



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE  
MÉXICO

---

---

CENTRO UNIVERSITARIO UAEM TEXCOCO

IDENTIFICACIÓN DE TUBERCULOSIS MEDIANTE  
PATRONES RADIOLÓGICOS CON EL USO DE LAS  
REDES NEURONALES CONVOLUCIONALES

T E S I S

QUE PARA OBTENER EL GRADO DE:

DOCTOR EN CIENCIAS DE LA COMPUTACIÓN

P R E S E N T A :

JESSICA SÁNCHEZ ARRAZOLA

DIRECTOR DE TESIS

ADRIÁN TRUEBA ESPINOSA

COO DIRECTOR

FARID GARCÍA LAMONT

TUTOR: JAIR CERVANTES CANALES

ESTADO DE MÉXICO, TEXCOCO, 2024



# Resumen

En este trabajo se realizó la clasificación de la tuberculosis (TB) utilizando redes neuronales artificiales. Debido a que aún no se ha podido erradicar esta enfermedad a pesar de que existe tratamiento. La Organización Mundial de la Salud (OMS), dio a conocer que en el año 2022, 1.3 millones de personas murieron, es por ello, que año con año, hacen un llamado a investigadores para que desarrollen estrategias y herramientas con la finalidad de eliminarla. Por lo anterior se desarrolló un modelo de red neuronal convolucional (*CNN* por sus siglas en inglés) que permite detectar los patrones de tuberculosis (se considera un patrón de tuberculosis a los diferentes daños presentes en los pulmones) a partir de imágenes de radiografías, con la finalidad de obtener un diagnóstico como apoyo a los médicos y así los pacientes puedan comenzar su tratamiento de forma oportuna.

Esta tesis está dividida en 4 experimentos, en el primer experimento se realizó la clasificación de enfermedades pulmonares, el segundo experimento se trabajó con patrones de tuberculosis y se propuso un modelo de CNN, en el tercero se utilizó la clasificación de patrones con transferencia de aprendizaje y en el cuarto, se analizaron las imágenes para hacer extracción de características y mejorar la clasificación de los patrones de TB con una red neuronal *Backpropagation*.

A continuación se describe brevemente cada experimento y sus resultados. En el primer experimento se realizó la clasificación de tuberculosis contra enfermedades pulmonares como Covid-19, neumonía viral y bacteriana. Se diseñó la arquitectura de una *CNN* para la clasificación de las enfermedades y se comparó con la red *ResNet50*. Los resultados en cuanto a la exactitud con la red propuesta es de 0.87 vs 0.71. En el segundo experimento, se obtuvo la base de datos del Instituto Nacional de Enfer-

medades Infecciosas, se realizó la clasificación de patrones radiológicos con presencia de TB en imágenes de rayos X, se observó que en las radiografías de los pacientes se presentan de dos a seis patrones (consolidación, fibrosis, opacidad, pleural, nódulos y cavitaciones). Es importante mencionar que los especialistas consideran el tipo de patrón de TB para poder dar un tratamiento adecuado. Cabe destacar que no en todos los centros médicos hay especialistas que interpretan de inmediato los patrones radiológicos. Considerando lo anterior se tiene como objetivo la clasificación de patrones mediante una red neuronal convolucional para ayudar a tener un diagnóstico más certero en las radiografías, para que los médicos recomienden un tratamiento de forma inmediata y así evitar el contagio a más personas.

Para la clasificación de patrones de tuberculosis se diseñó una CNN Metodo propuesto (MPro), y se comparó contra las arquitecturas *VGG16*, *InceptionV3* y la *ResNet50*. Los resultados que se obtuvieron en cuanto a la métrica *AUC-SVM*, *InceptionV3* y *VGG16* fue de 0.95, y la red propuesta y *ResNet50* con 0.80.

En el tercer experimento se utilizó la transferencia de aprendizaje y se tomaron los pesos del experimento uno, para transferirlos en la CNN propia y a si no comenzará a entrenar desde cero, también se realizó la transferencia de las arquitecturas *InceptionV3*, *VGG16* y *ResNet50*. Para validar la clasificación se usó la métrica *ROC* y obtuvo una precisión de 0.80 para el MPro, para *InceptionV3* 0.95, *VGG16* 0.95 y *ResNet50* 0.80. En el cuarto experimento se realizó la extracción de características de imágenes previamente segmentadas y con ellas se entrenó una red neuronal *Backpropagation* dando una *exactitud* de 0.80.

Analizando los experimentos la mejor clasificación se obtuvo con la transferencia de aprendizaje y con la arquitectura *InceptionV3* ya que obtuvo un *AUC-SVM* de 0.95.

# Índice general

<b>Agradecimientos</b>	III
<b>Resumen</b>	VII
<b>1. Introducción</b>	1
1.1. Planteamiento del problema	1
1.2. Justificación	2
1.3. Hipótesis	3
1.4. Objetivos	3
1.4.1. Objetivo general	3
1.4.2. Objetivo específico	3
1.5. Contribuciones de la investigación	4
<b>2. Estado del arte</b>	5
<b>3. Marco teórico</b>	12
3.1. Tuberculosis	12
3.1.1. Tratamiento	15
3.1.2. Rayos X	16
3.1.3. Patrones de tuberculosis en radiografías	16
3.2. Procesamiento de imágenes	20
3.2.1. Tratamiento de imagen	22
3.2.2. Segmentación de imágenes	24
3.3. Inteligencia artificial	25
3.3.1. Redes neuronales convolucionales	30
3.3.2. ResNet50	35
3.3.3. InceptionV3	37
3.3.4. VGG16	39
3.4. Características de los datos	41
3.5. Transferencia de aprendizaje	42
3.6. Maquinas de soporte vectorial	42
3.7. Métrica para medir el rendimiento de la red entrenada	45
3.7.1. Matriz de confusión	45
3.7.2. ROC-AUC	46

<b>4. Metodología</b>	<b>48</b>
4.1. Experimento 1	51
4.1.1. Base de datos	53
4.1.2. Entrenamiento	53
4.2. Experimento 2	54
4.2.1. Base de datos experimento 2	54
4.3. Experimento 3	64
4.3.1. Transferencia de aprendizaje (TF)	64
4.4. Experimento 4	65
4.4.1. Entrenamiento <i>Backpropagation</i>	65
<b>5. Resultados de los experimentos</b>	<b>68</b>
5.1. Resultados y discusiones del experimento 1	68
5.2. Resultados y discusiones experimento 2	71
5.2.1. SVM-Métrica ROC	75
5.3. Resultados y discusiones experimento 3	81
5.3.1. Transferencia de aprendizaje	81
5.3.2. SVM-Métrica ROC y Kernel	86
5.4. Resultados y discusiones del experimento 4	93
<b>6. Discusiones generales</b>	<b>95</b>
<b>7. Conclusiones</b>	<b>97</b>
<b>8. Artículos publicados</b>	<b>99</b>
<b>Bibliografía</b>	<b>100</b>

# Índice de figuras

2.1. Precisión de patrones de TB unitarios . . . . .	10
3.1. Personal de salud haciendo una Baciloscopia. . . . .	13
3.2. Cultivo del bacilo de la tuberculosis. . . . .	14
3.3. Radiografía de tórax con presencia de tuberculosis. . . . .	15
3.4. Tuberculosis ganglio pulmonar. . . . .	17
3.5. Tuberculosis pleural. . . . .	17
3.6. Tuberculosis miliar. . . . .	18
3.7. Afección del parénquima . . . . .	18
3.8. Fibrosis. . . . .	19
3.9. Tuberculosis pulmonar primaria (complejo de Ranke). . . . .	19
3.10. Tuberculoma . . . . .	20
3.11. Tuberculosis pulmonar cavitada. . . . .	20
3.12. Redes neuronales artificiales y redes neuronales biológicas. . . . .	27
3.13. Funciones de activación. . . . .	28
3.14. Aprendizaje supervisado . . . . .	29
3.15. Aprendizaje no supervisado . . . . .	29
3.16. Aprendizaje reforzado . . . . .	30
3.17. Arquitectura CNN . . . . .	31
3.18. Canales de las imágenes . . . . .	32
3.19. Max-pooling . . . . .	33
3.20. Ejemplo de conexión de salto. . . . .	35
3.21. Factorización en circunvoluciones más pequeñas . . . . .	37
3.22. Fuente: Elaboración propia . . . . .	37
3.23. Módulos de red InceptionV3 . . . . .	38
3.24. Proporción de los datos. . . . .	41
3.25. Separación entre clases. . . . .	42
3.26. Kernel . . . . .	44
4.1. Metodología . . . . .	49
4.2. Metodología para el desarrollo de la CNN . . . . .	50
4.3. Metodología para el diseño de la CNN . . . . .	50
4.4. Selección y etiquetado de las radiografías por clase. . . . .	54
4.5. Imágenes con ruido. . . . .	56
4.6. Segmentación pulmonar. . . . .	57

4.7. Segmentación pulmonar a mano.	58
4.8. Aumento de imágenes.	59
4.9. Clasificación de las radiografías.	60
4.10. Gráfica de la exactitud de los entrenamientos.	62
4.11. Transferencia de aprendizaje	64
5.1. Segmentación pulmonar.	68
5.2. Comparación de los entrenamientos.	71
5.3. Comparación de la Matriz de confusión de cada Arquitectura entrenada del experimento 2.	72
5.4. Curvas ROC del método propuesto.	76
5.5. Curvas ROC InceptionV3.	77
5.6. Curvas ROC VGG16.	78
5.7. Curvas ROC ResNet50.	79
5.8. Comparación de resultados de curva ROC,experimento 2.	80
5.9. Gráficas de entrenamiento experimento 3.	81
5.10. Matriz de confusión experimento 3	82
5.11. Curvas ROC del método propuesto.	87
5.12. Curva ROC Red InceptionV3.	88
5.13. Curva ROC Red VGG16.	89
5.14. Curva ROC Red ResNet50.	90
5.15. Comparación de Curvas ROC, experimento 3.	91
5.16. Grafica de entrenamiento de la red Backpropagation.	93
5.17. Matriz de confusión de la red Backpropagation.	93
8.1. Artículo publicado	99
8.2. Artículo en revisión	100

# Índice de tablas

2.1. Comparación entre literatura y propuestas por los autores . . . . .	7
3.1. Profundidad de color . . . . .	22
3.2. Profundidad de color . . . . .	23
3.3. Arquitectura de la red ResNet50 . . . . .	36
3.4. Arquitectura de la red InceptionV3 . . . . .	39
3.5. Fuente: [1] . . . . .	39
3.6. Arquitectura de la red VGG16 . . . . .	40
3.7. Fuente: [2] . . . . .	40
3.8. Diferencias de arquitecturas de CNN . . . . .	41
3.9. Fuente: Elaboración propia . . . . .	41
3.10. Matriz de confusión . . . . .	45
4.1. Arquitectura diseñada MPro1 . . . . .	52
4.2. Clasificación de patrones . . . . .	55
4.3. Combinación de patrones usadas . . . . .	56
4.4. Arquitectura diseñada Mpro2 . . . . .	61
4.5. Cambios de los parámetros para entrenamiento . . . . .	62
4.6. Resultados del experimento 1 con <i>TF</i> . . . . .	65
4.7. Arquitectura Backpropagation . . . . .	67
5.1. Evaluación del Mpro del experimento 1. . . . .	69
5.2. Evaluación del modelo ResNet50 del experimento 1. . . . .	69
5.3. Resultado del MPro2. . . . .	73
5.4. Resultado InceptionV3. . . . .	73
5.5. Resultado Vgg16. . . . .	74
5.6. Resultados ResNet50 experimento 3. . . . .	75
5.7. Resultados con las 4 arquitecturas con imágenes segmentadas . . . . .	75
5.8. Mejores resultados de las curvas ROC Experimento 2 . . . . .	80
5.9. Resultado del método propuesto . . . . .	83
5.10. Resultados InceptionV3 experimento 3. . . . .	84
5.11. Resultados VGG16 experimento 3. . . . .	85
5.12. Resultado ResNet50. . . . .	85
5.13. Resultados con las 4 arquitecturas <i>TF</i> . . . . .	86
5.14. Mejores resultados de las curvas ROC . . . . .	91

5.15. Resultados Backpropagation . . . . .	94
--	----

# 1 Introducción

Un padecimiento que aún no se ha podido erradicar en el mundo es la TB, es una infección bacteriana contagiosa que es causada por la bacteria *Mycobacterium o bacilo de Koch* [3]. A pesar de que esta enfermedad tiene cura y se puede prevenir aun siguen habiendo decesos. La Dirección General de Epidemiología de la Secretaria de Salud ha mencionado que en la cuarta semana de 2024 hay 400 casos nuevos de contagio.

La TB se transmite muy fácilmente, por ejemplo cuando un enfermo de TB tose, estornuda o escupe, expulsa bacilos tuberculosos en el ambiente y si una persona inhala esos bacilos puede quedar infectada [4]. Existen dos tipos de TB, la primaria es cuando se infectan por primera vez y la pos primaria es cuando la enfermedad se reactiva.

