo la Curva Característica Operativa del Receptor (AUROC).

La exploración del uso de CNN para la clasificación de radiografías de tórax comenzó alrededor de 2017, en [3][39][6] se muestra un estudio donde se empleó más de 1,000 imágenes con las CNN *AlexNet* y *GoogLeNet*, obteniendo como resultado un 0.99 AUROC al implementar un ensamble de ambas redes. En el estudio [40] llevado a cabo en el año 2018 se implementó 138 imágenes en distintos experimentos para probar el desempeño de las redes VGG19, InceptionV3, ResNet50, DenseNet121, InceptionResNetV2, obteniendo el mejor resultado de 0.9213 AUROC con la red VGG19. En este mismo año en el estudio [41] se utilizó 874 imágenes con 14 etiquetas y la red Quore.AI, obteniendo un resultado de 0.929 AUROC en esta clasificación multiclase.

En 2019, se llevó a cabo un estudio [42] implementando 1,000 imágenes con una CNN personalizada donde obtuvo 0.925 AUROC, mientras que en el estudio [43] del mismo año, se hizo uso de la red DenseNet121 con más de 800 imágenes obteniendo un resultado de 0.937 AUROC. En el estudio [44] llevado a cabo en el 2018, se hizo una comparación de métodos tradicionales de ML contra CNN, donde probó las redes *AlexNet*, *VGG16*, *GoogLeNet*, *ResNet50* con 2,000 imágenes de entrenamiento, encontrando que las CNNs superaban a los métodos tradicionales, el mejor resultado fue con la red *AlexNet* con 0.95 AUROC, posteriormente el mismo autor en el año 2020, realizó el estudio [45], donde se implementó más de 10,000 imágenes con las CNNs *VGG16*, *InceptionV3*, *inceptionResNetV2*, *Xception*, *DenseNet12*, explorando un ensamble con las redes *InceptionResNetV2*, *InceptionV3*, y *DenseNet121* con el que obtuvo un 0.995 AUROC.

Tabla 3. Resultados obtenidos con distintas redes y conjuntos de datos en la clasificación de radiografías de tórax para la detección de TB pulmonar

Autor	Imágenes	Mejor Resultado	Resultado	Referencia
Lakhani 2017	1,000	Ensamble AlexNet y GoogLeNet	0.99 AUROC	[39]
Becker 2018	138	VGG19	0.9213 AUROC	[40]
Singh 2018	874	Quore.AI	0.929 AUROC	[41]
Pasa 2019	1,000	CNN personalizada	0.925 AUROC	[42]
Gozes 2019	800	DenseNet121	0.937 AUROC	[43]
Rajaraman 2018	2,000	AlexNet	0.95 AUROC	[44]
Rajaraman 2020	10,000	Ensamble InceptionResNetV2, InceptionV3, y DenseNet121	0.995 AUROC	[45]
Heo 2019	2,000	VGG19	0.9213 AUROC	[37]
Karki 2021	135	CNN personalizada	0.66 precisión	[46]

Existen otras metodologías menos exploradas, tal es el caso del estudio [37], donde en 2019 se implementaron 2,000 imágenes además de incluir el uso de datos demográficos (peso, edad, género y altura) para el entrenamiento del modelo, obteniendo un 0.9213 AUROC con la red VGG19. Otro enfoque poco explorado fue el presentado en el estudio [46], donde se propuso la identificación de farmacoresistencia analizando únicamente radiografías, con lo que obtuvo un 66% de precisión utilizando una CNN, tomando la forma y textura de la imagen como características relevantes.

1.2. Justificación

La radiografía de tórax es una prueba de diagnóstico establecida para la detección y tamizaje de la TB pulmonar, esta tiene la ventaja de ser rápida y a un costo relativamente bajo en comparación de otras pruebas. La principal desventaja que presenta esta prueba es la baja especificidad al ser analizada por el personal de salud, sin embargo, se ha demostrado que implementando un sistema de diagnóstico asistido por computadora es posible alcanzar altos valores de sensibilidad y especificidad. Una de las aplicaciones más relevantes del *deep learning* es el desarrollo de estos sistemas de diagnóstico asistido por computadora utilizando grandes conjuntos de imágenes biomédicas. Una opción poco explorada es la implementación de datos clínicos además de imágenes, por lo que la creación de un conjunto de imágenes con datos clínicos relacionados podría representar una herramienta importante para mejorar los resultados de los sistemas CAD.

1.3. Hipótesis

Las técnicas de *deep learning* son una herramienta que permitirá la identificación de patrones radiológicos como apoyo para el diagnóstico de tuberculosis pulmonar.

1.4. Objetivo general

Desarrollar un modelo basado en técnicas de *deep learning* para la asistencia de la interpretación y diagnóstico de la tuberculosis, haciendo uso imágenes de radiografías de tórax y datos clínicos de pacientes.

1.5. Objetivos específicos

1. Evaluar técnicas de *deep learning* a utilizar para el desarrollo del método computacional.
2. Ajustar los hiperparámetros para generar un modelo computacional capaz de analizar y clasificar imágenes de radiografías de tórax.
3. Validar el modelo computacional utilizando datos independientes a los empleados para el desarrollo del modelo.

2. Metodología

2.1 Declaración de ética y manejo de los datos

Para llevar a cabo el presente estudio se realizó la construcción de un conjunto de datos que incluye radiografías de tórax e información clínica de pacientes divididos en tres grupos: grupo TB conformado por pacientes con diagnóstico positivo a TB pulmonar, grupo neumonía que incluye pacientes con diagnóstico positivo a neumonía y por último pacientes que no presentan anomalías en radiografía de tórax y prueba de diagnóstico, denominado grupo normal. Los datos correspondientes a los grupos neumonía y normal fueron obtenidos de conjuntos de datos públicos, el uso y manejo de estos datos se llevó a cabo conforme al acuerdo de uso para investigación estipulado por el autor de cada conjunto. La obtención de los datos del grupo TB se llevó a cabo con un estudio clínico en la Clínica de Tuberculosis de Tijuana (CTT), el estudio fue sometido y aprobado por el Comité de Bioética del Hospital General de Tijuana. A cada paciente participante se le informó el propósito del estudio, los métodos de manejo de su información y se brindó un consentimiento informado por escrito, con el cual se autoriza la colecta y manejo de los datos. Adicionalmente, los registros de los participantes fueron anonimizados previo al análisis de la información.

2.2 Criterios de inclusión y exclusión

Para cada uno de los grupos se definieron criterios de inclusión y exclusión en la integración de elementos al conjunto de datos, los cuales se describen en la Tabla 4.

Tabla 4. Criterios de inclusión y exclusión definidos para el estudio clínico.

Grupo	Criterios de inclusión	Criterios de exclusión
Tuberculosis	<ul style="list-style-type: none"> • Pacientes que hayan aceptado y firmado el consentimiento informado. • Pacientes con confirmación fenotípica y/o molecular del diagnóstico de TB. • Pacientes con radiografía de tórax con hallazgos de TB. • Pacientes que cuenten con expediente clínico. 	<ul style="list-style-type: none"> • Pacientes que no hayan aceptado participar en el estudio. • Pacientes sin radiografía de tórax o confirmación fenotípica y/o molecular del diagnóstico de TB. • Pacientes sin información clínica suficiente.
Normal	<ul style="list-style-type: none"> • Pacientes con radiografía de tórax sin hallazgos anormales. • Pacientes con prueba fenotípica y/o molecular negativa a TB. • Pacientes con información clínica común al grupo TB. 	<ul style="list-style-type: none"> • Pacientes con diagnóstico positivo a alguna enfermedad pulmonar. • Pacientes sin radiografía de tórax. • Pacientes sin información clínica suficiente.
Neumonía	<ul style="list-style-type: none"> • Pacientes con radiografía de tórax con hallazgos de neumonía. • Pacientes con información clínica común al grupo TB. 	<ul style="list-style-type: none"> • Pacientes con síntomas de neumonía por TB. • Pacientes sin radiografía de tórax. • Pacientes sin información clínica suficiente.

Para el grupo TB inicialmente se recolectó la información de todos los pacientes que aceptaron y firmaron el consentimiento de participación del estudio. Posteriormente en una segunda fase de selección se incluyeron únicamente los datos que cumplen estrictamente con los criterios de inclusión. Para los grupos normal y neumonía, se realizó una discriminación de los datos conforme a los criterios de inclusión y exclusión estipulados.

2.3 Recolección de información

Para el grupo de TB, la radiografía de tórax, información clínica y pruebas de laboratorio de los pacientes fueron recolectadas por medio de un estudio clínico llevado a cabo en la Clínica de TB del Hospital General Tijuana, Baja California, México. La información que se consideró de relevancia para la construcción de la base de datos consistió en datos generales del paciente, comorbilidades, toxicomanías, tratamientos previos, información de pruebas de diagnóstico, información de resistencia a antibióticos, química sanguínea e información de tratamiento actual. La información fue recuperada manualmente del expediente clínico del paciente, reemplazando el nombre del paciente con un identificador numérico para salvaguardar el anonimato. Se generó una matriz con los datos resultantes en un archivo XLSX y se almacenó en Google Drive.

2.4 Captura de las imágenes

En este estudio se consideró únicamente radiografías de tórax tomadas con técnica postero anterior. La totalidad de las radiografías se encontraron en formato físico, por lo que se realizó la toma de una fotografía al estudio, dicha captura de imágenes fue llevada a cabo por médicos prestadores de servicio social de la Clínica de TB de Tijuana. El

proceso de captura de imágenes se realizó utilizando cámaras de equipos telefónicos de uso cotidiano.

Se solicitó que la captura de las imágenes se hiciera con una resolución mínima de 5 MP, postrando la radiografía sobre un negatoscopio y colocando la cámara a una distancia fija de 45cm paralela al plano del negatoscopio. Con el propósito de estandarizar el proceso de la captura de imágenes y reducir la variabilidad en las tomas se desarrolló un dispositivo para facilitar la captura de las imágenes, este se conforma de una pantalla retroiluminada con función similar al de un negatoscopio con un soporte para la cámara que permite la toma a una distancia fija de 45 cm (Figura 1). Todos los elementos que no cumplieran las características mínimas solicitadas fueron descartados del estudio.



Figura 1. Dispositivo construido para la toma de fotografías a la radiografía de tórax.

2.5 Conjunto de datos

El conjunto de datos utilizado para el entrenamiento del modelo se conformó por los datos recabados del estudio clínico de la Clínica de Tuberculosis de Tijuana para el grupo denominado TB y dos conjuntos de datos públicos para los grupos normal y neumonía como se muestra en la Tabla 5. El conjunto de datos seleccionado para el grupo normal fue recuperado del Instituto Nacional de Salud de EE. UU., este consiste en 662 radiografías postero anteriores de tórax y anotaciones clínicas de pacientes sin hallazgo de enfermedades y pacientes positivos a TB, para la selección del conjunto de entrenamiento se discriminaron a todos los pacientes con diagnóstico positivos a TB. Para el conjunto de datos del grupo neumonía se utilizó el conjunto de datos público *CheXpert* [26] el cual ha sido publicado en el sitio del *Stanford ML Group* para investigación, este consta de 224,316 imágenes de radiografías de tórax postero anteriores, datos del paciente y anotaciones para 13 hallazgos con 3 etiquetas: positivo, negativo e incierto. Para la conformación del conjunto de entrenamiento se realizó un filtro de los datos, donde se seleccionaron únicamente elementos con la etiqueta positiva a la anotación de neumonía, excluyendo las etiquetadas como inciertas o marcadas con cualquier otra anotación. Para la conformación final del conjunto de entrenamiento se utilizó el 85% datos recabados para la clase TB, para el grupo neumonía se seleccionaron aleatoriamente el mismo número de elementos que en la clase TB, mientras que para el grupo normal se seleccionaron aleatoriamente el mismo número de elementos que la clase TB y la clase neumonía. Para el conjunto de prueba se tomó el 15% de los datos restantes del grupo TB, mientras que para los grupos normal y neumonía se tomaron muestras aleatorias de datos de tamaño igual al grupo TB.

Tabla 5. Conjuntos de datos utilizados para la conformación del conjunto de datos utilizado en este estudio.

Grupo	Fuente	Tamaño	Etiquetas	Cita
Tuberculosis	Clínica de Tuberculosis de Tijuana.	138 radiografías de tórax con técnica anteroposterior y datos clínicos.	TB positiva	Este trabajo
Normal	Conjunto de datos NLM por el Instituto Nacional de Salud de EE. UU.	662 radiografías de tórax con técnica anteroposterior y datos del paciente.	TB positivo y normal.	[22]
Neumonía	Conjunto de datos CheXpert por el Stanford ML Group.	224,316 radiografías de tórax con técnica anteroposterior y datos del paciente.	Sin hallazgos, engrandecimiento, cardiomegalia, lesión, opacidad, edema, consolidación, neumonía, atelectasia, neumotórax, efusión pleural, fractura, dispositivo de soporte y otros con etiquetas: positivo, negativo e incierto.	[26]

2.6 Tratamiento de las imágenes

Las imágenes obtenidas de la Clínica de TB de Tijuana fueron procesadas manualmente utilizando el software de fuente abierta *GIMP* [47]. Debido a que las imágenes fueron tomadas en diferentes resoluciones, estas fueron redimensionadas a un tamaño estándar 1280 x 800 píxeles, posteriormente se ajustaron los bordes de la imagen para que mostrara únicamente los bordes de la radiografía y se eliminaron otras figuras no correspondientes a la zona de interés. Posteriormente se realizó una anonimización de la imagen, donde se eliminó nombre y anotaciones en el estudio que pudieran relacionar al paciente con la imagen. A continuación, se realizó una conversión de formato RGB a escala de grises; El proceso completo del tratamiento de la imagen puede observarse en la Figura 2.

