

UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA
CARRERA DE BIOINGENIERÍA



**IMPLEMENTACIÓN DE MODELOS DE DEEP
LEARNING PARA LA DETECCIÓN DE
TUBERCULOSIS Y NEUMONÍA VIRAL Y/O
BACTERIANA EN RADIOGRAFÍAS TORÁCICAS**

TESIS

Para optar el título profesional de Bioingeniero

AUTORES

Franza Sofia Estevez Velarde (ORCID: 0009-0004-0747-9648)

Oscar Alfredo Gomez Vasquez (ORCID: 0009-0008-5514-7460)

ASESOR:

Jinmi Gregory Lezama Calvo (ORCID: 0000-0003-2906-9741)

Lima – Perú

2025

DECLARACIÓN JURADA

Yo, Jinmi Gregory Lezama Calvo, identificado con DNI N° 42294872, en mi condición de persona responsable de validar la autenticidad de los trabajos de investigación y tesis de la Universidad de Ingeniería y Tecnología (en adelante, la Universidad), **declaro bajo juramento** lo siguiente:

Que el trabajo de investigación/tesis denominado: "Implementación de modelos de Deep Learning para la detección de tuberculosis y neumonía viral y/o bacteriana en radiografías torácicas" ha sido elaborado bajo mi asesoría, por Franza Sofia Estevez Velarde, identificado con DNI N° 76296196 y por Oscar Alfredo Gomez Vasquez, identificado con DNI N° 73674359 , para obtener el título profesional de Bioingeniería.

El trabajo de investigación/tesis antes mencionado ha sido sometido a los mecanismos de control y sanciones anti plagio previstos en la normativa interna de la Universidad, encontrándose un porcentaje de similitud de 0%.

En fe de lo cual firmo la presente.

En Barranco, el 07 de enero de 2025



Firma del asesor

TABLA DE CONTENIDO

| | Pág. |
|--|------|
| RESUMEN | 8 |
| ABSTRACT | 9 |
| INTRODUCCIÓN | 10 |
| Presentación del tema de investigación | 10 |
| Descripción de la situación problemática | 10 |
| Formulación del problema | 14 |
| Objetivos de investigación | 14 |
| Justificación | 15 |
| Alcance y limitaciones | 16 |
| CAPÍTULO I | 17 |
| ANTECEDENTES | 17 |
| CAPÍTULO II | 19 |
| MARCO TEÓRICO | 19 |
| 2.1. Neumonía | 19 |
| 2.1.2. Descripción de la enfermedad | 19 |
| 2.1.2. Diagnóstico de neumonía y limitaciones en los métodos | 20 |
| 2.1.3. Identificación de características en radiografías de neumonía | 21 |
| 2.1.3. Diferenciación de neumonía bacteriana y viral en radiografías | 23 |

| | | |
|--------------------|--|----|
| 2.2. | Tuberculosis | 24 |
| 2.2.1. | Descripción de la enfermedad | 24 |
| 2.2.2. | Diagnóstico de neumonía y limitaciones en los métodos | 25 |
| 2.2.3. | Identificación de características en radiografías de tuberculosis | 27 |
| 2.3. | Inteligencia Artificial | 30 |
| 2.3.1. | Aprendizaje automático | 30 |
| 2.3.2. | Redes neuronales | 30 |
| 2.3.2.1. | ResNet | 34 |
| 2.3.2.2. | EfficientNet | 39 |
| 2.3. | Métricas de evaluación | 41 |
| CAPÍTULO III | | 44 |
| MARCO METODOLÓGICO | | 44 |
| 3.1. | Caracterización de investigación | 44 |
| 3.2. | Consideraciones éticas | 44 |
| 3.3. | Desarrollar dos modelos de deep learning para la detección de tuberculosis y dos modelos para la detección de neumonía (viral y/o bacteriana) en radiografías torácicas; utilizando las bases de datos del Instituto Nacional de Alergias y Enfermedades Infecciosas, y del Centro Médico de Mujeres y Niños de Guangzhou, respectivamente | 45 |
| 3.3.1. | Recopilación de imágenes médicas | 45 |
| 3.3.2. | Diseño del modelo de deep learning | 49 |
| 3.3.3. | Entrenamiento y validación del modelo de deep learning | 52 |

| | | |
|-------------|--|----|
| 3.3.4. | Afinación del modelo de deep learning | 53 |
| 3.3.5. | Prueba del modelo de deep learning | 54 |
| 3.4. | Evaluar el desempeño del mejor modelo de tuberculosis y del mejor modelo de neumonía (viral y/o bacteriana), utilizando las bases de datos del Hospital Nacional Edgardo Rebagliati Martins y del Hospital Nacional Guillermo Almenara Irigoyen | 54 |
| 3.5 | Desarrollar una aplicación web para el análisis de radiografías basada en los modelos de detección seleccionados | 56 |
| 3.5.1. | Back-End App | 56 |
| 3.5.2. | Front-End App | 59 |
| CAPÍTULO IV | | 60 |
| RESULTADOS | | 60 |
| 4.1. | Desarrollar dos modelos de deep learning para la detección de tuberculosis y dos modelos para la detección de neumonía (viral y/o bacteriana) en radiografías torácicas; utilizando las bases de datos del Instituto Nacional de Alergias y Enfermedades Infecciosas, y del Centro Médico de Mujeres y Niños de Guangzhou, respectivamente | 60 |
| 4.2. | Evaluar el desempeño del mejor modelo de tuberculosis y del mejor modelo de neumonía (viral y/o bacteriana), utilizando las bases de datos del Hospital Nacional Edgardo Rebagliati Martins y del Hospital Nacional Guillermo Almenara Irigoyen | 65 |
| 4.3. | Desarrollar una aplicación web para el análisis de radiografías basada en los modelos de detección seleccionados | 67 |
| DISCUSIÓN | | 71 |
| 5.1. | Desarrollar dos modelos de deep learning para la detección de tuberculosis y dos modelos para la detección de neumonía (viral y/o bacteriana) en radiografías torácicas; | |

utilizando las bases de datos del Instituto Nacional de Alergias y Enfermedades Infecciosas, y del Centro Médico de Mujeres y Niños de Guangzhou, respectivamente

71

5.2. Evaluar el desempeño del mejor modelo de tuberculosis y del mejor modelo de neumonía (viral y/o bacteriana), utilizando las bases de datos del Hospital Nacional Edgardo Rebagliati Martins y del Hospital Nacional Guillermo Almenara Irigoyen 73

5.3. Desarrollar una aplicación web para el análisis de radiografías basada en los modelos de detección seleccionados 75

CONCLUSIONES 77

RECOMENDACIONES 79

REFERENCIAS BIBLIOGRÁFICAS 80

ÍNDICE DE FIGURAS

| | Pág. |
|--|------|
| Figura 1: Neumonía por consolidación del lóbulo medio derecho | 21 |
| Figura 2: Neumonía del lóbulo superior derecho | 22 |
| Figura 3: Radiografía de tórax de pacientes adultos inmunocompetentes hospitalizados por neumonía adquirida en la comunidad según el agente causal | 23 |
| Figura 4: Complejo de Gohn en radiografía | 27 |
| Figura 5: Complejo de Rake | 27 |
| Figura 6: Radiografía con derrame pleural izquierdo | 28 |
| Figura 7: Alteraciones pulmonares típicas de TB postprimaria, con condensaciones y áreas sugestivas de excavación. | 29 |
| Figura 8: Alteraciones pulmonares asociadas a TB postprimaria observadas en TC y radiografías de tórax. | 29 |
| Figura 9: Neurona | 31 |
| Figura 10: Estructura de ANNs | 32 |
| Figura 11: Red Neuronal | 32 |
| Figura 12: Arquitectura CNNs | 33 |
| Figura 13: Estructura de redes recurrentes vs. de alimentación directa | 34 |
| Figura 14: Arquitectura de ResNet de 34 capas | 35 |
| Figura 15: Estructura de las convoluciones | 36 |
| Figura 16: A) Representación de la convolución. B) Max pooling | 37 |
| Figura 17: A) proceso de convolución. B) Proceso a escala de convolución | 38 |
| Figura 18: A) cambio de dimensionalidad usando el método de convolución 1x1 B) imagen global de la segunda capa del modelo | 39 |
| Figura 19: Arquitectura de EffcientNet con MBConv como bloques de construcción básicos | 41 |
| Figura 20: Matriz de confusión 2x2 | 41 |
| Figura 21: Matriz de confusión 3x3 | 43 |
| Figura 22: Importe de la librería de TensorFlow | 46 |

| | |
|--|----|
| Figura 23: Normalización | 48 |
| Figura 24: Parámetros Neumonía | 49 |
| Figura 25: Carga del modelo ResNet | 49 |
| Figura 26: Carga del modelo EfficientNet | 49 |
| Figura 27: Creación del modelo ResNet Neumonía | 50 |
| Figura 28: Compilación el modelo ResNet Neumonía | 52 |
| Figura 29: Entrenamiento y Validación del modelo ResNet Neumonía | 52 |
| Figura 30: Aumento de datos ResNet Neumonía | 53 |
| Figura 31: Segundo entrenamiento del modelo ResNet Neumonía | 54 |
| Figura 32: Librerías | 56 |
| Figura 33: Carga de modelo | 57 |
| Figura 34: Definición de rutas y vistas de la aplicación web | 57 |
| Figura 35: Función predict() | 59 |
| Figura 36: A) Parámetros entrenables de ResNet TB. B) Parámetros entrenables de ResNet neumonía. | 60 |
| Figura 37: A) Parámetros entrenables de EfficientNet TB. B) Parámetros entrenables para EfficientNet neumonía. | 61 |
| Figura 38: Comparaciones de Training Accuracy y Training Loss entre modelos de TB y neumonía en 5 épocas. | 62 |
| Figura 39: Comparaciones de Validation Accuracy y Validation Loss entre modelos de TB y neumonía en 5 épocas. | 63 |
| Figura 40: A) Comparaciones de Second Training Accuracy y Loss entre modelos de TB y neumonía en 10 épocas. | 64 |
| Figura 41: Comparaciones finales de Accuracy y Loss entre los modelos de TB y neumonía. | 65 |
| Figura 42: Predicciones de TB | 65 |
| Figura 43: Predicciones de neumonía viral y/o bacteriana | 66 |
| Figura 44: Página web | 67 |
| Figura 45: Identificación de imagen sin TB y resultado de exactitud | 68 |

| | |
|---|----|
| Figura 46: Identificación de radiografía de tórax con neumonía viral y resultado de exactitud | 68 |
| Figura 47: Ausencia de neumonía viral y/o bacteriana | 69 |
| Figura 48: Presencia de TB | 69 |
| Figura 49: Presencia de neumonía bacteriana | 70 |

RESUMEN

La neumonía y la tuberculosis (TB) son las enfermedades respiratorias más prevalentes en niños de uno a cinco años en Perú. Un diagnóstico rápido y eficaz es esencial para hospitales, clínicas e instituciones de salud, ya que mejora los resultados del paciente y optimiza la eficiencia operativa. El deep learning (DL) aplicado al análisis de imágenes médicas ha demostrado ser valioso para el diagnóstico, ayudando a los médicos a incrementar la exactitud y reducir el error humano. Este estudio desarrolló un modelo basado en la arquitectura ResNet para detectar neumonía y TB en radiografías de tórax pediátricas, utilizando una base de datos de 5856 imágenes del Centro Médico de Mujeres y Niños de Guangzhou (4100 de entrenamiento, 878 de validación y 878 de prueba) y del Instituto Nacional de Alergias y Enfermedades Infecciosas (988 de entrenamiento, 211 de validación y 211 de prueba).

La validación final mostró que ResNet superó a EfficientNet en la detección de TB y neumonía, logrando menor pérdida de validación (7,46% vs. 10,59% en TB y 41,73% vs. 46,91% en neumonía) y mayor exactitud (97,62% vs. 95,71% en TB y 86,06% vs. 83,03% en neumonía), confirmándolo como el modelo más adecuado. Asimismo, se implementó una plataforma digital en Python y HTML, facilitando su integración en la práctica clínica. Este estudio resalta el potencial de ResNet para mejorar el diagnóstico de TB y neumonía y enfatiza la importancia de optimizar los modelos de DL en el ámbito médico.

PALABRAS CLAVES:

Deep Learning; Neumonía; IA; Imágenes Médicas; Diagnóstico; Radiografías

ABSTRACT

PROPOSAL FOR TWO DEEP LEARNING MODELS FOR THE DETECTION OF TUBERCULOSIS AND VIRAL AND/OR BACTERIAL PNEUMONIA IN CHEST X-RAYS

Pneumonia and tuberculosis (TB) are among the most prevalent respiratory diseases in children aged one to five in Peru. Rapid and effective diagnosis is essential for healthcare institutions as it enhances patient outcomes and optimizes institutional efficiency and response capacity. Deep learning (DL) applied to medical image analysis has proven to be valuable for automatic pathology diagnosis, helping physicians increase diagnostic accuracy and reduce human error. This study developed a model based on the ResNet architecture to detect pneumonia and TB chest X-rays, using a database of 5856 medical images from the Guangzhou Women and Children's Medical Center and the National Institute of Allergy and Infectious Diseases.

The final validation results showed that ResNet outperformed EfficientNet in detecting TB and pneumonia, achieving lower validation loss (7,46% vs. 10,59% for TB and 41,73% vs. 46,91% for pneumonia) and higher accuracy (97,62% vs. 95,71% for TB and 86,06% vs. 83,03% for pneumonia), confirming ResNet as the more suitable model. Implemented on an interactive digital platform in Python and HTML, the model facilitates clinical access and integration. Overall, this study highlights ResNet's potential to enhance TB and pneumonia diagnosis and emphasizes the importance of further optimizing DL models in the medical field.

KEYWORDS:

Deep Learning; Machine Learning; Pneumonia; IA; Medical Images; Diagnosis; X-ray

INTRODUCCIÓN

Presentación del tema de investigación

La investigación presentada busca desarrollar un modelo de aprendizaje profundo para aumentar la exactitud en la detección de neumonía y TB en radiografías torácicas. Por consiguiente, se planea diseñar un sistema para ayudar a mejorar la atención médica y reducir posibles errores de diagnóstico. Para la implementación del modelo, se va a utilizar el lenguaje de programación Python. Asimismo, se emplearán bibliotecas como TensorFlow para la aplicación de redes neuronales convolucionales.

Descripción de la situación problemática

La neumonía es una infección, provocada ya sea por agentes bacterianos, virales o fúngicos, capaz de inflamar los sacos aéreos de uno o de ambos pulmones. Estos pueden llenarse de fluidos, provocando tos, fiebre, escalofríos y principalmente dificultad para respirar [1].

Según la Mayo Clinic [2], la neumonía es más grave en bebés y niños pequeños, personas mayores de 65 años, y personas con problemas pasados de salud o inmunitarios. En niños, la neumonía es la principal causa de muerte en menores de cinco años, con aproximadamente 700 000 fallecimientos anuales.

Globalmente, existen alrededor de 1400 casos de neumonía por cada 100 000 niños, con una gran incidencia en el sur de Asia, África occidental y central. De hecho, en el 2015, se obtuvieron cifras de hasta 2400 casos de muerte infantil cada día debido a la neumonía [3]. En adultos, la neumonía es la principal causa de hospitalización y muerte, particularmente para aquellos mayores de 65 años y aquellos con problemas de salud preexistentes [4].

La neumonía sigue siendo un grave problema de salud pública en el Perú y afecta principalmente a niños menores de cinco años. En el invierno de 2024 se registraron alrededor de 12000 casos de infección en niños menores de cinco años y en el primer semestre de ese año se registraron 75 muertes [5]. Factores como la desnutrición, el acceso limitado a los servicios de salud y la falta de vacunación contribuyen a una alta morbilidad. En el Perú, el esquema de vacunación relevante para la prevención de infecciones respiratorias agudas (IRA) en este grupo de edad, incluye las vacunas Antineumocócica, contra el Rotavirus, SPR (Sarampión, Paperas y Rubéola), BCG (Bacilo de Calmette-Guérin) y Pentavalente (DPT-HvB-Hib) [6]. Además, el suministro insuficiente de los medicamentos necesarios y la resistencia a los antibióticos dificultan el tratamiento [7].

En otra instancia, la tuberculosis (TB) es una enfermedad causada por la bacteria *Mycobacterium tuberculosis* que afecta principalmente a los pulmones, aunque también puede presentarse en otros órganos. Su forma de propagación es aérea, por lo que el contagio se da cuando una persona infectada tose, estornuda o inclusive habla. Una vez infectada la persona, se puede presentar el caso latente, en la cual la bacteria está inactiva; o se puede presentar la TB activa, en la cual se desarrollan síntomas, y puede ser altamente contagiosa y mortal. [8]

Según el reporte global de TB del 2023, desarrollado por la organización mundial de la salud (OMS) la TB sigue siendo una de las principales causas de morbilidad y mortalidad a nivel mundial. Es más, se estimó que en el 2021, 10,6 millones de personas padecieron de TB y 1,6 millones fallecieron de la enfermedad, esto adicionando las 187 000 personas que perecieron con TB asociada al VIH. Cabe resaltar la implicación de la pandemia ha exacerbado la crisis de TB, interrumpiendo los servicios y provocando una disminución en las notificaciones de casos de TB en muchas regiones [9].

En el Perú, la TB es una de las enfermedades que repercute significativamente en la salud nacional. Anualmente, se notifican alrededor de 27 mil casos nuevos de enfermedad activa y 17 mil casos nuevos de TB pulmonar frotis positivo, lo que nos convierte en uno de los países con mayor cantidad de casos de TB en América. De este modo, es relevante

resaltar el incremento de las formas drogoresistentes (TB DR) en el país. Estas son atribuibles a diagnósticos tardíos y el abandono de tratamientos [10]. Las formas de TB mencionadas anteriormente suelen presentar patrones radiológicos que pueden diferir de los casos nuevos, aún sensibles a los esquemas de primera línea que brinda el sistema nacional de salud [11].

Sin duda, el diagnóstico temprano y tratamiento oportuno de estas enfermedades es importante para aumentar la supervivencia, especialmente en pacientes que presentan enfermedades previas que suprimen al sistema inmunológico [12]. Igualmente, el diagnóstico temprano permite prevenir el desarrollo de complicaciones como la sepsis y la falla respiratoria [13]. A su vez, representa un factor importante al momento de evitar la expansión de las enfermedades hacia otras partes del cuerpo y mediante el uso de tratamiento con antibióticos evitar la severidad de la infección que puede terminar en consecuencias fatales [14,15].

No obstante, en países en desarrollo, la desigualdad entre los servicios de salud urbanos y rurales representa un desafío crítico. Según J. X. Guo y B. Li, la escasez de proveedores de atención médica calificados es una de las principales causas de la baja calidad y disponibilidad de atención en áreas rurales. En este contexto, la implementación de técnicas médicas asistidas por computadora o inteligencia artificial (IA) podría mejorar significativamente los resultados de la atención médica [16]. En el caso del Perú, la falta de personal médico y de enfermería capacitado en zonas rurales resalta la necesidad de soluciones innovadoras. Los modelos de IA, como el deep learning, pueden desempeñar un papel crucial al apoyar y capacitar al personal médico, compensando la ausencia de especialistas como radiólogos y médicos.

Adicionalmente, el tiempo de atención es un gran problema en el país. En los establecimientos del MINSA, según el Instituto Nacional de Estadística e Informática (INEI), el tiempo promedio de espera para emergencia es de 114 minutos y en EsSalud es de 58 minutos [17]. Sumado a esto, se ve el tiempo en el que se realizan las pruebas, como radiografías, y la prioridad en la que se obtienen los resultados. Por lo que conlleva a la implementación de herramientas de deep learning para el análisis automático de imágenes

médicas sea una solución efectiva para agilizar el tiempo de espera. El uso de deep learning puede ayudar a reducir los retrasos en la detección de imágenes médicas anormales, mejorando el flujo de trabajo en los centros médicos y así, mejorando significativamente la atención médica del país. Cabe resaltar que, el diagnóstico rápido y exacto de enfermedades, juega un papel crítico en áreas donde los servicios de atención médica son limitados. Esto es especialmente importante para el diagnóstico basado en radiografías de tórax, en la que el tiempo es fundamental en la progresión de las enfermedades. Cabe destacar que, según la Organización Mundial de la Salud, el 40% de todas las imágenes de diagnóstico realizadas en el mundo corresponden a imágenes del tórax [18].

Dentro de este marco, el sistema público de salud del Perú carece de suficiente financiamiento, por lo que la facilidad a los servicios de salud, y la calidad y disponibilidad de estos mismos es deficiente. El uso de deep learning para el diagnóstico de enfermedades puede ayudar a cerrar la brecha entre el sistema público y privado de salud. Por una parte, el costo de la atención médica se reduce gracias a la detección temprana y exacta. Por otra parte, al automatizar el análisis de imágenes médicas, la carga de trabajo del personal médico también se reduce, permitiendo que se enfoquen en labores que requieran más atención. La atención médica de alta calidad generalmente se asocia con el sistema privado. Entonces, al enriquecer la calidad de la atención médica en el sistema público, se puede reducir la brecha entre estos dos.