Durante el diagnóstico de TB en radiografías de tórax, se ha detectado que el bacilo es capaz de atacar varias zonas del pulmón formando diferentes patrones. Un patrón radiológico es la sucesión de anomalías presentes en una radiografía que se repite, los más comunes son: cavitaciones, nódulos, miliars, parénquima, pleural. Depende del patrón radiológico que desarrolle el paciente recibirá un tratamiento, con la finalidad de evitar que se siga propagando la enfermedad.

## 1.1. Planteamiento del problema

La OMS pronostica que una tercera parte de la población mundial tiene TB latente; es decir, hay personas infectadas por el bacilo, que no presentan síntomas de la enfermedad, pero son portadores del bacilo y pueden transmitir la infección [5].

Las personas que estuvieron expuestas al bacilo de TB tienen el riesgo de por vida de enfermarse de un 5 a 10 % [6]. Desde el 2015 la OMS está implementando un protocolo para erradicar esta enfermedad, para ello, se necesita que los diagnósticos de los médicos sean rápidos, certeros, y de bajo costo [7].

Hay estudios [8], [9], [10] los cuales demuestran que a partir de radiografías de tórax los médicos expertos son capaces de identificar y diagnosticar la TB y neumonía. Sin embargo existen regiones que carecen de estos expertos, lo cual dificulta el diagnóstico de TB aun teniendo las radiografías.

Por lo que se propone una metodología para clasificar más de un patrón de TB presente, que puede presentarse en un solo paciente con el uso de una *CNN*, con el fin de ayudar a un médico sin experiencia a detectar esta enfermedad y le ayude en la toma de decisiones para un diagnóstico adecuado y oportuno.

Existen trabajos con *CNN* que detectan patrones de TB [11], [12] esos trabajos sólo detectan un solo patrón por paciente, pero de acuerdo al conjunto de datos proporcionados por el laboratorio *National Institute of Allergy and Infectious Diseases (NIAID) TB Portals DEPOT* [13], en un paciente se pueden encontrar de dos a siete patrones al mismo tiempo, esto implica que el problema no es solo detectar un patrón de TB como en otros trabajos expuestos, también es detectar cuantos y cuales patrones se tienen en la radiografía del paciente.

## 1.2. Justificación

La OMS ha puesto en marcha una estrategia para ponerle fin a la TB, la cual tiene como objetivo reducir el número de muertes a un 90 %, esto en un periodo de entre 2020 y 2025, este objetivo se comenzó a implementar en mayo del 2014 por la Asamblea Mundial de la salud. Para poder lograr este objetivo la OMS necesita nuevos e innovadoras herramientas científicas que ayuden a prevenir y detectar de forma oportuna. [5]. Al realizar la clasificación de TB en radiografías se observó que un solo paciente puede tener mas de dos patrones de TB por lo que hace el diagnóstico de TB sea amplio y no se de el tratamiento adecuado. Para que los especialistas

generen un tratamiento adecuado deben tener en cuenta todos los patrones de TB es por ello que se propuso diseñar la clasificación utilizando las radiografías de pacientes que tienen más de un patrón en la radiografía.

## 1.3. Hipótesis

Será posible identificar los diferentes patrones de TB de tórax a partir de radiografías, usando algoritmos de inteligencia artificial como lo son las CNNs y Backpropagation.

## 1.4. Objetivos

### 1.4.1. Objetivo general

Clasificar los patrones (consolidación, fibrosis, opacidad, pleural, nódulos y cavitaciones) mediante el desarrollo de una CNN capaz de reconocer patrones asociados de TB en radiografías, para que los médicos recomienden un tratamiento de forma inmediata y así evitar el contagio a más personas. Dicha red, aprenderá a realizar la clasificación mediante ejemplos conformados por radiografías en las cuales se presentan los diferentes patrones de TB

### 1.4.2. Objetivo específico

- Diseñar una CNN para la clasificación de enfermedades pulmonares y compararla contra la arquitectura ResNet50.
- Obtener y organizar imágenes diagnosticadas con presencia de TB con los patrones (consolidación, fibrosis, opacidad, pleural, nódulos y cavitaciones), para ser procesadas y analizadas.
- Diseñar una CNN para la clasificación de los patrones de TB.
- Realizar la segmentación pulmonar para evitar el ruido en las imágenes.

- Evaluar los resultados obtenidos y compararlos con las arquitecturas *InceptionV3*, *VGG16* y *ResNet50*.
- Realizar transferencia de aprendizaje.
- Realizar la extracción de características en las imágenes sementadas.
- Realizar el entrenamiento con una red *Backpropagation*.
- Contrastar los resultados obtenidos con cada método.

## 1.5. Contribuciones de la investigación

- Clasificación de enfermedades pulmonares y patrones de tuberculosis para ayudar a los que no son especialistas médicos a tener una segunda opinión sobre el padecimiento de los enfermos y que puedan tener un tratamiento inmediato.
- Desarrollar una red convolucional que se enfoque en imágenes radiológicas.
- Mejorar los resultados de la clasificación con el uso de la transferencia de aprendizaje.

## 2 Estado del arte

A continuación se presentan los procedimientos más relevantes que corresponden al experimento 1 y se colocó una breve explicación de ellos, con el fin de tomar parte de la experiencia de otros autores, lo cual ha sido de gran interés para la creación del modelo que se propone.

Para la detección de TB en general [9], se construyeron dos conjuntos de datos, el primero se formó con una técnica de segmentación de imágenes, el segundo sin segmentar las imágenes. Para clasificar las imágenes usaron redes neuronales convolucionales profundas (DCNN). Los resultados obtenidos demuestran que la técnica es aceptable, la precisión de clasificación no excedió el 90 % con imágenes segmentadas y sin la segmentación 85 %.

En [14] desarrollaron una red neuronal, capaz de dar un diagnóstico de la TB, el objetivo fue presentar un estudio comparativo de clasificadores para la realización del diagnóstico utilizando *redes neuronales multicapa (MLNN)*. Se utilizaron dos estructuras *MLNN* diferentes. También se usó una *red neuronal de regresión general (GRNN)* para realizar el diagnóstico de TB para posteriormente compararlas. En conclusión, el análisis realizado muestra que el modelo tiene una concordancia del 60.72 % con el diagnóstico del médico con una precisión del 57.14 %.

En la universidad de Chulalongkorn Tailandia [10] se evaluaron tres redes neuronales convolucionales (CNN): *AlexNet*, *VGG-16* y *CapsNet*; estas se utilizaron para clasificar la TB en rayos X, utilizaron una muestra de 1,390 elementos y emplearon el muestreo aleatorio para incrementar las imágenes a 1,986; posteriormente se creó

un tercer conjunto con aumento de imágenes, generando 3,310 imágenes; para cada conjunto de imágenes se utilizaron el 80 % para entrenamiento y 20 % de prueba. Realizaron 3 entrenamientos por cada conjunto de datos y los compararon. El clasificador que mejor desempeño tuvo fue el VGG16 para el conjunto de 3,310; logrando una sensibilidad de 92.83 %.

Stefan Jaeger [15] diseñó un sistema CAD (Diagnóstico asistido por computadora) con el cual realizó la segmentación pulmonar basada en el corte gráfico para la clasificación de enfermedades. Usaron máquinas de soporte vectorial para detectar en las radiografías patrones anormales. El área bajo la curva (AUC) es de 87 % - 90 % y la precisión fue de 78.3 % - 84 %.

Otro trabajo similar lo realizaron [16] logrando una exactitud del 0.674 % - 0.83 %.

En Brasil [17], diseñaron tres propuestas de pre entrenamiento, para redes neuronales convolucionales, con la finalidad de extraer características para poder entrenar un clasificador SVM (Máquinas de soporte vectorial). Los resultados se muestran en la tabla 2.1 y al mismo tiempo hacen una comparación con otros trabajos y sus resultados. P1 y P2 (se refieren a sus deducciones con su metodología), EP1 y EP2 (se refieren a los conjuntos de datos creados utilizando los clasificadores GoogLeNet y ResNet de otros autores). La propuesta P2 y EP2 superan la exactitud con respecto a los otros dos autores. Los resultados sugieren que el uso de máquinas de soporte vectorial contribuye a tener mejores resultados.

Conjunto de datos	literatura		Propuesta			
	[15]	[16]	P1	P2	EP1	EP2
Montgomery	0.783	0.674	0.782	0.826	0.760	0.8226
Shenzhen	0.840	0.837	0.8334	0.847	0.846	0.846.

Tabla 2.1: Comparación entre literatura y propuestas por los autores

En [18] se documentó la identificación de enfermedades pulmonares en radiografías de rayos X de COVID, Neumonía y TB. El COVID-19, es una nueva enfermedad que ha afectado al mundo entero, requiere de un diagnóstico médico rápido y al igual que la TB se ha posicionado como una enfermedad importante de diagnosticar; ambas pueden ser diagnosticadas por medio de rayos X. Para obtener un diagnóstico más rápido se planteó una investigación donde se utilizaron 14 redes neuronales profundas pre entrenadas de donde se extrajeron los vectores de características, se clasificaron 5 categorías de imágenes de radiografías del tórax, las cuales son: Covid-19, TB, Neumonía bacteriana y Neumonía viral. El número de imágenes que se utilizó en cada categoría fue balanceado, dando un total de 2,186 imágenes. Se realizaron 14 entrenamientos y estos fueron clasificados con tres métodos: SVM cuadrático, SVM gaussiano medio y el conjunto de clasificadores discriminantes sub espaciales. La red de entrenamiento que dio una mejor precisión fue la ResNet50 con una precisión del 95%. En este mismo trabajo se usó la red ResNet50, para clasificar solo 3 clases las cuales fueron: Covid-19, normal y TB obteniendo una precisión en general del 98%.

En el artículo [19] se trabajó con la detección de neumonía y TB con un conjunto de datos de rayos X, crearon una arquitectura propia de CNN. Realizaron los siguientes pasos de preprocesamiento al conjunto de datos *CXR (Chest X-ray)*: Filtrado para la preservación de los bordes y eliminar el ruido, redimensión de pixeles a 224 x 224, escalado para restringir los pixeles en el rango de [0, 1] y normalización y estandarización para asegurar un rango similar a las características extraídas. Se diseñaron 3 modelos para aprender características específicas; en el primer modelo solo usaron conjuntos de datos que tenían neumonía RSNA, neumonía pediátrica y colecciones

radiografías pertenecientes de Indiana para aprender las características específicas de las radiografías del tórax. En el segundo modelo se utilizaron conjuntos de datos con TB y en el tercer modelo se usaron el primer modelo ya previamente entrenado y se ajustaron con el segundo modelo. Cada modelo fue entrenado por seis clasificadores (CNN personalizada, *VGG16*, *InceptionV3*, *InceptionResNetV2*, *Xception* y *DenseNet211*), de ellos se obtuvieron valores de precisión, *AUC*. Teniendo un *AUC* de ente 0.94 a .99. Aunque no muestran resultados específicos por enfermedad, se puede apreciar que los resultados expuestos mejoran sustancialmente los reportados por otros autores que van de .90 [15], 0.92 [16] y 0.92 [17].

En otro trabajo propusieron un método para detectar; TB, neumonía y cáncer de pulmón con la ayuda de una red neuronal artificial, [20] extrajeron dos clases: normal y anormal. Se consideraron las características geométricas y las de contraste para la certificación. El método que propusieron es el siguiente: Procesamiento de imagen: Eliminar datos irrelevantes, recuperar información útil y fortalecer del área de interés. Realizaron el filtrado para eliminar el ruido en las imágenes y se uso el filtro paso alto para agudizar las imágenes. Segmentación de la imagen: Utilizaron el método de detección de los bordes, funciona en píxeles individuales y convierte la imagen en binario. Extracción de características: para esta fase, se requiere de la fase anterior de la segmentación de la imagen y las características que se consideran son las áreas geométricas, perímetros diámetros y desviación estándar. La base de datos de radiografía de tórax fue para 80 pacientes y se obtuvo una precisión del 92%.

En el siguiente experimento [21] se utilizaron 3 bases de datos (*Montgomery, Shenzhen, Belarus* TB, con un total de 1,007 imágenes) las imágenes se re-dimensionan a una matriz de 256 x 256 y utilizaron 2 diferentes DCNN (*Deep convolutional neural network*) *AlexNet* y *GoogLeNet* y se obtuvo una *AUC* de 0.99 y una sensibilidad del 97.3%. En este trabajo los autores mencionan 3 patrones de TB (pleural, miliar y con cavitaciones) pero al finalizar del trabajo solo hablan sobre la clasificación de dos categorías normal y con TB en general, sin especificar si se consideraron los 3 diferentes patrones.

En [22] comentan que las imágenes de rayos X tienen mucho potencial al momento

de dar diagnósticos. Ellos utilizaron cinco modelos CNN ya entrenados los cuales son: *VGG16*, *VGG19*, *ResNet18*, *ResNet50* y *ResNet101*. Posteriormente usaron el método SVM para clasificar con cuatro funciones del *kernel*, *Lineal*, *Cuadrático*, *Cúbico* y *Gaussiano*. El valor de  $\epsilon$  con las funciones del *kernel lineal y cuadrático* se estableció en 0.04 y 0.02 y para *kernel cúbico* y la función gaussiana se estableció en 0.01. Las características obtenidas del modelo *ResNet50* y el clasificador SVM con la función del *kernel lineal* obtuvo una precisión de 94.7% y el modelo *ResNet50* ajustado del 92.6%.