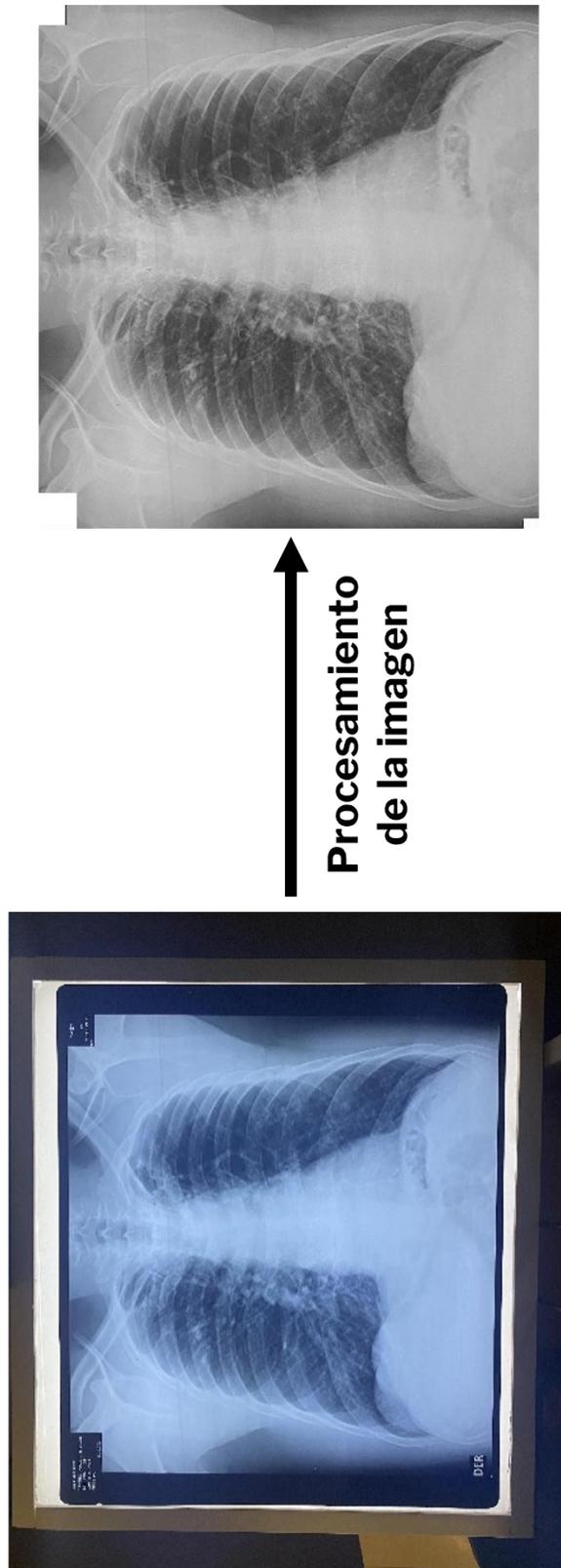


Figura 2. Comparación de la fotografía de una radiografía antes y después del procesamiento.

2.7 Preprocesamiento de los datos

El preprocesamiento previo al entrenamiento de las redes se llevó a cabo utilizando la librería de *Keras v.2.8*, se realizó una normalización de los datos redimensionando todas las imágenes a una dimensión de 128X128 píxeles y un reescalado de 1/255. Al tratarse de un conjunto de datos con un número limitado de elementos, se aplicaron distintas técnicas de aumentación de datos como el estiramiento, rotación, translación, corte y otras deformaciones de forma aleatoria. Esto permite generar múltiples elementos que el modelo reconocerá como nuevos a partir de un número limitado de elementos de entrenamiento, consiguiendo un proceso de aprendizaje más robusto. El proceso de aumentación de datos se llevó a cabo con la librería de aumentación de datos de *Keras*, en la Figura 3 puede observarse una muestra de 12 elementos productos de aumentación de datos con su diagnóstico correspondiente, siendo [1,0,0] para TB, [0,1,0] para normal y [0,0,1] para neumonía.

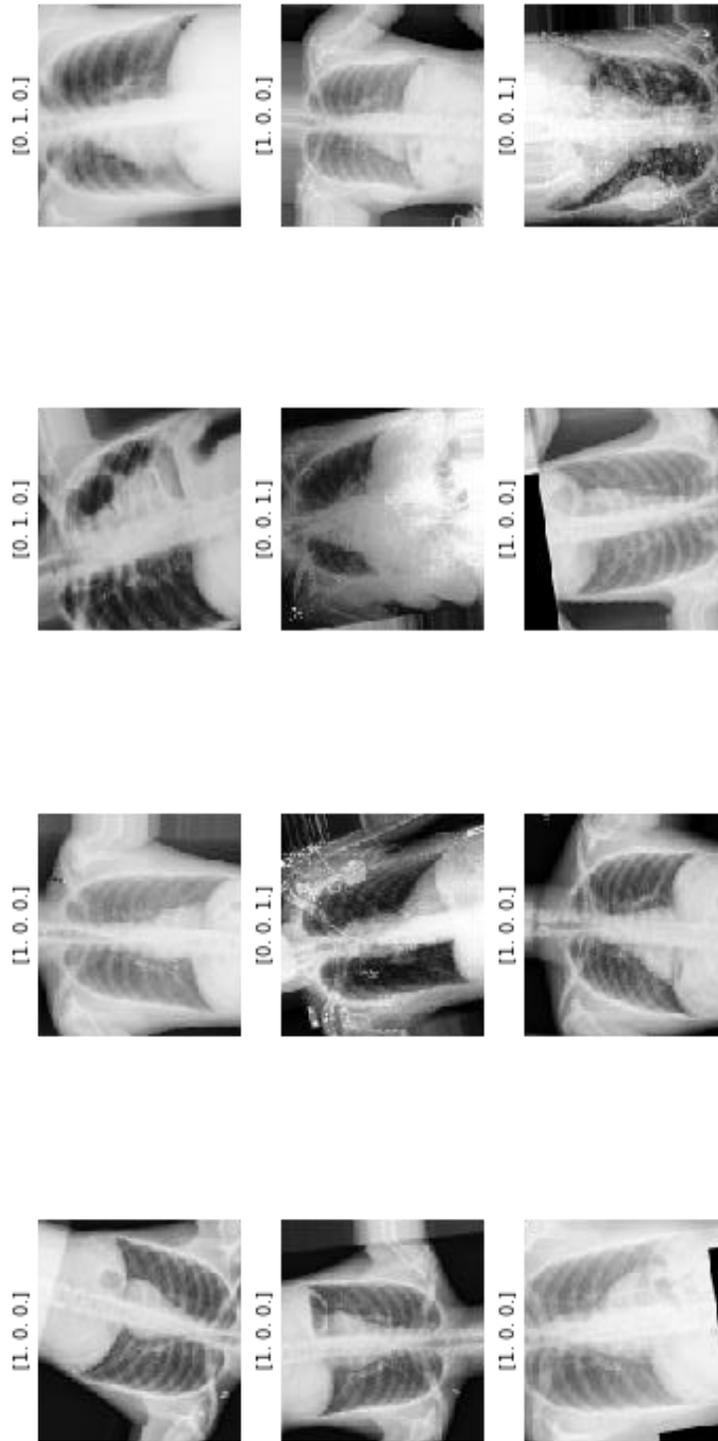


Figura 3. Elementos generados con aumentación de datos, la etiqueta superior corresponde al diagnóstico de la imagen; [1,0,0] para normal, [0,1,0] para TB y [0,0,1] para neumonía.

2.8 Estructura general del modelo

Para la construcción de todos los modelos se siguió la metodología que se muestra en la Figura 4. Se comenzó realizando un análisis exploratorio para conocer la naturaleza y dimensión del conjunto de datos, después se aplicaron las técnicas de preprocesamiento descritas en el punto 2.6, el porcentaje de elementos destinados para los conjuntos de entrenamiento, validación y prueba fueron uniformes en cada experimento.

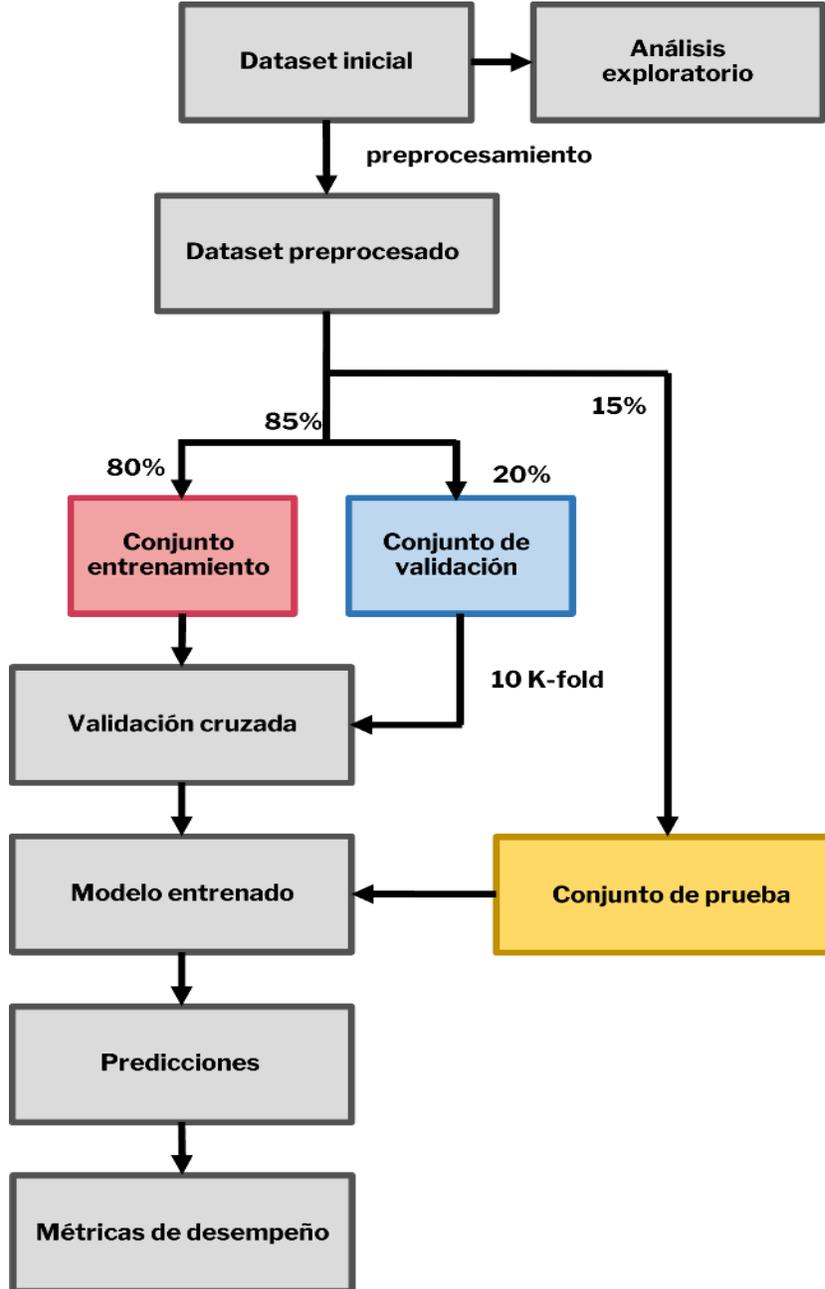


Figura 4. Metodología general para el entrenamiento y validación de un modelo clasificador.

Se tomó una muestra aleatoria de la totalidad de los datos (15%), manteniendo una proporción equitativa entre clases, esta muestra conforma el conjunto de prueba, este conjunto es independiente a cualquier proceso de entrenamiento, con lo cual se asegura la calidad de las predicciones. El 85% de los datos restantes fueron destinados a los conjuntos de entrenamiento y validación, en porcentajes de 80 y 20% para cada conjunto respectivamente. Debido al limitado número de elementos en el conjunto de datos se optó por usar *k-fold cross validation*, finalmente se realizaron predicciones con el conjunto de prueba, con base en el resultado de este se obtuvieron las métricas de desempeño.