En consecuencia, el uso de herramientas tecnológicas para la detección y diagnóstico de enfermedades será de gran utilidad en los centros médicos dentro del país. El desarrollo de modelos de deep learning para el diagnóstico de neumonía y TB cobran importancia debido a la gran necesidad de diagnósticos exactos y rápidos, y a la baja calidad de la atención médica en el país.

Formulación del problema

La investigación busca abordar el desafío de diagnósticos erróneos y tardíos de neumonía y TB, que puede tener un impacto significativo en la atención médica y el pronóstico de los pacientes. Por lo que nos planteamos lo siguiente: Desarrollo de dos modelos de deep learning para aumentar la exactitud en la detección y diagnóstico de neumonía y TB en radiografías torácicas.

Objetivos de investigación

Objetivo General

- Implementar dos modelos de deep learning para mejorar la exactitud en la detección de tuberculosis y neumonía (viral y/o bacteriana), utilizando bases de datos internacionales para el desarrollo y nacionales para la evaluación.

Objetivos Específicos

- Desarrollar dos modelos de deep learning para la detección de tuberculosis y dos modelos para la detección de neumonía (viral y/o bacteriana) en radiografías torácicas; utilizando las bases de datos del Instituto Nacional de Alergias y Enfermedades Infecciosas, y del Centro Médico de Mujeres y Niños de Guangzhou, respectivamente.
- Evaluar el desempeño del mejor modelo de tuberculosis y del mejor modelo de neumonía (viral y/o bacteriana), utilizando las bases de datos del Hospital Nacional Edgardo Rebagliati Martins y del Hospital Nacional Guillermo Almenara Irigoyen.
- Desarrollar una aplicación web para el análisis de radiografías basada en los modelos de detección seleccionados.

Justificación

La detección temprana y exacta de patologías y anomalías en las radiografías de tórax es esencial para garantizar un tratamiento adecuado en los pacientes con enfermedades respiratorias, y de esta forma poder mejorar sus resultados médicos. Sin embargo, al momento de evaluar las imágenes médicas, los radiólogos y los médicos pueden pasar por alto detalles críticos y/o experimentar fatiga visual debido a la gran cantidad de imágenes para analizar, lo que en algunos casos puede llevar a un diagnóstico tardío o inexacto. Para abordar este problema, se han propuesto modelos de deep learning para ayudar a los radiólogos y a los médicos a detectar y diagnosticar patologías y anomalías en imágenes médicas.

Dentro de este orden de ideas, los algoritmos de deep learning pueden aprender a detectar patrones en grandes conjuntos de datos y, cuando se entrenan, pueden identificar de manera inmediata, patologías y anomalías en las radiografías de tórax. Igualmente, pueden mejorar significativamente la detección patológica en las radiografías de tórax. De hecho, algunos modelos pueden superar en exactitud de diagnóstico a los radiólogos profesionales. Rajpurkar et al. (2017) desarrollaron un modelo de red neuronal convolucional (CNN) en base al conjunto de datos de rayos X de tórax del Instituto Nacional de Salud de Estados Unidos, que logró un puntaje F1 (media armónica de precisión y sensibilidad) de 0,435 en la detección de patologías en radiografías de tórax, superando a los radiólogos profesionales con un promedio del 0.387 [19].

Dentro del contexto peruano, la implementación de modelos de deep learning para la detección y diagnóstico de enfermedades respiratorias, como la neumonía y TB, servirá para mejorar la calidad de la atención médica. De este modo, los radiólogos y los médicos peruanos mejorarán su eficiencia.

Alcance y limitaciones

El modelo de deep learning desarrollado será capaz de identificar anomalías en las radiografías del área torácica. De esta forma, se podrá distinguir si es que los pacientes presentan neumonía. Para desarrollar el algoritmo, este se entrenará con 5856 radiografías

de cohortes retrospectivas de pacientes pediátricos de uno a cinco años de edad que son específicas a esta patología, proporcionadas por el Centro Médico de Mujeres y Niños de Guangzhou para la parte de neumonía. Para el modelo de TB se utilizará una base de datos de imágenes de rayos X de tórax para casos positivos de TB (TB) y normales, creada por un equipo de investigadores de la Universidad de Qatar, Doha, Qatar, y la Universidad de Dhaka, Bangladesh, en colaboración con médicos de la Hamad Medical Corporation y Bangladesh. Esta base de datos está conformada por 700 imágenes de TB accesibles públicamente y 2800 imágenes que pueden descargarse desde el portal NIAID TB firmando un acuerdo, junto con 3500 imágenes normales. Por lo tanto, otro tipo de patologías que se encuentren en la radiografía o que sean de otros órganos no podrán ser evaluadas. Cabe resaltar que, como es una base de datos curada, es posible que el modelo desarrollado no pueda proporcionar resultados exactos fuera de este conjunto de datos. Finalmente, una primera validación se realizará con radiografías tomadas en el Perú.

La implementación de este modelo no solo beneficiará a los pacientes, que recibirán un diagnóstico más rápido y exacto, sino que también aliviará la carga de trabajo del personal médico al agilizar el proceso de identificación de esta enfermedad. En consecuencia, esta herramienta podría tener un impacto significativo en la mejora en la eficiencia y accesibilidad de la atención médica, beneficiando a los pacientes y al sistema de salud peruano.

CAPÍTULO I

ANTECEDENTES

B. F. Julca Villarreal [20] en su tesis de titulación “Aplicación de Deep Learning sobre imágenes topográficas para mejorar la precisión del diagnóstico de queratocono en una clínica de Lima”, Perú, tuvo como objetivo diseñar un modelo de deep learning para mejorar la precisión del diagnóstico de queratocono. Para lograrlo, utilizó algoritmos de redes neuronales convolucionales. La muestra estuvo compuesta por 2073 imágenes topográficas de pacientes con y sin queratocono. Los resultados muestran una precisión de 92.04% en el diagnóstico de queratocono usando el modelo de deep learning. Por tanto, el uso de modelos de deep learning en el procesamiento de imágenes topográficas, sí sirve para el diagnóstico de queratocono.

L. A. Loyola y R. M. Chamorro [21] en su tesis de titulación “Implementación de un sistema de diagnóstico clínico aplicando un modelo predictivo de Machine Learning para la detección de neumonía en el Hospital Villa Rebagliati de EsSalud, 2021”, Perú, tuvo como objetivo implementar un modelo de machine learning de redes neuronales convolucionales para ayudar a la detección de neumonía. La muestra consistió en un conjunto de 4023 radiografías de tórax de pacientes con y sin neumonía. Los resultados indican que el modelo de machine learning implementado tuvo una precisión del 86,32% en la detección de neumonía. En definitiva, la implementación del modelo tiene un impacto positivo en el diagnóstico clínico de los pacientes con neumonía del Hospital Villa Rebagliati.

C. S. Rojas [22] en su tesis de titulación "Precisión del uso del Deep Learning en la síntesis, detección y segmentación de signos ultrasonográficos de pulmón sugerentes a COVID-19", Perú, tuvo como objetivo evaluar un modelo de deep learning para la detección de signos ultrasonográficos de pulmón relacionados a COVID-19. Utilizó una metodología de redes neuronales convolucionales para la detección de las imágenes. La muestra consistió en un conjunto de 1892 ultrasonografías pulmonares. Los resultados indican que el modelo de deep learning implementado tuvo una exactitud de 92% en la

detección. En consecuencia, se obtuvo una alta exactitud en la detección de signos relacionados a COVID-19.

S. Bharati, P. Podder, y M. M. Mondal [23] en su artículo "Hybrid deep learning for detecting lung diseases from X-ray images", Bangladesh, tuvo como objetivo desarrollar un sistema de detección de enfermedades pulmonares utilizando radiografías. Utilizó una combinación de redes neuronales convolucionales y redes neuronales recurrentes. La muestra consistió de 5606 radiografías de pacientes con y sin enfermedades pulmonares. Los resultados muestran una exactitud del 73% en la detección de enfermedades pulmonares utilizando el modelo propuesto. Por consiguiente, se logró una exactitud alta para la detección de enfermedades pulmonares a partir de radiografías.

D. Kvak et al. [24] en su artículo "Chest X-ray Abnormality Detection by Using Artificial Intelligence: A Single-Site Retrospective Study of Deep Learning Model Performance", República Checa, tuvo como objetivo analizar el desempeño de un modelo de machine learning en la detección de anomalías en radiografías de tórax. Desempeñaron el modelo de redes neuronales convolucionales, basado en un conjunto de datos retrospectivos de 127 radiografías de tórax. Los resultados del modelo mostraron una sensibilidad del 92,5% y una especificidad del 64,4%, mientras que los radiólogos tuvieron una sensibilidad del 66,1% y una especificidad del 80,3%. Es así que, el software propuesto tuvo un rendimiento significativamente mejor que el de los radiólogos, con un índice de probabilidad de falso negativo mucho menor.

R. Mehrrotraa et al. [25] en su artículo "Ensembling of Efficient Deep Convolutional Networks and Machine Learning Algorithms for Resource Effective Detection of TB Using Thoracic (Chest) Radiography", India, tuvo como objetivo desarrollar un algoritmo de redes neuronales convolucionales en base al conjunto de datos Tuberculosis X-Ray (TBX11K) para detectar TB utilizando radiografías torácicas. La muestra consistió en un conjunto de 6678 radiografías de tórax. Los resultados indican que se obtuvo una exactitud de 99,10%. Por lo tanto, el ensamblado de redes convolucionales profundas utilizando radiografías torácicas resulta en una detección efectiva de TB.

CAPÍTULO II

MARCO TEÓRICO

2.1 Neumonía

2.1.1 Descripción de la enfermedad

La neumonía es una infección en los pulmones. Esto causa que los alvéolos (unidad funcional del tejido pulmonar que funcionan como “sacos de aire”) se inflamen, dificultando la difusión del oxígeno hacia el torrente sanguíneo. Existen distintos tipos de neumonía, que se clasifican según la causa y la zona o región del pulmón donde se ha producido [26, 27].

- Neumonía bacteriana: Es causada por distintas bacterias, siendo la más común la *Streptococcus pneumoniae*. Normalmente ocurren debido a que la respuesta inmunológica puede estar comprometida, ya sea por alguna enfermedad o mala nutrición. Algunas bacterias atípicas son *Mycoplasma pneumoniae*, *Chlamyphila pneumoniae* y *Legionella pneumophila*. Son consideradas atípicas porque estos organismos podrían causar diferentes síntomas o aparecer de manera distinta en las radiografías.
- Neumonía viral: Es causada por virus que afectan el tracto respiratorio superior, los más comunes son SARS-CoV-2, el virus de la influenza, y, sobre todo, el virus sincitial respiratorio.
- Neumonía fúngica: Es causada por hongos como *Pneumocystis jirovecii*. Normalmente afecta a personas que poseen sistemas inmunológicos débiles.
- Neumonía andante: Este tipo de neumonía es causada por la bacteria *Mycoplasma pneumoniae*. Generalmente causa una neumonía leve y generalizada que afecta a todos los grupos de edad.
- Neumonía química: Este tipo de neumonía es causada por la inhalación de vapores químicos, como los de productos de limpieza domésticos o químicos industriales.

- Neumonía adquirida en el hospital: Este tipo de neumonía se adquiere durante la estancia en un hospital. Puede llegar a ser grave porque las bacterias que causan la neumonía pueden ser resistentes a los antibióticos.

2.1.2 Diagnóstico de neumonía y limitaciones en los métodos

Para el diagnóstico de la neumonía se emplea una variedad de métodos, incluyendo el historial médico, en el cual se realizan preguntas sobre los síntomas típicos, recientes contactos o viajes; los exámenes físicos, en los cuales se escucha a los pulmones con el estetoscopio y se evalúan las señales vitales; las radiografías torácicas, en las cuales se identifican posibles inflamaciones en los pulmones, proporcionando información crucial para determinar la extensión y ubicación de la infección; los exámenes de sangre, en los cuales se identifican el tipo de organismo que está generando la infección; y las broncoscopias, en las cuales se obtienen muestras del pulmón para su procesamiento correspondiente [28, 29].

Algunas limitaciones que se pueden presentar en el diagnóstico de la neumonía incluyen:

- Historial médico: El historial médico es subjetivo y depende de la capacidad del paciente para recordar e informar sus síntomas con exactitud. Esto puede ser un desafío, especialmente en niños pequeños o pacientes de edad avanzada, que pueden tener dificultades para comunicar sus síntomas.
- Exámenes físicos: Estos también son subjetivos ya que dependen de la capacidad del médico para detectar sonidos anormales u otros signos de neumonía. Sin embargo, es posible que algunos pacientes no presenten ningún signo físico de neumonía, especialmente en las primeras etapas de la infección.
- Radiografías torácicas: Si bien las radiografías son una herramienta de diagnóstico común para la neumonía, tienen limitaciones. Por ejemplo, no se puede diferenciar

entre neumonía bacteriana y viral, y es posible que no se detecte neumonía en etapa temprana o neumonía en ciertas áreas de los pulmones.

- Exámenes de sangre: Los análisis de sangre pueden ayudar en algunos casos a confirmar una infección e identificar el tipo de organismo que la causa.
- Bronoscopias: La broncoscopia es un procedimiento invasivo que conlleva algunos riesgos, como sangrado o infección. Por lo general, se reserva para casos en los que otros métodos de diagnóstico han fallado.

2.1.3 Identificación de características en radiografías de neumonía

La identificación de características en una radiografía puede clasificarse según el instrumento utilizado, como una radiografía simple, tomografía computarizada, e incluso ultrasonido. En el caso de una radiografía simple, el tejido infectado aparece como manchas blancas en el fondo más oscuro de los pulmones, lo que se conoce como infiltrados. Estos infiltrados son identificados por los radiólogos y los médicos, y se utilizan para diagnosticar la neumonía y determinar la gravedad de la infección (Figura 1) [30].

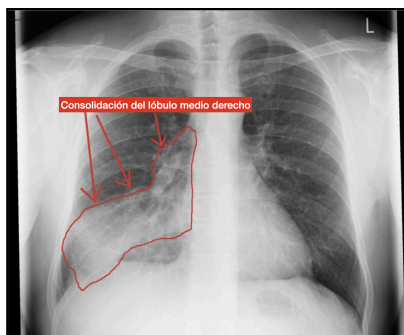


Figura 1: Neumonía por consolidación del lóbulo medio derecho [31]

La apariencia radiográfica de una neumonía puede variar dependiendo de la cantidad y distribución de los espacios aéreos afectados. Puede presentarse como una opacidad parenquimatosa confluyente (en lóbulo o segmento) o simplemente como una opacidad parcheada. Si el intersticio está predominantemente comprometido, puede manifestarse como un patrón reticulonodular. La presencia de broncogramas aéreos

confirmaría un proceso alveolar. Es importante destacar que, el volumen pulmonar no debería reducirse, e incluso podría aumentar. Por lo general, todas las anomalías radiográficas deberían desaparecer después de 6 semanas de tratamiento antibiótico adecuado. No obstante, la neumonía puede complicarse con la formación de abscesos o empiema (Figura 2) [32].



Figura 2: Neumonía del lóbulo superior derecho [32]

En relación con la diferenciación entre neumonía viral y bacteriana, los hallazgos radiográficos mostraron similitudes entre las neumonías bacterianas, virales, aquellas asociadas a microorganismos atípicos y las de origen desconocido. No obstante, el patrón alveolar fue más común en las neumonías bacterianas y de causa desconocida, mientras que los patrones intersticial y mixto fueron más frecuentes en las neumonías virales y las causadas por microorganismos atípicos. El patrón alveolar con distribución lobar fue raro en las neumonías virales, en contraste con el patrón intersticial en vidrio esmerilado, que predominó en las neumonías virales y las asociadas a microorganismos atípicos [33].

2.1.4 Diferenciación de neumonía bacteriana y viral en las radiografías.

La neumonía viral y la bacteriana pueden diferenciarse a través de sus características radiográficas en las radiografías de tórax. Neumonía bacteriana: se presenta típicamente con un patrón de opacidad homogénea, que refleja consolidación pulmonar, generalmente de distribución lobar o segmentaria. Esto se debe a la acumulación de

exudado en los alvéolos, lo que resulta en nódulos acinares visibles como pequeñas opacidades mal definidas. Además, la neumonía bacteriana puede mostrar un aumento en el puntaje radiológico, lo que indica una mayor gravedad y extensión de la infección.

Neumonía viral: por otro lado, suele manifestarse con un patrón intersticial o mixto, donde predominan las opacidades en vidrio esmerilado y un engrosamiento de los septos interlobulillares. Este patrón es menos específico y puede variar considerablemente entre los pacientes, lo que dificulta su identificación precisa en comparación con la neumonía bacteriana. En general, la neumonía viral tiende a ser menos grave y puede no mostrar las mismas características de consolidación que se observan en las infecciones bacterianas [34]. En la Figura 3 se puede ver una comparación de las diferentes radiografías con neumonía teniendo presente diferentes patógenos.

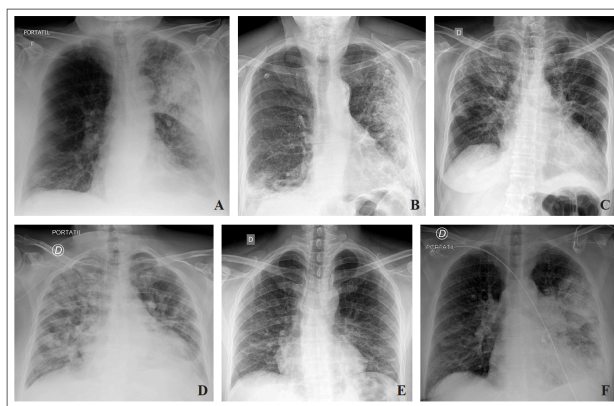


Figura 3: Radiografía de tórax de pacientes adultos inmunocompetentes hospitalizados por neumonía adquirida en la comunidad según el agente causal. A) Neumonía neumocócica; B) Neumonía por bacilos gram negativos; C) Neumonía por virus influenza; D) Neumonía por coronavirus SARS-CoV-2; E) Neumonía por *Chlamydia pneumoniae*; F) Neumonía por *Legionella pneumophila* [35]

2.2 Tuberculosis

2.2.1 Descripción de la enfermedad

La tuberculosis (TB) es una infección bacteriana transmitida por el aire, causada por el *Mycobacterium tuberculosis*, que principalmente afecta los pulmones, aunque puede

involucrar otros órganos y tejidos. Esta enfermedad puede manifestarse de dos formas principales: TB pulmonar y TB extrapulmonar.

La TB pulmonar es la más común y contagiosa, afectando los pulmones y la garganta. Puede presentarse como infección primaria, que aparece inmediatamente después de la infección o como enfermedad post-primaria o del adulto, que surge al menos dos años después de la infección inicial y es más agresiva, causando lesiones pulmonares graves y facilitando la diseminación de la enfermedad en el cuerpo.

Por otro lado, la TB extrapulmonar afecta partes del cuerpo como la columna vertebral, los huesos, los ganglios linfáticos o la piel. Ejemplos de esta forma incluyen la TB miliar, que se disemina por la sangre causando múltiples opacidades en los pulmones; la TB meníngea, que afecta el sistema nervioso central; y la TB renal, que involucra los riñones. Aunque esta forma no es tan contagiosa como la pulmonar, representa un riesgo significativo para la salud [36].

A nivel global, alrededor de 1.8 mil millones de personas están infectadas con TB, pero la mayoría presenta TB latente, que no es activa. De estos, aproximadamente 10 millones desarrollan TB activa cada año. Los grupos más vulnerables incluyen a quienes tienen sistemas inmunológicos debilitados, como personas con VIH, niños, ancianos, y aquellos en tratamiento para enfermedades crónicas. Si bien la TB puede ser grave, es tratable y curable si se siguen las indicaciones médicas adecuadas [37].

2.2.2 Diagnóstico de tuberculosis y limitaciones en los métodos

Para el diagnóstico de TB se implementan una variedad de métodos que incluyen [38]:

- **Baciloscopia del Esputo:** La baciloscopia es un examen microscópico directo que permite detectar bacilos ácido-alcohol resistentes (BAAR) en muestras de esputo.

Es un método económico y accesible, utilizado comúnmente en el diagnóstico inicial.

- Cultivo de Muestras: El cultivo es considerado el estándar de oro para el diagnóstico de TB. Permite el aislamiento del *Mycobacterium tuberculosis* y la realización de pruebas de sensibilidad a fármacos.
- Prueba de Tuberculina (Mantoux): Esta prueba cutánea consiste en la inyección de tuberculina y la evaluación de la reacción en la piel después de 48 a 72 horas.
- Pruebas de Sangre (IGRA): Las pruebas de interferón gamma (IGRA) son análisis de sangre que detectan la respuesta inmune a antígenos específicos de *Mycobacterium tuberculosis*.
- Radiografía de Tórax: Se utiliza para visualizar cambios en los pulmones que pueden indicar TB activa.
- Métodos Moleculares: Las técnicas como la PCR (reacción en cadena de la polimerasa) permiten la detección rápida de *Mycobacterium tuberculosis* en muestras clínicas.