Para trabajar con patrones de TB, se busco literatura que abordara el tema pero no se encontraron trabajos .

En la investigación [23], se usan rayos X para detectar la TB por medio de patrones radiológicos con redes neuronales convolucionales (Faster RCNN, EPN y una red neuronal propuesta por los autores) considerando los patrones; miliar, pleural encapsulado, nódulos, calcificación y exudación. Los resultados se muestran en la figura 2.1, donde se observan las precisiones que obtuvieron por patrón, considerando tres métodos de clasificación; como se puede apreciar la precisión más alta se obtuvo en el patrón miliar con el método FPN (Feature Piramy Network).

Method	Mean Average Precision	Exudation	Calcification	Nodule	Miliary TB	Free Pleural Effusion	Encapsulated Pleural Effusion
Faster RCNN	22.66	41.49	5.52	0.37	40.57	45.65	2.36
FPN	50.96	52.89	42.84	17.50	87.72	52.67	52.13
Method	53.74	55.54	41.82	22.72	84.62	64.92	52.79

Figura 2.1: Precisión de patrones de TB unitarios

También se realizó un método para evaluar la etapa de la segmentación, mediante dos indicadores de evaluación (*Dice*, *IoU*). *Dice*: mide la relación entre el área de inserción y el área total de las imágenes reales (G) y las imágenes segmentadas (O). *Dice* obtuvo un valor de 0.978 con el conjunto de datos de *Montgomery* y un *IoU* del 0.957.

En otro trabajo [24] se menciona que el tema de la TB sigue siendo un tema urgente sobre todo en los países de bajos recursos. Por ello proponen desarrollar una red neuronal convolucional, la cual dará un diagnóstico fiable, por medio de un celular, se hace la captura de la radiografía y en ella se pueda mostrar un pre-diagnóstico más rápido. En este trabajo se propone una metodología para clasificar más de un patrón de TB en una sola radiografía del tórax, con una red neuronal convolucional profunda, con el fin de ayudar a un médico sin experiencia a detectar esta enfermedad y la toma de decisiones sea pronta para un diagnóstico adecuado. No hay evidencias de que la metodología fue implementada y no exponen resultados concretos.

En [25] evaluaron la viabilidad de la detección y clasificación de patrones de TB, basados en el aprendizaje profundo. El experimento consiste en obtener una fotografía de la radiografía, la fotografía es analizada por una red profunda y la clasifica en las siguientes clases; cavidad, consolidación, derrame, cambios intersticiales, patrón miliar o examen normal. La clasificación se evaluó mediante el análisis de las características de cada patrón a identificar dando un AUC de 0.82%.

En otra investigación [26] dividieron la radiografía del tórax en 3 partes: lóbulo superior, lóbulo medio y región inferior(bilateral). Y clasifican las imágenes en 4 cate-

gorías: cavidad, fibróticas, agudas y miliars. La imagen digitalizada media  $512 \times 512$  píxeles y estaban en formato *.jpg* Posteriormente se redujo la imagen original a  $50 \times 50$  píxeles. Los autores desean tener la tasa de error más baja por lo que desarrollan su propia red neuronal, que contiene los siguientes parámetros: Capas ocultas 1, neuronas ocultas 5, tasa de aprendizaje 0.2, y con esos parámetros obtienen una tasa de error mínima de 0.05.

## 3 Marco teórico

### 3.1. Tuberculosis

“La TB es una infección bacteriana contagiosa producida por *bacilo de Koch o Mycobacterium*” [27]. Si tiene cura y se puede prevenir. Se transmite de una persona a otra, por el aire, basta con que una persona inhale unos pocos bacilos para que pueda quedar infectada [4].

En el 2023 la OMS [28] informo que 10.6 millones de personas están infectadas de TB a nivel mundial y 1.6 millones de personas fallecieron.

El Dr. Miguel Ángel Lezana [29] diseñó los estándares de atención de la TB en México, los cuales se encuentran sustentados por la Norma “NOM-006-SSA2-1993”, esta ayudará a los médicos a identificar pacientes que puedan tener tuberculosis, a continuación, se detallan las personas que pueden contraer TB, el riesgo de contraer TB, los síntomas, los estándares de diagnóstico, exámenes de diagnóstico y el tratamiento.

#### **Personas propensas a contraer tuberculosis:**

- Las personas de edad avanzada.
- Niños menores de edad (5 años).
- Personas con el sistema inmunitario debilitado, VIH/Sida, personas tratadas con quimioterapia, diabetes).

**Síntomas:** La TB en muchos pacientes no causa síntomas sobretodo en la fase primaria. [30] Pero cuando si llegan a presentar, pueden incluir: dolor en el pecho, tos, dificultad al respira, hemoptisis, sudoración, fatiga, fiebre y perdida de peso.

**Estándares y diagnóstico** Los doctores realizará un examen físico donde deben de analizar los siguientes puntos:

- Dedos hipocráticos, alteración que engrosa el tejido en dedos y manos.
- Inflamación de los ganglios linfáticos.
- Derrame pleural pulmonar y crepitaciones al respirar.

**Los exámenes que se solicitan incluyen:** [31]

- Broncoscopia y Tomografía computarizada del tórax.
- Radiografía del tórax.
- Prueba de sangre QFT-Gold o cultivo de esputo .
- Prueba cutánea con tuberculina o prueba PPD.
- Biopsia del tejido afectado.

**Práctica médica nacional** La baciloscopia, prueba de esputo es el procedimiento de diagnóstico más útil en los pacientes con TB, es el método más barato, simple, rápido y específico, en la Figura [3.1] se observa a una doctora ayudando al paciente para obtener una muestra de la mucosidad que hay en el pulmón.



Figura 3.1: Personal de salud haciendo una Baciloscopia.

Fuente: [29]

El cultivo es el mejor método para el diagnóstico, pero la desventaja es que tarda mucho tiempo en dar un resultado, en la Figura 3.2 se ven a dos especialistas identificando los bacilos que hay en la muestra.



Figura 3.2: Cultivo del bacilo de la tuberculosis.

Fuente: 29

Las radiografías del tórax son útiles para la localización de lesiones, pero las imágenes también son muy inespecíficas, en la Figura 3.3 se muestra una radiografía del tórax, obsérvese como un pulmón se ve con colores más difuminados.

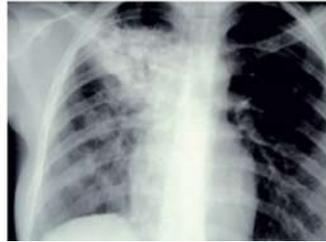


Figura 3.3: Radiografía de tórax con presencia de tuberculosis.

Fuente: [29]

En el artículo [15] explica que una de las pruebas más importantes para detectar la tuberculosis es la muestra de esputo, la cual es considerada como la mejor prueba, sin embargo esta prueba puede tardar hasta 8 semanas en poder tener un resultado ya que son varias las pruebas que se realizan y requieren de tiempo. Sin embargo, en las radiografías se puede detectar enfermedades de tuberculosis de forma masiva, además que son usadas obligatoriamente para la evaluación de la tuberculosis. Una estrategia para acabar con la TB es el diagnóstico oportuno es por eso que se recomienda el uso de radiografías de tórax [32].

### 3.1.1. Tratamiento

El tratamiento interrumpe la cadena de transmisión y logra la curación. Los pacientes toman los medicamentos hasta ver los resultados radiológicos o de laboratorio. Si las personas no siguen correctamente el tratamiento, la infección puede complicarse aún más y puede ser más difícil de tratar. Las bacterias de la TB pueden adquirir resistencia a los medicamentos, lo que implica que los tratamientos dejan de ser efectivos. En caso de que el paciente suspenda el tratamiento deberá iniciar de nuevo. [33].

### 3.1.2. Rayos X

Se emplean rayos X para crear imágenes de los tejidos corporales. Al atravesar el cuerpo, los rayos X son absorbidos de manera distinta por los diferentes tejidos, lo que provoca contrastes en las imágenes radiológicas. Por ello, los huesos se ven más blancos, mientras que los tejidos blandos y el aire en los pulmones se muestran en tonos grises. Las radiografías son imágenes en blanco y negro en las que las estructuras que bloquean la radiación se ven en color blanco y las que permiten el paso de la radiación se muestran en color negro [34].

La radiografía del tórax es una forma frecuente de diagnosticar enfermedades. Además, ayuda a los especialistas a determinar las causas y poder dar el mejor tratamiento. [35]

### 3.1.3. Patrones de tuberculosis en radiografías

En [36] describen que la tuberculosis torácica tiene ocho patrones de presentación: ganglio pulmonar, pleural, miliar, afección del parénquima (Consolidación) o linfadenopatía, Broncoestenosis (fibrosis), granuloma pulmonar, tuberculomas y cavitaciones.

**Tuberculosis ganglio pulmonar** Los ganglios linfáticos hiliares permiten diferenciar entre tuberculosis primaria y post-primaria. En la Figura [3.4] se muestra la radiografía del tórax con el patrón ganglio pulmonar extendido infiltrado, que es la inflamación de los ganglios [36].

**Tuberculosis pleural** Pablo Molina [36] menciona que el derrame pleural es más frecuente que se presente al inicio. Es consecuencia de la inflamación pleural granulomatosa aguda. Por lo general, se presenta como un derrame pequeño a moderado. En la Figura [3.5] se muestra una radiografía con signos de tuberculosis pleural, en el pulmón de lado izquierdo.

**Tuberculosis miliar** Se debe a una amplia diseminación pulmonar y se presenta en la tuberculosis primaria y secundaria. Esta consiste en que en la radiografía se

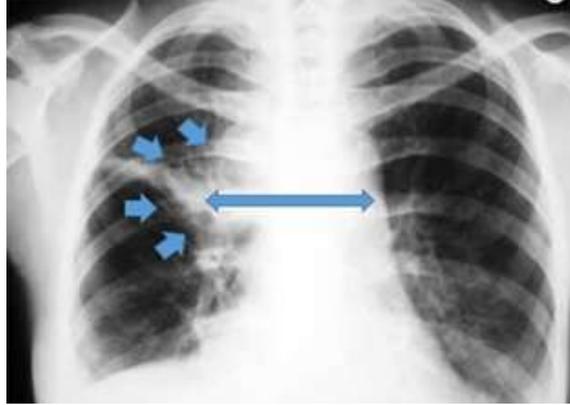


Figura 3.4: Tuberculosis ganglio pulmonar.

Fuente: [36]

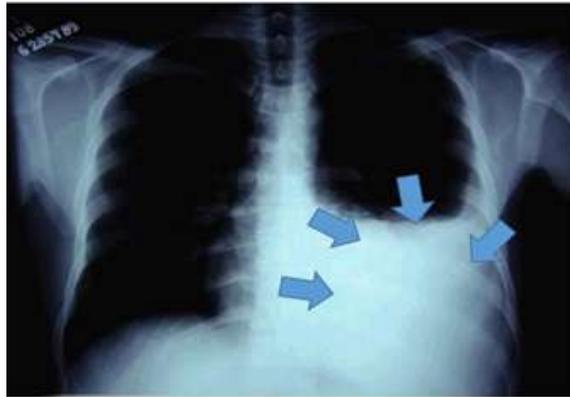


Figura 3.5: Tuberculosis pleural.

Fuente: [36]

presentan una multitud de nódulos pequeños los cuales varían entre 1 y 3 mm de diámetro. En la Figura [3.6] se observan pequeños nódulos [35].

**Afección del parénquima (Consolidación) o linfadenopatía.** Por lo general afecta la totalidad del lóbulo medio y es causada por la invasión directa de los bacilos, este patrón se observa aproximadamente en el 25% de los casos. Algunas veces es el resultado de la combinación de consolidación y atelectasia. El parénquima pulmonar se le conoce como neumonía tuberculosa. En la Figura [3.7] se muestra como está difuminado el lóbulo superior izquierdo [3].

**Broncoestenosis (fibrosis).** La fibrosis y broncoestenosis son consecuencia de la

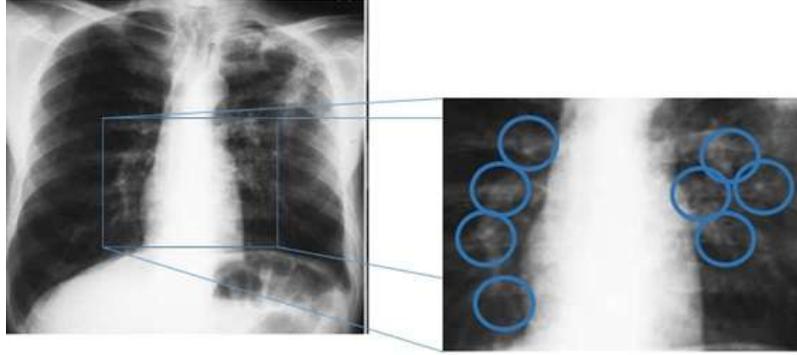


Figura 3.6: Tuberculosis miliar.