2.9 Construcción de los modelos

Para los modelos se implementó una red neuronal convolucional que fue entrenada previamente con la base de datos *ImageNet*; una base de imágenes que aloja más de 14 millones de imágenes naturales en más de 20,000 categorías y que es ampliamente utilizada para el entrenamiento de redes neuronales para el reconocimiento de objetos. Se seleccionaron las redes *VGG16*, *AlexNet*, *DenseNet121*, *DenseNet169*, *DenseNet201*, *InceptionV3* y *ChexNet* como posibles candidatos para la generación del modelo. Con el objetivo de comparar entre el uso de datos clínicos, se generaron tres modelos distintos durante el desarrollo del experimento.

2.8.1. Modelo de clasificación binaria

El primer modelo generado utilizó únicamente imágenes divididas en dos clases (TB y normal) como elementos de entrada, esto quiere decir que solo habrá dos posibles resultados en la clasificación, como se observa en la Figura 5.

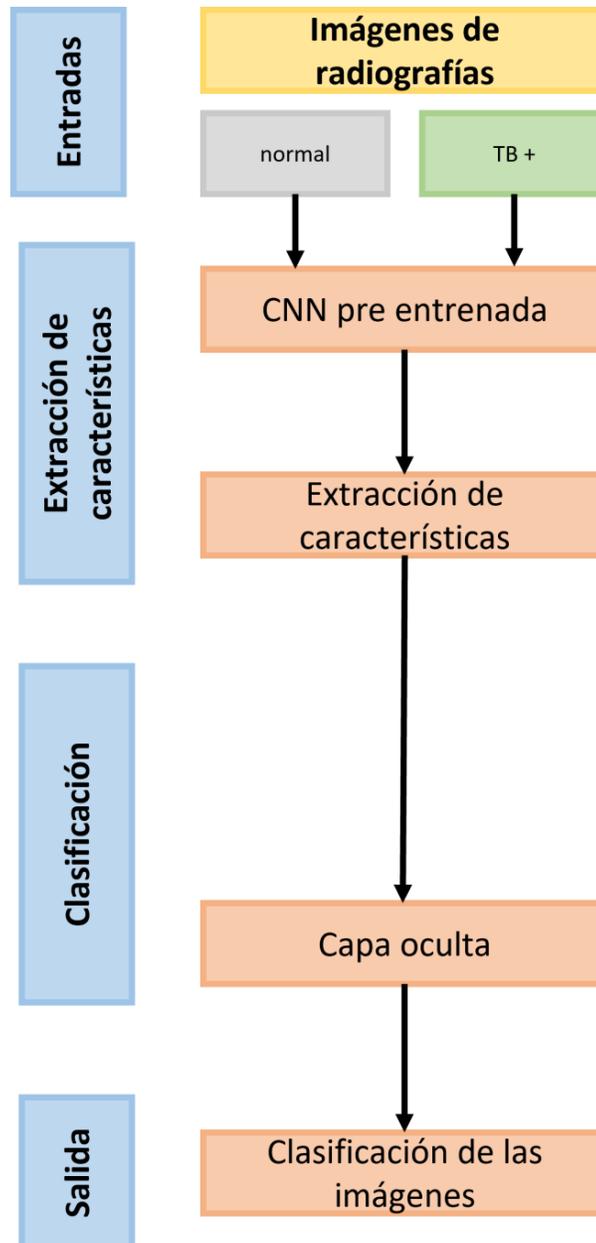


Figura 5. Diagrama de flujo de la red para el modelo de clasificación binaria.

2.8.2. Modelo de clasificación multiclase

El segundo modelo utilizó imágenes divididas en tres clases (TB, neumonía y normal) como variables de entrada (Figura 6), la estructura es similar al modelo de clasificación binaria.

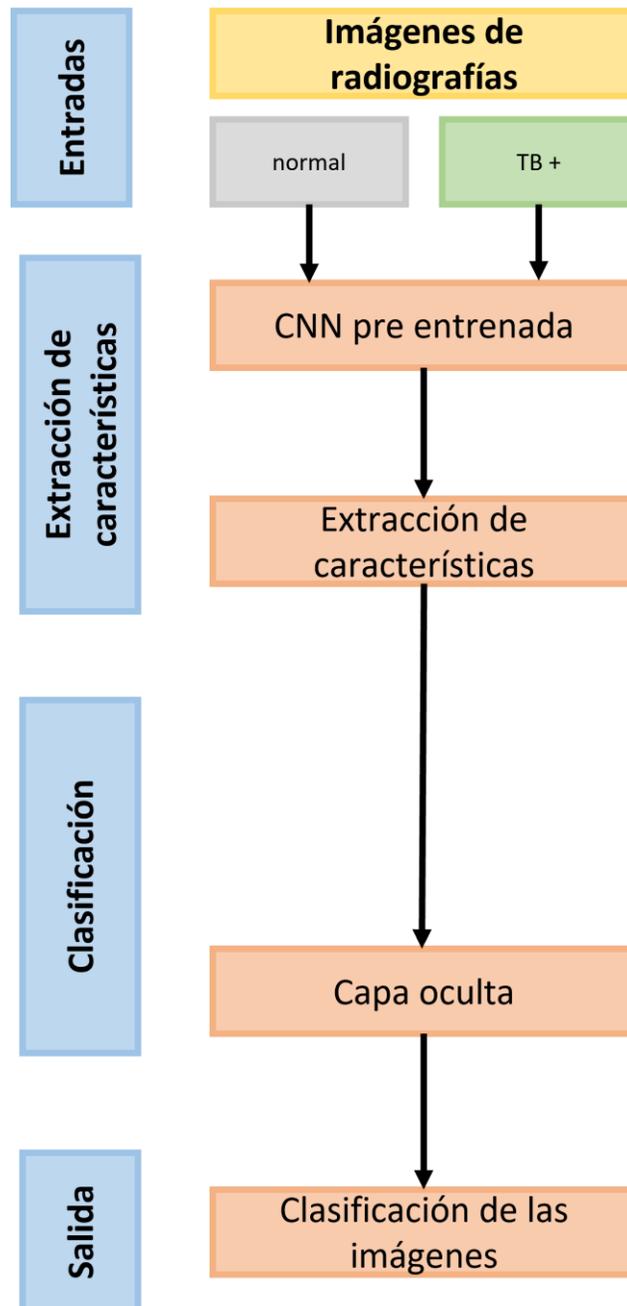


Figura 6. Diagrama de flujo de la red para el modelo de clasificación multiclase.

2.8.3. Modelo de clasificación multiclase con datos numéricos.

Por último, el tercer modelo utilizó imágenes divididas en tres clases además de datos relacionados del paciente. El proceso de extracción de características es el mismo que en los otros modelos para las imágenes, para los datos numéricos la extracción de características funciona como un proceso independiente, uniendo el resultado de cada uno de estos procesos en una capa denominada concatenación. Posteriormente, la clasificación se realiza en una capa oculta, obteniendo como resultado la clasificación en tres categorías correspondientes a las etiquetas de entrada (Figura 7).

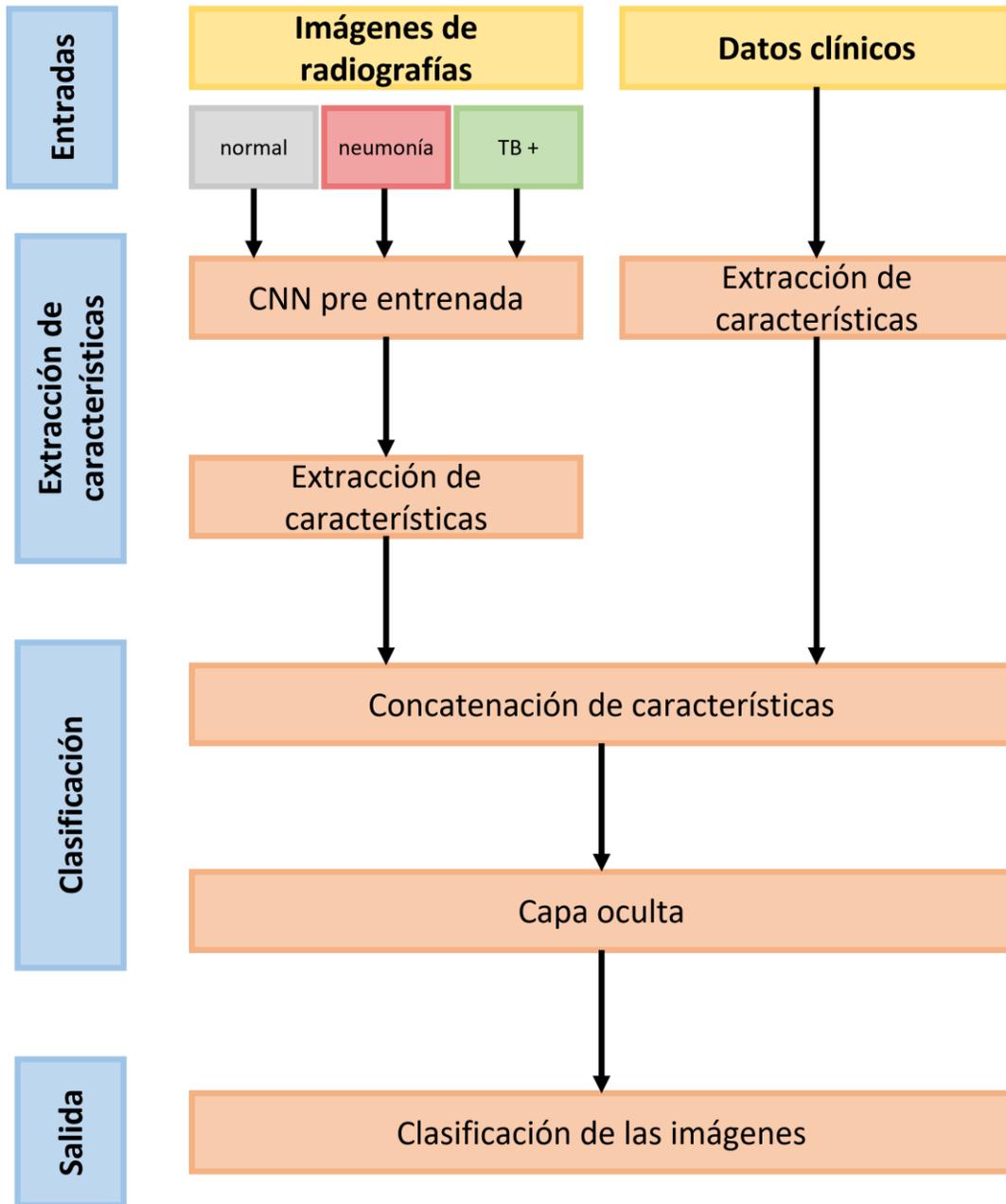


Figura 7. Diagrama de flujo de la red para el modelo de clasificación multiclase con datos numéricos.

2.10 Optimización de hiperparámetros

En *machine learning*, un hiperparámetro es un parámetro o variable cuyo valor permite controlar el proceso de aprendizaje, el cambio de estos valores puede afectar directamente los resultados alcanzados y el tiempo de entrenamiento. Idealmente se busca llegar al valor mínimo de pérdida en el conjunto de prueba salvaguardando los recursos implementados para el entrenamiento, cada modelo requiere valores distintos de hiperparámetros para alcanzar su máximo potencial. El proceso de optimización de estos hiperparámetros fue realizado de manera semiautomática.

2.11 10-Fold Cross Validation

El proceso de entrenamiento de una CNN implica un cierto grado de aleatoriedad, esto quiere decir que el resultado puede variar entre entrenamientos, incluso utilizando los mismos parámetros. Una forma de reducir esta variabilidad de los resultados es el *K-fold cross validation*, este proceso se lleva a cabo dividiendo el conjunto de entrenamiento en K número de conjuntos, posteriormente en cada entrenamiento se utiliza una combinación diferente para los conjuntos de entrenamiento y validación. El resultado final será el promedio de los resultados de cada corrida como se muestra en la Figura 8. Al implementar *10-fold cross validation* fue posible asegurar que la totalidad de los datos fueran utilizados en el proceso de entrenamiento.



Figura 8. Ilustración del funcionamiento *del 10-fold cross validation*.

2.12 Métricas de evaluación

Para evaluar el desempeño de cada modelo se utilizaron distintas métricas de evaluación, estas fueron calculadas con base en los resultados obtenidos con el conjunto de datos de prueba. Las métricas implementadas se describen a continuación.

Matriz de confusión: En el campo del ML, una matriz de confusión es una herramienta en diseño de tabla que permite la visualización del rendimiento de un clasificador. Cada fila de la matriz representa las instancias de las predicciones de una clase, mientras que cada columna representa las instancias de la clase real. En la clasificación binaria se genera una matriz de 2x2 como se muestra en la Figura 9, donde TP (verdadero positivo) y TN (verdadero negativo) corresponden a las imágenes de las clases TB y sanos que son clasificadas correctamente, mientras que FP (falso positivo) y FN (falso negativo) representan las imágenes de las clases TB y

sanos que son clasificados erróneamente. El comportamiento esperado para un rendimiento óptimo del clasificador es donde todos los valores se posicionan en las casillas de la diagonal con inclinación a la derecha.