Algunas limitaciones que se pueden presentar en el diagnóstico de la TB incluyen:

- Baciloscopia del Esputo
 - Sensibilidad baja: La baciloscopia tiene una sensibilidad limitada, especialmente en pacientes con formas extrapulmonares de TB o en aquellos con baja carga bacteriana [39].
 - Falsos negativos: Puede dar resultados negativos en casos de TB activa, lo que requiere la confirmación mediante otros métodos.
- Cultivo de muestras
 - Tiempo de Espera: El cultivo puede tardar de 2 a 8 semanas en producir resultados, lo que retrasa el inicio del tratamiento.
 - Requiere condiciones específicas: Necesita un laboratorio equipado y personal capacitado, lo que puede ser un obstáculo en áreas con recursos limitados.
- Prueba de tuberculina

- Falsos Positivos/Negativos: Puede dar resultados falsos positivos en personas que han recibido la vacuna BCG o han estado expuestas a otras micobacterias. Los falsos negativos son comunes en individuos inmunocomprometidos.
- No Distingue Infección Activa de Latente: No permite diferenciar entre TB activa y latente.
- Pruebas de sangre (IGRA)
 - Costo: Suelen ser más costosas que la prueba de tuberculina y requieren laboratorios especializados.
 - No Indican Enfermedad Activa: Al igual que la prueba de tuberculina, no pueden confirmar la presencia de TB activa.
- Radiografía de tórax
 - Interpretación Subjetiva: Los hallazgos radiológicos pueden ser ambiguos y requieren confirmación con métodos microbiológicos.
 - No Especificidad: Puede mostrar cambios en otras enfermedades pulmonares, lo que no confirma la presencia de TB.
- Métodos moleculares:
 - Costo y Accesibilidad: Estas pruebas son más costosas y requieren equipos sofisticados, lo que limita su uso en áreas con pocos recursos.
 - Sensibilidad Variable: Aunque son más rápidas, su sensibilidad puede ser inferior a la del cultivo en algunos casos.

2.2.3 Identificación de características en radiografías de tuberculosis

Las manifestaciones de la enfermedad van a depender del grado de respuesta inflamatoria defensiva del paciente. Se pueden presentar tres presentaciones de la TB pulmonar en las radiografías, primaria, primaria progresiva y postprimaria.

Las presentaciones de neumonía primaria y primaria progresiva, comparten similitudes a la hora de ser identificadas en las radiografías. Se puede observar el complejo de Ghon. Este se presenta como resistencia al TB y localizada en forma de tuberculomas bien definidos con depósitos de fibrina y colágeno (Figura 4) [40].

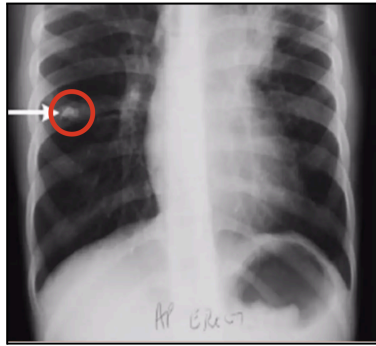


Figura 4: Complejo de Ghon en radiografía [40]

El complejo de Rake consiste en la asociación entre la lesión de Ghon y la calcificación de los ganglios linfáticos hiliares o paratraqueales [41]. Las flechas rojas señalan ganglios calcificados en la región paratraqueal e hilio pulmonar derechos. Las flechas blancas señalan nódulos calcificados que corresponden a granulomas (Figura 5).

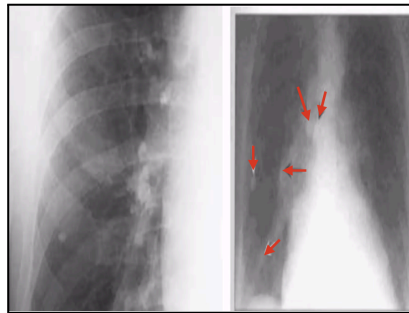


Figura 5: Complejo de Rake [40]

El derrame pleural se presenta como respuesta de hipersensibilidad retardada cuando escasos bacilos llegan a la cavidad pleural. Por esto, el líquido suele caracterizarse por ausencia del bacilo. El derrame generalmente es unilateral, la cantidad de líquido suele ser de escasa a moderada y sin tabicación (Figura 6) [42].



Figura 6: Radiografía con derrame pleural izquierdo [40]

En cuanto a la TB postprimaria, es la forma más frecuente de enfermedad en adultos y se presenta en personas que han desarrollado inmunidad celular y una hipersensibilidad tardía al *Mycobacterium tuberculosis*. En la mayoría de los casos de infección tuberculosa latente, el sistema inmunológico logra controlar la infección. Sin embargo, en algunos individuos, el microorganismo puede reactivarse y multiplicarse, dando lugar a la TB postprimaria [43].

Aunque las manifestaciones radiográficas de la TB postprimaria pueden coincidir con las de la enfermedad primaria, existen varios rasgos distintivos, como la predilección por los lóbulos superiores, la ausencia de linfadenopatías y la tendencia a la formación de cavidades. Además, la consolidación del espacio aéreo es el patrón más común de la enfermedad (Figura 7).

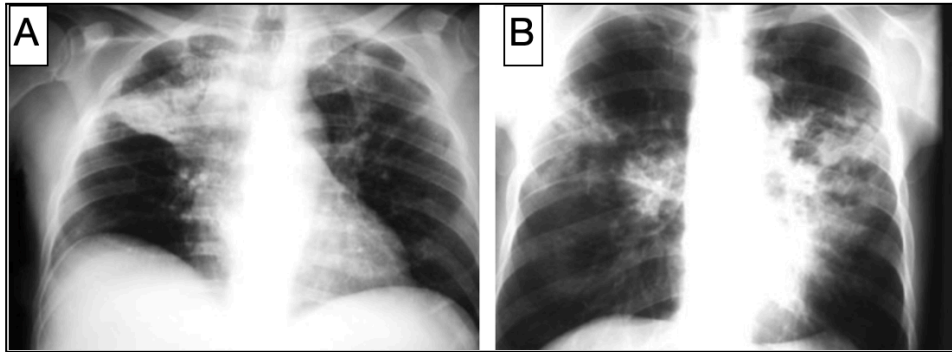


Figura 7: A) Condensación del lóbulo superior derecho, lugar frecuente de afectación en la TB postprimaria. B) Condensaciones múltiples, con un área sugere de excavación, como manifestación de TB [44]

En la mayoría de los casos, se observa una combinación de patrones radiográficos, como la presencia de opacidades lineales, reticulares y nodulares. La formación de cavidades es también un rasgo característico importante de la TB postprimaria. Estas cavidades se desarrollan debido a la necrosis caseosa y, por lo general, contienen una mayor concentración de micobacterias que cualquier otra lesión tuberculosa (Figura 8) [44].

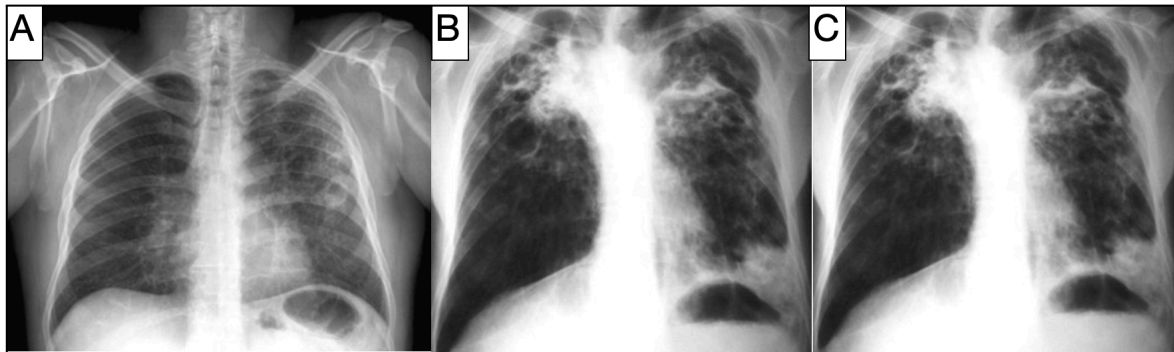


Figura 8: A) Presencia de un nódulo excavado e imágenes parenquimatosas nodulares por diseminación broncogénica. B) TC de Tórax donde se observan condensaciones con excavaciones, en un paciente con TB postprimaria. C) Reducción de volumen del lóbulo superior derecho con tracción del mediastino, tractos fibrosos y excavaciones [44]

2.3 Inteligencia Artificial

La inteligencia artificial o artificial intelligence (AI), término acuñado por John McCarthy en 1955, se define como “la ciencia e ingeniería de hacer máquinas

inteligentes”. En otras palabras, es el desarrollo de algoritmos dedicados a ejecutar tareas complejas que normalmente requieren de la inteligencia humana.

2.3.1 Aprendizaje Automático

El aprendizaje automático, más conocido como machine learning (ML), es un tipo de inteligencia artificial. Es la capacidad que tienen las computadoras de aprender a programarse a sí mismas a través de la experiencia. Es importante destacar que, los datos e información recopilada, se preparan para usarse como datos de entrenamiento. Cuantos más datos, mejor es el programa [45].

Existen tres subcategorías de machine learning, las cuales son aprendizaje supervisado, aprendizaje no supervisado, y aprendizaje por refuerzo. Por una parte, los modelos de aprendizaje supervisado se entrenan con conjuntos de datos etiquetados, lo que permite que los modelos aprendan y se vuelvan más exactos con el tiempo. Por otra parte, los modelos de aprendizaje no supervisado, buscan patrones en datos no etiquetados. Este tipo de aprendizaje puede encontrar patrones o tendencias que no se buscan explícitamente. En cambio, los modelos de aprendizaje por refuerzo entrenan a las máquinas a través de prueba y error para tomar la mejor acción mediante el establecimiento de un sistema de recompensas [46].

2.3.2 Redes Neuronales

El cerebro es una red compleja de neuronas que procesan información a través de un sistema de varias neuronas interconectadas [47]. Cabe resaltar que, una neurona se define como una célula cerebral que transmite señales eléctricas y químicas. En la neurona presináptica, la señal eléctrica se propaga a lo largo del axón hasta llegar a los terminales presinápticos. En los terminales, la señal eléctrica se transforma en una señal química mediante la liberación de neurotransmisores, los cuales son recibidos por las dendritas de

la neurona postsináptica. De tal modo, cuando la neurona postsináptica recibe una cantidad suficiente de neurotransmisores en pocos milisegundos, se generan impulsos eléctricos propios, y el proceso se repite (Figura 9) [48].

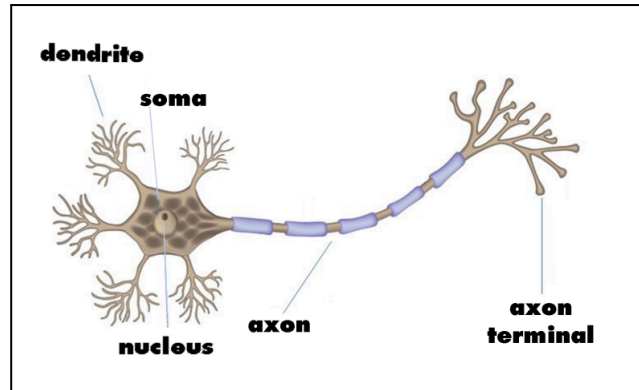


Figura 9: Neurona [49]

Las redes neuronales artificiales o artificial neural networks (ANNs) son modelos de machine learning que sirven para el procesamiento de información. De grosso modo, los ANNs son inspirados en las redes neuronales biológicas o biological neural networks (BNNs), que se encuentran en los cerebros. De manera similar a la estructura de los BNNs, los ANNs definen a la neurona como una unidad central de procesamiento, que es capaz de efectuar operaciones matemáticas para generar una salida, o output, a partir de una entrada, o input. En los ANNs, el output de la neurona artificial es una función de la suma ponderada de las entradas, más el sesgo o bias. Cada neurona realiza una operación que consiste en activarse si es que el input total de la señal recibida excede al umbral de activación, al igual que en los BNNs [49].

La estructura de los ANNs se compone de diferentes capas. En primer lugar se encuentra la capa de entrada, también conocida como input layer, que recibe los inputs del sistema. En segundo lugar se encuentra la capa oculta, o hidden layer, que realiza el procesamiento de la información. En último lugar, se encuentra la capa de salida, o output layer, que proporciona los outputs o resultados finales de la red (Figura 10).

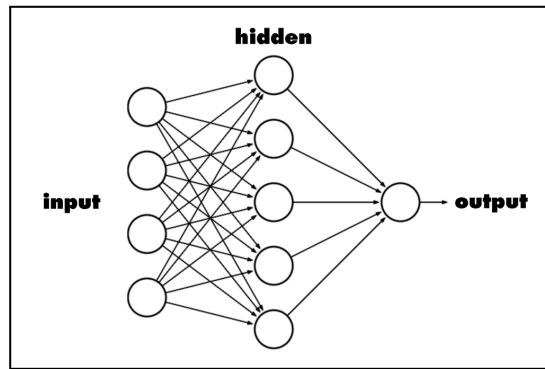


Figura 10: Estructura ANNs [47]

A grandes rasgos, en las redes neuronales, los inputs (x) atraviesan las neuronas, a las cuales se les asignan pesos (w). Posteriormente, los pesos participan en la operación de suma ponderada, que ocurre implícitamente. En esta operación, se realiza una multiplicación entre cada peso y el input correspondiente de la capa, y se suman al sesgo o bias. Finalmente, se aplica la función de activación, la cual es crucial para evitar que la red neuronal se reduzca a una simple regresión lineal. El resultado de la función de activación constituye el output del modelo, representando la salida final de la neurona (Figura 11).

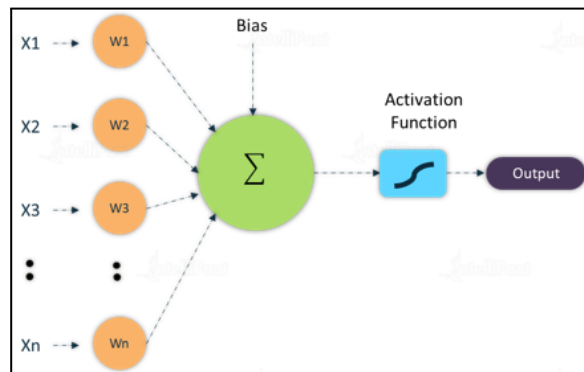


Figura 11: Red Neuronal [50]

Algo semejante ocurre con las redes neuronales convolucionales o convolutional neural networks (CNNs). Los CNNs son redes neuronales de alimentación directa o feedforward, al igual que los ANNs. La diferencia entre ambos recae en que los CNNs se usan específicamente para resolver problemas relacionados a visión computarizada. Las capas convolucionales de los CNNs son un conjunto de filtros, cuyos parámetros se aprenden a lo largo del entrenamiento. De esta manera, cada filtro se aplica deslizándose por toda la imagen de entrada, lo que resulta en la creación de un mapa de activación [51].

Específicamente, la arquitectura de los CNN consta de tres tipos de capas: capa convolucional, capa de pooling y capa totalmente conectada. La capa convolucional se encarga de reconocer características en los píxeles de la imagen, la capa de pooling se encarga de hacer que estas características sean más abstractas y la capa totalmente conectada se encarga de utilizar las características adquiridas para realizar predicciones. En este contexto, las capas convolucionales de la red analizan las radiografías torácicas y extraen automáticamente características relevantes a través de filtros convolucionales. Estos filtros convolucionales, combinados con las operaciones de pooling, permiten detectar patrones y características visuales en diferentes niveles de abstracción. A medida que los datos pasan a través de las capas convolucionales y de pooling, se van extrayendo y resumiendo características cada vez más complejas y abstractas. Entonces, al utilizar una estructura de capas convolucionales predefinidas, el modelo aprende a reconocer características discriminativas sin la necesidad de una extracción manual de características (Figura 12).

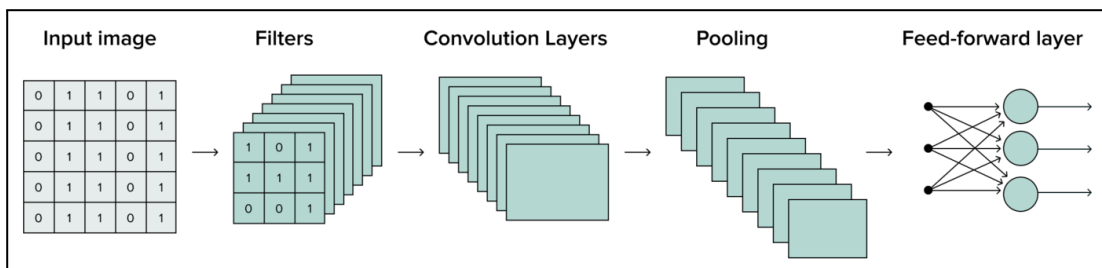


Figura 12: Arquitectura CNNs [52]

Adicionalmente, las redes neuronales recurrentes o recurrent neural networks (RNNs), son redes neuronales que poseen un estado interno, en el que se encuentran parte de los inputs ya alimentados a la red. Este estado cambia constantemente para incorporar los nuevos inputs. Por tal motivo, el output es una combinación de su estado interno y del último input. Debido a estas propiedades, las redes recurrentes se usan en problemas de datos secuenciales, como texto o datos de series temporales.

A continuación, se puede observar la diferencia en estructura entre las redes neuronales recurrentes y las redes neuronales de alimentación directa (Figura 13):

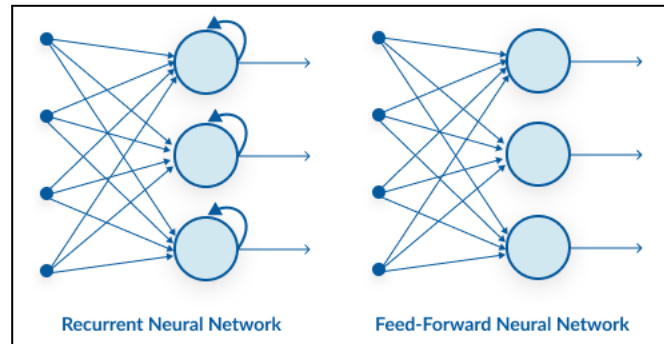


Figura 13: Estructura de redes recurrentes vs. de alimentación directa [53]

Con respecto al aprendizaje profundo o deep learning (DL), es primordial indicar que es un tipo de machine learning. De hecho, es un conjunto complejo de redes neuronales con dos o más capas de procesamiento, hidden layers, que desarrollan altos niveles de abstracción. Típicamente se utilizan para tareas complicadas, como reconocimiento de imágenes, clasificación de imágenes e identificación de escritura a mano [53]. Existen varias clases de deep learning, que se distinguen según su arquitectura de red, entre las cuales se encuentran los CNNs y RNNs.

2.3.2.1 ResNet

ResNet es un tipo específico de red neuronal residual que se usa para el uso de deep learning en aplicaciones de imágenes computarizadas. Se introdujo por primera vez en el 2015 por Kaiming He, Xiangyu Zhang, Shaoqing Ren y Jian Sun en su artículo "Deep Residual Learning for Image Recognition". ResNet está diseñado para abordar el problema del entrenamiento de redes neuronales muy profundas [54]. ResNet utiliza conexiones residuales, para permitir que la información omita ciertas capas y pase directamente a capas más profundas de la red. Esto ayuda a prevenir el problema del gradiente que desaparece, y permite la creación de redes muy profundas con cientos de capas [55]. A su vez, ResNet se compone de bloques de construcción, incluidos bloques básicos, bloques de cuello de botella y bloques de preactivación, que constan de capas convolucionales y conexiones residuales (Figura 14) [56].

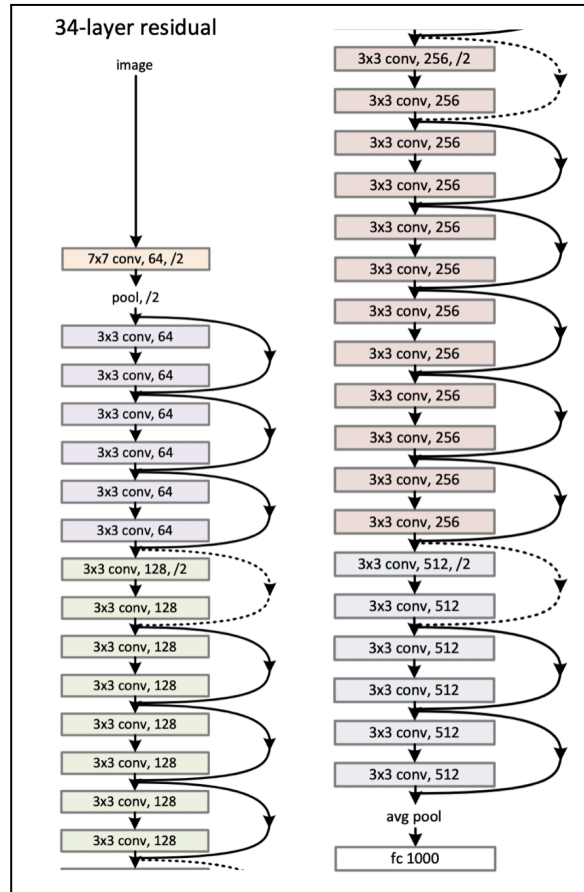


Figura 14: Arquitectura de ResNet de 34 capas [54]

En el ejemplo mostrado se realiza una convolución 3x3 con una dimensión de mapa de características fija [4, 128, 256, 512], omitiendo la entrada cada 2 convoluciones. Las dimensiones de ancho y alto permanecen constantes durante toda la capa. La línea punteada representa un cambio en la dimensión del volumen de entrada, dada la reducción por la convolución. La reducción entre capas se logra mediante un aumento de la zancada, de 1 a 2, en la primera circunvolución de cada capa.