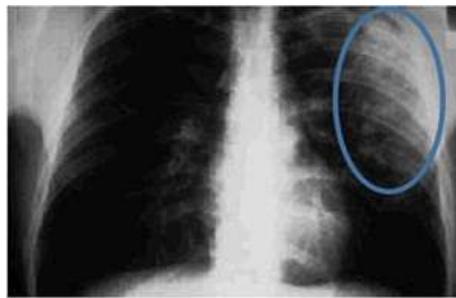


Figura 3.7: Afección del parénquima

cicatrización. La fibrosis grave conlleva la pérdida de volumen de lóbulos superiores, se observa aproximadamente en el 29 % de los casos. En la Figura 3.8 se muestra la presencia de la fibrosis [3].

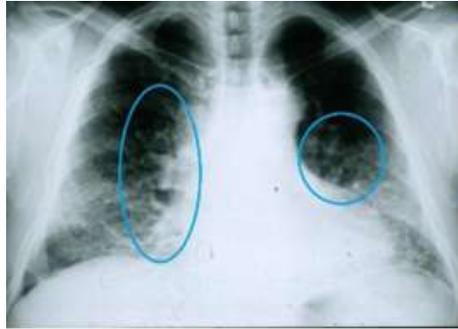


Figura 3.8: Fibrosis.

**Granuloma pulmonar calcificado, lesión de Ghon y complejo de Ranke.** Aproximadamente el 30 % de los casos presentan calcificación. En casos de TB primaria, los síntomas no son visibles de inmediato, puede tardar entre 6 meses y dos años en manifestarse. La cicatriz parenquimatosa de Ghon se produce en alrededor del 15 al 17% de los casos. El complejo de Ranke se forma por la combinación de la calcificación y la lesión de Ghon. En la Figura ?? se puede observar una lesión de complejo de Ranke en una sección del pulmón derecho. [3]



Figura 3.9: Tuberculosis pulmonar primaria (complejo de Ranke).

**Tuberculomas.** Son opacidades nodulares ya sea de forma redonda u ovalada que se sitúan en el lóbulo superior, los nódulos son regulares y bien definidos tiene un diámetro de entre 3 cm hasta 5 cm. En la Figura 3.10 se aprecia un tuberculoma [3].

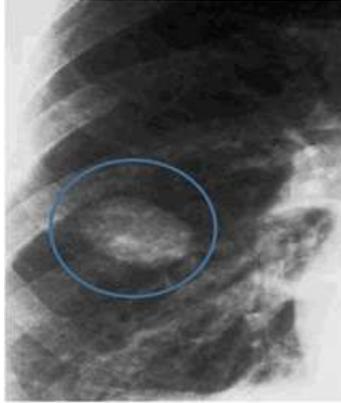


Figura 3.10: Tuberculoma

**Cavitación.** Por lo general es una lesión de contenido aéreo con una pared gruesa que es superior a 4 mm y al principio suelen ser pequeñas y múltiples y conforme pasa el tiempo se va haciendo la cavidad más grande. Se presenta entre un 7 y 29% en adultos. En la Figura 3.11 se muestra un patrón de cavitación en el pulmón derecho [35].



Figura 3.11: Tuberculosis pulmonar cavitada.

Fuente: [35]

## 3.2. Procesamiento de imágenes

### Imágenes

Se define como una función  $I(x, y)$  de dos dimensiones en el plano  $(x, y)$ , el modelo más común de representar una imagen es por medio de una matriz, cuyos valores

escalados pueden ir en rangos de  $[0, 255]$  [37].

$$\begin{pmatrix} I(1,1) & I(1,2) & \cdots & I(1,M) \\ I(2,1) & I(2,2) & \cdots & I(2,M) \\ \vdots & \vdots & \ddots & \vdots \\ I(N,1) & I(N,2) & \cdots & I(N,M) \end{pmatrix} \quad (3.1)$$

Una imagen se representa por un conjunto finito de  $M \times N$  elementos, los cuales se le denomina píxel.

Las imágenes digitales son descritas por los bits que representan cada píxel, una imagen con 1 bit es monocroma, si contiene 8 bits contiene 256 colores o tonos [37].

Las imágenes se clasifican en dos tipos: Imágenes vectoriales e imágenes de mapas de bits. Las imágenes vectoriales son formadas por objetos geométricos como líneas, curvas y polígonos, etc., este tipo de imágenes tienen escalabilidad y no pierden calidad. Las imágenes de mapas de bits están conformadas por píxeles, esta tiene una gama mayor de tonos.

Un píxel (picture element) guarda información de la intensidad de un solo color, en el caso de la profundidad de tonos, es decir la cantidad de bits para guardar información, ya sea de 1 bit es igual a 2 colores, 2 bits es igual a 4 colores y 8 bits igual a 256 tonos.

Cuando se tiene una imagen a blanco y negro, es una matriz de 2 dimensiones y los valores van de 0 a 255, pero cuando se tiene una imagen a color, se tiene una matriz de 3 dimensiones con valores de 0 a 255, cada dimensión representa una capa de color RGB (rojo, verde y azul).

### Modelos de color

Es un conjunto de colores primarios a partir de los cuales se generan nuevas mezclas hasta cubrir todo el espectro visible.

RGB: este modelo es aditivo ya que la suma de los colores primarios produce el blanco y la ausencia de color genera el negro, cuenta con 3 canales.

También existen otros formatos de color como CMYK es un modelo sustractivo y se utiliza en la impresión a partir de pigmentos de tres colores básicos, comprende los colores Cian, Magenta, Yellow y Key, siendo este último color el color negro.

### Profundidad del color

Dependiendo de la cantidad de colores utilizados en la imagen influye en el tamaño del archivo que la contiene en la Tabla 3.1 se muestra los tonos posibles de acuerdo a los bits por pixel.

Profundidad de color	Tonos(colores)	
1 bit	2 tonos	Modo mapa de bits.
4 bit	16 tonos	Escala de grises.
8 bit	256 tonos	Color indexado y escala de grises.
16 bit	65.536 tonos	High color.
24 bit	16.777.216 tonos	RGB.
32 bit	4.294.967.296 tonos	MCYK.

Tabla 3.1: Profundidad de color

### Modelo HSV y HSL

Estos modelos incluyen otros parámetros adicionales al matiz o croma para obtener el color, HSL (H- hue o matiz, S- saturation o saturación, L- luminosity o luminosidad), HSV (H- hue o matiz, S- saturation o saturación, V-value o valor). La diferencia entre HSV Y HSL es que HSV la saturación va desde el negro al color y el HSL va del negro al blanco.

#### 3.2.1. Tratamiento de imagen

**Vecindad** Un pixel  $p(x,y)$  tiene 4 vecinos horizontales y verticales en las coordenadas  $(x,y)$  tiene cuatro vecinos horizontales y verticales, también cuenta con 4 vecinos diagonales, en la Tabla 3.2 se muestra las coordenadas. 38

	Posición	Coordenada
Coordenada horizontal y vertical		
	Superior	$(x - 1, y)$
	Inferior	$(x + 1, y)$
	Izquierdo	$(x, y - 1)$
	Derecho	$(x, y + 1)$
Coordenadas diagonales		
	Superior izquierdo	$(x - 1, y - 1)$
	Superior derecho	$(x - 1, y + 1)$
	Inferior izquierdo	$(x + 1, y - 1)$
	Inferior derecho	$(x + 1, y + 1)$

Tabla 3.2: Profundidad de color

A los píxeles horizontales y verticales se le conocen como  $4 - vecino$  denotado por  $N_4(p)$  y los vecinos diagonales como  $N_D(p)$ .

Se le llama a  $N_D(p) \cup N_4(p)$  los  $8 - vecinos$  y se denota  $N_8(p)$ .

A estos se le llaman los  $8 - vecinos$  de  $p$  y se escriben  $N_8(p)$ .

La conductividad entre píxeles sirve para establecer los límites en objetos y regiones de componentes de una imagen, para ello se necesita conocer la adyacencia. Por ejemplo, si una imagen binaria puede tener *vecindadde* 4, pero se consideran conectados si tienen el mismo valor.

### Adyacencia

Un píxel  $p(x, y)$  tienen un vector de intensidades:

$$p = \langle p_1, p_2, \dots, p_R \rangle \quad (3.2)$$

Donde  $R =$  número de filtros canales de la imagen.

Los píxeles adyacentes  $p$  y  $q$  son adyacentes si presentan algún tipo de relación 38

### 3.2.2. Segmentación de imágenes

Consiste en dividir la imagen en partes o regiones con el fin de resaltar áreas que tengan características específicas. La clasificación de los píxeles también puede ser vista como la asignación de clases a las que pertenecen. Los algoritmos de segmentación de imágenes blancas y negras se fundamentan en una de dos propiedades, ya sea discontinua o similaridad. [39]. Las propiedades discontinuas, divide la imagen debido a los cambios bruscos del nivel de gris, las propiedades que se buscan son [39]:

- La detección de puntos aislados , líneas y bordes de una imagen.

Similaridad, se presenta el nivel de gris, las propiedades con las que cuenta son:

- Umbralización, para separar el fondo.
- Crecimiento de región, para destacar.
- División y fusión de regiones utilizando una representación piramidal.

Segmentación binaria por umbral:

$$I(x, y) = \begin{cases} 0, & I(x, y) < u \\ 255, & I(x, y) \geq u \end{cases} \quad (3.3)$$

#### Detección de bordes

Es el punto de transición entre dos tonalidades distintas de gris en una imagen. Las discontinuidades se identifican utilizando derivadas de primer y segundo orden. El operador gradiente se emplea para las derivadas de primer orden, mientras que el operador laplaciano se utiliza para las de segundo orden. La función  $f(x, y)$  que define el gradiente de una imagen es un vector bidimensional. El gradiente es un vector que se define como la derivada en un punto determinado.

$$\nabla z(x, y) = \begin{bmatrix} Z_x, Z_y \end{bmatrix} = \begin{bmatrix} \frac{\partial z(x, y)}{\partial x} \\ \frac{\partial z(x, y)}{\partial y} \end{bmatrix} \quad (3.4)$$

$$Mag[\nabla f(x, y)] = \sqrt{\left(\frac{\partial z(x, y)}{\partial x}\right)^2 + \left(\frac{\partial z(x, y)}{\partial y}\right)^2} \quad (3.5)$$

$$\theta = \operatorname{arctag} \frac{\frac{\partial z(x,y)}{\partial x}}{\frac{\partial z(x,y)}{\partial y}} \quad (3.6)$$

El laplaciano de una función bidimensional  $f(x, y)$  es un operador diferencial denotado por  $\nabla^2$ , se utiliza para la minimización de ciertas magnitudes y se representa en la ecuación.

$$\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial^2 x} + \frac{\partial^2 f(x, y)}{\partial^2 y} \quad (3.7)$$

**La umbralización** Un método de gran importancia es la conversión de imágenes de tonalidades variadas a blanco y negro, conocido como escala de grises. Dada una imagen original  $f(x, y)$ , la imagen umbralizada se denota como  $g(x, y)$  y se determina fijando un umbral específico. La operación de umbralizado se define como:

$$\begin{aligned} g(x, y) &= 225 & f(x, y) &> \text{Umbral} \\ g(x, y) &= 0 & f(x, y) &\leq \text{Umbral} \end{aligned} \quad (3.8)$$

Umbral global, es una técnica de segmentación que permite separar los pixeles de las imágenes, el umbral fijo o global  $T$ , es único sobre la imagen [39].

$$b(r, c) = \begin{cases} t_0 & \text{si } l(r, c) < T \\ t_1 & \text{si } l(r, c) \geq T \end{cases} \quad (3.9)$$

### 3.3. Inteligencia artificial

La inteligencia artificial es la construcción de sistemas como si fuera echo por una persona, y esa acción se diría que es inteligente. [40] También se puede definir cómo el tratar de imitar el comportamiento humano ya que se le caracteriza por aprender por medio de la experiencia, tiene la capacidad de memorizar y recibe conocimiento genético a partir de un conjunto de datos [41].

El nombre de Inteligencia Artificial surgió en 1955 por John McCarthy quien es considerado como el padre de la inteligencia artificial. Un primer acercamiento a la inteligencia artificial fue el matemático Alan Turing quien argumentaba que si una máquina puede actuar como un humano entonces se puede decir que es inteligente [42].

### Redes neuronales artificiales

Las redes neuronales ofrecen medios para ayudar a resolver problemas complejos. Se basa en la analogía del comportamiento del cerebro humano [41].

Las neuronas en los seres humanos son la unidad estructural y funcional del sistema nervioso, se compone de tres partes: *las dendritas, el axón y la sinapsis* [43].

- Las *dendritas* se dividen formando ramificaciones como si fuera un árbol, tienen la función de recibir impulsos de otras neuronas y enviarlos hasta la sinapsis.
- *Axón* es la propagación de la sinapsis que va a dirección opuesta a las dendritas y conduce un impulso nervioso desde la sinapsis hacia otra neurona.
- La *sinapsis* conecta el axón de una neurona a las dendritas, es la zona de transición de impulsos eléctricos entre dos neuronas.

La estructura de las redes neuronales artificiales consiste [44] en un conjunto de *entradas (axón)*, un conjunto de pesos (*sinapsis*), un umbral (*cuerpo*), una *función de activación (cuello del axón)* y una *salida (axón)*.

En primera instancia se tienen las entradas en la cual se reciben los datos o parámetros, estas se multiplica por el peso, el peso es determinado por el algoritmo de aprendizaje, posteriormente al resultado se le suma la función no lineal llamada *función de activación*, si la salida de la función excede al valor de la entrada está genera una salida, de lo contrario cambia los pesos y nuevamente realiza las operaciones hasta llegar al valor deseado [41].