		Real	
		Tuberculosis (1)	Sanos (0)
Predicción	Tuberculosis (1)	TP	FN
	Sanos (0)	FP	TN

Figura 9. Matriz de confusión para una clasificación binaria.

Para la clasificación de tres clases se genera una matriz de 3x3, con un comportamiento esperado similar a la matriz de confusión de clase binaria. En este caso existen nueve casos posibles que pueden resultar del clasificador como se muestra en la Figura 10. En este caso la casilla (0,0) representa las imágenes de la clase sanos que fueron clasificadas correctamente, mientras que las casillas (0,1) y (0,2) representan las imágenes de la misma clase que fueron clasificadas erróneamente. De igual manera, la casilla (1,1) representa las imágenes de la clase TB clasificadas correctamente, mientras que las casillas (1,0) y (1,2) representan las imágenes de esta misma clase que fueron clasificadas erróneamente. Por último, la casilla (2,2) representa las imágenes de la clase neumonía clasificadas

correctamente y las casillas (2,0) y (2,1) representan las imágenes de esta misma clase clasificadas erróneamente.

		Real		
		Sanos (0)	Tuberculosis (1)	Neumonía (2)
Predicción	Sanos (0)	T _{0,0}	F _{1,0}	F _{2,0}
	Tuberculosis (1)	F _{0,1}	T _{1,1}	F _{2,1}
	Neumonía (2)	F _{0,2}	F _{1,2}	T _{2,2}

Figura 10. Matriz de confusión para una clasificación de tres clases.

Exactitud: Es la medida de rendimiento más intuitiva y representa una relación entre la clasificación correcta y el total de observaciones. Es una medida útil en conjuntos de datos simétricos, esta métrica responde a la pregunta: ¿Cuántas imágenes, independiente de la clase, fueron clasificadas correctamente?

$$Exactitud = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Precisión: Es la relación entre el número de clasificaciones positivas realizadas correctamente y el total de clasificaciones positivas realizadas. Una alta precisión se

relaciona con la baja tasa de falsos positivos. La precisión responde a la pregunta: ¿Cuántas imágenes clasificadas como TB eran realmente de pacientes con TB?

$$Precisión = \frac{TP}{TP + FP}$$

(2)

Sensibilidad: Es la proporción de clasificaciones positivas realizadas correctamente para todos los datos en la clase real. La sensibilidad responde a la pregunta: ¿De todas las imágenes de pacientes con TB, cuántas fueron clasificadas correctamente como TB?

$$Sensibilidad = \frac{TP}{TP + FN}$$

(3)

Especificidad: Es una medida que indica el número de clasificaciones negativas que fueron clasificadas correctamente como negativas. Responde a la pregunta: ¿De todas las imágenes de pacientes sanos, cuántos fueron clasificados correctamente como sanos?

$$Especificidad = \frac{TN}{TN + FP}$$

(4)

F1-Score: Es el promedio ponderado de precisión y sensibilidad. Por lo tanto, esta puntuación tiene en cuenta tanto los falsos positivos como los falsos negativos. Otras métricas de evaluación como la precisión y sensibilidad pueden verse afectadas

cuando no se cuenta con un número equitativo de elementos por las clases evaluadas, el F1-Score se diseñó dando igual valor a dos métricas de evaluación, lo cual permite obtener un resultado preciso incluso cuando no se cuente con un número equitativo de elementos por clase, tal es el caso de la red 2 y 3 de este trabajo.

$$F1 - score = \frac{2TP}{2TP + FP + FN}$$

(5)

3. Resultados y Discusiones

3.1 Colecta de información y construcción del conjunto de datos

Durante el estudio clínico realizado en la Clínica y Laboratorio de TB de Tijuana, se proporcionaron un total de 159 imágenes. Ya que en este estudio únicamente se consideraron radiografías con técnica postero anterior, 30 radiografías fueron excluidas por tratarse de tomas laterales.

De las 129 imágenes restantes se consideró que 21 no cumplían con las características de calidad mínimas solicitadas, por lo que de igual manera fueron excluidas. Se obtuvieron como resultado un total de 108 fotografías de radiografías de tórax con técnica postero anterior que cumplen con los criterios de inclusión, correspondientes a 45 pacientes, las cuales conformaron el grupo TB (Figura 11).

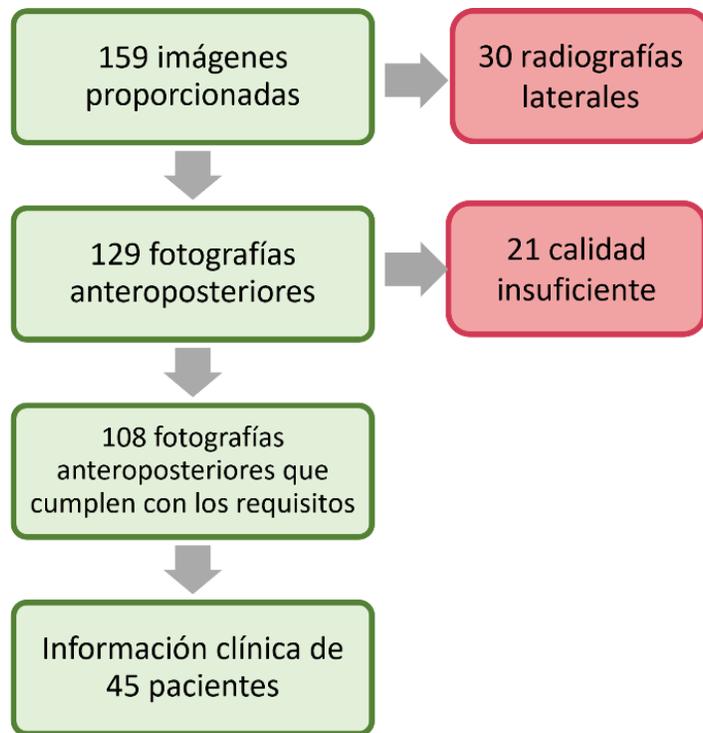


Figura 11. Proceso de selección y discriminación de imágenes recolectadas durante el estudio clínico.

En un primer experimento de clasificación de dos clases; TB y normal, se seleccionaron las 108 imágenes resultantes del estudio clínico para el grupo TB y 108 imágenes fueron seleccionadas aleatoriamente del conjunto de datos Shenzhen como grupo normal (Figura 12).

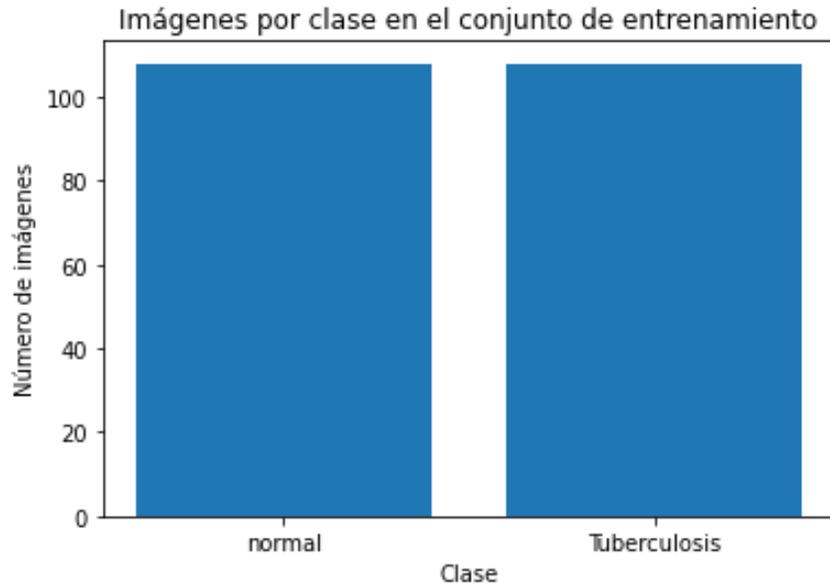


Figura 12. Distribución balanceada de los datos para 2 clases; normal y TB

En un segundo experimento con tres clases, se tomaron las 108 imágenes resultantes del estudio clínico para el grupo TB y para el grupo neumonía fueron seleccionadas 108 imágenes aleatorias resultantes del proceso de selección del conjunto de datos *CheXpert*. Por último, para el grupo normal en un principio se seleccionaron 108 imágenes aleatorias del conjunto de datos Shenzhen (Figura 12), sin embargo, el rendimiento en la clasificación de esta clase resultaba relativamente bajo, por lo que se optó por duplicar el número de imágenes destinadas al entrenamiento y validación para esta clase, la distribución final de los datos se muestra en la figura 13.

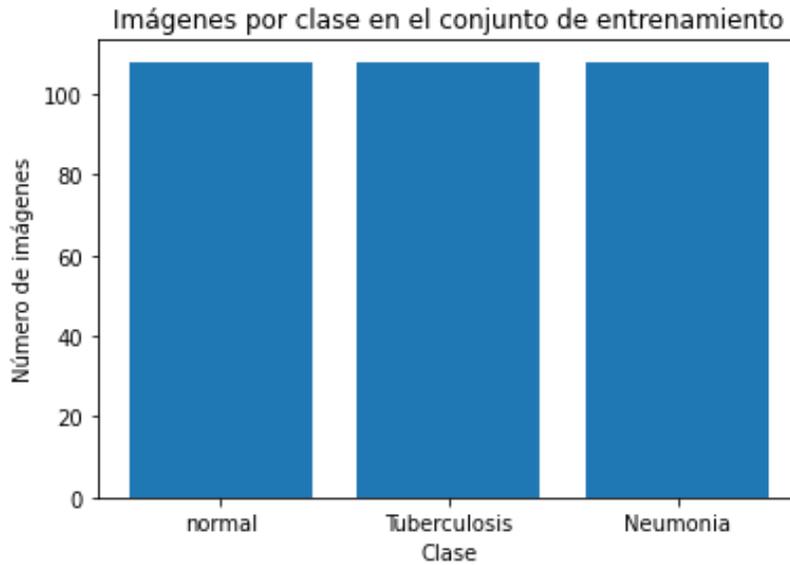


Figura 13. Distribución balanceada de los datos para las 3 clases; normal, TB y neumonía

Finalmente, para el grupo normal fueron seleccionadas 198 imágenes aleatorias del mismo conjunto de datos. Un total de 414 imágenes fueron implementadas para el experimento, la distribución de los elementos por clase se muestra en la Figura 14.

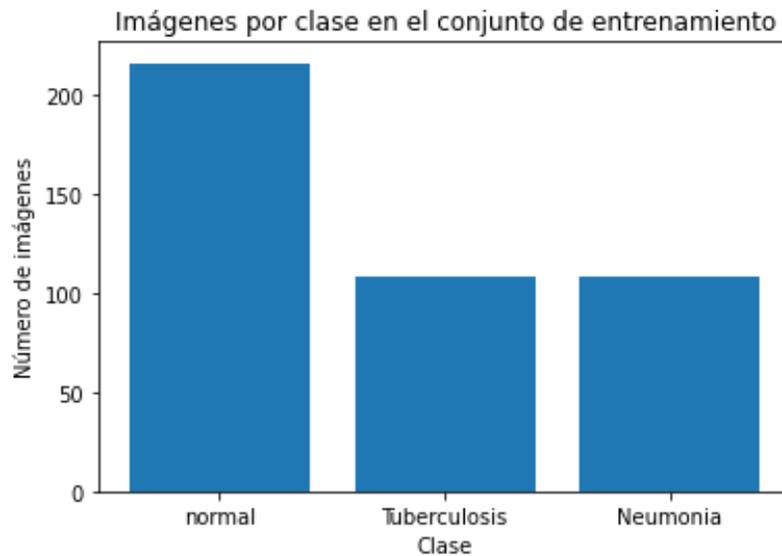


Figura 14. Distribución final de los datos para cada clase; normal, tuberculosis y neumonía.

Para el conjunto de prueba fueron seleccionadas aleatoriamente 18 imágenes de cada clase, a este conjunto se le aplicó reescalamiento y normalización de los datos. Las 360 imágenes restantes fueron divididas en conjuntos de entrenamiento con 288 imágenes y conjunto de validación con 72 imágenes, a estos conjuntos se les aplicó normalización de los datos, reescalamiento y aumentación de datos.

Tabla 6. Número de elementos por clase utilizados para los conjuntos de entrenamiento, validación y prueba.

Conjunto	Entrenamiento	Validación	Prueba
Tuberculosis (n=108)	72	18	18
Neumonía (n=108)	72	18	18
Normal (n=198)	144	36	18
Total (n=414)	288	72	54

3.2 Comparación entre arquitecturas de red

Con base en la revisión bibliográfica destacaron las redes *AlexNet*, *DenseNet121*, *DenseNet169*, *DenseNet201*, *InceptionV3* y *CheXnet* como posibles candidatos por sus resultados en la clasificación de radiografías de tórax en otras investigaciones. Para comparar su desempeño con el conjunto de datos construido se realizó un entrenamiento con clasificación binaria con el conjunto de datos Shenzhen.