Se tiene el siguiente cuadro (Figura 15) sobre la estructura de las convoluciones y el tamaño de los outputs, correspondientes a cada uno de los modelos dependiendo de la cantidad de capas que se estén implementando.

| layer name | output size | 18-layer | 34-layer | 50-layer | 101-layer | 152-layer |
|------------|-------------|---|---|---|--|--|
| conv1 | 112×112 | 7×7, 64, stride 2 | | | | |
| | | 3×3 max pool, stride 2 | | | | |
| conv2_x | 56×56 | $\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$ | $\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$ | $\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$ | $\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$ | $\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$ |
| conv3_x | 28×28 | $\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$ | $\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$ | $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$ | $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$ | $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$ |
| conv4_x | 14×14 | $\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$ | $\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$ | $\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$ | $\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$ | $\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$ |
| conv5_x | 7×7 | $\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$ | $\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$ | $\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$ | $\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$ | $\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$ |
| | 1×1 | average pool, 1000-d fc, softmax | | | | |
| FLOPs | | 1.8×10^9 | 3.6×10^9 | 3.8×10^9 | 7.6×10^9 | 11.3×10^9 |

Figura 15: Estructura de las convoluciones [54]

Antes de que se entre al comportamiento normal de las capas en ResNet, se tiene una primera capa de convolución, la cual consta de 4 partes:

- La capa convolucional, que realiza la extracción de características de la entrada.
- La normalización por lotes, que normaliza las activaciones de la capa convolucional, mejorando la estabilidad y velocidad del entrenamiento.
- La activación de ReLU, que introduce no linealidad en la red. Esta sigue a la capa de normalización por lotes y permite que el modelo aprenda patrones complejos.
- La operación de agrupación máxima o max pooling, que reduce las dimensiones espaciales del mapa de características, captura las características más destacadas en cada región de la imagen (Figura 16).

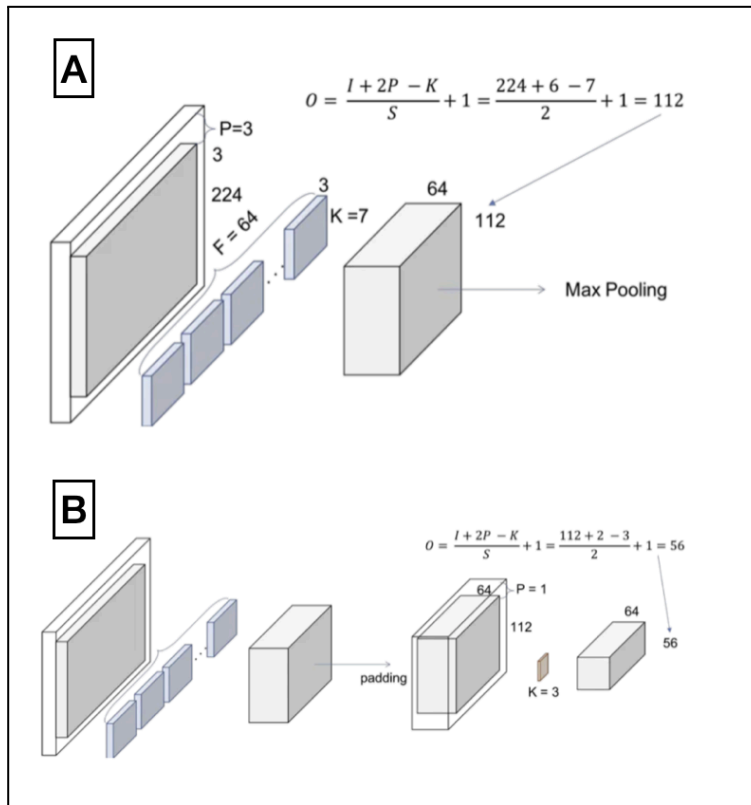


Figura 16: A) Representación de la convolución. B) Max pooling [56]

Para las siguientes capas se repite el mismo proceso. Se puede verificar que se está usando el kernel [3x3, 64] y el tamaño de salida de [56x56]. Se puede observar cómo el tamaño del volumen no cambia dentro de un bloque. Esto se debe a que se utiliza un padding=1 y una zancada también de 1. Seguidamente se mostrará cómo esto se extiende a un bloque completo, para cubrir las dos convoluciones [3x3, 64] (Figura 17).

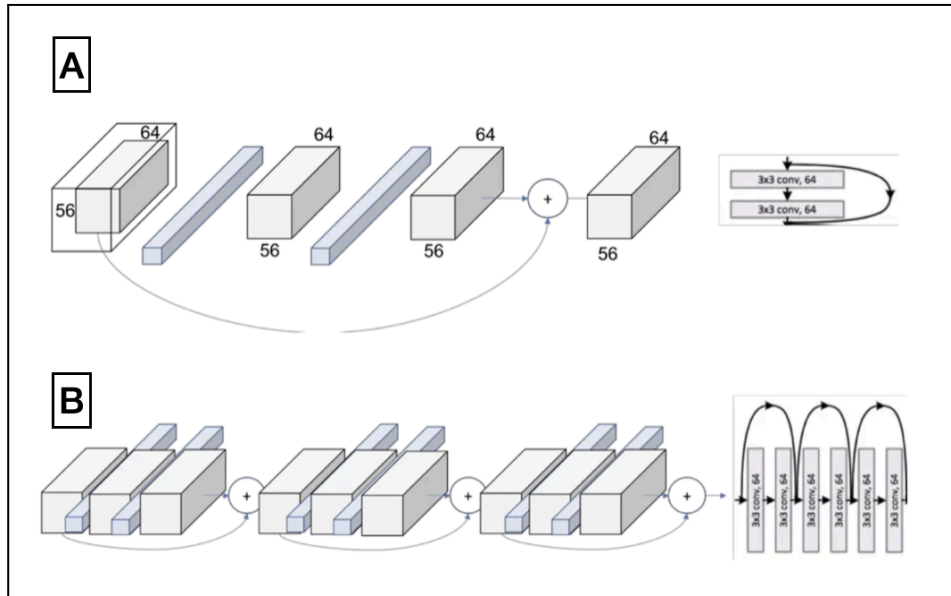


Figura 17: A) proceso de convolución. B) Proceso a escala de convolución [56]

La siguiente parte corresponde al escalamiento de este proceso de bloques para las capas. Se tiene en cuenta que para cada inicio de capa, el paso usado es de 2 en vez de 1. El muestreo reducido del volumen a través de la red se logra aumentando el paso, no usando una operación de agrupación. También, la capa de puntos que representa el cambio de dimensionalidad (Figura 18), es un patrón que se repite sobre las capas de ResNet. La primera operación de cada capa es reducir la dimensión, por lo que también se necesita cambiar el tamaño del volumen que pasa por la conexión de salto, para poder ser agregados.

Esta diferencia en las conexiones de salto se denomina “Atajo de Identidad” y “Atajo de Proyección”. El atajo de identidad es el que se ha comentado previamente, es el paso del volumen de entrada al operador de suma. El acceso directo de proyección realiza una operación de convolución para garantizar que los volúmenes en esta operación de suma sean del mismo tamaño. En este ejemplo, se opta por realizar convoluciones 1x1.

Asimismo, se puede ver (Figura 18) la imagen global de toda la segunda capa. El comportamiento es exactamente el mismo para las siguientes capas, tres y cuatro, cambiando solo las dimensiones de los volúmenes entrantes.

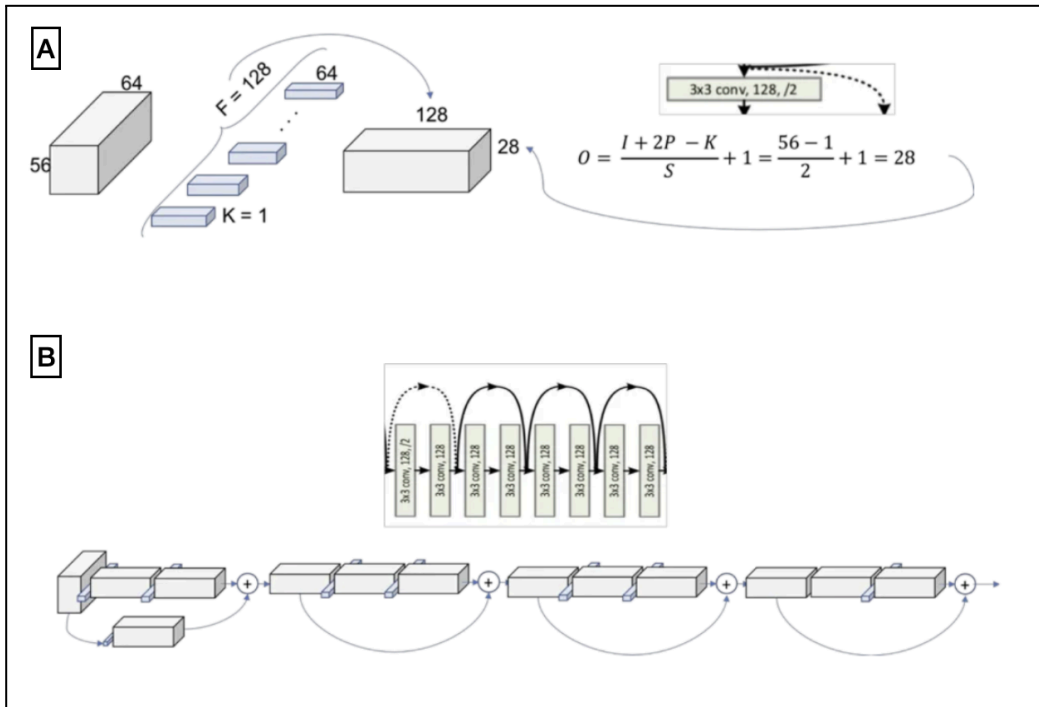


Figura 18: A: cambio de dimensionalidad usando el método de convolución 1x1 B: imagen global de la segunda capa del modelo [56]

2.3.2.2 EfficientNet

EfficientNet es una red neuronal convolucional construida sobre un concepto llamado "escala compuesta". Este concepto aborda la relación entre el tamaño del modelo, la exactitud y la eficiencia computacional. La idea detrás de la escala compuesta es ajustar tres dimensiones esenciales de una red neuronal: ancho, profundidad y resolución [57].

- Ancho: La escala de ancho se refiere al número de canales en cada capa de la red neuronal. Al aumentar el ancho, el modelo puede capturar patrones y características más complejas, lo que resulta en una mejora en la exactitud. Por otro lado, reducir el ancho conduce a un modelo más liviano, adecuado para entornos con recursos limitados.
- Profundidad: La escala de profundidad se refiere al número total de capas en la red. Por un lado, los modelos más profundos pueden capturar representaciones más intrincadas de los datos, pero también requieren más recursos computacionales. Por

otro lado, los modelos más superficiales son más eficientes computacionalmente, pero pueden tener una exactitud reducida.

- Resolución: La escala de resolución implica ajustar el tamaño de la imagen de entrada. Las imágenes de alta resolución proporcionan información más detallada, lo que potencialmente conduce a un mejor rendimiento. Sin embargo, también requieren más memoria y poder computacional. En cambio, las imágenes de baja resolución, consumen menos recursos pero pueden llevar a una pérdida de detalles finos.

En cuanto a la arquitectura de EfficientNet, la capa MBConv es una parte bastante importante del modelo. Esta se basa principalmente en los bloques residuales invertidos de MobileNetV2. La capa MBConv consta de una convolución en profundidad, una convolución puntual (1x1) que aumenta el número de canales, y una convolución 1x1 que reduce el número de canales al inicial. El modelo aprende de manera eficiente con este diseño de cuello de botella. Además de las capas MBConv, EfficientNet incorpora el bloque SE, que ayuda al modelo a aprender a concentrarse en las características esenciales y eliminar las menos relevantes. El mapa de características del bloque SE se reduce en un solo canal y dos capas completamente conectadas mediante la agrupación promedio global. Estas capas permiten que el modelo aprenda las dependencias de características de los canales y cree pesos de atención que se multiplican con el mapa de características original, para así enfatizar la información importante (Figura 19).

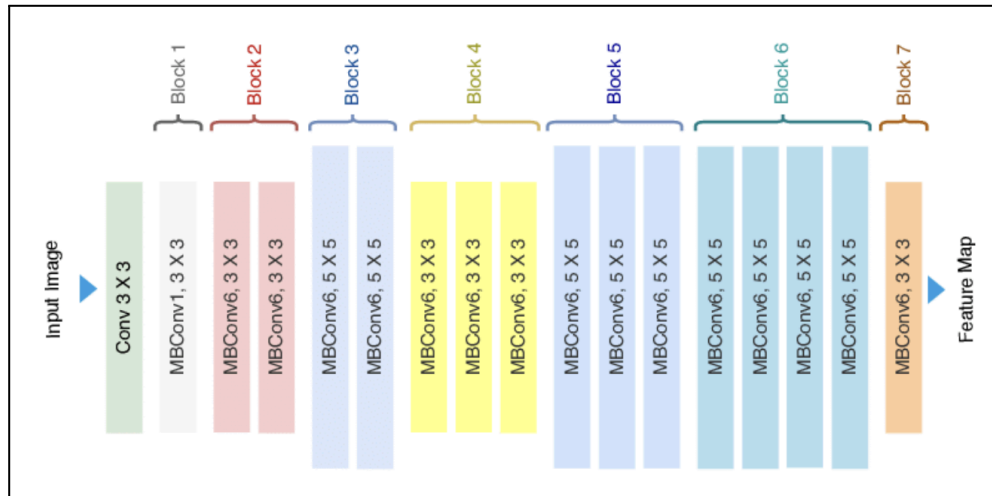


Figura 19: Arquitectura de EfficientNet con MBConv como bloques de construcción básicos [58]

2.4 Métricas de Evaluación

En un problema binario de clasificación, solo existen dos clases posibles: la clase 1, llamada clase positiva, y la clase 0, llamada clase negativa. De este modo, la matriz de confusión es un arreglo bidimensional que contiene la cantidad de instancias clasificadas como 0 o 1, consideradas como valor estimado, y la cantidad de instancias que realmente son 0 y 1, consideradas valor real. Con esta notación, la matriz de confusión se evidencia de la siguiente manera (Figura 20):

| | | | | |
|------------|---------|----------------|---------------|-------|
| | | Valor estimado | | |
| | | $\hat{y} = 1$ | $\hat{y} = 0$ | |
| Valor real | $y = 1$ | TP | FN | n_P |
| | $y = 0$ | FP | TN | n_N |
| | | \hat{n}_p | \hat{n}_N | |

Figura 20: Matriz de confusión 2x2 [59]

- Verdadero Positivo o True Positive (TP): Representan las instancias que han sido clasificadas como 1, siendo su valor real 1.
- Verdadero Negativo o True Negative (TN): Representan las instancias que han sido clasificadas como 0, siendo su valor real 0.

- Falso Positivo o False Positive (FP): Representan las instancias que han sido clasificadas como 1, siendo su valor real 0.
- Falso Negativo o False Negative (FN): Representan las instancias que han sido clasificadas como 0, siendo su valor real 1.

Se debe agregar que, estos elementos de la matriz de confusión son comúnmente utilizados para evaluar el comportamiento de los clasificadores. A continuación se muestran cuáles son las métricas más utilizadas.

- Exactitud o Accuracy (A): Representa la cantidad de instancias correctamente clasificadas con respecto al total de instancias, $(TP+TN)/n$.
- Precisión (P): Indica cuántas instancias, de todas las que han sido clasificadas como positivas, realmente son positivas, $TP/(TP+FP)$.
- Sensibilidad o Recall (R): Indica cuántas instancias que realmente son positivas han sido clasificadas como positivas, $TP/(TP+FN)$.
- Especificidad o Specificity (S): Representa la cantidad de instancias negativas que han sido correctamente clasificadas como negativas, $TN/(FP+TN)$.

Es relevante mencionar que, en un problema de clasificación multiclase, como en el caso de 3 clases, la matriz de confusión es un arreglo tridimensional (Figura 21).

| | | Predicted | | |
|--------|---|-----------|--------|--------|
| | | A | B | C |
| Actual | A | TP_A | FN_A | FN_A |
| | B | FP_A | TN_A | TN_A |
| | C | FP_A | TN_A | TN_A |

Fig. 2: Building blocks for class A in 3-class classification

| | | A | B | C |
|---|--------|--------|--------|--------|
| | | Actual | A | TN_B |
| B | FN_B | | TP_B | FN_B |
| C | TN_B | | FP_B | TN_B |

Fig. 3: Building blocks for class B in 3-class classification

| | | A | B | C |
|---|--------|--------|--------|--------|
| | | Actual | A | TN_C |
| B | TN_C | | TN_C | FP_C |
| C | FN_C | | FN_C | TP_C |

Fig. 4: Building blocks for class C in 3-class classification

Figura 21: Matriz de confusión 3x3 [60]
c

En este caso, las fórmulas para calcular la exactitud, precisión, sensibilidad, y especificidad siguen siendo las mismas que se mencionaron anteriormente para un arreglo bidimensional. No obstante, para cada clase, cambian las posiciones de los TP, TN, FP, y FN, como se evidencia en la imagen.

CAPÍTULO III

MARCO METODOLÓGICO

3.1 Caracterización de la investigación

La presente investigación es de tipo aplicada, puesto que tiene una finalidad práctica y utilitaria, orientada a la mejora de detección y diagnóstico de neumonía y TB en la atención médica. Además, cuenta con un enfoque cuantitativo, dado que se utilizarán técnicas cuantitativas para analizar los datos y entrenar el modelo. El diseño es no experimental, con un alcance descriptivo-propositivo, en el cual se analizarán las características de las radiografías de neumonía y TB. Por lo que, se desarrollará un modelo de deep learning para la detección de neumonía y TB en radiografías torácicas, utilizando un enfoque aplicado, y cuantitativo; con un diseño no experimental y un alcance descriptivo-propositivo.

3.2 Consideraciones éticas

En relación a la confidencialidad, se asegurará la protección de la información privada, así como la preservación de la identificación de los pacientes involucrados en el conjunto de datos utilizado. Se garantizará el respeto a la propiedad intelectual, citando y referenciando correctamente las fuentes utilizadas, y evitando el uso de material o información sin permiso. Se enfatizará la importancia de la honestidad, sin falsificar ni manipular los datos obtenidos en el proceso de desarrollo y evaluación del modelo. Por último, se garantizará la objetividad en la evaluación de la hipótesis planteada, basándose en hechos e información contrastable.

3.3 Desarrollar dos modelos de deep learning para la detección de tuberculosis y dos modelos para la detección de neumonía (viral y/o bacteriana) en radiografías torácicas; utilizando las bases de datos del Instituto Nacional de Alergias y Enfermedades Infecciosas, y del Centro Médico de Mujeres y Niños de Guangzhou, respectivamente

Se siguieron cinco fases para desarrollar los modelos de deep learning:

- Recopilación y procesamiento de imágenes médicas: Se recolectaron y procesaron imágenes de neumonía y TB para entrenar los modelos.
- Diseño del modelo de deep learning: Se crearon las arquitecturas de los modelos utilizando las redes neuronales ResNet y EfficientNet.
- Entrenamiento y validación del modelo de deep learning: Se entrenaron los modelos con las imágenes procesadas, y se validaron para medir su rendimiento.
- Afinación del modelo de deep learning: Se realizaron ajustes y mejoras en los modelos para optimizar su rendimiento.
- Prueba del modelo de deep learning: Se evaluó la efectividad de los modelos para la clasificación de las enfermedades.

En total, se desarrollaron cuatro modelos: “ResNet Neumonía”, “EfficientNet Neumonía”, “ResNet TB”, y “EfficientNet TB”, con el objetivo de seleccionar el mejor modelo para cada patología. Se utilizó Google Colab como entorno de desarrollo para programar y ejecutar los modelos en Python .

3.3.1 Recopilación y procesamiento de imágenes médicas

En primer lugar, se montó Google Drive en el entorno de Google Colab, lo que permitió acceder a los archivos y directorios almacenados en Google Drive directamente desde Colab. “from google.colab import drive” importa el módulo drive de la biblioteca google.colab. “drive.mount('/content/drive')” llama a la función mount() del módulo drive,

que monta Google Drive en el directorio especificado ('/content/drive'). Al ejecutar esta línea, se permite que Google Colab acceda a Google Drive.