En la Figura 3.12 se muestra un diagrama describiendo la estructura de una neurona biológica y la artificial, se hace mención a la arquitectura de un perceptrón simple.

El perceptrón simple fue introducido por Rosenblatt en 1958 quien propuso una red que resuelve problemas linealmente separables [41]. Matemáticamente la salida de la neurona se puede describir de la siguiente manera.

$$\text{Salida} = f(wx + b) \tag{3.10}$$

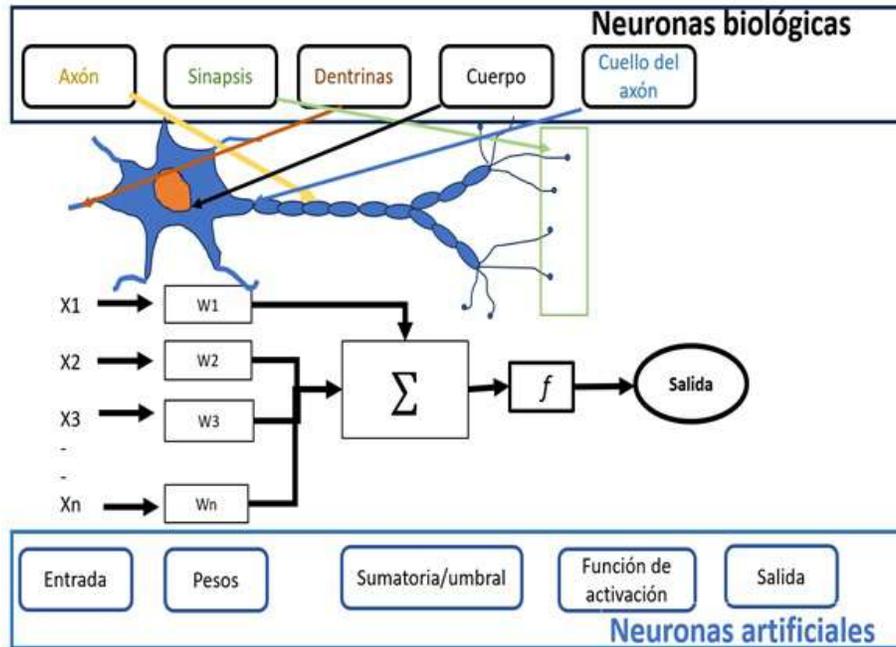


Figura 3.12: Redes neuronales artificiales y redes neuronales biológicas.

Fuente: Elaboración propia

El *bias* es parecido a los pesos solo que tiene una entrada constante de 0 y 1.

*Funciones de activación:* Las funciones de activación calculan el estado de actividad de una neurona, cuyo valor de activación va de (0) a 1 se denota por:

$$y_i = \begin{cases} 1 & \text{si } \sum_{j=1}^n w_{ij}x_{ij} \geq \theta_i \\ 0 & \text{si } \sum_{j=1}^n w_{ij}x_{ij} < \theta_i \end{cases} \quad (3.11)$$

En la Figura [3.13](#) se mencionan algunas funciones de activación que se utilizan para el entrenamiento de la red neuronal, las funciones de activación se usan dependiendo de las entradas [41](#) , [45](#).

### Algoritmos de aprendizaje

**Supervisado:** Son aquellos que aprenden con funciones asociadas entre las entradas y etiquetas para poder aprender. Se suelen usar en problemas de clasificación o de regresión (lineal, polinomial, árboles de decisión, árboles aleatorios (*random forest*), logística, etc.)

Este algoritmo también se suele utilizar en el modelo del *perceptrón* y algunas

<b><i>Función</i></b>	<b>Ecuación</b>	<b>Grafica</b>
<i>Linear</i>	$f(x) = x$ Rango $[-\infty, +\infty]$	
<i>Sigmoide</i>	$f(x) = \frac{1}{1+e^{-x}}$ Rango $[0,+1]$ $y = tgh(x)$ Rango $[-1,+1]$	
<i>Relu</i>	$f(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{otro caso} \end{cases}$ Rango $[0,1]$	
<i>Tang</i>	$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$	

Figura 3.13: Funciones de activación.

Fuente: [44]

de las características son: Permite que la red aprenda problemas complejos ya que se desactivan una parte de las neuronas de entrada y salida, permitiendo que la red aprenda [46].

En la Figura 3.14 se muestra un ejemplo en el cual se le da al algoritmo de aprendizaje la imagen en este caso es la radiografía del tórax y la etiqueta a la que corresponde al nombre a la que pertenece la radiografía (RX con Tuberculosis), el algoritmo lo aprende y genera un modelo, que cuando posteriormente se le dé una imagen ya la podrá clasificar.

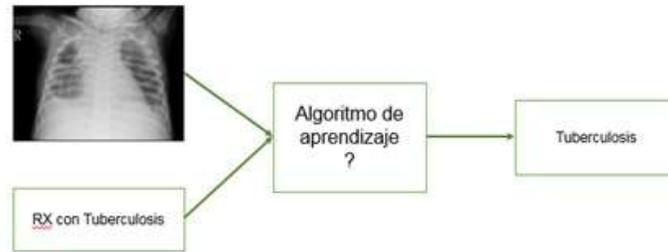


Figura 3.14: Aprendizaje supervisado

Fuente: Elaboración propia

**No supervisado:** Los datos de entrenamiento consisten sólo en los patrones de entrada y aprende de las experiencias recogidas de los patrones de entrenamiento anteriores. Se usa en problemas de análisis de grupos (clustering), agrupamientos de coocurrencias o profiling [46].

En la Figura 3.15 se muestra un ejemplo en cual deja que el algoritmo determine los patrones de datos por sí mismo. Este tipo de algoritmo tiende a reestructurar los datos en nuevas características o nuevos valores no correlacionados por ejemplo se pueden dar círculos, rectángulos y triángulos, y este lo agrupa por medio de las similitudes que presenta. Así cuando se le dé una imagen nueva esta la agrupa.



Figura 3.15: Aprendizaje no supervisado

Fuente: Elaboración propia

**Reforzado:** El algoritmo tiene que decidir cómo actuar para realizar la tarea, el algoritmo aprende de la experiencia, recoge los ejemplos de entrenamiento a través de ensayo y error mientras intenta su tarea, el objetivo es maximizar la recompensa a largo plazo. En la Figura 3.16 se tiene un ejemplo en el cual se le da una muestra de una imagen y el algoritmo la caracteriza y da una respuesta, pero si no es correcta se le indica al algoritmo hasta que de la respuesta sea correcta.

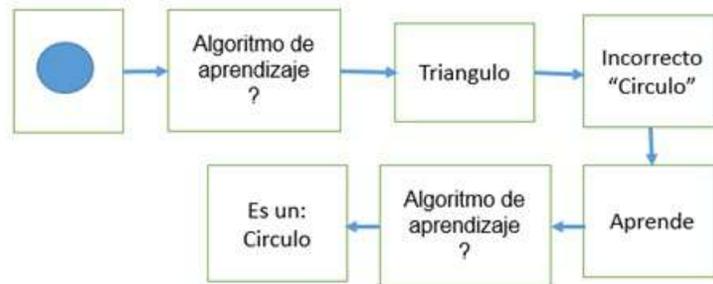


Figura 3.16: Aprendizaje reforzado

Fuente: Elaboración propia

### 3.3.1. Redes neuronales convolucionales

Las redes neuronales convolucionales o también llamadas ConvNet son un método de aprendizaje profundo, el cual es fundamental cuando se quiere analizar una imagen, esta se compone de capas de convolución, capas agrupadas y capas completamente conectadas, fue diseñada para aprender de forma automática y adaptativa [47] [48].

- Las dos primeras capas de convolución y agrupación se utilizan para realizar la extracción de característica y la tercera capa mapea las características extraídas [44]. Una convolución puede describirse como una función que trabaja con dos funciones ( $f$  y  $g$ ) las cuales son transformadas en una tercera función después de desplazar una de ellas una distancia  $t$ .

Si  $f$  y  $g$  son funciones ent, entonces la convolución es un intervalo finito integrado por [44]:

$$(f * g) = \int_{-\infty}^{\infty} f(\tau) g(t - \tau) d\tau \quad (3.12)$$

Las propiedades de los diferentes operadores son:

1.  $f * g = g * f$  (Ley conmutativa)
2.  $f * (g + h) = f * g + f * h$  (Ley distributiva)
3.  $(f * g) * h = f * (g * h)$  (Ley asociativa)

Para hacer el reconocimiento de imágenes se utilizan las redes neuronales convolucionales (CNNs) con el fin de extraer características de una imagen, la cual filtra las imágenes colocando diferentes máscaras, esta utiliza un entrenamiento supervisado y no supervisado, en la Figura 3.17 se muestra la arquitectura en general de una CNN.

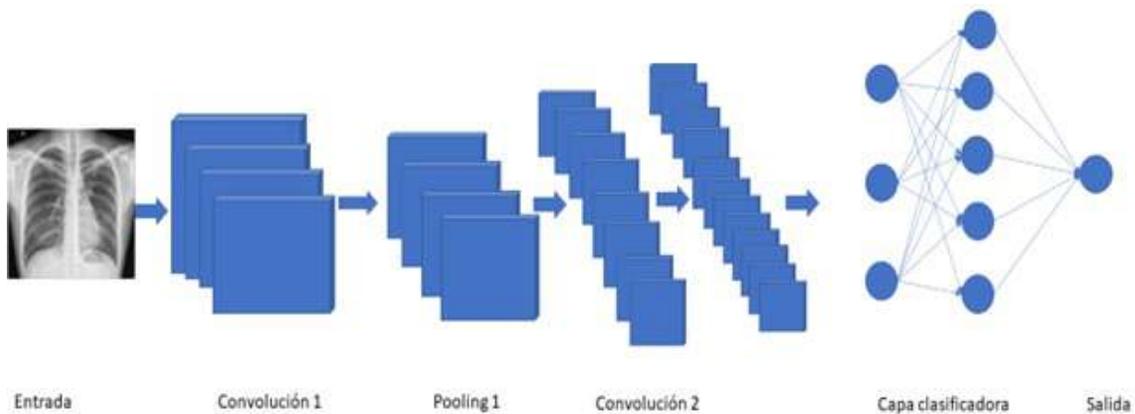


Figura 3.17: Arquitectura CNN

Fuente: Elaboración propia

Entrada de una red convolucional [49]  $x^l$  en forma de un tensor de 3er orden donde  $x^l \in R^{(M_i \times N_j \times K_d)}$  en donde  $M$  es la altura y  $N$  es el ancho de la imagen y tres canales son; rojo, verde, y azul, en este caso  $M = 256$ ,  $N = 256$ , como se muestra en la figura 3.18. Cada elemento es designado por el índice  $(i; j; d)$ , donde  $0 \leq i < M, 0 \leq j < N$  y  $0 \leq d < 3$ .

Las CNN toma pixeles cercanos de la imagen de entrada y opera contra una pequeña matriz de  $3 \times 3$  la cual recorre todas las neuronas de entrada de izquierda a derecha, de arriba-abajo y genera una nueva matriz de salida, la cual será la nueva capa de neurona. A este detector de propiedades se le conoce como *Kernel* o detector de características. A las nuevas matrices de salida se le conoce como mapeo de características [50], [51].

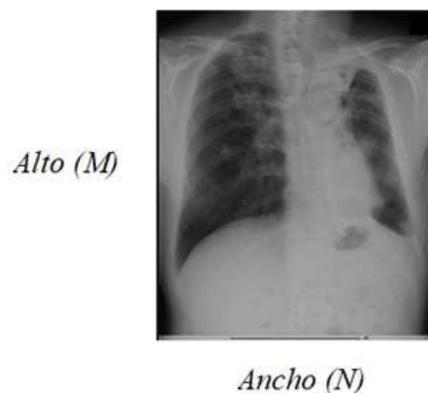


Figura 3.18: Canales de las imágenes

Fuente: Elaboración propia

### El modelo general de la CNN:

**Capa de convolución:** Primero se proporciona la imagen a clasificar, es este caso es la entrada y la salida es la etiqueta de la clase [44], posteriormente se realiza la extracción de características mediante el campo receptivo, la cual se asocia a una capa conectada de la capa anterior y forma un vector de peso, como las neuronas tienen el mismo peso, las características son similares, por ello los datos de entrada se pueden detectar [52], [45].

En esta capa se extrae  $N$  números de características y debido a la convolución el número de parámetros entrenables son reducidos significativamente [52]. En esta capa  $w^l$  se usan múltiples kernels y convoluciones.

El *kernel* es asumido como  $K$  y cada *kernel*  $M \times N$  usados son denotados como  $f$ , donde  $f$  es un tensor de cuarto orden  $R^{M \times N \times k^l \times k}$  y la variable índice  $0 \leq i < M, 0 \leq j < N, 0 \leq d^l < K^l, 0 \leq d < K$  se usan para señalar el elemento del núcleo. *Stride* es el paso del *kernel* que va a atravesar la imagen, donde su valor  $s = 1$ . El proceso de convolución se expresa:

$$y_{i^l, j^{l+1}, d} = \sum_{i=0}^M \sum_{j=0}^N \sum_{d^l=0}^{d^l} f_{i, j, d^l, d} \times x_{i^l+1+i, j^{l+1}+j, d^l} + b_k \quad (3.13)$$

Donde el *bias* constante  $b_k$  es agregado a la ecuación con el valor de 1.