Los mejores resultados fueron obtenidos en la red *DenseNet121*, por lo que fue utilizada en la construcción de los modelos clasificadores.

Tabla 7. Rendimientos alcanzados en clasificación binaria con distintas arquitecturas de CNN.

Arquitectura	Train Accuracy	Train Loss	Validation Accuracy	Validation Loss	Test Accuracy
AlexNet	0.8828	0.3475	0.7636	0.3681	0.8611
DenseNet121	0.9722	0.1306	0.9167	0.2561	0.9722
DenseNet169	0.9236	0.2565	0.7222	0.5128	0.8055
DenseNet201	0.9931	0.1137	0.5833	0.6959	0.9444
InceptionV3	0.7083	0.8427	0.8333	0.4661	0.9166
ChexNet	0.8656	0.6842	0.8106	0.3946	0.75

3.3 Construcción del modelo clasificador

La estructura de los diferentes modelos clasificadores resulta muy similar para la extracción de características en imágenes, en todos los casos se utilizó la red preentrenadas *DenseNet121* con los pesos de la base de datos *ImageNet*. Primero se definió una capa de entrada con dimensiones de 128x128x3, los primeros valores corresponden a las dimensiones de las imágenes de entrada de 128x128, mientras que el último valor corresponde a los canales de la imagen, en este caso tres ya que la red funciona con imágenes en formato RGB (composición de colores rojo, verde y azul), aun si las imágenes de entrenamiento se encuentran en escala de grises, esto debido a que la red procesa imágenes de tres canales preentrenada utiliza imágenes de tres canales únicamente. Posteriormente se aplicó una capa *flatten*, donde se transformaron las dimensiones del arreglo a un vector de 49,152 elementos, luego se utilizó una capa

dense para transformar las dimensiones de a 64 elementos, a continuación, una capa *dropout* seguido de otra capa *dense* y luego otra capa *dropout*, donde se continua con 64 elementos, seguido de una capa *dense* donde se transforma la dimensión a 32 elementos con una capa *dropout* enseguida. Una capa *dense* hace de nuevo la transformación a 16 elementos, finalmente una capa *dense* realiza la clasificación, obteniendo una salida de dos o tres elementos, correspondientes a las clases de entrada. En el caso del clasificador con datos clínicos se realiza la transformación de los datos y posteriormente se juntan con los datos de las imágenes con una capa de concatenación, la clasificación se realiza de la misma manera. Un diagrama del modelo clasificador multiclase utilizando datos clínicos se muestra en la Figura 15.

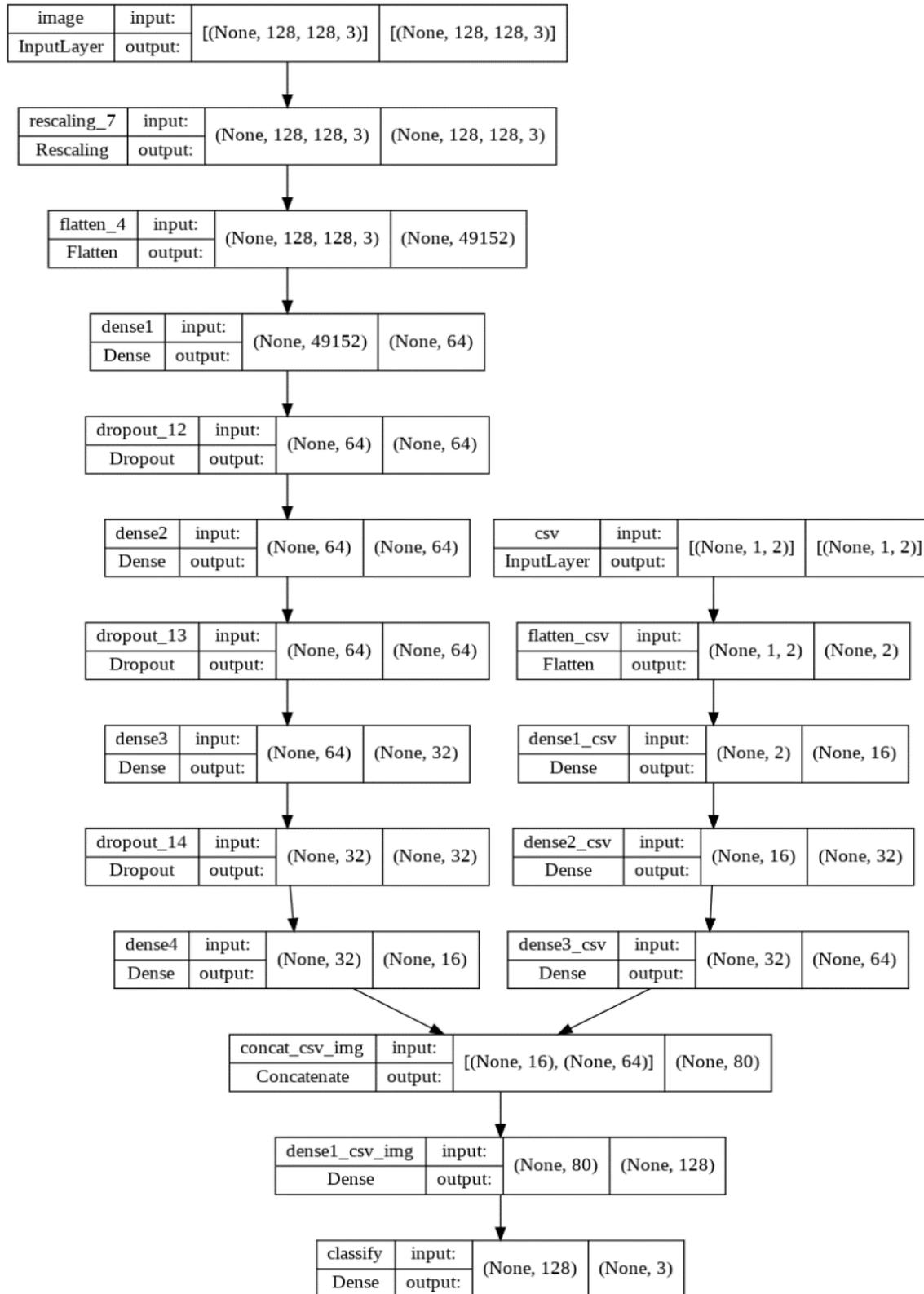


Figura 15. Diagrama del clasificador multiclase.

3.4 Resultados del clasificador con dos clases

El entrenamiento del clasificador para dos clases se llevó a cabo con 50 épocas, implementando un *early stopping*, esta herramienta permite evitar un sobreentrenamiento de la red y un ahorro de recursos computacionales al detener el proceso de aprendizaje de manera automática una vez que se detecta que no hay cambios considerables en el aprendizaje del modelo. Se implementó una reducción de la tasa de aprendizaje automático, herramienta que permite reducir automáticamente la rapidez con la que el modelo aprende, esto resulta benéfico cuando el aprendizaje del modelo no muestra mejora considerable durante un número definido de épocas consecutivas, de tal manera que puede continuar el proceso de entrenamiento sin activar el *early stopping*.

Durante el entrenamiento existen dos métricas que nos dan un indicador del desempeño del modelo en cada época; la precisión y la pérdida, la precisión indica el número de predicciones correctas obtenidas mientras que la pérdida son los valores que indican la diferencia con respecto a los estados deseados. Posteriormente se lleva a cabo un proceso de validación donde se utilizan datos independientes al entrenamiento para dar evidencia del desempeño del modelo en cada época. Idealmente se busca que los valores de precisión en la etapa de entrenamiento y validación vayan en aumento, sin alcanzar el 1 absoluto, ya que esto da un indicio de sobreentrenamiento, por otro lado, se busca que la pérdida vaya en decremento, siendo lo más cercano a 0 en las etapas finales del entrenamiento del modelo.

En el entrenamiento se obtuvo una precisión final de 0.9722 y una pérdida de 0.1306, el entrenamiento paró a la quinta época debido al *early stopping*. Los valores de precisión y pérdida alcanzada en cada época se pueden observar en la Figura 16.

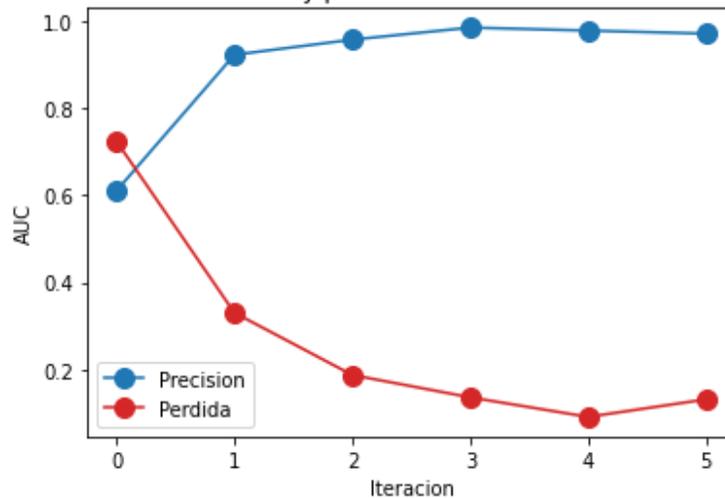


Figura 16. Precisión y pérdida durante el entrenamiento del clasificador binario.

Por otro lado, durante la validación se obtuvo una precisión final de 0.9167 y una pérdida de 0.2561. Los valores de precisión y pérdida alcanzada en cada época se pueden observar en la Figura 17.

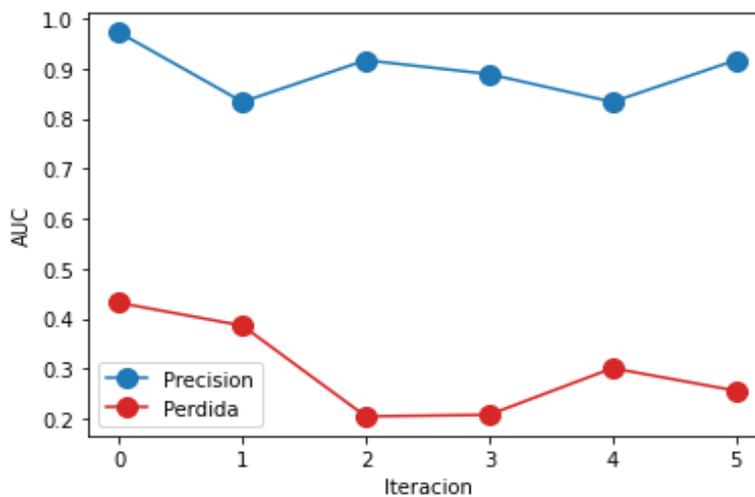


Figura 17. Precisión y pérdida durante la validación del clasificador binario.

Utilizando el conjunto de prueba sobre el modelo entrenado se obtuvo la matriz de confusión de la Figura 18.

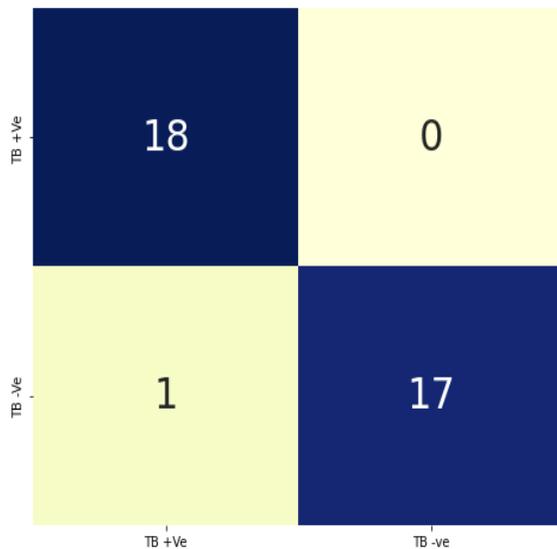


Figura 18. Matriz de desempeño para clasificador binario.

Con base en los valores obtenidos en la clasificación del conjunto de prueba, se obtuvieron las métricas que se muestran en la Tabla 8.

Tabla 8. Métricas alcanzadas por el modelo de clasificación binaria.

Exactitud	Precisión	Sensibilidad	Especificidad	F1-score
0.9722	0.9473	1	0.9444	0.9729

3.5 Clasificador con tres clases

El entrenamiento del clasificador para tres clases se llevó a cabo con 100 épocas, implementando una reducción de la tasa de aprendizaje automático.

En el entrenamiento se obtuvo una precisión final de 0.9965 y una pérdida de 0.0293. Los valores de precisión y pérdida alcanzada en cada época se pueden observar en la Figura 19.