En segundo lugar, se importaron las bibliotecas necesarias para trabajar con TensorFlow y procesar las imágenes (Figura 22). “import tensorflow as tf” importa la biblioteca TensorFlow y la nombra como tf, lo que permite acceder a todas las funciones y clases de TensorFlow. “from tensorflow.keras.preprocessing.image import ImageDataGenerator” importa la clase ImageDataGenerator del módulo keras.preprocessing.image de TensorFlow. Esta clase se utiliza para realizar aumentos de datos en imágenes, lo que ayuda a mejorar la generalización del modelo al introducir variaciones en los datos de entrada, como rotaciones y traslaciones. “import tensorflow_hub as hub” importa TensorFlow Hub y la renombra como hub, lo que permite utilizar modelos de deep learning preentrenados, como ResNet y EfficientNet, de manera fácil y eficiente. “import os” importa el módulo os, que proporciona funciones para interactuar con el sistema operativo, y se utiliza comúnmente para manipular rutas de archivos y directorios.

```
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import tensorflow_hub as hub
import os
```

Figura 22: Importe de la librería de Tensorflow

En tercer lugar, se recopilaron las radiografías torácicas de neumonía y TB de Kaggle, y se cargaron a Google Colab. La base de datos de neumonía consistió de 5856 radiografías de tórax del Centro Médico de Mujeres y Niños de Guangzhou, China [61]. Y la base de datos de TB consistió de 1410 imágenes de la Oficina de Infraestructura Cibernética y Biología Computacional (OCICB) del Instituto Nacional de Alergias y Enfermedades Infecciosas (NIAID) de Bethesda, Maryland, Estados Unidos [62]. Las bases de datos se dividieron en tres conjuntos de datos distintos: entrenamiento, validación, y prueba. Los conjuntos de neumonía constaron de 4100, 878, y 878 imágenes, respectivamente. Y los de TB de 988, 211, y 211 imágenes. Es importante destacar que, en ambas bases de datos, el 70% de las imágenes se utilizó para el entrenamiento, el 15% para la validación, y el 15% para la prueba, aproximadamente. El 70% de imágenes para el

entrenamiento proporciona suficientes ejemplos para aprender patrones relevantes, el 15% de imágenes para la validación permite ajustar hiperparámetros y monitorear el rendimiento durante el entrenamiento, evitando el sobreajuste, y, el 15% de imágenes para la prueba garantiza una evaluación objetiva en datos no vistos, lo que asegura que el resultado final sea lo suficientemente robusto y representativo. En la tabla 1, se presenta un resumen de los conjuntos de datos:

| Base de Datos | Conjunto de Datos | | |
|--|-------------------------------------|----------------------------------|------------------------------|
| | Número de Imágenes de Entrenamiento | Número de Imágenes de Validación | Número de Imágenes de Prueba |
| Centro Médico de Mujeres y Niños de Guangzhou | 4100 | 878 | 878 |
| Instituto Nacional de Alergias y Enfermedades Infecciosas | 988 | 211 | 211 |

Tabla 1: Resumen de conjunto de datos.

De este modo, después de subir las imágenes a Google Drive, se establecieron los directorios o rutas de acceso para cada conjunto de datos en Google Colab (Figura 23). En cuarto lugar, se normalizaron todas las imágenes. “`train_datagen = ImageDataGenerator(rescale=1./255)`” crea un objeto `train_datagen` que aplica un escalado a las imágenes de entrenamiento. La operación `rescale=1./255` normaliza los valores de píxeles de las imágenes, que inicialmente están en un rango de 0 a 255, a un rango de 0 a 1, estandarizando así todas las imágenes. Esta normalización es fundamental, ya que mejora la convergencia del modelo durante el entrenamiento, facilitando el aprendizaje de patrones y reduciendo el riesgo de que el modelo se vea afectado por grandes variaciones en la magnitud de los datos. De manera similar, las líneas de código que crean los objetos para las imágenes de prueba y validación (`test_datagen` y `valid_datagen`) también utilizan `ImageDataGenerator(rescale=1./255)`.

```
train_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)
valid_datagen = ImageDataGenerator(rescale=1./255)
```

Figura 23: Normalización.

En quinto lugar, se definieron varios parámetros cruciales para el entrenamiento del modelo (Figura 24). “train_data = train_datagen.flow_from_directory” crea un generador de datos que carga imágenes desde un directorio específico. Al invocar flow_from_directory, se permite que el objeto train_datagen, que es una instancia de ImageDataGenerator, acceda a imágenes organizadas en carpetas, donde cada subcarpeta representa una clase distinta. “directory=train_dir” especifica la ruta al directorio que contiene las imágenes de entrenamiento. En este caso, train_dir hace referencia a la carpeta donde están almacenadas las imágenes organizadas en subcarpetas según sus clases. “batch_size=32” define el tamaño del lote, lo que indica la cantidad de imágenes procesadas simultáneamente en cada iteración; en este caso, se procesaron 32 imágenes a la vez. Este tamaño estándar se considera óptimo para el entrenamiento de redes neuronales porque ofrece un equilibrio entre velocidad, requisitos de memoria y estabilidad; evitando alimentar todas las imágenes del conjunto de datos al modelo de una sola vez, lo que podría resultar en un uso excesivo de memoria y un aumento en el tiempo de procesamiento. Durante cada época, el modelo avanza a través de todos los lotes disponibles, y al finalizar el procesamiento de todos ellos, se completa la época. Este enfoque contribuye a un aprendizaje más eficiente del modelo. “target_size=(224, 224)” establece el tamaño al que se redimensionarán todas las imágenes cargadas. En este caso, todas las imágenes se ajustarán a un tamaño de 224x224 píxeles, un tamaño común para modelos preentrenados que captura detalles significativos sin aumentar excesivamente la complejidad computacional. “class_mode='categorical’” o “class_mode='binary’” especifica el tipo de problema de clasificación que se está abordando: categorical se utiliza para problemas de clasificación multiclase, como la distinción entre “Presencia de Neumonía Viral”, “Presencia de Neumonía Bacteriana”, y “Ausencia de Neumonía Viral y/o Bacteriana”, mientras que binary se aplica a problemas de clasificación binaria, como la detección de “Presencia de TB” o “Ausencia de TB”. “shuffle=True” mezcla

aleatoriamente las imágenes en cada época de entrenamiento, lo que garantiza que el modelo vea lotes únicos y fomenta una mejor generalización. Esto evita que el modelo memorice el orden de los datos y promueve el aprendizaje de patrones generales, reduciendo el riesgo de sobreajuste.

```
train_data = train_datagen.flow_from_directory(directory=train_dir,
                                              batch_size=32,
                                              target_size=(224,224),
                                              class_mode='categorical',
                                              shuffle=True)

test_data = test_datagen.flow_from_directory(directory=test_dir,
                                             batch_size=32,
                                             target_size=(224,224),
                                             class_mode='categorical',
                                             shuffle=True)

valid_data = valid_datagen.flow_from_directory(directory=val_dir,
                                              batch_size=32,
                                              target_size=(224,224),
                                              class_mode='categorical',
                                              shuffle=True)
```

Figura 24: Parámetros Neumonía.

3.3.2 Diseño del modelo de deep learning

Se diseñaron los dos diferentes modelos de deep learning adecuados para el análisis de imágenes médicas, ResNet y EfficientNet. Ambos modelos son destacados en la extracción de características en imágenes. La elección de estos modelos radica en su capacidad para aprender representaciones complejas, adaptarse a diferentes escalas y tamaños de imágenes, y su eficacia en tareas de clasificación.

En primer lugar, se cargó la arquitectura del modelo utilizando una URL proporcionada (Figura 25 y Figura 26).

```
resnet_url = 'https://tfhub.dev/google/imagenet/resnet_v2_50/feature_vector/5'
```

Figura 25: Carga del modelo ResNet

```
efficientnet_url = 'https://tfhub.dev/tensorflow/efficientnet/b0/feature-vector/1'
```

Figura 26: Carga del modelo EfficientNet

“resnet_url = 'https://tfhub.dev/google/imagenet/resnet_v2_50/feature_vector/5'” asigna un URL a la variable resnet_url, la cual apunta a un modelo preentrenado de ResNet

disponible en TensorFlow Hub. Este modelo se utiliza como un vector de características, lo que significa que puede extraer características relevantes de las imágenes. De la misma manera, se cargó la arquitectura del modelo EfficientNet. Es relevante destacar que, cada uno de estos modelos fue programado por separado.

En segundo lugar, se creó un modelo basado en la arquitectura previamente cargada (Figura 27). Es relevante señalar que, las arquitecturas de ResNet/EfficientNet han sido preentrenadas en grandes conjuntos de datos con millones de imágenes clasificadas en miles de categorías diferentes. Estas imágenes son de objetos comunes y no están específicamente relacionadas con enfermedades pulmonares. Aunque las arquitecturas de ResNet/EfficientNet no están especializadas en la detección de enfermedades pulmonares, su capacidad para extraer características generales de objetos comunes se puede aprovechar mediante transfer learning. Por tal motivo, se creó una capa de extracción de características configurada como no entrenable, lo que permite preservar los pesos preentrenados de ResNet/EfficientNet. De tal modo, esta capa identifica y extrae características relevantes de las radiografías, sin modificar los pesos aprendidos. Es así que, esta capa de extracción de características se incorporó a un modelo secuencial de Keras, mediante el cual las capas se apilan de manera secuencial. Adicionalmente, se añadió una capa densa o fully connected, que establece conexiones entre cada neurona de una capa con cada neurona de la siguiente capa. Esta capa está diseñada para aprender y adaptarse a la relación entre las características extraídas y las etiquetas reales del conjunto de datos de entrenamiento.

```
def create_model(mode_url, num_classes=3):
    feature_extractor_layer = hub.KerasLayer(mode_url,
                                             trainable=False,
                                             name='feature_extraction_layer',
                                             input_shape=(224,224,1))

    model = tf.keras.Sequential()
    model.add(feature_extractor_layer)
    model.add(tf.keras.layers.Dense(num_classes, activation='softmax', name='output_layer'))
    return model

resnetmodel = create_model(resnet_url, num_classes=3)
resnetmodel.summary()
```

Figura 27: Creación del modelo ResNet Neumonía.

Por un lado, para las predicciones de neumonía, el modelo incluye una capa densa de salida con 3 neuronas, utilizando la activación softmax, que asigna una probabilidad a cada una de las tres clases: “Presencia de Neumonía Viral”, “Presencia de Neumonía Bacteriana”, y “Ausencia de Neumonía Viral y/o Bacteriana”. La función softmax asegura que la suma de las probabilidades de las tres clases sea 1, y la clase con mayor probabilidad es la predicción final. Por otro lado, para la detección de TB, el modelo incluye una única neurona en la capa de salida, utilizando la activación sigmoide, que produce una probabilidad en el rango de 0 a 1, indicando la probabilidad de que la radiografía examinada presente TB. El umbral de probabilidad se establece en 0.5; esto se debe a que, en un problema de clasificación binaria, este valor marca el punto de equilibrio: si la salida es mayor o igual a 0,5, la predicción se clasifica como positiva (“Presencia de TB”), y si es menor a 0,5, se clasifica como negativa (“Ausencia de TB”). Este enfoque, comúnmente denominado transfer learning, permite a los modelos ResNet y EfficientNet aprovechar el conocimiento aprendido previamente en tareas similares, acelerando y mejorando el rendimiento del entrenamiento en la nueva tarea específica de detección de neumonía y TB.

En tercer lugar, se compiló el modelo creado (Figura 28). Se definieron las funciones de pérdida como `CategoricalCrossentropy` o `BinaryCrossentropy`, apropiados para problemas de clasificación categórica y binaria, respectivamente. Y, se seleccionó el optimizador Adam, reconocido por su eficacia en diversas tareas de optimización, para ajustar los pesos del modelo durante el entrenamiento. Por un lado, durante el proceso de entrenamiento, el modelo realiza predicciones basadas en sus pesos actuales, y estas predicciones se comparan con las etiquetas utilizando la función de pérdida, en este caso, `CategoricalCrossentropy` para neumonía y `BinaryCrossentropy` para TB. Por otro lado, el algoritmo de optimización Adam ajusta los pesos del modelo durante el entrenamiento para minimizar la pérdida. Adam adapta automáticamente las tasas de aprendizaje para cada peso y gestiona el historial de los gradientes, lo que le permite producir predicciones más exactas. Este proceso se repite iterativamente para mejorar la capacidad del modelo para hacer predicciones exactas. Finalmente, se estableció la métrica de exactitud para

evaluar el rendimiento del modelo. Cabe resaltar que, se siguieron los mismos pasos para el diseño de los modelos EfficientNet, solo cambió el URL proporcionado.

```
resnetmodel.compile(loss=tf.keras.losses.CategoricalCrossentropy(),
                    optimizer=tf.keras.optimizers.Adam(),
                    metrics=['accuracy'])
```

Figura 28: Compilación el modelo ResNet Neumonía.

3.3.3 Entrenamiento y validación del modelo de deep learning

Para el entrenamiento de los modelos ResNet y EfficientNet, se utilizaron los conjuntos de datos de entrenamiento. Durante este proceso, estos modelos aprendieron a reconocer y extraer características relevantes de las radiografías torácicas mediante las arquitecturas de ResNet y EfficientNet, respectivamente. Cada uno de estos modelos se entrenó durante 5 épocas, permitiendo así el ajuste de sus pesos y parámetros para mejorar la identificación de neumonía. Al optar por 5 épocas, se buscó asegurar que los modelos tuvieran suficientes oportunidades para aprender los patrones esenciales de las radiografías torácicas sin prolongar innecesariamente el tiempo de entrenamiento ni sobreajustar los datos. Simultáneamente, se llevó a cabo el proceso de validación para cada una de las 5 épocas. Este proceso implicó utilizar los conjuntos de datos de validación. Los modelos se validaron para poder medir su capacidad de realizar predicciones exactas en nuevas radiografías. Esto es esencial para garantizar que los modelos no solo memoricen los datos de entrenamiento, sino que sean capaces de aplicar su aprendizaje a imágenes diferentes (Figura 29).

```
history = resnetmodel.fit(train_data,
                          epochs=5,
                          steps_per_epoch=len(train_data),
                          validation_data=valid_data,
                          validation_steps=len(valid_data),
                          callbacks=[create_tensorboard_callback(dir_name='tensor_hub',
                                                                experiment_name='resnet50V2_pneumonia')])
```

Figura 29: Entrenamiento y Validación del modelo ResNet Neumonía.

3.3.4 Afinación del modelo de deep learning

Se llevó a cabo un aumento de datos para mejorar el rendimiento del modelo. Esta técnica es fundamental para potenciar la capacidad de generalización de los modelos y permitirles aprender patrones más robustos en las imágenes. De este modo, se creó un generador de aumento de datos encargado de aplicar una serie de transformaciones a las imágenes del conjunto de entrenamiento, incluyendo rotaciones, zoom, desplazamientos horizontales y verticales, y volteo horizontal. Estas transformaciones generaron un nuevo conjunto de datos. En este conjunto, las imágenes son versiones transformadas de las originales, lo que incrementa significativamente la variabilidad y diversidad de los ejemplos en el conjunto de entrenamiento (Figura 30).

```
train_datagen_augmented = ImageDataGenerator(rescale=1/255.,
                                             rotation_range=0.2,
                                             shear_range=0,
                                             zoom_range=0.2,
                                             width_shift_range=0.2,
                                             height_shift_range=0.2,
                                             horizontal_flip=True)

train_data_augmented = train_datagen_augmented.flow_from_directory(train_dir,
                                                                    target_size=(224,224),
                                                                    batch_size=32,
                                                                    class_mode='categorical',
                                                                    shuffle=True)
```

Figura 30: Aumento de datos ResNet Neumonía.

Es así que, los modelos fueron nuevamente sometidos a un nuevo entrenamiento, esta vez utilizando el conjunto de datos aumentado, durante un total de 10 épocas. Se decidió usar 10 épocas porque, tras el entrenamiento inicial con 5, se identificó que el modelo aún tenía margen para mejorar su rendimiento y capacidad de generalización. Incrementar el número de épocas permitió al modelo ajustar sus pesos de manera más exacta, aprovechando el conjunto de datos aumentado para aprender características más complejas y sutiles de las imágenes (Figura 31).

```

history2 = resnetmodel.fit(train_data_augmented,
                           epochs=10,
                           steps_per_epoch=len(train_data_augmented),
                           callbacks=[create_tensorboard_callback(dir_name='tensor_hub',
                                                                    experiment_name='resnet50V2_pneumonia_augmented')])

```

Figura 31: Segundo entrenamiento del modelo ResNet Neumonía.

3.3.5 Prueba del modelo de deep learning

Finalmente, se evaluaron los modelos con el conjunto de prueba. Esta evaluación permitió obtener resultados definitivos sobre la capacidad de los dos modelos para detectar neumonía y tuberculosis en radiografías. En la tabla 2, se presenta el resumen de los resultados correspondientes:

| Modelo | ResNet Neumonía | EfficientNet Neumonía | ResNet TB | EfficientNet TB |
|-----------|-----------------|-----------------------|-----------|-----------------|
| Exactitud | 86,06% | 83,33% | 97,62 % | 95,71% |

Tabla 2: Resumen de resultados.

3.4 Evaluar el desempeño del mejor modelo de tuberculosis y del mejor modelo de neumonía (viral y/o bacteriana), utilizando las bases de datos del Hospital Nacional Edgardo Rebagliati Martins y del Hospital Nacional Guillermo Almenara Irigoyen

Se llevó a cabo una evaluación adicional de los dos modelos ResNet utilizando dos bases de datos provenientes de hospitales peruanos. Para evaluar el modelo de detección de neumonía viral y/o bacteriana, se utilizaron 60 imágenes: 30 radiografías con presencia de neumonía viral y/o bacteriana, y 30 con ausencia de neumonía viral y/o bacteriana. De la misma manera, para evaluar el modelo de detección de TB, se utilizaron 89 imágenes: 59 radiografías con presencia de TB, y 30 con ausencia de TB.

Estas imágenes fueron proporcionadas por el Hospital Nacional Edgardo Rebagliati Martins y el Hospital Nacional Guillermo Almenara Irigoyen. Las radiografías de ambos hospitales fueron capturadas entre 2019 y 2023 como parte de la rutina diaria de atención

médica. La base de datos del Hospital Nacional Edgardo Rebagliati Martins incluye 15 radiografías torácicas con presencia de neumonía viral y/o bacteriana y 15 con ausencia de neumonía viral y/o bacteriana. Por otro lado, la base de datos del Hospital Nacional Guillermo Almenara Irigoyen contiene 59 radiografías torácicas con presencia de TB, 15 con presencia de neumonía viral y/o bacteriana, y 45 radiografías con ausencia de TB y neumonía viral y/o bacteriana.

Al evaluar los modelos ResNet desarrollados con estas radiografías de pacientes peruanos, se obtuvo un análisis más amplio y un resultado más representativo del modelo, validando su capacidad de generalización a través de una base de datos diversa. La inclusión de esta variedad de casos es crucial, ya que brinda una comprensión más holística de la aplicabilidad de los modelos en la población peruana, que podría presentar diferencias genéticas y demográficas en comparación con las poblaciones de China y Estados Unidos.

La evaluación consistió básicamente en comparar las etiquetas reales de las 60 radiografías con presencia y ausencia de neumonía viral y/o bacteriana, con las predicciones del modelo ResNet (“Neumonía viral”, “Neumonía bacteriana”, o “Ausencia de Neumonía Viral y/o Bacteriana”). Igualmente, se compararon las etiquetas reales de las 89 radiografías con presencia y ausencia de TB, con las predicciones del modelo ResNet (“Tuberculosis” o “Ausencia de Tuberculosis”).

En el modelo de detección de neumonía viral y/o bacteriana, la etiqueta predicha corresponde a la clase con la mayor probabilidad entre las tres. Por ejemplo, si el modelo predice una probabilidad de 0,4 para la clase 01 (Presencia de Neumonía Viral), 0,5 para la clase 02 (Presencia de Neumonía Bacteriana) y 0,1 para la clase 03 (Ausencia de Neumonía), el resultado final sería “Neumonía Bacteriana”. En el modelo de TB, si la probabilidad predicha es mayor o igual a 0,5, se clasifica como “Tuberculosis”; de lo contrario, se clasifica como “Ausencia de Tuberculosis”. Por ejemplo, si el modelo predice 0,7, el resultado final sería “Presencia de Tuberculosis”.

3.5 Desarrollar una aplicación web para el análisis de radiografías basada en los modelos de detección seleccionados

Para el desarrollo de la interfaz se implementó una plataforma enfocada en brindar servicio para los pacientes y sus radiografías. Se elaboraron dos partes, la parte interna de la plataforma (Back-End), y la parte que se expondrá a los pacientes (Front-End). Para su desarrollo se hizo uso del lenguaje de programación Python .