**Capa de agrupación o Pooling Capas:** proporciona una operación de reduc-

ción, reduce la dimensionalidad en el plano de los mapas de características. Este genera otro vector de salida. Se usan para reducir las dimensiones para disminuir los tiempos del procesado, aunque al reducir el tamaño conlleva a la pérdida de información.

La operación que más se utiliza es *max-pooling*, que divide la imagen en rectángulos y cada uno de ellos conserva el valor más grande [53]. Por ejemplo en la Figura 3.19 se aprecia una matriz de 4x4 la cual se reduce a una matriz de 2 x 2 y esta tiene los valores máximos de la primera matriz.

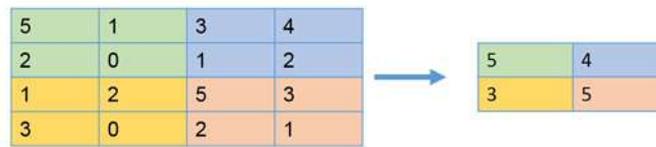


Figura 3.19: Max-pooling

Fuente: Elaboración propia

La ecuación de la salida de esta capa es:

$$y_j^l = f(\beta_j \text{down}(y_j^l - 1) + b_j) \quad (3.14)$$

Representa la función de submuestreo.

El agrupamiento  $L_p$  = Fue creado por medio del proceso de agrupamiento biológico en las células complejas [44], este método de una mejor generalización que es el *max-pooling*. Se representa de la siguiente manera:

$$y_{i,j,k} = \left[ \sum_{(m,n) \in R_{i,j}} (a_{m,n,k})^p \right]^{\frac{1}{p}} \quad (3.15)$$

$y_{i,j,k}$  = es la salida del operador en la posición  $(i, j)$

$a_{m,n,k}$  = el valor de las características en la posición  $(m, n)$ .

$R_{i,j}$  = Es la agrupación de  $K$  – *esimo* mapa de características.

Capa de agrupación o *Pooling* Capas. Proporciona una operación de reducción; reduce la dimensión en el plano de los mapas de características. Este genera otro vector de salida, la función se denota [44] [49]:

$$\max : y_{i^{l+1}, j^{l+1}, d} = \max x_{i^{l+1} \times M + i, j^{l+1} \times N + j, d^l} \quad (3.16)$$

$$0 \leq i < M, 0 \leq j < N \quad (3.17)$$

Donde  $0 \leq i^{l+1} < M^l, 0 \leq j^{l+1} < N$  and  $0 \leq d < K^{l+1} = K^l$

**Los parámetros de una convolución son:**

**Numero de filtros o kernel:** Es el tamaño del kernel que define el campo de visión de la convolución. Se desliza sobre el vector de entrada para generar el mapa de características. Este filtro por lo general es de  $3 \times 3$  o de  $5 \times 5$ .

**Depth:** Es la cantidad de filtros que deseamos usar.

**Stride:** Se define el tamaño del paso del kernel que va a atravesar la imagen. Su valor predeterminado suele ser 1.

**Padding:** Coloca ceros en el borden de la imagen, esto permite controlar el tamaño espacial de los volúmenes de salida.

**Output Capas.** Una vez que se tiene esta nueva capa se le aplica la función llamada Softmax que conecta con la capa de salida, esta contiene la cantidad de neuronas que se están clasificando. La función lógica de  $r - dimensionado(z)$  con valor entre  $[0; 1]$ . Se representa [49] :

$$\sigma : R^r \rightarrow [0, 1]^r \quad \sigma(z) = \frac{e^{z_j}}{\sum_{r=1}^r e^{z_r}} \text{ for } j = 1, \dots, r \quad (3.18)$$

Donde  $z$  es la entrada a la capa de la salida y  $r$  es la dimensión del vector  $z$  y  $j$  son los índices de la unidad de salida [49].

Para el entrenamiento de la red la dimensión del modelo es de  $256 \times 256$ , el tamaño del *batch size* es de 10 y el valor de la taza de aprendizaje es de 0.0001, para la función de perdida se usó el optimizador *Adam* [54].

### 3.3.2. ResNet50

Red residual es una red neuronal presentada por Kaiming He, Xiangyu Zhang, Shaoqing Ren y Jian Sun [55] cuenta con 50 capas de redes neuronales, las capas adicionales ayudan a resolver problemas complejos de manera más eficiente, ya que las diferentes capas pueden entrenarse para diferentes tareas para obtener resultados de alta precisión. Si bien la cantidad de capas apiladas pueden enriquecer las características del modelo, una red más profunda puede mostrar el problema de degradación, esto significa que los niveles de precisión pueden saturarse y degradarse. Es por ello que en esta red se creó para abordar ese tema [56].

*ResNet* introdujo el concepto de conexión de salto, en la Figura 3.20 en la parte izquierda se muestra las capas convolucionales apiladas juntas una contra otras y en la parte derecha igual se apilan capas de convolución, pero ahora también se agrega la capa original de la entrada a la salida del bloque de la convolución.

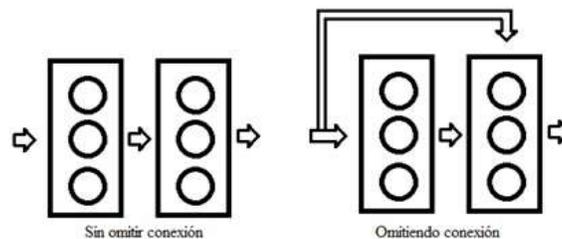


Figura 3.20: Ejemplo de conexión de salto.

Fuente: Elaboración propia

En la Tabla 3.3 se muestra las capas con las que cuenta la arquitectura ResNet50, al igual que los pasos y los *Kernel* [55]

Capas	Salida	Kernel
Conv1	(112, 112, 64, stride 2)	7
Maxpooling	(56,56,3)	3
Conv 2	(56,56,64)	1
x3 Conv 2	(56,56,64)	3
Conv 2	(56,56,256)	1
Maxpooling	(56,56)	
Conv 3	(28,28,128)	1
x4 Conv 3	(28,28,128)	3
Conv 3	(28,28,512)	1
Maxpooling	(28,28,32)	
Conv 4	(14,14,256)	1
x5 Conv 4	(14,14,256)	3
Conv 4	(14,14,1024)	1
Maxpooling	(14,14)	
Conv 5	(7,7,512)	1
x6 Conv 5	(7,7,512)	3
Conv 5	(7,7,2048)	1
Average pool,	1000-d	FC, softmax
FLOPs	$3.8 \times 10^9$	

Tabla 3.3: Arquitectura de la red ResNet50

### 3.3.3. InceptionV3

Es una arquitectura de red neuronal convolucional, se entrenó con un conjunto de datos de 1000 clases, el conjunto de original de imágenes con más de 1 millón de imágenes de entrenamiento.

Se lanzó en el 2015 y contiene un total de 42 capas. Las modificaciones que se realizaron son: en esta se realiza la factorización en circunvoluciones más pequeñas: Para hacerlo aún mejor, las circunvoluciones más grandes del modelo se factorizaron en circunvoluciones más pequeñas. Por ejemplo, si se tiene una capa de convolución de  $5 \times 5$  esta es computacionalmente cara, para corregir esto se pueden tomar dos capas de  $3 \times 3$  para reducirlo [56], [1]

Para entenderlo mejor, se puede visualizar en las Figuras 3.21 3.24, el proceso de usar dos convoluciones  $3 \times 3$  reduce la cantidad de parámetros.

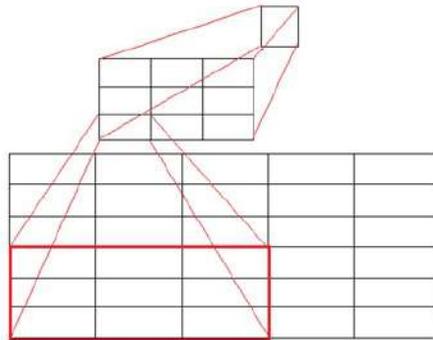


Figura 3.21: Factorización en circunvoluciones más pequeñas

Figura 3.22: Fuente: Elaboración propia

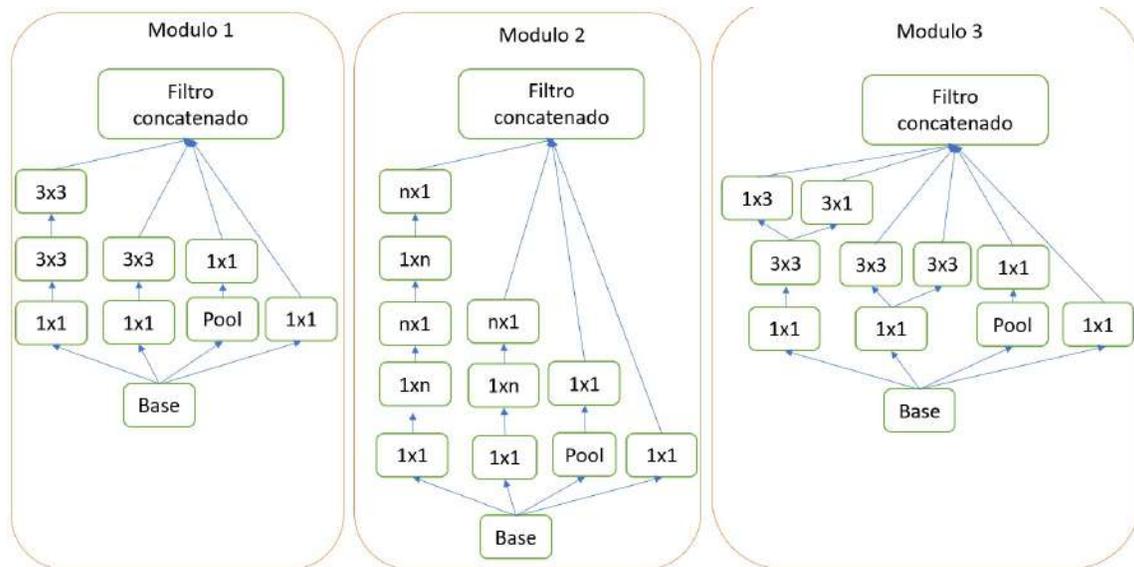


Figura 3.23: Módulos de red InceptionV3

Como resultado del número reducido de parámetros, los costos computacionales también se reducen.

Factorización espacial en circunvoluciones asimétricas estas son de la forma  $n * 1$ , en la cual para remplazar una convolución de  $3 \times 3$  se hace con dos convoluciones la primera de  $1 * 3$  y la otra de  $3 \times 1$ . Lo cual hará que el costo computacional sea más barato.

En la Tabla 3.5 se muestra la arquitectura de red InceptionV3 [\[1\]](#).

Capas	Salida	Kernel
Conv1	(299, 299, 3)	3
Conv	(149,149,32)	3
Conv padded	(147,147,32)	3
Pool	(147,147,64)	3
Conv	(73,73,64)	3
Conv	(71,71,80)	3
Conv	(35,35,288)	3
3 x Inception	(35,35,228)	Modulo 1
5 x Inception	(17,17,768)	Modulo 2
2 x Inception	(8,8,1280)	Modulo 3
Pool	(8,8,2048)	8
Linear	(1,1,2048)	
Softmax	(1,1,1000)	

Tabla 3.4: Arquitectura de la red InceptionV3

Tabla 3.5: Fuente: [1]

### 3.3.4. VGG16

Es una red neuronal convolucional con 16 capas, fue entrenada con el data set ImageNet, fue desarrollada por Karen Simonyan y Andrew Zisserman de la Universidad de Oxford en el año 2014 [2]. En la Tabla 3.7 se muestra la arquitectura de la red VGG16.

Capas	Salida	Kernel
imagen	(224, 224, 3)	3
2 x Conv	(224,224,64)	3
Max pooling	(112,112,64)	3
2 x Conv	(112,112,128)	3
Max pooling	(56,56,128)	3
2 x Conv	(56,56,256)	3
Max pooling	(28,28,256)	3
3 x Conv	(28,28,512)	3
Max pooling	(14,14,512)	3
3 x Conv	(14,14,512)	3
Max pooling	(7,7,512)	
FC	(25088)	
FC	(4096)	
FC	(4096)	
FC	(1000)	

Tabla 3.6: Arquitectura de la red VGG16

Tabla 3.7: Fuente: [2]

**Comparativo de arquitecturas de algunas CNN** Existen diferentes arquitecturas de redes neuronales convolucionales las cuales son propuestas para mejorar ya sea los tiempos de entrenamiento o la precisión, en la Tabla 3.9 se hace mención de algunas de ellas, y se colocaron las diferencias que hay de una red a otra [56].