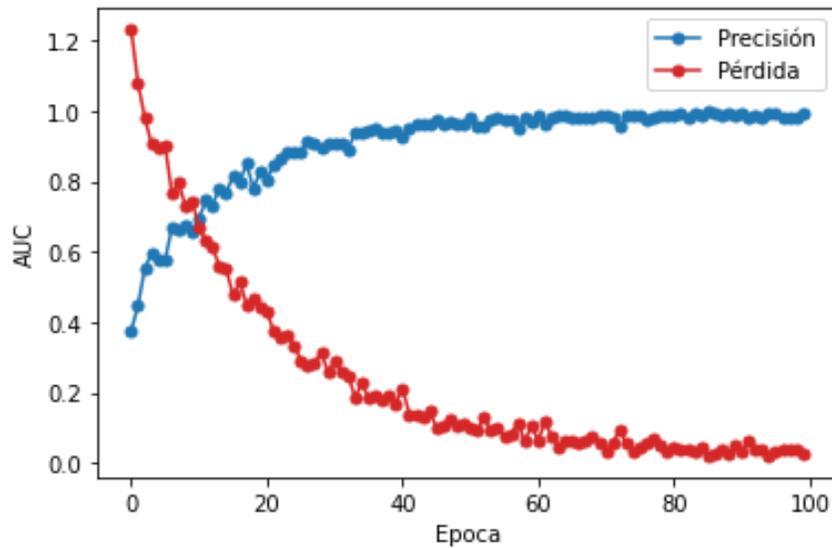
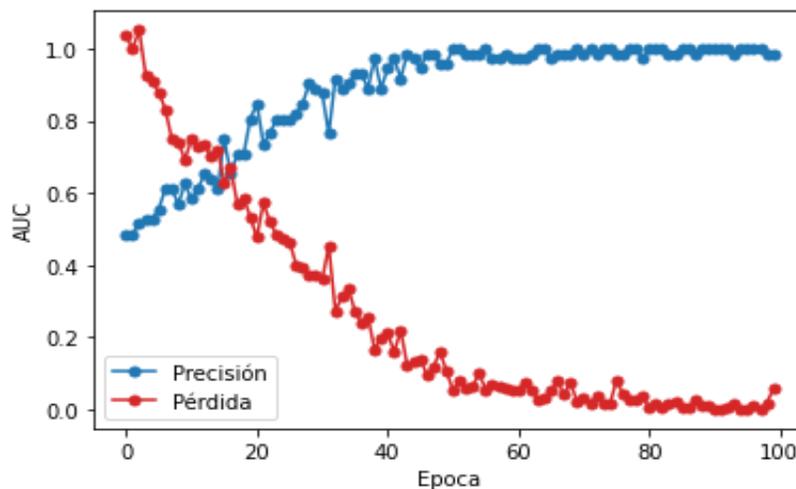


Figura 19. Precisión y pérdida en durante el entrenamiento del clasificador multiclase.

Por otro lado, durante la validación se obtuvo una precisión final de 0.9861 y una pérdida de 0.0587. Los valores de precisión y pérdida alcanzada en cada época se pueden observar en la Figura 20.



6

Figura 20. Precisión y pérdida en durante la validación del clasificador multiclase.

Utilizando el conjunto de prueba sobre el modelo entrenado se obtuvo la matriz de confusión de la Figura 21.

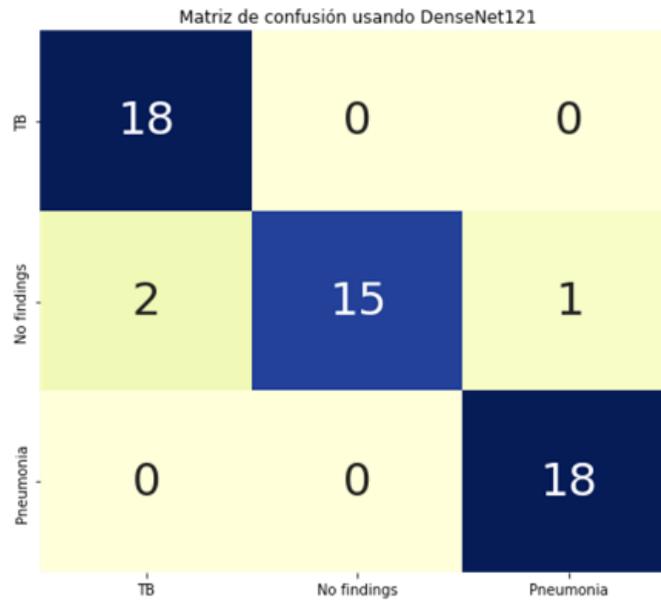


Figura 21. Matriz de desempeño para clasificador multiclase.

Con base en los valores obtenidos en la clasificación del conjunto de prueba, se obtuvieron las métricas que se muestran en la Tabla 9.

Tabla 9. Métricas alcanzadas por el modelo de clasificación multiclase.

Exactitud	Precisión	Sensibilidad	Especificidad	F1-score
0.9622	0.90	1	0.9428	0.9473

3.6 Comparación entre modelos

Finalmente, con las métricas obtenidas en las pruebas de cada modelo se realizó una comparación de todos los modelos, los valores se reportan en la Tabla 10.

Tabla 10. Comparación de métricas de desempeño de cada modelo.

Clasificador	Exactitud	Precisión	Sensibilidad	Especificidad	F1-score
2 Clases	0.9722	0.9473	1	0.9444	0.9729
3 Clases	0.9622	0.90	1	0.9428	0.9473

Durante el desarrollo de este estudio, se llevaron a cabo dos experimentos, la principal diferencia entre estos fue la utilización de dos y tres grupos de clasificación, como puede observarse en la Tabla 9, el clasificador de dos grupos obtuvo considerablemente mejores resultados que el clasificador de tres grupos, sin embargo, uno de los problemas que representa este enfoque de únicamente dos clases es la posible clasificación errónea en caso de introducir datos que no pertenezcan a ninguna de las dos clases. Por esta razón se desarrolló el segundo en el cual se agregó un tercer grupo donde se engloban varias patologías que producen neumonía pulmonar. Uno de los resultados esperados con este enfoque es una reducción notoria en la precisión de las predicciones del clasificador, sin embargo, con el ajuste correcto de los hiperparámetros fue posible mantener un buen rendimiento.

4. Conclusión y trabajo futuro

En este trabajo de investigación se construyeron dos modelos capaces de clasificar imágenes de radiografías de tórax en tres clases; pacientes con TB pulmonar, pacientes con neumonía no tuberculosa y pacientes sin patologías pulmonares utilizando técnicas de *machine learning*. Luego de la construcción del modelo este fue entrenado, validado y posteriormente se realizaron pruebas para probar su rendimiento con un conjunto de datos independientes al entrenamiento. Para la TB pulmonar el modelo obtuvo una exactitud de 96.22%, precisión de 90%, sensibilidad del 100%, especificidad de 94.28% y un F1-score del 94.73%. A partir de los valores obtenidos de las métricas de evaluación se puede determinar que el modelo es capaz de clasificar satisfactoriamente radiografías de tórax de pacientes con TB pulmonar a las de pacientes sanos y con otras patologías. Es necesario mencionar que se identificó una fuente de sesgo consistente en el presente estudio, ya que la totalidad de las imágenes utilizadas se encontraron en formato físico y fu necesaria su digitalización.

Otra aportación importante de esta investigación es el conjunto de datos obtenidos a partir del estudio clínico que se llevó a cabo en la clínica de TB de Tijuana, donde se recabaron 159 radiografías de tórax en conjunto con información clínica que formaron parte del conjunto de entrenamiento para esta investigación, además de otros datos obtenidos de bases de datos públicas. La generación de una nueva base de datos abre las puertas a futuras investigaciones.

Finalmente se puede concluir que los objetivos planteados para este trabajo fueron alcanzados con éxito.

La generación de este modelo deja una puerta abierta en esta línea de investigación, es posible obtener resultados más robustos y posiblemente rendimientos más altos al aumentar el número de elementos de entrenamiento para cada clase. Por otro lado, es posible aumentar la especificidad a otras patologías agregando otros grupos, existen distintos conjuntos de datos públicos de radiografías de tórax con etiquetas que no fueron implementadas en esta investigación. Para que el modelo pueda ser implementado y utilizado es necesario que pueda ejecutarse en un ambiente más amigable con el usuario, por lo que el desarrollo de una aplicación móvil o web es imperativo.

Por otro lado, se generó una base de datos con radiografías de tórax y datos clínicos de pacientes con TB pulmonar, esta base de datos incluye además anotaciones de las zonas donde el experto de la salud encontró las lesiones que indican la presencia de TB, esto permite generar un modelo que además de clasificar la TB, genere una predicción de la zona con posible presencia de lesiones indicativas de TB pulmonar.

5. Referencias

- [1] World Health Organization, “Global tuberculosis report,” 2021. [Online]. Available: <https://www.who.int/teams/global-tuberculosis-programme/tb-reports/global-tuberculosis-report-2021>.
- [2] World Health Organization (WHO), *WHO operational handbook on tuberculosis. Module 3: Diagnosis - Rapid diagnostics for tuberculosis detection 2021 update*. 2021.
- [3] A. J. Brent *et al.*, “Performance of the MGIT TBc Identification Test and Meta-Analysis of MPT64 Assays for Identification of the Mycobacterium tuberculosis Complex in Liquid Culture,” *J. Clin. Microbiol.*, vol. 49, no. 12, pp. 4343–4346, Dec. 2011, doi: 10.1128/JCM.05995-11.
- [4] D. Helb *et al.*, “Rapid detection of Mycobacterium tuberculosis and rifampin resistance by use of on-demand, near-patient technology,” *J. Clin. Microbiol.*, vol. 48, no. 1, pp. 229–237, Jan. 2010, doi: 10.1128/JCM.01463-09.
- [5] E. Çallı, E. Sogancioglu, B. van Ginneken, K. G. van Leeuwen, and K. Murphy, “Deep learning for chest X-ray analysis: A survey,” *Med. Image Anal.*, vol. 72, p. 102125, 2021, doi: 10.1016/j.media.2021.102125.
- [6] E. Maclean, L. Mckenna, and M. Ruhwald, “Pipeline Report » 2021,” 2021.
- [7] N. Kumar, S. K. Bhargava, C. S. Agrawal, K. George, P. Karki, and D. Baral, “Chest radiographs and their reliability in the diagnosis of tuberculosis.,” *JNMA. J. Nepal Med. Assoc.*, vol. 44, no. 160, pp. 138–142, 2005, doi: 10.31729/jnma.447.
- [8] R. Singhal and V. P. Myneedu, “Microscopy as a diagnostic tool in pulmonary tuberculosis,” *Int. J. Mycobacteriology*, vol. 4, no. 1, pp. 1–6, 2015, doi: 10.1016/j.ijmyco.2014.12.006.
- [9] FIND, “Digital Chest Radiography and Computer-Aided Detection (CAD) Solutions for Tuberculosis Diagnostics: Technology Landscape Analysis,” 2021.
- [10] M. R. A. van Cleeff, L. E. Kivihya-Ndugga, H. Meme, J. A. Odhiambo, and P. R. Klatser, “The role and performance of chest X-ray for the diagnosis of tuberculosis: A cost-effective analysis in Nairobi, Kenya,” *BMC Infect. Dis.*, vol. 5, pp. 1–9, 2005, doi: 10.1186/1471-2334-5-111.
- [11] S. Graham *et al.*, “Chest radiograph abnormalities associated with tuberculosis: reproducibility and yield of active cases.,” *Int. J. Tuberc. Lung Dis. Off. J. Int. Union against Tuberc. Lung Dis.*, vol. 6, no. 2, pp. 137–142, Feb. 2002.
- [12] M. Nijjati *et al.*, “Deep learning assistance for tuberculosis diagnosis with chest radiography in low-resource settings,” *J. Xray. Sci. Technol.*, vol. 29, no. 5, pp. 785–796, 2021, doi: 10.3233/XST-210894.
- [13] I. Kononenko, “Machine learning for medical diagnosis: History, state of the art