3.5.1 Back-End

Primero, se importan las librerías necesarias para desarrollar la aplicación web y trabajar con los modelos de deep learning. Utilizamos Flask, un micro-framework de Python para aplicaciones web. Este permite crear y gestionar rutas, manejar solicitudes HTTP, y renderizar plantillas HTML. A su vez, se importan funciones de TensorFlow y Keras para manejar los modelos de deep learning. La aplicación web se inicializa con `app = Flask(__name__)`, la cual define el contexto de la aplicación web, de manera que permite conocer dónde buscar recursos como archivos HTML, modelos, y otros elementos requeridos (Figura 32).

```
from flask import Flask, render_template, request, jsonify
from tensorflow import keras
from tensorflow.keras.applications.resnet50 import preprocess_input,
decode_predictions
from tensorflow.keras.preprocessing import image
import tensorflow_hub as hub
import numpy as np
import base64
app = Flask(__name__)
```

Figura 32: Librerías.

Posteriormente, se cargarán dos modelos pre entrenados, utilizando la función `keras.models.load_model`. Los modelos están instruidos para identificar neumonía (bacteriana y viral) y TB a partir de radiografías, respectivamente. La función también incorpora capas personalizadas de TensorFlow Hub (`KerasLayer`), que permiten reutilizar modelos o capas complejas definidas externamente, optimizando así el rendimiento y la

exactitud. Esta carga inicial de los modelos facilita la ágil ejecución de predicciones cuando la aplicación web recibe solicitudes (Figura 33).

```
neumonia_model =
keras.models.load_model('resnet_pneumonia_model.h5',
custom_objects={'KerasLayer': hub.KerasLayer})
tuberculosis_model = keras.models.load_model('resnet_tb_model.h5',
custom_objects={'KerasLayer': hub.KerasLayer})
```

Figura 33: Carga de modelo.

En esta sección, se definirá las rutas y vistas del marco de trabajo de Flask:

- Ruta Principal (/): Es la ruta base que se accede cuando el usuario visita la página inicial. Acepta solicitudes tanto GET como POST. En esta ruta, simplemente se renderiza la plantilla app.html, que es la interfaz principal para interactuar con la aplicación web.
- Ruta /predict: Esta ruta maneja solicitudes POST y se utiliza para realizar predicciones basadas en las imágenes enviadas. Al recibir una solicitud, llama a la función predict(), que procesa la imagen y ejecuta el modelo correspondiente. La respuesta se devuelve en formato JSON, incluyendo la predicción, el nivel de confianza y cualquier error potencial (Figura 34).

```
@app.route('/', methods=['POST', 'GET'])
def index():
    return render_template('app.html')
@app.route('/predict', methods=['POST'])
def upload():
    label, confidence, error = predict()
    return jsonify({'prediction': label, 'confidence': confidence,
'error': error})
```

Figura 34: Definición de rutas y vistas de la aplicación web.

Finalmente, la función predict() (Figura 35) es la responsable de procesar las imágenes y realizar las predicciones. Primero, verifica si la solicitud incluye un archivo de imagen. En el caso de haber una imagen válida, se guarda temporalmente y se preprocesa para ajustar las dimensiones y la escala de valores, preparándola para ser interpretada por los modelos.

- Predicción de Neumonía: Si el tipo solicitado es "neumonía", el modelo correspondiente se utiliza para realizar la predicción. La función identifica la clase con la mayor probabilidad entre "Neumonía Bacteriana", "Ausencia de Neumonía", y "Neumonía viral". Luego, realiza el cálculo correspondiente y devuelve tanto la etiqueta predicha como la exactitud de la predicción.
- Predicción de TB: Si el tipo es "Tuberculosis", se utiliza el modelo de TB. La predicción devuelve la probabilidad de la clase "Tuberculosis". En consecuencia, dependiendo de si la probabilidad supera un umbral establecido (0,5), la imagen se clasifica como "Tuberculosis" o "Ausencia de Tuberculosis". También devuelve tanto la etiqueta predicha como la exactitud de la predicción.

Si ninguno de los criterios propuestos con anterioridad se cumple, el programa detectará el error y nos mostrará un enunciado para que volvamos a probar con otra imagen válida.

```
def predict():
    if 'imagen' in request.files:
        imagen = request.files['imagen']
        type = request.form.get('type')

        if imagen.filename != '':
            imagen.save(imagen.filename)
            img = image.load_img(imagen.filename, target_size=(224, 224))
            img_array = image.img_to_array(img)
            img_array /= 255.0

            img_array = np.expand_dims(img_array, axis=0)

            if type == 'neumonia':
                class_names = ['Bacteria', 'Normal', 'Virus']
                preds = neumonia_model.predict(img_array)
                predicted_class = np.argmax(preds, axis=1)

                predicted_label = class_names[predicted_class[0]]
                confidence = np.max(preds) * 100
            else:
                prediction = tuberculosis_model.predict(img_array)[0][0]

                threshold = 0.5

                if prediction > threshold:
                    predicted_label = "Tuberculosis"
                    confidence = prediction * 100
                else:
                    predicted_label = "Normal"
                    confidence = (1 - prediction) * 100

            print(f'Predicción: {predicted_label}')
            return predicted_label, confidence, None

    return None, None, "No se seleccionó una imagen válida."
```

Figura 35: Función predict().

3.5.2 Front-End App

El Front-End de la aplicación web se implementó en HTML. Esta parte tiene que ser fácil de entender e intuitiva para el usuario. A su vez, debe permitir visualizar los datos y resultados requeridos para cada paciente según su necesidad, mostrando rápidamente la presencia o ausencia de neumonía y de TB. De tal modo, el usuario puede subir las imágenes desde el dispositivo en uso e inmediatamente obtener el resultado de predicción, mostrando la predicción con respecto a la imagen y el porcentaje de exactitud.

CAPÍTULO IV

RESULTADOS

4.1 Desarrollar dos modelos de deep learning para la detección de tuberculosis y dos modelos para la detección de neumonía (viral y/o bacteriana) en radiografías torácicas; utilizando las bases de datos del Instituto Nacional de Alergias y Enfermedades Infecciosas, y del Centro Médico de Mujeres y Niños de Guangzhou, respectivamente

Primero se imprimió el resumen de cada modelo (Figura 36 y 37) y se evidenciaron los siguientes resultados. El modelo ResNet cuenta con un total de 2049 parámetros entrenables para TB y 6147 para neumonía, mientras que EfficientNet posee únicamente 1281 parámetros entrenables para TB y 3843 para neumonía. Entonces, se puede afirmar que la diferencia en el número de parámetros entre ambos modelos definitivamente tuvo implicaciones en términos de capacidad de aprendizaje y eficiencia computacional, puesto que resnet obtuvo mejor rendimiento.

| A | | |
|---|--------------|----------|
| Model: "sequential" | | |
| Layer (type) | Output Shape | Param # |
| feature_extraction_layer (KerasLayer) | (None, 2048) | 23564800 |
| output_layer (Dense) | (None, 1) | 2049 |
| ===== | | |
| Total params: 23566849 (89.90 MB) | | |
| Trainable params: 2049 (8.00 KB) | | |
| Non-trainable params: 23564800 (89.89 MB) | | |

| B | | |
|---|--------------|----------|
| Model: "sequential" | | |
| Layer (type) | Output Shape | Param # |
| feature_extraction_layer (KerasLayer) | (None, 2048) | 23564800 |
| output_layer (Dense) | (None, 3) | 6147 |
| ===== | | |
| Total params: 23570947 (89.92 MB) | | |
| Trainable params: 6147 (24.01 KB) | | |
| Non-trainable params: 23564800 (89.89 MB) | | |

Figura 36: A) Parámetros entrenables de ResNet TB. B) Parámetros entrenables de ResNet neumonía.

| A | | |
|--|--------------|---------|
| Model: "sequential" | | |
| Layer (type) | Output Shape | Param # |
| feature_extraction_layer (KerasLayer) | (None, 1280) | 4049564 |
| output_layer (Dense) | (None, 1) | 1281 |
| Total params: 4050845 (15.45 MB) | | |
| Trainable params: 1281 (5.00 KB) | | |
| Non-trainable params: 4049564 (15.45 MB) | | |
| B | | |
| Model: "sequential" | | |
| Layer (type) | Output Shape | Param # |
| feature_extraction_layer (KerasLayer) | (None, 1280) | 4049564 |
| output_layer (Dense) | (None, 3) | 3843 |
| Total params: 4053407 (15.46 MB) | | |
| Trainable params: 3843 (15.01 KB) | | |
| Non-trainable params: 4049564 (15.45 MB) | | |

Figura 37: A) Parámetros entrenables de EfficientNet TB. B) Parámetros entrenables para EfficientNet neumonía.

Durante la primera parte del primer entrenamiento, el modelo ResNet presentó una exactitud de entrenamiento de 96,36% en TB y de 83,09% en neumonía. El modelo de EfficientNet presentó una exactitud de entrenamiento de 97,98% en TB y de 80,45% en neumonía. En segunda instancia, el modelo ResNet presentó una pérdida de entrenamiento de 12,56% en TB y de 46,04% en neumonía. El modelo de EfficientNet presentó una pérdida de entrenamiento de 12,87% en TB y de 45,90% en neumonía (Figura 38).

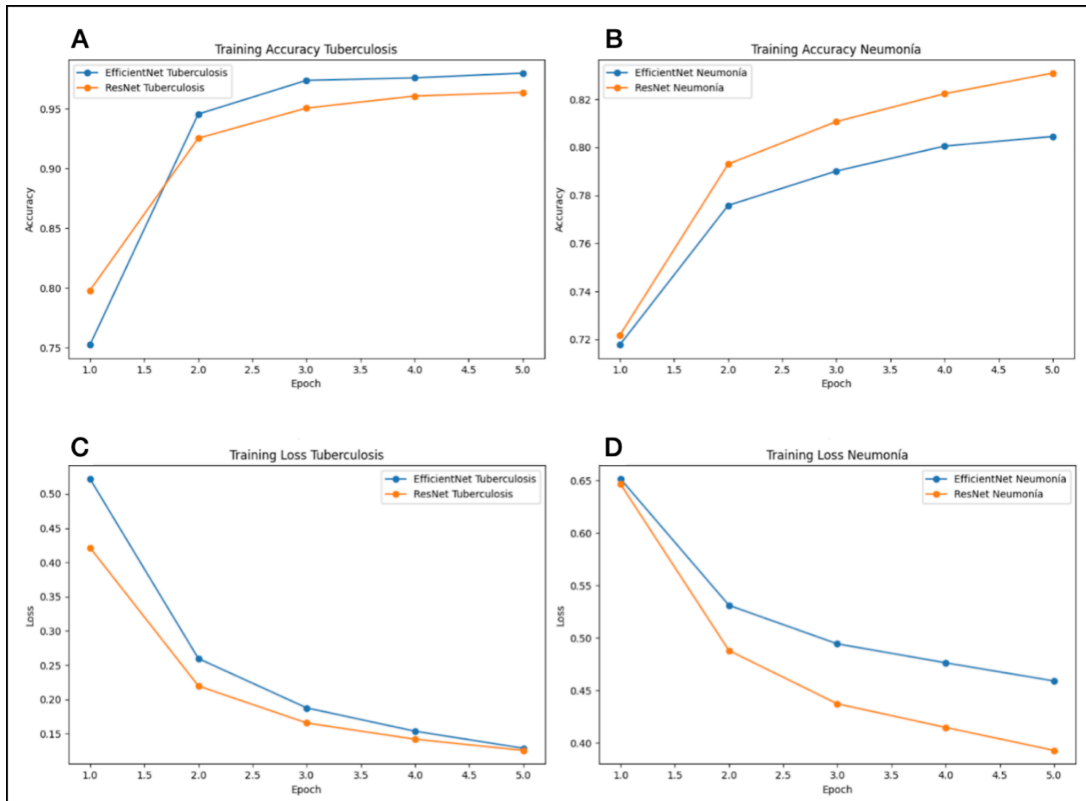


Figura 38: A) Comparación de resultados de Training accuracy entre los modelos de TB en 5 épocas. B) Comparación de resultados de Training accuracy entre los modelos de neumonia en 5 épocas. C) Comparación de resultados de Training Loss entre los modelos de TB en 5 épocas. D) Comparación de resultados de Training Loss entre los modelos de neumonia en 5 épocas.

Asimismo, se obtuvieron resultados de exactitud de validación y pérdida de validación. El modelo de ResNet presentó una exactitud de validación de 97,14% en TB y de 72,22% en neumonia. El modelo de EfficientNet presentó una exactitud de validación de 96,19% en TB y de 72,22% en neumonia. Posteriormente, el modelo ResNet presentó una pérdida de validación de 11,04% en TB y de 87,67% en neumonia. El modelo EfficientNet presentó una pérdida de validación de 12,78% en TB y de 55,59% en neumonia (Figura 39).

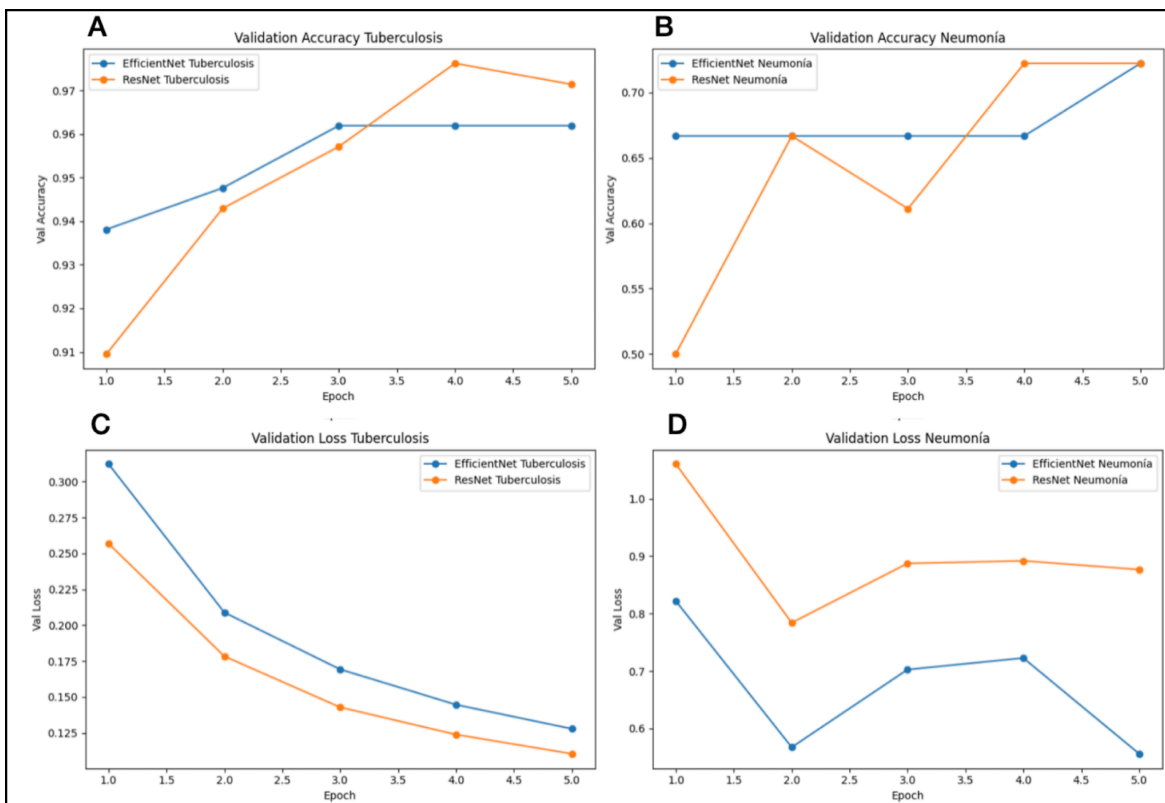


Figura 39: A) Comparación de resultados de Validation Accuracy entre los modelos de TB en 5 épocas. B) Comparación de resultados de Validation Accuracy entre los modelos de neumonía en 5 épocas. C) Comparación de resultados de Validation Loss entre los modelos de TB en 5 épocas. D) Comparación de resultados de Validation Loss entre los modelos de neumonía en 5 épocas.

Posteriormente, durante el segundo entrenamiento tras un aumento de datos se obtuvieron resultados de exactitud y de pérdida. El modelo de ResNet presentó una exactitud de 95,96% en TB y de 80,45% en neumonía. El modelo de EfficientNet presentó una exactitud de 96,46% en TB y de 80,83% en neumonía. Asimismo, el modelo ResNet presentó una pérdida de 11,04% en TB y de 46,04% en neumonía. El modelo de EfficientNet presentó una pérdida de 11,96% en TB y de 45,66% en neumonía (Figura 40).

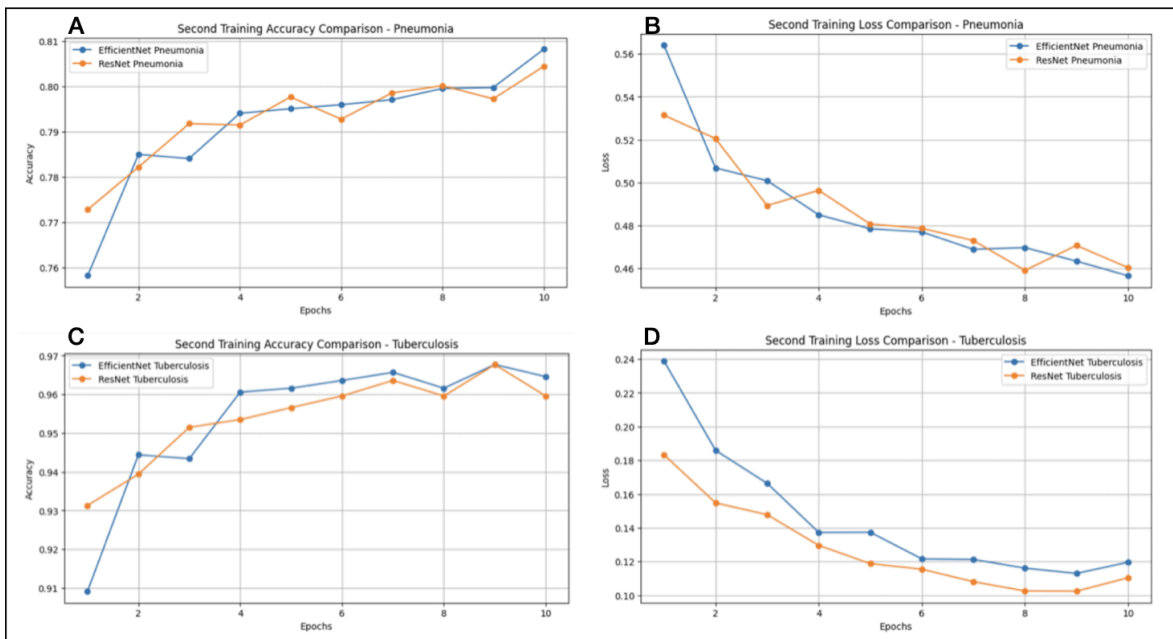


Figura 40: A) Comparación de resultados de Second Training Accuracy entre los modelos de neumonía en 10 épocas. B) Comparación de resultados de Second Training Loss entre los modelos de neumonía en 10 épocas. C) Comparación de resultados de Second Training Accuracy entre los modelos de TB en 10 épocas. D) Comparación de resultados de Second Training Loss entre los modelos de TB en 10 épocas.

Finalmente se realizó una prueba final donde se obtuvieron los resultados de validación pérdida de y de exactitud finales. El modelo de ResNet presentó una pérdida de validación de 7,46% en TB y de 41,73% en neumonía. El modelo de EfficientNet presentó una pérdida de validación de 10,59% en TB y de 46,91% en neumonía. En cuanto a la exactitud de validación, ResNet obtuvo 97,62% en TB y 86,06% en neumonía, mientras que EfficientNet obtuvo un valor de 95,71% en TB y 83,03% en neumonía (Figura 41).

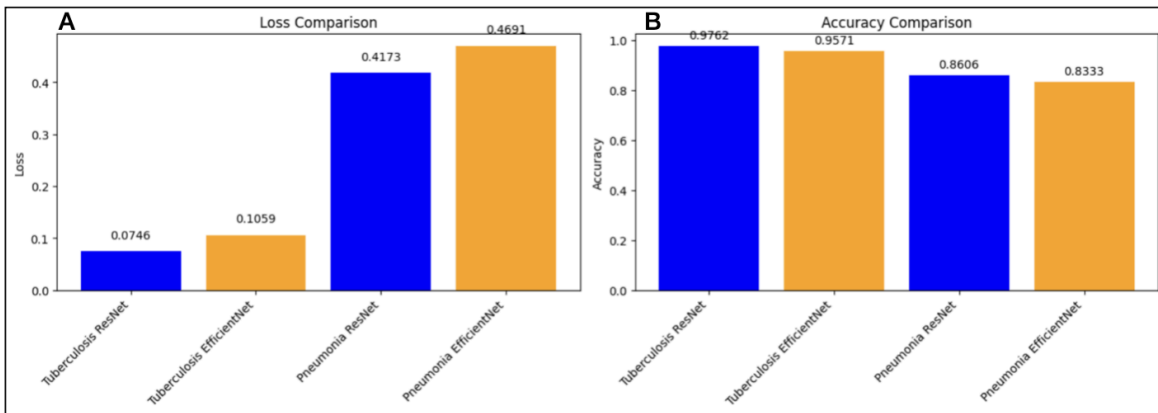


Figura 41: A) Comparación de resultados finales de Accuracy entre los modelos de neumonía y TB. B) Comparación de resultados finales de Loss entre los modelos de neumonía y TB.