Red	Año	Capas	Función	Filtro o kernel	Stride	Maxpooling	FC	Softmax
CapsNet	2011	6	ReLU	9*9	1	3	1	
Alexnet	2012	8	ReLU	11*11	2,3	3 Maxpooling	3	1
VGG16	2014	16	ReLU	3*3	2	4 Maxpooling	2	1
GoogLeNet	2015	100	ReLU	1*1, 3*3 y 5*5	2	16 pooling	1	1
InceptionV3	2015	42	ReLU	5*5 y 3*3	n*1	Maxpooling	1	1
ResNet50	2016	50	ReLU	1*1, 3*3	2	Maxpooling	1	1
CapsNet	2017	9	ReLU	9*9, 5*5 y 1*1	2,1		3	1
FPN	2017	3	ReLU	1*1 y 3*3	1	Maxpooling	1	1

Tabla 3.8: Diferencias de arquitecturas de CNN

Tabla 3.9: Fuente: Elaboración propia

### 3.4. Características de los datos

Los datos deben estar balanceados para poder hacer el entrenamiento, es decir, en todas las categorías tengan la misma cantidad de imágenes y estos datos se dividen en 3 conjuntos, el primer conjunto es el de entrenamiento, el segundo de validación y el tercero de muestra o prueba [3.24](#).



Figura 3.24: Proporción de los datos.

Se usa los conjuntos de validación y de prueba por separado porque los modelos ajustan sus hiperparámetros.

### 3.5. Transferencia de aprendizaje

Es el aprendizaje transferido, se refiere al conjunto de métodos que permiten transferir conocimientos adquiridos gracias a la resolución de problemas, con ello se pueden desarrollar rápidamente modelos eficaces y resolver problemas complejos de *Deep Learning* [57].

Al usar el aprendizaje transferido se pueden entrenar modelos con menor datos etiquetados, esto se logra reutilizando modelos de uso habitual que ya se han entrenado con el conjunto de datos de libre uso *ImageNet*, esta base de datos visual cuenta con 14 millones de imágenes y contiene más de 20.000 categorías. El aprendizaje transferido reduce el tiempo de entrenamiento y los recursos informáticos, puesto que los pesos no se aprenden desde cero ya que el modelo fue previamente entrenado y estos están guardados en un archivo.

### 3.6. Maquinas de soporte vectorial

Son algoritmos que fueron desarrollados por Vladimir Vapnik en 1988 [11], permite encontrar la forma más óptima de clasificar de entre varias clases, por lo que maximizan el margen de separación de las clases, en la figura 3.25 se muestra la separación de dos clases, en este caso los puntos color azul y los puntos naranjas.

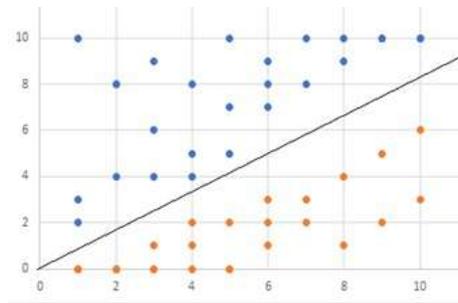


Figura 3.25: Separación entre clases.

*Support Vector Machine (SVM)* representar un conjunto de datos multidimensionales en un espacio donde los elementos de datos pertenecientes a diferentes clases están separados por un hiperplano. El clasificador *SVM* tiene la capacidad de minimizar el error de generalización en datos invisibles. El hiperplano de separación también se denomina hiperplano óptimo. El clasificador *SVM* se entrena inicialmente con nuevas características generadas automáticamente de imágenes de entrenamiento. Finalmente, el clasificador *SVM* entrenado se usa para reconocer las clases [37].

Se utilizan diferentes tipos de kernel: *Linear, Gaussian, RBF, Sigmoid, Polynomial, con las SVM*. El reconocimiento de patrones de los datos de entrenamiento se proporciona de la forma:

$$(x_1, y_1) \cdots (x_n, y_n) \in R_n \times \{+1, -1\} \quad (3.19)$$

$x_1$  = Son los patrones n-dimensiones (vectores).

$y_1$  = Son sus etiquetas.

Para separar las dos clases de puntos de datos se pueden elegir muchos hiperplanos, pero el objetivo es encontrar un plano que tenga la distancia máxima. Los hiperplanos son límites de decisión que ayudan a clasificar los puntos de datos. La dimensión de un hiperplano depende del número de características la ecuación es la siguiente:

$$x = w_0 + w_1 a_1 + w_2 a_2 + \cdots + w_k + a_k = \underline{w} \cdot \underline{a} \quad (3.20)$$

Aquellos puntos  $x$  que se encuentran en el hiperplano de separación satisfacen la siguiente relación:

$$x * w + b = 0 \quad (3.21)$$

Las restricciones que tiene este conjunto de entrenamiento son:

$$x_k * w + b \geq +1 \quad y_k = +1 \quad (3.22)$$

$$x_k * w + b \leq -1 \quad y_k = -1 \quad (3.23)$$

Las SVM son consideradas como clasificadores lineales y en caso que no sean linealmente separables se utiliza el truco de kernel ya que este añade una dimensión nueva para poder separar más fácilmente las clases, en la Figura [3.26] se muestra la nueva

dimensión mediante la cual ayuda a separar las dos clases.

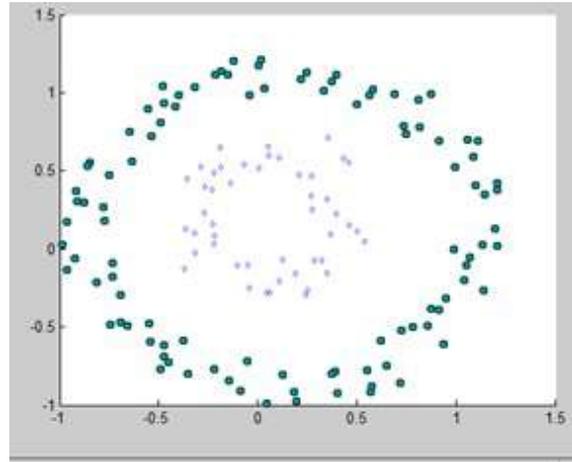


Figura 3.26: Kernel

Las funciones *kernel* que se utilizan por medio de SVM son: [58](#):

**Lineal:** Aprendizaje de dos clases.

$$K(x, x') = x * x' \quad (3.24)$$

**Polinómico**

$$K(x, x') = (x * x' + c)^d \quad (3.25)$$

Si  $d > 1$ , se generan límites de decisión no lineales, no se recomiendan valores mayores a 5 por problemas de overfitting y  $c = 0$

**Gaussiano**

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \quad (3.26)$$

Donde  $\sigma$  representa la altura del *Kernel*.

**Sigmoide**

$$K(x_i, x_j) = \tanh(\beta_0 x_i x_j + \beta_1) \quad (3.27)$$

**RBF**

$$K(x, x') = \exp(-\gamma \|x - x'\|^2) \quad (3.28)$$

## 3.7. Métrica para medir el rendimiento de la red entrenada

En esta sección se detallan dos métricas en las cuales se puede observar cual es el resultado de realizar el entrenamiento para saber cuál es la fiabilidad de la red.

### 3.7.1. Matriz de confusión

El matriz de confusión o de error se muestra en la tabla 3.10 que evalúa el rendimiento de un modelo de clasificación, esta compara los valores reales con los predichos. [59], [60].

		Positivo	Negativo
Valores	Positivo	TP	FP
	Negativo	FN	TN

Tabla 3.10: Matriz de confusión

TP (Verdadero positivo): Son el número de muestras que el modelo identifica correctamente como casos positivos y coinciden, con el valor real.

TN (Verdadero negativo): El resultado del modelo y predice correctamente la clase negativa.

FP (Falso positivo): El modelo predice incorrectamente la clase positiva, cuando en realidad es negativa.

FN (Falso negativo): El modelo predice incorrectamente la clase negativa cuando en realidad es positiva.

#### Métricas de la matriz de confusión

Para evaluar el desempeño de los modelos entrenados, se consideran 4 parámetros: exactitud, precisión, sensibilidad y puntuación F1 [56].

**Exactitud o Accuracy** Es igual a la proporción de predicciones que el modelo

clasifico correctamente.

$$Exactitud = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.29)$$

**Precisión** La precisión son las imágenes clasificadas correctamente sobre el número total de imágenes. Es el valor predictivo positivo

$$Precision = \frac{TP}{TP + FP} \quad (3.30)$$

**Sensibilidad *Recall*** Mide la proporción de negativos reales que se identifican correctamente

$$Especificidad = \frac{TP}{TP + FN} \quad (3.31)$$

**Puntuación F1 o F1-score** Es una medida de precisión y recuperación, puede tener una puntuación máxima de 1 o mínima de 0.

$$PuntuacionF1 = \frac{2TP}{2TP + FP + FN} \quad (3.32)$$

Para poder saber si nuestro entrenamiento fue bueno se tienen que analizar los siguientes parámetros:

Precisión y sensibilidad, si son valores altos significa que el entrenamiento es bueno y hace una buena clasificación. Cuando se tiene una alta precisión y una baja sensibilidad, significa que no se está detectando bien las clases, pero cuando la red clasifica las clases de manera correcta tiene una alta fiabilidad. Baja precisión y alta sensibilidad, detecta bien las clases, pero se equivoca colocando muestras en otras clases. Baja precisión y baja sensibilidad no se logra hacer una buena clasificación.

### 3.7.2. ROC-AUC

La curva ROC es una media del rendimiento de la clasificación  $f(x)$ . Se forma mediante las etiquetas verdaderas positivas y los falsos positivos. En un problema de clasificación se presenta de la siguiente manera: [12].

$$T = \{(x_1^+, 1), \dots, (x_1^+, 1), (x_1^- - 1), \dots, (x_1^-, -1)\} \in (R^n \times \{-1, 1\})^l, \quad (3.33)$$

### 3.7. MÉTRICA PARA MEDIR EL RENDIMIENTO DE LA RED ENTRENADA 47

Donde  $l = l^+ + l^-$ . Tomando en cuenta que la función de decisión tiene forma  $y = \text{sgn}(f(x)) = \text{sgn}((w \cdot x) + b)$ , donde  $y = 1$  si  $f(x) \geq 0$ , o  $y = -1$

El  $AUC$  es la probabilidad que de  $f(x)$  asigne un valor mas alto a un resultado positivo extraído al azar.

Cuando el  $AUC$  es igual a 1, el clasificador logra una precisión perfecta. [61]

$$AUC(f) = Pr(f(x^+) > f(x^-)) \quad (3.34)$$

La curva traza dos parámetros:  $TPR$  (Tasa de verdaderos positivos) y  $FPR$  (Tasa de falsos positivos)

$TPR$  es sinónimo del sensibilidad y se define por:

$$TPR = \frac{TP}{TP + FN} \quad (3.35)$$

$FPR$  es sinónimo de especificidad:

$$FPR = \frac{FP}{FP + TN} \quad (3.36)$$

## 4 Metodología

Se trabajo de acuerdo con la disposición de datos en diferentes tiempos y se plantearon los siguientes experimentos como se muestra en la Figura [4.1](#):

1. Se clasificaron enfermedades pulmonares que se diagnostican a partir de radiografías (rayos X), siendo; COVID-19, Tuberculosis, Neumonía viral y bacteriana. Con redes neuronales convolucionales.
2. Clasificación de patrones de tuberculosis, considerando 6 diferentes patrones pulmonares. Usando redes neuronales convolucionales.
3. Clasificación de patrones de tuberculosis, considerando 6 diferentes patrones. Usando redes neuronales convolucionales, con transferencia de aprendizaje.
4. Clasificación de patrones de tuberculosis, considerando 6 diferentes patrones. Usando imágenes segmentadas y una red neuronal *backpropagation*.

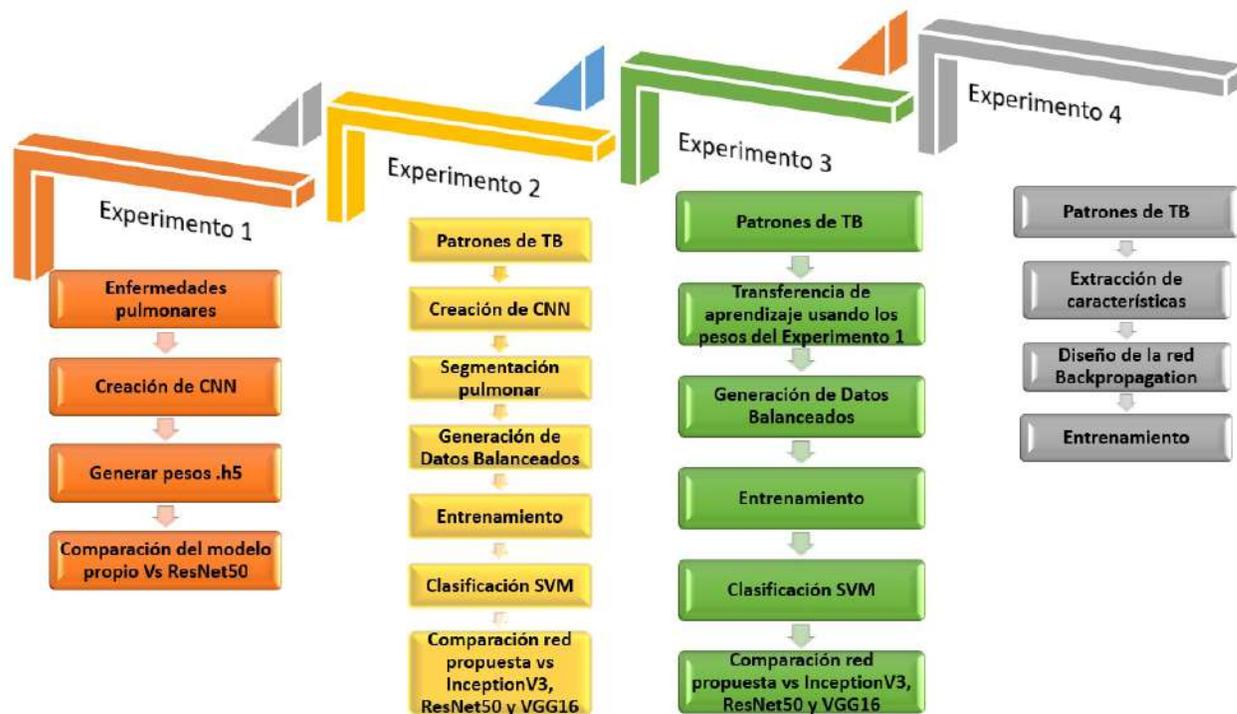


Figura 4.1: Metodología

La metodología a seguir para cada uno de los experimentos se aprecia en la siguiente Figura [4.2](#). Primero se busco la base de datos de imágenes y sus etiquetas, después se normalizan las imágenes a las dimensiones de  $256 \times 256$  para el método propuesto,  $229 \times 229$  para la red *InceptionV3*,  $224 \times 224$  *VGG16* y *ResNet50*.