- and perspective,” *Artif. Intell. Med.*, vol. 23, no. 1, pp. 89–109, 2001, doi: 10.1016/S0933-3657(01)00077-X.
- [14] K. Fukushima and S. Miyake, “Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Visual Pattern Recognition,” pp. 267–285, 1982, doi: 10.1007/978-3-642-46466-9_18.
- [15] Y. LeCun and Y. Bengio, “Convolutional networks for images, speech, and time series,” *Handb. brain theory neural networks*, vol. 3361, no. January 1995, pp. 255–258, 1995, [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.32.9297&rep=rep1&type=pdf>.
- [16] D. A. Moses, “Deep learning applied to automatic disease detection using chest X-rays,” *J. Med. Imaging Radiat. Oncol.*, vol. 65, no. 5, pp. 498–517, 2021, doi: 10.1111/1754-9485.13273.
- [17] M. Dewey and P. Schlattmann, “Deep learning and medical diagnosis,” *Lancet*, vol. 394, no. 10210, pp. 1710–1711, Nov. 2019, doi: 10.1016/S0140-6736(19)32498-5.
- [18] WHO, *Consolidated Guidelines on Tuberculosis Treatment*. 2020.
- [19] A. Al Amin, S. Parvin, M. A. Kadir, T. Tahmid, S. K. Alam, and K. Siddique-E Rabbani, “Classification of breast tumour using electrical impedance and machine learning techniques,” *Physiol. Meas.*, vol. 35, no. 6, pp. 965–974, 2014, doi: 10.1088/0967-3334/35/6/965.
- [20] A. Hosny, C. Parmar, J. Quackenbush, L. H. Schwartz, H. J. W. L. Aerts, and H. H. Edu, “Artificial intelligence in radiology HHS Public Access,” *Nat Rev Cancer*, vol. 18, no. 8, pp. 500–510, 2018, doi: 10.1038/s41568-018-0016-5.Artificial.
- [21] M. E. H. Chowdhury *et al.*, “Wearable Real-Time Heart Attack Detection and Warning System to Reduce Road Accidents.,” *Sensors (Basel)*, vol. 19, no. 12, Jun. 2019, doi: 10.3390/s19122780.
- [22] J. Bai, R. Posner, T. Wang, C. Yang, and S. Nabavi, “Applying deep learning in digital breast tomosynthesis for automatic breast cancer detection: A review,” *Med. Image Anal.*, vol. 71, p. 102049, 2021, doi: 10.1016/j.media.2021.102049.
- [23] T. Rahman *et al.*, “Reliable tuberculosis detection using chest x-ray with deep learning, segmentation and visualization,” *arXiv*, vol. 8, 2020, doi: 10.1109/access.2020.3031384.
- [24] P. Afshar, S. Heidarian, F. Naderkhani, A. Oikonomou, K. N. Plataniotis, and A. Mohammadi, “COVID-CAPS: A capsule network-based framework for identification of COVID-19 cases from X-ray images,” *Pattern Recognit. Lett.*, vol. 138, pp. 638–643, 2020, doi: 10.1016/j.patrec.2020.09.010.
- [25] T. Agrawal and P. Choudhary, “FocusCovid: automated COVID-19 detection using deep learning with chest X-ray images,” *Evol. Syst.*, no. Grech 2020, 2021, doi: 10.1007/s12530-021-09385-2.

- [26] I. S. Masad, A. Alqudah, A. M. Alqudah, and S. Almashaqbeh, "A hybrid deep learning approach towards building an intelligent system for pneumonia detection in chest x-ray images," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 6, pp. 5530–5540, 2021, doi: 10.11591/ijece.v11i6.pp5530-5540.
- [27] RSNA, "RSNA Pneumonia Detection Challenge," *Kaggle*, 2018. <https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/data> (accessed Apr. 20, 2022).
- [28] J. Shiraishi *et al.*, "Development of a digital image database for chest radiographs with and without a lung nodule: Receiver operating characteristic analysis of radiologists' detection of pulmonary nodules," *Am. J. Roentgenol.*, vol. 174, no. 1, pp. 71–74, 2000, doi: 10.2214/ajr.174.1.1740071.
- [29] S. Jaeger, S. Candemir, S. Antani, Y.-X. J. Wang, P.-X. Lu, and G. Thoma, "Two public chest X-ray datasets for computer-aided screening of pulmonary diseases," *Quant. Imaging Med. Surg.*, vol. 4, no. 6, p. 475, Dec. 2014, doi: 10.3978/J.ISSN.2223-4292.2014.11.20.
- [30] S. Ryoo and H. J. Kim, "Activities of the Korean Institute of Tuberculosis," *Osong Public Heal. Res. Perspect.*, vol. 5, no. Suppl, p. S43, Dec. 2014, doi: 10.1016/J.PHRP.2014.10.007.
- [31] D. Demner-Fushman *et al.*, "Preparing a collection of radiology examinations for distribution and retrieval," *J. Am. Med. Inform. Assoc.*, vol. 23, no. 2, pp. 304–310, Mar. 2016, doi: 10.1093/JAMIA/OCV080.
- [32] I. Pan, S. Agarwal, and D. Merck, "Generalizable Inter-Institutional Classification of Abnormal Chest Radiographs Using Efficient Convolutional Neural Networks," *J. Digit. Imaging*, vol. 32, no. 5, p. 888, Oct. 2019, doi: 10.1007/S10278-019-00180-9.
- [33] J. Irvin *et al.*, "CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison," *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 01, pp. 590–597, Jul. 2019, doi: 10.1609/AAAI.V33I01.3301590.
- [34] A. E. W. Johnson *et al.*, "MIMIC-CXR-JPG, a large publicly available database of labeled chest radiographs," vol. 14, pp. 1–7, 2019, [Online]. Available: <http://arxiv.org/abs/1901.07042>.
- [35] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "ChestX-ray8: Hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 3462–3471, 2017, doi: 10.1109/CVPR.2017.369.
- [36] A. Bustos, A. Pertusa, J.-M. Salinas, and M. de la Iglesia-Vayá, "PadChest: A large chest x-ray image dataset with multi-label annotated reports," *Med. Image Anal.*, vol. 66, Jan. 2019, doi: 10.1016/j.media.2020.101797.
- [37] S. J. Heo *et al.*, "Deep learning algorithms with demographic information help to

- detect tuberculosis in chest radiographs in annual workers' health examination data," *Int. J. Environ. Res. Public Health*, vol. 16, no. 2, 2019, doi: 10.3390/ijerph16020250.
- [38] T. Pande, C. Cohen, M. Pai, and F. Ahmad Khan, "Computer-aided detection of pulmonary tuberculosis on digital chest radiographs: a systematic review," *Int. J. Tuberc. Lung Dis.*, vol. 20, no. 9, Sep. 2016, doi: 10.5588/IJTLD.15.0926.
- [39] P. Lakhani and B. Sundaram, "Deep learning at chest radiography: Automated classification of pulmonary tuberculosis by using convolutional neural networks," *Radiology*, vol. 284, no. 2, pp. 574–582, 2017, doi: 10.1148/radiol.2017162326.
- [40] A. S. Becker *et al.*, "Detection of tuberculosis patterns in digital photographs of chest X-ray images using Deep Learning: Feasibility study," *Int. J. Tuberc. Lung Dis.*, vol. 22, no. 3, pp. 328–335, 2018, doi: 10.5588/ijtld.17.0520.
- [41] R. Singh *et al.*, "Deep learning in chest radiography: Detection of findings and presence of change," *PLoS One*, vol. 13, no. 10, pp. 1–12, 2018, doi: 10.1371/journal.pone.0204155.
- [42] F. Pasa, V. Golkov, F. Pfeiffer, D. Cremers, and D. Pfeiffer, "Efficient Deep Network Architectures for Fast Chest X-Ray Tuberculosis Screening and Visualization," *Sci. Rep.*, vol. 9, no. 1, pp. 2–10, 2019, doi: 10.1038/s41598-019-42557-4.
- [43] O. Gozes and H. Greenspan, "Deep Feature Learning from a Hospital-Scale Chest X-ray Dataset with Application to TB Detection on a Small-Scale Dataset," *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, pp. 4076–4079, 2019, doi: 10.1109/EMBC.2019.8856729.
- [44] S. Rajaraman *et al.*, "A novel stacked generalization of models for improved TB detection in chest radiographs," *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, vol. 2018-July, pp. 718–721, 2018, doi: 10.1109/EMBC.2018.8512337.
- [45] S. Rajaraman and S. K. Antani, "Modality-Specific Deep Learning Model Ensembles Toward Improving TB Detection in Chest Radiographs," *IEEE Access*, vol. 8, pp. 27318–27326, 2020, doi: 10.1109/ACCESS.2020.2971257.
- [46] M. Karki *et al.*, "Identifying Drug-Resistant Tuberculosis in Chest Radiographs: Evaluation of CNN Architectures and Training Strategies," *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, pp. 2964–2967, 2021, doi: 10.1109/EMBC46164.2021.9630189.
- [47] P. Whitt, "An Overview of GIMP 2.8," *Pro Photo Color. with GIMP*, pp. 3–25, 2016, doi: 10.1007/978-1-4842-1949-2_1.

6. Apéndices

A. Código fuente para modelo clasificador multiclase con 10-K fold Cross Validation utilizando DenseNet121.

▾ Librerías

Se importan las paqterias necesarias

```
[ ] import os
import csv
import pandas as pd
import cv2
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
import tensorflow.keras
from tensorflow.keras.layers import Input,Dense,Flatten, Dropout
from tensorflow.keras.models import Model
from tensorflow.keras.callbacks import ModelCheckpoint
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow import keras
from tensorflow.keras import layers
import seaborn as sn

from sklearn import preprocessing, datasets, metrics
from sklearn.metrics import confusion_matrix, classification_report, hamming_loss, matthews_corrcoef
from sklearn.metrics import roc_curve
from sklearn.metrics import roc_auc_score
from sklearn.metrics import cohen_kappa_score
from sklearn.datasets import make_blobs
```

▾ Importacion del Dataset

Los datos se encuentran almacenados en Drive, por lo que se da permisos a Colab para acceder a archivos de Drive

```
[ ] from google.colab import drive
drive.mount("/content/gdrive")
```

Una vez montado el drive, se descomprimen los datos de entrenamiento indicando la direccion del archivo .zip y se almacenan en la direccion indicada

```
[ ] !unzip gdrive/My\ Drive/DATOS/data_all.zip -d data/ #En este caso guardamos TODOS los datos en la misma carpeta,
#las iteraciones del K-fold haran automaticamente los conjuntos de train y test
```

▼ Preprocesamiento de los datos

Todos los archivos siguen el formato '###CXR_###_#.png', donde

- Las primeras ### representan el origen de la imagen. Por ejemplo CHN pertenece a China, TJH representa Hospital de Tijuana.
- Los #### representan un identificador numérico de 4 dígitos consecutivo.
- La # del último carácter representa el diagnóstico de la imagen, siendo 0 sin presencia de TB y 1 para TB.

Hacemos la lectura del archivo .xlsx, este contiene una relación del nombre del archivo, categoría, edad y sexo del paciente. A partir de este archivo generamos un dataframe con esta información.

```
[ ] read_file = pd.read_excel(r'/content/gdrive/MyDrive/DATOS/Dataset2.xlsx') #lectura del archivo .xlsx
#read_file = pd.read_excel(r'/content/drive/MyDrive/Colab Notebooks/DATOS/Dataset2.xlsx')
read_file.to_csv(r'/content/data/datos.csv', index = None, header=True) #conversion de formato .xlsx a csv

train = pd.read_csv(r'/content/data/datos.csv', header=None) #generamos el dataframe train
train.columns = ['filename', 'category', 'sex', 'age'] #agregamos el header de las columnas
train = train.astype({'category': int, 'sex': int, 'age': int}, errors='raise') #por default los valores son tipo objeto,
#se hace la conversión a int
print(train.shape) #shape del dataframe train
print("\n\n",train.tail()) #impresión de las últimas 5 filas de df train
```

Hacemos un conteo de observaciones por categoría y generamos una gráfica.

```
[ ] x = np.arange(3)
y = train['category'].value_counts()
y = y
y[0:2]
print(y)

fig, ax = plt.subplots()
plt.bar(x, y)
plt.xticks(x, ('normal', 'TB positivo', 'Pneumonia'))
ax.set_xlabel('Clase')
ax.set_ylabel('Número de imágenes')
plt.title('Imágenes por clase en el conjunto de entrenamiento')
plt.show()
```

▼ Generación de los conjuntos de entrenamiento, validación y prueba

Para la generación de los conjuntos de entrenamiento y prueba usaremos la función `ImageDataGenerator` de la librería de `keras`. Esta requiere un dataframe que relacione la dirección de las imágenes y la etiqueta.

```
[ ] def prepend_path_train(fn): #definimos la funcion para generar la direccion de las imagenes
    return "/content/data/"+fn

train['filename'] = train['filename'].apply(prepend_path_train)

df_data = train.applymap(str)

df_data.tail() #se imprimen las ultimas 5 filas
```

Declaramos el target size y el batch size

```
[ ] auto_target_size = (128, 128)
    auto_batch_size = 32
```

Creamos un generador de imágenes basado en la función `ImageDataGenerator`. Esta función permite implementar aumentación de datos por lo que definimos los parámetros correspondientes.

```
[ ] train_datagen = ImageDataGenerator(
    rescale=1./255, #factor de reescala. El valor predeterminado es Ninguno. Si es Ninguno o 0, no se aplica el factor
    #de reescala; de lo contrario, multiplicamos los datos por el valor proporcionado (después de aplicar todas las demás
    #transformaciones).
    zoom_range=0.1, #Float. Rango para zoom aleatorio. Si es float, [inferior, superior] = [1-zoom_range, 1+zoom_range].
    rotation_range=20, #Int. Rango de grados para rotaciones aleatorias.
    width_shift_range=0.1, #Genera un corte aleatorio de la imagen horizontalmente.
    height_shift_range=0.1, #Genera un corte aleatorio de la imagen verticalmente.
    horizontal_flip=True, #boolean. Voltea aleatoriamente las entradas horizontalmente.
    vertical_flip=True) #boolean. Voltea aleatoriamente las entradas verticalmente.
```