4.2 Evaluar el desempeño del mejor modelo de tuberculosis y del mejor modelo de neumonía (viral y/o bacteriana), utilizando las bases de datos del Hospital Nacional Edgardo Rebagliati Martins y del Hospital Nacional Guillermo Almenara Irigoyen

Para la evaluación de los modelos, se utilizó la matriz de confusión, una herramienta crucial en la clasificación. En la siguiente imagen se muestra la matriz de confusión del modelo de detección de TB (Figura 42).

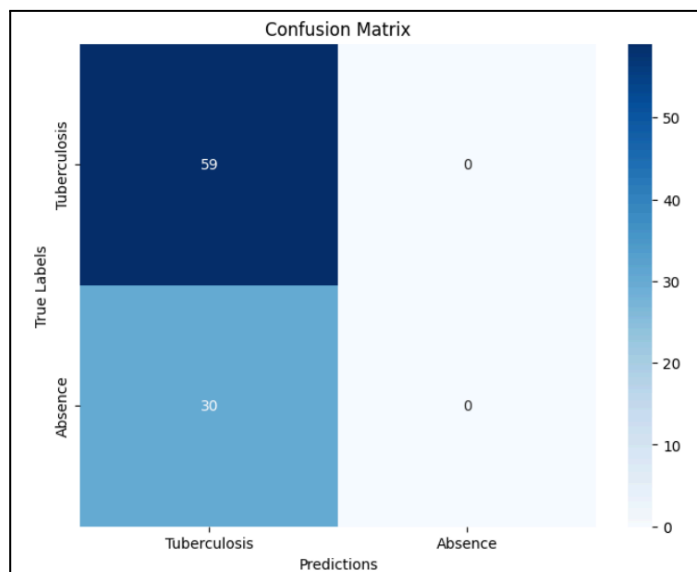


Figura 42: Predicciones de TB.

A partir de esta matriz, se obtienen los siguientes resultados:

- Exactitud = $(TP + TN) / n = (59 + 0) / (59 + 0 + 30 + 0) = 62,29\%$
- Precisión = $TP / (TP + FP) = 59 / (59 + 0) = 62,29\%$
- Sensibilidad = $TP / (TP + FN) = 59 / (59 + 0) = 100\%$
- Especificidad = $TN / (TN + FP) = 0 / (0 + 30) = 0\%$

Y en la Figura 43 se muestra la matriz de confusión del modelo de detección de neumonía viral y/o bacteriana.

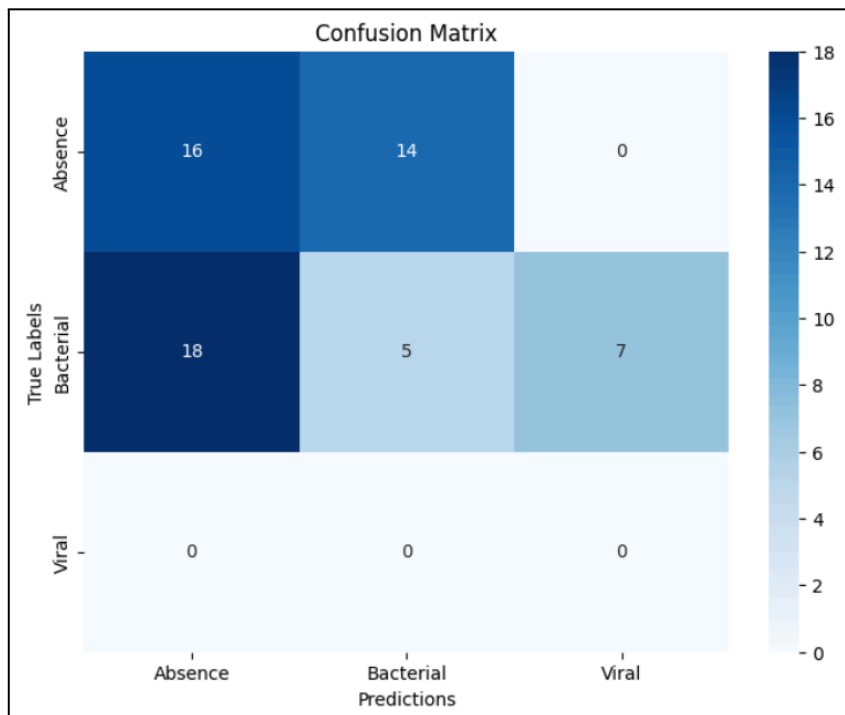


Figura 43: Predicciones de neumonía viral y/o bacteriana.

Y a partir de esta matriz, se obtienen los siguientes resultados:

- Exactitud = $(TP + TN) / n = 21 / 60 = 35\%$
- Precisión (Clase 0) = $TP / (TP + FP) = 16 / 34 = 47,06\%$
- Precisión (Clase 1) = $TP / (TP + FP) = 5 / 19 = 26,32\%$
- Precisión (Clase 2) = $TP / (TP + FP) = 0\%$
- Sensibilidad (Clase 0) = $TP / (TP + FN) = 8 / 15 = 53,33\%$

- Sensibilidad (Clase 1) = $TP / (TP + FN) = 1 / 6 = 16,67\%$
- Sensibilidad (Clase 2) = $TP / (TP + FN) = 0\%$
- Especificidad (Clase 0) = $TN / (TN + FP) = 5 / 19 = 40,00\%$
- Especificidad (Clase 1) = $TN / (TN + FP) = 16 / 30 = 53,33\%$
- Especificidad (Clase 2) = $TN / (TN + FP) = 53 / 60 = 88,33\%$

4.3 Desarrollar una aplicación web para el análisis de radiografías basada en los modelos de detección seleccionados

La página web diseñada (Figura 44) presenta cuatro botones principales: uno para cargar imágenes desde los archivos locales, otros dos para seleccionar si se desea analizar TB o neumonía y uno más para examinar y realizar el correspondiente análisis de la imagen brindada.



Figura 44: Página web.

En primera instancia, se eligió la imagen que se deseaba evaluar. Se optó por analizar una radiografía de tórax que evidenciaba un estado de salud no sintomático de TB. Esta imagen fue recuperada de una carpeta local que contiene diagnósticos conocidos de radiografías, obtenidas de diversas fuentes. Cuando se evaluó con respecto a la opción de TB, nos brindó un porcentaje de exactitud del 97,64% (Figura 45).



Figura 45: Identificación de imagen sin TB y resultado de exactitud.

En segunda instancia se evaluó el modelo para determinar su eficacia en la detección de neumonía viral. Se seleccionó una radiografía de tórax con un diagnóstico confirmado de neumonía viral, la cual fue procesada a través del modelo previamente desarrollado. El sistema arrojó un resultado de exactitud del 86,36% (Figura 46).



Figura 46: Identificación de radiografía de tórax con neumonía viral y resultado de exactitud.

Seguidamente se procedió a evaluar el modelo para determinar su eficacia para detectar la ausencia de neumonía. Se seleccionó una radiografía de tórax con un diagnóstico confirmado de ausencia de neumonía, la cual fue procesada a través del

modelo previamente desarrollado, obteniendo un resultado de exactitud de 99,37% (Figura 47).



Figura 47: Ausencia de neumonía viral y/o bacteriana.

Posteriormente se volvió a realizar otra evaluación del modelo para determinar su eficacia para detectar la presencia de TB. Se seleccionó una radiografía de tórax con un diagnóstico confirmado de presencia de TB, la cual fue procesada a través del modelo previamente desarrollado, obteniendo un porcentaje de 99,14% de exactitud (Figura 48).

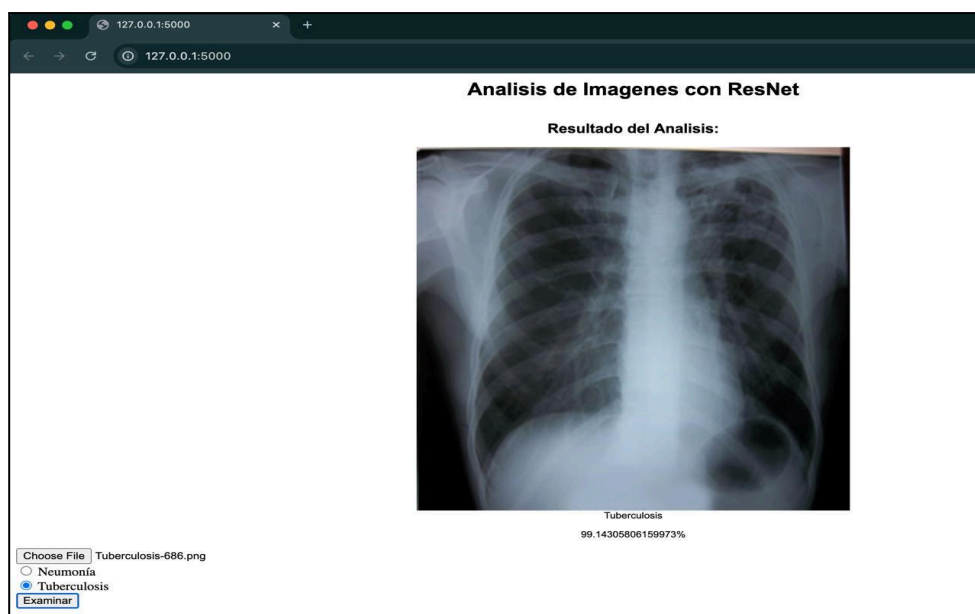


Figura 48: Presencia de TB.

Finalmente, se realizó una última evaluación del modelo para determinar su eficacia para detectar la presencia de neumonía bacteriana. Se seleccionó una radiografía de tórax con un diagnóstico confirmado de presencia de neumonía bacteriana, la cual fue procesada a través del modelo previamente desarrollado, obteniendo un porcentaje de 85,69 % de exactitud (Figura 49).

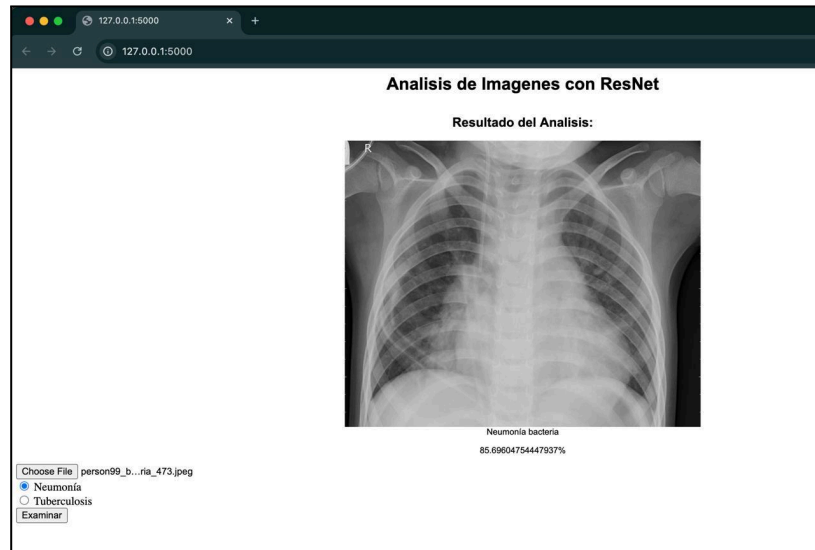


Figura 49: Presencia de neumonía bacteriana.

DISCUSIÓN

5.1. Desarrollar dos modelos de deep learning para la detección de tuberculosis y dos modelos para la detección de neumonía (viral y/o bacteriana) en radiografías torácicas; utilizando las bases de datos del Instituto Nacional de Alergias y Enfermedades Infecciosas, y del Centro Médico de Mujeres y Niños de Guangzhou, respectivamente

En el presente estudio, se compararon dos arquitecturas de deep learning, ResNet y EfficientNet, para la detección de TB y neumonía a partir de radiografías de tórax. Los resultados obtenidos reflejan diferencias en el desempeño de ambos modelos, tanto en términos de exactitud como de pérdida, durante las fases de entrenamiento, validación y pruebas finales.

Antes del entrenamiento, se realizó imprimió un resumen de cada modelo donde se pudo visualizar que el modelo de ResNet trabaja con una mayor cantidad parámetros entrenables que en EfficientNet, ya sea en TB o en neumonía. Por lo que se puede afirmar que la diferencia en el número de parámetros entre ambos modelos definitivamente tuvo implicaciones en términos de capacidad de aprendizaje, puesto que ResNet obtuvo mejor rendimiento.

En un primer entrenamiento ambos modelos mostraron un alto rendimiento en la clasificación de TB. EfficientNet tuvo mejores resultados con respecto a ResNet en cuanto a TB, teniendo el primero un porcentaje de 97,98% y el segundo un porcentaje de 96,36%. Sin embargo, ResNet obtuvo mejores resultados en el ámbito de neumonía (83,09%) en comparación a EfficientNet (80,45%). En cuanto a la pérdida de entrenamiento, ResNet tuvo un menor resultado (12,56%) en comparación a EfficientNet (12,87%) pero un mayor resultado (46,04%) para neumonía que en la segunda arquitectura (45,90%). Es importante que los porcentajes de pérdida se mantengan bajos porque se refiere a qué tan bien está ajustando el modelo a los datos de entrenamiento. Un valor bajo de pérdida de entrenamiento indica que el modelo está aprendiendo bien a clasificar los datos en ese conjunto.

A continuación, se realizó la primera validación donde se obtuvieron unos resultados bastante altos en cuanto a la pérdida para el ámbito de neumonía, obteniendo en el modelo de EfficientNet 55,59% y en el modelo ResNet 87,67%. El alto porcentaje de pérdida se debe a la poca cantidad de épocas que se tuvo en este primer entrenamiento, también que no haya tenido suficiente tiempo para ajustar adecuadamente los pesos y captar patrones relevantes en los datos en las pocas épocas en las que se entrenó.

Seguidamente se realizó un segundo entrenamiento donde se implementaron 10 épocas. En EfficientNet presentó una exactitud ligeramente mayor (96,46%) en comparación a ResNet (95,96%) en TB, mientras que en neumonía Resnet (80,45%) es ligeramente mayor a EfficientNet (80,83%). Por el lado de la pérdida, ResNet (46,04%) presentó una disminución mayor en comparación al primer entrenamiento de neumonía. Esto se debe principalmente al aumento de épocas de entrenamiento, donde el modelo tiene más oportunidades de refinar los pesos de las capas y mejorar su capacidad para capturar patrones más complejos en las imágenes de neumonía.

Finalmente, los resultados obtenidos tras la validación final, se observó que el modelo de ResNet demostró un mejor rendimiento comparado con EfficientNet en la detección tanto de TB como de neumonía, con una menor pérdida de validación final (7,46% vs. 10,59% en TB y 41,73% vs. 46,91% en neumonía) y una mayor exactitud (97,62% vs. 95,71% en TB y 86,06% vs. 83,03% en neumonía).

Comparando estos resultados con estudios previos, como el de Mehrrotra et al. [25], quienes lograron una precisión del 99,10% en la detección de TB, el rendimiento de ResNet es competitivo, aunque el ligero descenso en la exactitud podría deberse a diferencias en los conjuntos de datos utilizados y en las características específicas de la muestra. Asimismo los resultados de ResNet en neumonía son comparables con los de Loyola y Chamorro [21], quienes obtuvieron un 86,32%, lo que refuerza el impacto del uso de redes profundas como ResNet para mejorar los resultados diagnósticos en neumonía.

El modelo de EfficientNet, aunque menos preciso que ResNet en este caso, mostró un desempeño robusto, similar al observado por S. Bharati et al. [23], quienes lograron una

precisión del 73% en la detección de enfermedades pulmonares. Los resultados sugieren que EfficientNet es eficaz, pero ResNet maneja mejor la complejidad del conjunto de datos y la detección simultánea de ambas patologías, justificando su elección en este contexto.

Por lo tanto, aunque ambos modelos presentan un rendimiento adecuado, ResNet se destaca por ofrecer mejores resultados finales, especialmente en TB, lo que coincide con estudios como el de Kvak et al. [24], donde las redes convolucionales también superaron a los radiólogos en la identificación de anormalidades en radiografías de tórax. Este rendimiento más consistente de ResNet en distintas pruebas respalda su elección como modelo preferido para el diagnóstico de TB y neumonía en este estudio.

5.2. Evaluar el desempeño del mejor modelo de tuberculosis y del mejor modelo de neumonía (viral y/o bacteriana), utilizando las bases de datos del Hospital Nacional Edgardo Rebagliati Martins y del Hospital Nacional Guillermo Almenara Irigoyen

En la evaluación de los modelos utilizando las bases de datos peruanas, el modelo de detección de TB presenta una serie de métricas que son cruciales para su interpretación clínica. A continuación se detallan los resultados obtenidos:

La exactitud del 62,29% sugiere que, en general, el modelo logra identificar correctamente aproximadamente 62 de cada 100 casos en la población estudiada. Sin embargo, esta cifra debe interpretarse con cautela, ya que se tiene un número significativo de falsos positivos y/o negativos que pueden influir en la validez del diagnóstico en un contexto clínico. Esto es especialmente relevante en la detección de TB, donde un diagnóstico erróneo puede tener consecuencias graves para el paciente y la salud pública.

La precisión del 62,29% indica que, de todos los casos que el modelo clasificó como positivos para TB, el 62,29% realmente son positivos. En términos clínicos, esto significa que hay una probabilidad considerable de que un diagnóstico positivo sea erróneo. Esto es crítico, ya que un resultado falso positivo podría llevar a un tratamiento innecesario, efectos secundarios y preocupaciones adicionales para el paciente.

La sensibilidad del 100% es altamente positiva en el contexto de la detección de TB, lo que significa que el modelo identifica correctamente todos los casos verdaderos de TB en la muestra analizada. En la práctica clínica, esto es crucial, ya que ningún paciente con TB sería pasado por alto, minimizando el riesgo de transmisión de la enfermedad y permitiendo un tratamiento oportuno.

La especificidad del 0% es preocupante, ya que indica que el modelo no identifica correctamente ningún caso negativo (ausencia de TB). En la práctica clínica, esto puede resultar en un alto número de falsos positivos, donde pacientes sin TB podrían ser diagnosticados erróneamente con TB. Esto podría conducir a tratamientos innecesarios y ansiedad en los pacientes, además de poner en riesgo la salud pública al generar confusiones sobre la prevalencia de la enfermedad.

Y para la evaluación del modelo de detección de neumonía viral y/o bacteriana, se detallan los siguientes resultados obtenidos:

La exactitud de 35%, sugiere que hay un alto margen de error, lo cual es preocupante en un contexto clínico, ya que un diagnóstico inexacto puede tener serias repercusiones en la salud del paciente.

La precisión del 47,06% indica que, de los casos clasificados como negativos (ausencia de neumonía), sólo alrededor del 47% son realmente negativos, lo cual resalta la alta posibilidad de falsos negativos. La precisión del 26,32%, indica que el modelo clasifica incorrectamente más del 70% de los casos de neumonía bacteriana. Esto significa que un diagnóstico positivo puede ser poco confiable. La precisión de 0% indica que el modelo no identifica ningún caso de neumonía viral correctamente, lo que representa un fallo crítico en el diagnóstico, ya que ningún paciente con esta condición sería detectado.

La sensibilidad del 53,33%, indica que el modelo identifica más de la mitad de los casos negativos correctamente, pero todavía hay un considerable número de falsos negativos. La sensibilidad del 16,67% indica que el modelo es incapaz de detectar adecuadamente los casos de neumonía bacteriana, lo que plantea riesgos significativos para el tratamiento del paciente y la salud pública. La sensibilidad del 0% indica que el modelo

no identifica ningún caso de neumonía viral, lo que podría llevar a un brote de esta enfermedad sin un diagnóstico oportuno.

La especificidad del 40% indica que el modelo no discrimina correctamente entre la ausencia y presencia de neumonía viral y/o bacteriana, lo que puede llevar a un alto número de falsos positivos. La especificidad del 53,33% indica que el modelo no está logrando una correcta identificación de los negativos en esta clase, aumentando el riesgo de diagnósticos erróneos. La especificidad del 88,33%, indica que el modelo muestra un desempeño relativamente mejor en la identificación de negativos en esta clase, lo que sugiere que un alto porcentaje de los pacientes sin neumonía viral son correctamente clasificados.

Los resultados obtenidos con el modelo ResNet muestran una alta exactitud en la identificación de neumonía bacteriana y viral, así como en la detección de TB en radiografías. Sin embargo, es importante destacar que, aunque las cifras de exactitud son prometedoras, el modelo ha sido evaluado con un conjunto de imágenes limitado. Además, factores como la calidad y el posicionamiento de las radiografías pueden influir significativamente en el rendimiento del modelo. Por lo tanto, es fundamental ampliar las pruebas con un conjunto de datos más variado para validar mejor su eficacia en distintos entornos clínicos.

Este aspecto subraya que, a pesar del rendimiento alto actual, el modelo debe ser sometido a un análisis más exhaustivo para determinar su robustez frente a variaciones en las condiciones de las imágenes, como cambios en la resolución o incluso en el tipo de equipo radiológico utilizado, lo cual puede influir negativamente en la exactitud y confiabilidad del modelo.

5.3. Desarrollar una aplicación web para el análisis de radiografías basada en los modelos de detección seleccionados

La interfaz del programa ha sido diseñada para presentar los resultados de exactitud de manera clara y accesible, permitiendo a los profesionales de la salud interpretar

rápidamente la información. La visualización de las imágenes radiográficas junto con los resultados de exactitud proporciona una mejor comprensión del diagnóstico. Este enfoque visual es relevante en un entorno clínico, donde la rapidez y la exactitud son esenciales.

Además, la intuitividad de la interfaz facilita la navegación y minimiza la curva de aprendizaje, lo que maximiza la eficiencia en su uso. Al integrar criterios clínicos junto con los resultados generados por el modelo, la plataforma se convierte en una buena herramienta para los médicos.

CONCLUSIONES

El análisis comparativo de los modelos ResNet y EfficientNet revela que ResNet es el más adecuado para la detección de TB y neumonía en radiografías. Si bien ambos modelos demostraron una alta exactitud, ResNet presentó un desempeño superior en términos de menor pérdida de valores y mayor precisión, destacándose especialmente en la detección de TB. Estos hallazgos son consistentes con estudios previos, lo cual confirma que el uso de redes profundas como ResNet optimiza de manera significativa el diagnóstico clínico de estas enfermedades.

La evaluación del modelo de detección de TB muestra tanto fortalezas como áreas de mejora en su desempeño clínico. La sensibilidad del 100% garantiza que todos los casos de TB sean correctamente identificados; sin embargo, la exactitud y precisión del 62,29% indican un riesgo considerable de falsos positivos, lo cual podría comprometer la validez diagnóstica y llevar a intervenciones innecesarias. Además, la falta de especificidad (0%) evidencia una preocupación significativa, ya que sugiere que pacientes sin TB podrían ser erróneamente diagnosticados, afectando tanto a la salud individual como a la salud pública.

En cuanto al modelo de detección de neumonía viral y/o bacteriana, los resultados revelan un desempeño limitado que subraya la necesidad de mejorar su eficacia clínica. Con una exactitud del 35%, el modelo muestra un alto margen de error, señalando que solo una fracción de los casos es correctamente identificada. Las bajas métricas de precisión, sensibilidad y especificidad para la neumonía bacteriana reflejan que el modelo tiene dificultades para diferenciar entre presencia y ausencia de la condición, lo cual representa un riesgo significativo en la práctica clínica. Es importante destacar que, debido a la falta de imágenes de neumonía viral, la precisión y sensibilidad en esta categoría son del 0%. A pesar de ello, la especificidad del 83,33% indica que el modelo puede identificar correctamente imágenes sin neumonía viral, lo cual es alentador. Para mejorar estas métricas, será fundamental disponer de una mayor cantidad de datos.

Por lo tanto, aunque el modelo ResNet ofrece un avance prometedor para el diagnóstico de TB y neumonía, la optimización de los modelos actuales y la inclusión de más datos son esenciales para perfeccionar su capacidad diagnóstica y maximizar su aplicabilidad en el contexto clínico.

Finalmente, a diferencia de modelos existentes como el de Loyola y Chamorro, enfocado exclusivamente en la detección de neumonía, o el de Mehrrotra et al., diseñado únicamente para tuberculosis, esta investigación presenta dos modelos que diagnostican de manera independiente tuberculosis y neumonía (viral y bacteriana) a partir de radiografías. Además, mientras que los estudios mencionados se limitan a desarrollar los modelos sin avanzar hacia su implementación práctica, esta investigación incluye una aplicación web que permite utilizar ambos modelos en entornos clínicos. Esta plataforma facilita su uso por parte de médicos y especialistas al ofrecer una interfaz intuitiva que integra las capacidades diagnósticas, optimizando los tiempos y recursos en la práctica médica.

RECOMENDACIONES

Para optimizar la eficacia del modelo ResNet y fortalecer su aplicabilidad clínica en la detección de TB y neumonía, se recomienda, ampliar el conjunto de datos utilizado, lo cual permitirá mejorar la generalización del modelo y reducir el riesgo de sobreajuste. En este sentido, incluir una mayor cantidad de radiografías de distintas calidades, posiciones y, en particular, de pacientes peruanos, enriquecerá la base de datos y facilitará una evaluación representativa y fiable. La diversidad de datos no solo fortalecerá el modelo, sino que también contribuirá a una detección más exacta de estas patologías en el contexto clínico local.

Asimismo, se recomienda que los modelos de detección de TB y neumonía se integren en una única arquitectura. Este enfoque optimizaría los recursos computacionales, reduciría la complejidad operativa, y facilitaría su implementación en entornos clínicos. Sin embargo, para lograr esta integración, sería fundamental contar con un mayor número de imágenes de TB y neumonía, lo que garantizaría una diferenciación más exacta y confiable de ambas patologías.

Por último, en términos de accesibilidad, se recomienda mejorar la interfaz de la página web para que su uso sea más intuitivo y eficiente. Además, la migración de parte del procesamiento a servicios de la nube podría hacer que el modelo sea más accesible y escalable que en su implementación actual, que depende de un servidor local. Esto permitiría ampliar su disponibilidad y reducir la carga sobre el servidor, facilitando el acceso para profesionales de la salud desde distintos lugares y dispositivos.

REFERENCIAS BIBLIOGRÁFICAS

[1] “Pneumonia”. Home | American Lung Association. Accedido el 2 de noviembre de 2024. Disponible:

<https://www.lung.org/lung-health-diseases/lung-disease-lookup/pneumonia>

[2] “Neumonía - Síntomas y causas - Mayo Clinic”. Top-ranked Hospital in the Nation - Mayo Clinic. Accedido el 2 de noviembre de 2024. [En línea]. Disponible: <https://www.mayoclinic.org/es/diseases-conditions/pneumonia/symptoms-causes/syc-20354204>.

[3] UNICEF, “Pneumonia in Children - UNICEF Data,” UNICEF DATA, 2018. Disponible: <https://data.unicef.org/topic/child-health/pneumonia/>

[4] World Health Organization, “Pneumonia,” www.who.int, 2023. Disponible: https://www.who.int/health-topics/pneumonia#tab=tab_1

[5] S. Campó. “Invierno 2024: se registran alrededor de 12 mil casos de neumonía en menores de cinco años a nivel nacional”. [infobae](http://infobae.com). Disponible: <https://www.infobae.com/peru/2024/07/09/invierno-2024-se-registran-alrededor-de-12-mil-casos-de-neumonia-en-menores-de-cinco-anos-a-nivel-nacional/>

[6] Ministerio de Salud, Vacunas del Esquema Nacional de Vacunación en el Perú. Plataforma del Estado Peruano, 2024. [Online]. Disponible: <https://www.gob.pe/22037-vacunas-del-esquema-nacional-de-vacunacion-en-el-peru>.

[7] “Reportan cerca de 12 mil casos de neumonía en menores de cinco años”. El Comercio Perú. Accedido el 2 de noviembre de 2024. Disponible:

<https://elcomercio.pe/peru/reportan-cerca-de-12-mil-casos-de-neumonia-en-menores-de-cinco-anos-noticia/>

[8] “Tuberculosis-Tuberculosis - Síntomas y causas - Mayo Clinic”. Mayo Clinic. Disponible:

<https://www.mayoclinic.org/es/diseases-conditions/tuberculosis/symptoms-causes/syc-20351250>

[9] Organización Mundial de la Salud, "Informe Mundial sobre la Tuberculosis 2023," Ginebra, 2023. Disponible: <https://iris.who.int/>.

[10] MINSA, "Perú da pasos significativos en la identificación temprana de casos y el tratamiento de la tuberculosis resistente," Ministerio de Salud, 2-nov-2024. Disponible: <https://www.gob.pe/institucion/minsa/noticias/1050728-peru-da-pasos-significativos-en-la-identificacion-temprana-de-casos-y-el-tratamiento-de-la-tuberculosis-resistente>. [Accedido: 5-ene-2025].

[11] “Vigilancia de tuberculosis - CDC MINSA”. CDC MINSA. Disponible: <https://www.dge.gob.pe/portalnuevo/vigilancia-epidemiologica/vigilancia-de-tuberculosis/>

[12] “Pneumonia in children”. World Health Organization (WHO). Disponible: <https://www.who.int/news-room/fact-sheets/detail/pneumonia>

[13] K. Kallander, D. H. Burgess, and S. A. Qazi, “Early identification and treatment of pneumonia: a call to action,” *The Lancet Global Health*, vol. 4, no. 1, pp. e12–e13, enero de 2016, doi: [https://doi.org/10.1016/s2214-109x\(15\)00272-7](https://doi.org/10.1016/s2214-109x(15)00272-7).

[14] American Lung Association, “Pneumonia symptoms and diagnosis,” www.lung.org, 03 de agosto de 2023. Disponible:

<https://www.lung.org/lung-health-diseases/lung-disease-lookup/pneumonia/symptoms-and-diagnosis>

[15] "Department of Surgery," froemkelab.med.nyu.edu. Disponible: <https://froemkelab.med.nyu.edu/surgery/content?ChunkIID=889654>.

[16] J. X. Guo y B. Li, "The Application of Medical Artificial Intelligence Technology in Rural Areas of Developing Countries", *Health equity*, vol. 2, n.o 1, pp. 174-181, agosto 2018, doi: 10.1089/heq.2018.0037.

[17] "Instituto Nacional de Estadística e Informática". Instituto Nacional de Estadística e Informática - INEI - Plataforma del Estado Peruano. Disponible: <https://m.inei.gob.pe/prensa/noticias/usuarios-esperan-mas-de-una-hora-para-ser-atendidos-en-un-establecimiento-de-salud-7823/#:~:text=En%20los%20establecimientos%20del%20Ministerio,las%20clínicas%20esperan%2035%20minutos>.

[18] M. Annarumma, S. J. Withey, R. Bakewell, E. Pesce, V. Goh, y G. Montana, "Automated Triaging of Adult Chest Radiographs with Deep Artificial Neural Networks", *Radiology*, vol. 291, n.o 1, pp. 196-202, enero 2019, doi: 10.1148/radiol.2018180921.

[19] P. Rajpurkar, "CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning", *arXiv.org*, 14 de noviembre de 2017. Disponible: <https://arxiv.org/abs/1711.05225>

[20] B. F. Julca Villarreal, "Aplicación de deep learning sobre imágenes topográficas para mejorar la precisión del diagnóstico de queratocono en una clínica de Lima", Tesis para optar el Título Profesional de Ingeniero Informático y de Sistemas, Facultad de Ingeniería, Carrera de Ingeniería Informática y de Sistemas, Lima, Perú, 2020.

[21] L. A. Loyola y R. M. Chamorro, "Implementación de un sistema de diagnóstico clínico aplicando un modelo predictivo de Machine Learning para la detección de neumonía en el

Hospital Villa Rebagliati de EsSalud," Tesis, Universidad Tecnológica del Perú, Lima, Perú, 2021.

[22] C. S. Rojas "Precisión del uso del Deep Learning en la síntesis, detección y segmentación de signos ultrasonográficos de pulmón sugerentes a Covid-19" Tesis, Universidad Nacional Mayor de San Marcos, Lima, Perú, 2022.

[23] S. Bharati, P. Podder, y M. M. Mondal, "Hybrid deep learning for detecting lung diseases from X-ray images", Informatics in Medicine Unlocked, vol. 20, p. 100391, enero 2020, doi: 10.1016/j.imu.2020.100391.

[24] D. Kvak et al., "Chest X-ray Abnormality Detection by Using Artificial Intelligence: A Single-Site Retrospective Study of Deep Learning Model Performance", BioMedInformatics, vol. 3, n.o 1, pp. 82-101, ene. 2023, doi: 10.3390/biomedinformatics3010006.

[25] "Ensembling of Efficient Deep Convolutional Networks and Machine Learning Algorithms for Resource Effective Detection of Tuberculosis Using Thoracic (Chest) Radiography", IEEE Journals & Magazine | IEEE Xplore, 2022. Disponible: <https://ieeexplore.ieee.org/document/9841572>

[26] "Pneumonia". Yale Medicine. Disponible: <https://www.yalemedicine.org/conditions/pneumonia>

[27] "Types of Pneumonia". WebMD. Disponible: <https://www.webmd.com/lung/pneumonia-types>

[28] Cleveland Clinic, "Pneumonia," Cleveland Clinic, 15 de Junio de 2020. Disponible <https://my.clevelandclinic.org/health/diseases/4471-pneumonia>

[29] Mayo Clinic, "Pneumonia - Diagnosis and treatment - Mayo Clinic," Mayoclinic.org, 13 de Junio de 2020. Disponible: <https://www.mayoclinic.org/diseases-conditions/pneumonia/diagnosis-treatment/drc-20354210>

[30] H. P. Ltd, "How Chest Xrays Are Used To Diagnose Pneumonia," HealthMatch, 30 de Noviembre de 2022. Disponible: <https://healthmatch.io/pneumonia/pneumonia-chest-xray#risks-of-a-chest-x-ray>

[31] S. A. Sorrentino, "Pneumonia - right middle lobe | Radiology Case | Radiopaedia.org," Radiopaedia. Disponible: <https://radiopaedia.org/cases/pneumonia-right-middle-lobe-4>.

[32] K. G. Danielsson. "X-ray Atlas: Chest X-ray | GLOWM". The Global Library of Women's Medicine | GLOWM. Disponible: <https://www.glowm.com/atlas-page/atlasid/chestXray.html>

[33] L. Moreno, E. Bujedo, H. Robledo, R. Conci, I. Marqués, L. Mosca, y D. Quiroga, "Validez de la radiografía de tórax para diferenciar etiología bacteriana de viral en niños hospitalizados con neumonía," Archivos argentinos de pediatría, vol. 104, no. 2, pp. 109-113, 2006.

[34] "Radiología de las infecciones pulmonares", en PRO-RADIO Año 2. Experto en Radiología Cardiorádica y Radiología Vascular e Intervencionista. España: Ed. Medica Panam., pp. 1-13.

[35] F. Saldías, E. Herrera, N. Nieto, F. Fernández, F. Manzur, y I. Leiva, "Características radiográficas de la neumonía adquirida en la comunidad en el adulto inmunocompetente hospitalizado según el agente causal," Revista chilena de enfermedades respiratorias, vol. 38, no. 2, pp. 96-105, 2022.

[36] “Guía de tuberculosis | Aurora Health Care”. Aurora Health Care. Disponible: <https://es.aurorahealthcare.org/services/heart-vascular/conditions/chest-lung-conditions/tuberculosis-tb>

[37] “Learn About Tuberculosis”. Home | American Lung Association. Disponible: <https://www.lung.org/lung-health-diseases/lung-disease-lookup/tuberculosis/learn-about-tuberculosis>

[38] “Tuberculosis-Tuberculosis - Diagnóstico y tratamiento - Mayo Clinic”. Mayo Clinic. Disponible: <https://www.mayoclinic.org/es/diseases-conditions/tuberculosis/diagnosis-treatment/drc-20351256>

[39] Manual para el diagnóstico bacteriológico de la tuberculosis. Parte 1: Manual de actualización de la baciloscopía (*cursiva*), Programa “Fortalecimiento de la Red de Laboratorios de Tuberculosis en la Región de las Américas”, ORAS - CONHU, Lima, Perú, 2018, pp. 88.

[40] M. Martínez Rojas, A. Flores Arroyo, y G. B. Salinas Molina, Tuberculosis pulmonar. Instituto Politécnico Nacional Escuela Superior de Medicina, Presentación en PowerPoint, mayo 2015. Disponible: <https://www.slideshare.net/slideshow/tuberculosis-pulmonar-56434654/56434654#26>.

[41] A. Castañera, MR. López Pedreira, MJ. Pena Rodríguez y M. Liñares Iglesias, “Manifestaciones radiológicas de la tuberculosis pulmonar”, Elsevier, vol. 39, n.º 5, pp. 192–206, marzo de 2002.

[42] L. Ferreiro, E. San José, y L. Valdés, "Derrame pleural tuberculoso," Archivos de Bronconeumología, vol. 50, no. 10, pp. 435-443, 2014.

[43] B. E. Jones, R. Ryu, Z. Yang, et al., "Chest radiographic findings in patients with tuberculosis with recent or remote infection," *Am. J. Respir. Crit. Care Med.*, vol. 156, pp. 1270-1273, 1997.

[44] G. Miranda G, J. C. Díaz P, P. Arancibia H, M. Antolini T, C. Díaz G y A. Vidal F, "MANIFESTACIONES RADIOGRAFICAS DE LA TUBERCULOSIS PULMONAR", *Rev. Chil. Radiol.*, vol. 10, n.º 4, 2004. Disponible: <https://doi.org/10.4067/s0717-93082004000400006>

[45] MIT Sloan, "Machine learning, explained | MIT Sloan", 21 de abril de 2021. Disponible: <https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained>

[46] I. Vasilev, D. Slater, G. Spacagna, P. Roelants, y V. Zocca, "Python Deep Learning: Exploring deep learning techniques and neural network architectures with PyTorch, Keras, and TensorFlow, 2nd Edition", Packt Publishing Ltd, 2019.

[47] G. Ciaburro y B. Venkateswaran, "Neural Networks with R: Smart models using CNN, RNN, deep learning, and artificial intelligence principles," 2017. Disponible: <https://cds.cern.ch/record/2294110>

[48] M. J. Caire, "Physiology, Synapse", StatPearls - NCBI Bookshelf, 26 de marzo de 2022. Disponible: <https://www.ncbi.nlm.nih.gov/books/NBK526047/#:~:text=The%20presynaptic%20terminal%20is%20at,away%20and%20contains%20specialized%20receptors.>

[49] M. Douglass, "Book Review: Hands-on Machine Learning with Scikit-Learn, Keras, and Tensorflow, 2nd edition by Aurélien Géron", *Physical and Engineering Sciences in Medicine*, vol. 43, n.o 3, pp. 1135-1136, sep. 2020, doi: 10.1007/s13246-020-00913-z.

[50] Role of bias in Neural Networks (2019) Intellipaat Community. Disponible: <https://intellipaat.com/community/253/role-of-bias-in-neural-networks>.

[51] G. Carneiro, J. C. Nascimento, y A. P. Bradley, "Deep Learning Models for Classifying Mammogram Exams Containing Unregistered Multi-View Images and Segmentation Maps of Lesions", This work is an extension of the paper published by the same authors at the Medical Image Computing and Computer-Assisted Intervention (MICCAI 2015) [1].", en Elsevier eBooks, 2017, pp. 321-339. doi: 10.1016/b978-0-12-810408-8.00019-5.

[52] Gavrilova, Y. (2021) What are convolutional neural networks?, Serokell Software Development Company. Disponible: <https://serokell.io/blog/introduction-to-convolutional-neural-networks>.

[53] A. Pai, "CNN vs. RNN vs. ANN – Analyzing 3 Types of Neural Networks in Deep Learning", Analytics Vidhya, mayo 2023. Disponible en: <https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/>.

[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778, 2016.

[55] Khan, M.R. (2018) Residual Networks (resnets), Medium. Disponible: <https://towardsdatascience.com/residual-networks-resnets-cb474c7c834a>.

[56] Ruiz, P. (2019) Understanding and visualizing ResNets, Medium. Disponible: <https://towardsdatascience.com/understanding-and-visualizing-resnets-442284831be8>.

[57] Potrimba, P. (2023) What is EfficientNet? The Ultimate Guide., Roboflow Blog.
Disponibile:

<https://blog.roboflow.com/what-is-efficientnet/#:~:text=EfficientNet%20Architecture,further%20enhance%20the%20model's%20performance>.

[58] Acm

[59] O. Ramos. "Tema 4: Métricas de Evaluación". Tópicos de Ingeniería Mecatrónica, Universidad de Ingeniería y Tecnología - UTEC, Departamento de Ingeniería Mecatrónica, 2020.

[60] M. Fahmy Amin, "Confusion Matrix in Three-class Classification Problems: A Step-by-Step Tutorial," *Journal of Engineering Research*, vol. 7, no. 1, Article 26, 2023.
Disponibile: <https://digitalcommons.aaru.edu.jo/erjeng/vol7/iss1/26>

[61] D. S. Kermany, M. Goldbaum, W. Cai, C. C. Valentim, H. Liang, S. L. Baxter, et al., "Identifying medical diagnoses and treatable diseases by image-based deep learning," *Cell*, vol. 172, no. 5, pp. 1122-1131, 2018.

[62] S. Jaeger, A. Karargyris, S. Candemir, L. Folio, J. Siegelman, F. Callaghan, Z. Xue, K. Palaniappan, R. K. Singh, S. Antani, G. Thoma, Y. X. Wang, P. X. Lu, y C. J. McDonald, "Automatic tuberculosis screening using chest radiographs," *IEEE Trans Med Imaging*, vol. 33, no. 2, pp. 233-245, Feb. 2014. doi: 10.1109/TMI.2013.2284099. PMID: 24108713