De igual manera creamos un generador de imágenes para el conjunto de prueba, en este generador no se implementa aumentación de datos.

```
[ ] test_datagen = ImageDataGenerator(
    rescale=1./255) #unicamente se implementa el factor de reescala.
```

Se llama al generador de imágenes de entrenamiento desde como ejemplo, se observa que encuentra 414 imágenes correspondientes a 3 clases.

```
[ ] eg_gen = train_datagen.flow_from_dataframe(
    dataframe = df_data,
    subset='training',
    x_col= 'filename',
    y_col= 'category',
    batch_size=auto_batch_size,
    shuffle=True,
    class_mode="categorical",
    target_size=auto_target_size,
    color_mode = "rgb")
```

```
def plots(ims, figsize=(12,6), interp=False, titles=None): #Se imprimen 12 elementos desde el generador de imagenes eg_gen, como
#se puede observar las transformaciones de la aumentacion de datos se aplicaron correctamente.
    f = plt.figure(figsize=figsize)
    for i in range(0,12):
        sp = f.add_subplot(3, 4, i+1)
        sp.axis('Off')
        if titles is not None:
            sp.set_title(titles[i], fontsize=8)
        plt.imshow(ims[i], interpolation=None if interp else 'none') #La etiqueta de la imagen corresponde al diagnostico.

    imgs, labels = next(eg_gen) # de esta manera el generador de imagenes entrega iteraciones de imagenes y la etiqueta correspondiente.

    plots(imgs, titles=labels)
```

▼ Cross Validation

Declaramos el numero de k-folds para la validacion cruzada.

```
[ ] from sklearn.model_selection import KFold
    k_folds=10
```

Creamos un bucle para la validacion cruzada, donde se utilizaran diferentes particiones del dataset para el entrenamiento y prueba respectivamente. Dentro del bucle se divide el dataframe train en dos conjuntos; un conjunto para entrenamiento y otro para prueba, despues entrena el modelo y se reportan resultados. Este proceso se repite k numero de veces.

```
[ ] #Entrenamiento conwith K-fold cross validation
kf = KFold(n_splits=k_folds, random_state=None, shuffle=True)
X= np.array(df_data["filename"])
i = 1
for train_index, test_index in kf.split(X): #se crea un index para hacer la particion del df train, este sera diferente en cada itera
    trainData = X[train_index]
    testData = X[test_index]

#se genera un split de los del df train en dos df: train_df y test_df
train_df = df_data.loc[df_data["filename"].isin(list(trainData))]
test_df = df_data.loc[df_data["filename"].isin(list(testData))]

all_labels = [ "0" , "1" , "2"]

train_gen = train_datagen.flow_from_dataframe( #Se llama al generador de imagenes de entrenamiento utilizando el df_train como di
    dataframe = train_df,
    subset='training',
    x_col= 'filename',
    y_col= 'category',
    batch_size=auto_batch_size,
    shuffle=True,
    class_mode="categorical",
    target_size=auto_target_size,
    color_mode = "rgb")

test_gen = test_datagen.flow_from_dataframe( #Tambien se llama al generador de imagenes de prueba con el df_test como directorio
    dataframe = test_df,
    x_col= 'filename',
    y_col= 'category',
    batch_size=auto_batch_size,
    shuffle=False,
    class_mode="categorical",
    target_size=auto_target_size,
    color_mode = "rgb")
```

▼ Modelo con DenseNet121

Se definen las dimensiones de los datos

```
[ ] img_data_shape = (128, 128, 3) #dimensiones de la imagen, este valor corresponde a las entradas aceptadas por DenseNet121
    csv_data_shape = (1, 2) #Este valor corresponde a los dos datos (edad y sexo) que tenemos por cada imagen
    num_classes = 3 #las 3 etiquetas correspondientes a sano, TB+ y neumonia
```

Se carga el modelo desde una API funcional de Keras, ya que esta acepta dos inputs.

```
[ ] dn121_model = tf.keras.applications.DenseNet121(
    include_top=False,
    weights="imagenet",
    input_tensor=None,
    input_shape=img_data_shape,
    pooling=max)
```

Se generan las capas del modelo

```
[ ] # Definimos dos capas de entrada, una para imagenes, otro para datos clinicos
img_input = tf.keras.layers.Input(shape=img_data_shape, name="image")
csv_input = tf.keras.layers.Input(shape=csv_data_shape, name="csv")

# Se crea una red que usara como input las imagenes, esta es la red que utilice en otros experimentos y ha tenidos buenos resultados,
act = tensorflow.keras.layers.LeakyReLU(alpha=0.3)

x1 = (dn121_model.output)
x1 = tf.keras.layers.experimental.preprocessing.Rescaling(1./255)(img_input)
x1 = tf.keras.layers.Flatten()(x1)
x1 = tf.keras.layers.Dense(64, activation=act, name='dense1')(x1)

x1 = tf.keras.layers.Dropout(0.2)(x1)
x1 = tf.keras.layers.Dense(64, activation=act, name='dense2')(x1)

x1 = tf.keras.layers.Dropout(0.2)(x1)
x1 = tf.keras.layers.Dense(32, activation=act, name='dense3')(x1)

x1 = tf.keras.layers.Dropout(0.2)(x1)
x1 = tf.keras.layers.Dense(16, activation=act, name='dense4')(x1)

# Se crea una red para el que utilice como input los datos clinicos
x2 = tf.keras.layers.Flatten(name="flatten_csv")(csv_input)
x2 = tf.keras.layers.Dense(16, activation='relu', name="dense1_csv")(x2)
x2 = tf.keras.layers.Dense(32, activation='relu', name="dense2_csv")(x2)
x2 = tf.keras.layers.Dense(64, activation='relu', name="dense3_csv")(x2)

# concatenamos ambas redes
x = tf.keras.layers.concatenate([x1,x2], name="concat_csv_img")
#x1 = tf.keras.layers.Dense(128, activation='relu', name="dense1_csv_img")(x1)

x = tf.keras.layers.Dense(128, activation='relu', name="dense1_csv_img")(x)

# Se agrega una capa para la clasificacion de las imagenes
#output = tf.keras.layers.Dense(num_classes, activation='softmax', name="classify")(x1)
output = tf.keras.layers.Dense(num_classes, name="classify")(x)

# se genera el modelo con 2 entradas y 1 salida
model = tf.keras.models.Model(inputs=[img_input, csv_input], outputs=output)
#model = tf.keras.models.Model(inputs=[img_input], outputs=output)
model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

[ ] print(model.summary())

[ ] from keras.models import Sequential
    from keras.layers import Dense
    from keras.utils.vis_utils import plot_model
    plot_model(model, to_file='model_plot.png', show_shapes=True, show_layer_names=True)
```

▼ Feats

Creamos un dataset de tensorflow para los datos clinicos del CSV, unicamente utilizaremos la particion de los datos que nos entrega el kfold bajo del dataframe train_df

```
[ ] # Se declaran los feautres de relevancia
features_names = ['sex', 'age']
features = train_df[features_names].astype(float)

# utilizamos un pop
#target = train_df.pop('category').astype(int)
```

```
[ ] data=tf.convert_to_tensor(features) #se convierte el dataframe a tensor
```

```
[ ] dataset = tf.data.Dataset.from_tensor_slices(data).batch(auto_batch_size) #se genera un dataset de tensorflow
```

```
[ ] for row in dataset.take(1):
    print(row)
```

Para la entrada del modelo es necesario introducir el batch de imagenes y datos clinicos, por lo cual se implementa un generador que llama al generador de imagenes e itera el dataset de datos.

```
[ ] def my_gen(subset):
    while True:
        if subset == "training":
            for i in train_gen.next():
                img_batch = i
            for j in dataset.take(1):
                csv_batch = j
            for k in dataset.take(1):
                labels_batch = k
        else:
            for i in test_gen.next():
                img_batch = i
            for j in test_gen.take(1):
                csv_batch = j
            for k in dataset.take(1):
                labels_batch = k

        yield((img_batch, csv_batch),labels_batch)

gen_train = my_gen("training")
gen_valid = my_gen("validation")
```

```
[ ] model.fit(train_gen, epochs=2, steps_per_epoch=3)
```

```
[ ] nn_train_generator.reset()
    nn_valid_generator.reset()
```

```
[ ] history_1
```

```
[ ] #Generamos una grafica de la precision y la perdida alcanzada por la red las etapas de entranamiento y validacion, comenzamos con la
acc = history_1.history['accuracy']
val_acc = history_1.history['val_accuracy']
loss = history_1.history['loss']
val_loss = history_1.history['val_loss']

epochs = range(len(acc))

plt.plot(epochs, acc, marker = 'o', markersize = 5, label='Precisión', color='C0')
plt.plot(epochs, loss, marker = 'o', markersize = 5, label='Pérdida', color='C3')
plt.xlabel("Epoca")
plt.ylabel("AUC")
plt.title('Presición y pérdida en conjunto de entrenamiento')
plt.legend()

plt.figure()

plt.plot(epochs, val_acc, marker = 'o', markersize = 5, label='Precisión', color='C0')
plt.plot(epochs, val_loss, marker = 'o', markersize = 5, label='Pérdida', color='C3')
plt.title('Presición y pérdida en conjunto de validación')
plt.xlabel("Epoca")
plt.ylabel("AUC")
plt.legend()

plt.show()
```

Clasificador

```
[ ] #Generamos un nuevo, para lo cual volvemos a cargar el modelo DenseNet121
loaded_model_1 = DenseNet121(include_top=False,
                             weights='imagenet',
                             input_tensor=None,
                             input_shape=(128, 128, 3),
                             pooling=max)
```

```
[ ] #Al igual que en la etapa anterior, congelamos las últimas 4 capas y agregamos una capa de salida softmax con 2 outputs
act=tensorflow.keras.layers.LeakyReLU(alpha=0.3)

out = (loaded_model_1.output)
out = Flatten()(out)
out = Dense(64, activation=act, name='dense1')(out)

out = Dropout(0.2)(out)
out = Dense(64, activation=act, name='dense2')(out)

out = Dropout(0.2)(out)
out = Dense(32, activation=act, name='dense3')(out)

out = Dropout(0.2)(out)
out = Dense(16, activation=act, name='dense4')(out)

out = Dense(3, activation='softmax')(out)

loaded_model_1 = Model(loaded_model_1.input, out)
```

```
[ ] #Cargamos los pesos obtenidos en el entrenamiento anterior
loaded_model_1.load_weights('labeller_model_1.hdf5')

[ ] #Se compila el modelo con un optimizador RMSprop
optimizer = tensorflow.keras.optimizers.RMSprop(learning_rate=1e-4)

loaded_model_1.compile(optimizer=optimizer,
                        loss='binary_crossentropy',
                        metrics=['accuracy'])

[ ] #Se realiza una prueba con el generador de prueba
NN_STEP_SIZE_test = np.ceil(nn_validation_test_generator.samples/nn_validation_test_generator.batch_size)

[ ] nn_validation_test_generator.reset()
results_VALID_1 = loaded_model_1.evaluate_generator(nn_validation_test_generator, NN_STEP_SIZE_test, verbose=1)

[ ] #Se obtienen los resultados de la validacion
results_VALID_1[1]
```

Metricas de desempeño

```
[ ] #Comenzamos haciendo un reset del generador de imagenes de prueba
nn_validation_test_generator.reset()

[ ] #Modelo 1
validation_predictions_1 = loaded_model_1.predict_generator(nn_validation_test_generator,
                                                           NN_STEP_SIZE_VALID,
                                                           workers=1,
                                                           use_multiprocessing=False,
                                                           verbose=1)

index_array_of_images_1 = nn_validation_test_generator.index_array[:validation_predictions_1.shape[0]]

actual_labels_1 = np.array(nn_validation_test_generator.classes)[index_array_of_images_1]
predicted_labels_1 = np.argmax(validation_predictions_1[index_array_of_images_1], axis=1)

[ ] #Realizamos pruebas de clasificacion para el modelo 1
print('modelo 1_ResNet121')
actual_labels_1[:10]
predicted_labels_1[0:10]

print(predicted_labels_1.shape)
print(np.array(actual_labels_1).shape)

print("Numero total de imagenes clasificadas correctamente por DenseNet121 = ",np.sum(actual_labels_1 == predicted_labels_1))
print("Accuracy = ", np.sum(actual_labels_1 == predicted_labels_1) / predicted_labels_1.shape )

[ ] Confusn_matrix_validation_set_1 = confusion_matrix(actual_labels_1, predicted_labels_1)

print("Se genera una predicci3n con",np.sum(Confusn_matrix_validation_set_1),"im3genes independientes.\n")
plt.figure(figsize = (10,7))
plt.title("Matriz de confusi3n usando DenseNet121")
sn.heatmap(Confusn_matrix_validation_set_1,
            annot=True,fmt='g',
            annot_kws={"size": 32},
            xticklabels = ['TB', 'No findings','Pneumonia'],
            yticklabels = ['TB', 'No findings','Pneumonia'],
            cmap="YlGnBu")
```