Se realizo la sobre ejemplificación de las imágenes para el balanceo, esto significa que a las imágenes se le puede realizar rotaciones, o realizar cambios de color para poder generar una nueva imagen y que cada carpeta contenga la misma cantidad de imágenes a lo cual se le conoce como balanceo de imágenes.

División de las imágenes para entrenamiento mediante validación cruzada. Las imágenes son divididas en 3 carpetas para entrenamiento, validación y test.



Figura 4.2: Metodología para el desarrollo de la CNN

### Metodología para el diseño de la CNN

Se tiene que configurar el número de capas que tendrá la red, cuantas convoluciones se realizarán por capa, el tamaño del *kernel* que se utilizará, el número de capas de agrupación y el número de *Dropout* que se usara por capa. De igual manera se configura el optimizador que se desea junto con su tasa de aprendizaje y por último se configura las funciones de activación. Cuando la red está entrenando se configura cuantas épocas va a entrenar y los pasos que va a realizar durante el entrenamiento. Posteriormente se analiza la red con diferentes parámetros como número de convoluciones. La metodología para el diseño de red neuronal convolucional se muestra en la siguiente Figura 4.3:



Figura 4.3: Metodología para el diseño de la CNN

Considerando lo anterior se describirá la metodología de cada experimento.

## 4.1. Experimento 1

### Clasificación de enfermedades pulmonares

En el primer experimento se trabajó en la clasificación de enfermedades pulmonares que son recurrentes en la población y que pueden detectarse a partir de radiografías de rayos X, las enfermedades que se analizaron son: COVID-19 que amenaza la vida y la salud, también la TB que sigue siendo un problema de salud mundial, la neumonía viral y bacteriana que presentan síntomas iniciales parecidos al COVID-19 y TB.

Para la clasificación se propuso y diseño una red (Mpro), para valorar el desempeño de esta, se entreno la red *ResNet50* con el mismo conjunto de datos.

En la Tabla [4.1](#), muestra la configuración de la red propuesta, considerando las capas, dimensión de imagen, número de convoluciones, tamaño *kernel* y parámetros resultantes. Así como el arreglo de las funciones *Dropout*, *Maxpooling*, *Fully connected* y *Softmax* y se realizó el aplanado *flatten*. Para la construcción de la red propuesta se trabajó en conjunto con búsqueda en maya [\[62\]](#) para determinar cuáles son los valores óptimos de la red. Se utilizo validación cruzada para evaluar el desempeño del modelo creado.

Capas	Salida	Kernel	Parametros
imagen	(256, 256, 64)	3	1792
1	(256,256,55)	5	88055
1	(256,256,55)	5	44032
Dropout	(256,256,32)		0
Max pooling	(128,128,32)		0
2	(128,128,32)	5	25635
2	(128,128,32)	3	9248
2	(128,128,32)	3	9248
Dropout	(128,128,32)		0
Max pooling	(64,64,32)	3	0
3	(64,64,32)	5	25632
3	(62,62,32)	3	9248
Dropout	(62,62,32)		0
Max pooling	(31,31,32)	3	0
4	(31,31,32)	5	25632
4	(29,29,64)	3	18496
Dropout	(29,29,64)		0
Max pooling	(14,14,64)	3	0
Dropout	(14,14,64)		0
Flatten	(12544)	3	0
Fully Connected	(100)		1254500
Dropout	(100)	3	0
Fully Connected	(190)		19190
Dropout	(190)	0	0
Softmax		5	955

Tabla 4.1: Arquitectura diseñada MPro1

### 4.1.1. Base de datos

Las radiografías de rayos X del tórax fueron obtenidas de varias bases de datos que a continuación se listan:

- El primero *Montgomery*, este conjunto contiene 138 radiografías.
- El segundo fue recopilado del *hospital de Shenzhen China*, cuenta con 326 radiografías saludables y 336 radiografías con tuberculosis;
- El tercer conjunto de la categoría de tuberculosis se obtuvo de *Kaggle* el cual cuenta con 700 radiografías [63].
- De igual manera se obtuvieron cuatro conjuntos de imágenes de la página de *Kaggle* correspondientes a COVID-19 [64], Neumonía viral y neumonía bacteriana e imágenes sanas [65]. Las imágenes de tuberculosis que fueron extraídas de *Kaggle* se depuraron; se eliminaron las imágenes repetidas ya que en ese conjunto de datos se tenían imágenes pre procesadas.

Para valorar el desempeño de la red propuesta y considerando que en el estado del arte se encontró la investigación [18] de CNN con la arquitectura ResNet50 donde clasifican enfermedades pulmonares como Covid-19, tuberculosis y neumonía reportando una precisión en promedio de 91.52 %. Con este antecedente, se consideró correr el mismo experimento con el mismo conjunto de datos con la red ResNet50 y con nuestra propuesta.

### 4.1.2. Entrenamiento

Las imágenes se almacenaron en 5 carpetas de acuerdo a cada clase (tuberculosis, COVID-19, neumonía viral, neumonía bacteriana y sanas).

Para el balanceo de las imágenes, se consideró el número de imágenes de COVID-19. En cada categoría se colocaron 1870 imágenes en total. Considerando que las imágenes tenían diferentes resoluciones, se realizó la reducción de las imágenes a una escala de 256 x 256 para normalizar los datos.

Los conjuntos de imágenes se dividieron en 3 carpetas; la primera de entrenamiento con 1456 imágenes; en la segunda se usaron 364 para validación y en la tercera 50 para las pruebas (la selección de imágenes se efectuó al azar).

Se etiquetaron 5 clases; considerando la selección de cada una de ellas al azar, en cada clase (0 = Normal; 1=Tuberculosis; 2= COVID-19; 3 =Neumonía viral; 4 = Neumonía bacteriana) se muestra en la Figura 4.4.

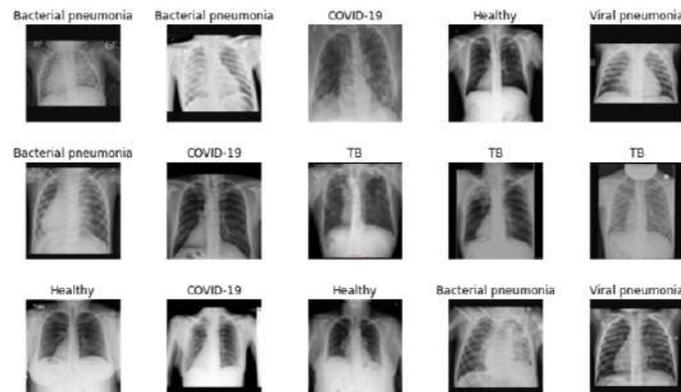


Figura 4.4: Selección y etiquetado de las radiografías por clase.

Fuente: Elaboración propia

El entrenamiento se realizó en una laptop modelo *Inspiron 15 500 series*, con procesador de *CORE i7, 7th. Generación*, con memoria RAM de 6 GB y disco duro de 1 TB.

La red se programó con *Python*, con las librerías de *machine learning: tensorflow, Sklearn, numpy y pandas*.

## 4.2. Experimento 2

### 4.2.1. Base de datos experimento 2

El conjunto de imágenes para este experimento fue proporcionado por el Instituto Nacional de Enfermedades Infecciosas <https://depot.tbportals.niaid.nih.gov/>, es de destacar que en este instituto el diagnóstico de la TB fue corroborado con la

prueba de esputo y con radiografías de rayos X, por lo que, se tiene la certeza del diagnóstico.

A partir del diagnóstico de las imágenes se logró determinar que un paciente con TB pueden presentar hasta siete tipos de patrones de la enfermedad al mismo tiempo. Esto es de suma importancia ya que en el estado del arte no se encontró ningún trabajo con este enfoque.

El conjunto de datos que se obtuvieron fue de 4001 imágenes en formato *DICOM* y *JPG*. Las imágenes que se tomaron en cuenta son las radiografías que están avaladas por la prueba de esputo, como se ha mencionado es considerada la mejor prueba, dando un total de 1982 imágenes.

Durante el análisis de las radiografías se observó que en los metadatos estaba registrado más de un patrón de TB por radiografía, en las radiografías no estaban marcados los patrones, sin embargo, si estaban registradas. Por ello que se realizó la clasificación de acuerdo de los patrones existentes y de las combinaciones de ellos. En la Tabla 4.2 se detalla las combinaciones encontradas en el conjunto de datos.

Número de patrones	Combinación de patrones	Imágenes
2 patrones	Fibrosis y opacidad	72
2 patrones	Consolidación y opacidad	122
3 Patrones	Fibrosis, opacidad y nódulos	220
3 Patrones	Consolidación, opacidad y nódulos	190
4 Patrones	Consolidación, fibrosis, opacidad y nódulos	491
5 Patrones	Consolidación, fibrosis, opacidad, nódulos y cavitaciones	480
5 Patrones	Consolidación, fibrosis, opacidad, pleural, nódulos	117
6 Patrones	Consolidación, fibrosis, opacidad, pleural, nódulos y cavitaciones	154
6 Patrones	Consolidación, fibrosis, opacidad, pleural, nódulos y linopatas	136

Tabla 4.2: Clasificación de patrones

A partir de esas imágenes solo se tomaron en cuentas los patrones que tenían más imágenes, en la Tabla 4.3 se puede apreciar la combinación de patrones que se

Número de patrones	Combinación de patrones	Imágenes
Sin patrones	Saludables	500
2 patrones	Consolidación y opacidad	122
3 Patrones	Fibrosis, opacidad y nódulos	220
4 Patrones	Consolidación, fibrosis, opacidad y nódulos	491
5 Patrones	Consolidación, fibrosis, opacidad, nódulos y cavitaciones	480
6 Patrones	Consolidación, fibrosis, opacidad, pleural, nódulos y cavitaciones	154

Tabla 4.3: Combinación de patrones usadas

tomaron en cuenta.

Se analizaron las imágenes de la base de datos y como se puede observar en la Figura 4.5, las radiografías tienen partes cortadas, en esas partes, se encontraban datos del paciente por lo que se eliminaron, por privacidad y confidencialidad que tienen los pacientes. Al eliminar esos datos, se creó ruido.

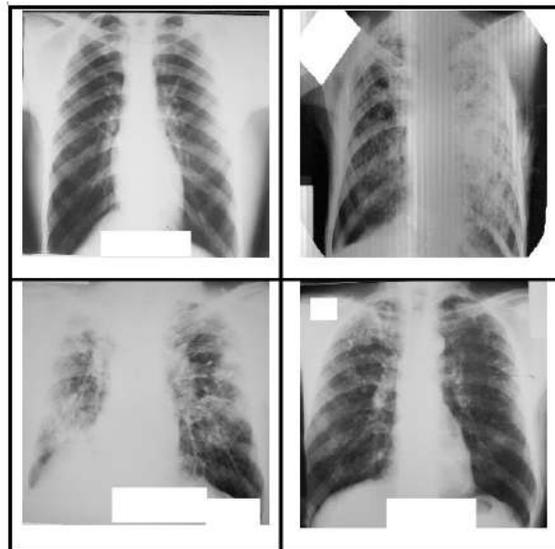


Figura 4.5: Imágenes con ruido.

Fuente: Elaboración propia

Para eliminar ese tipo de ruido se plantea hacer la segmentación de los pulmones.

En la Figura 4.6 se muestra un ejemplo de segmentación de pulmones a partir del método otsu [66], como se puede observar en la figura a y b si realiza la segmentación de manera correcta debido a que se muestra en la radiografía original los dos pulmones. Sin embargo, en la figura c y d al haber una opacidad en el pulmón y no verse el pulmón, la segmentación no toma en cuenta el pulmón izquierdo, pero en realidad si está el pulmón, simplemente hay un patrón radiológico presente que oculta el pulmón.

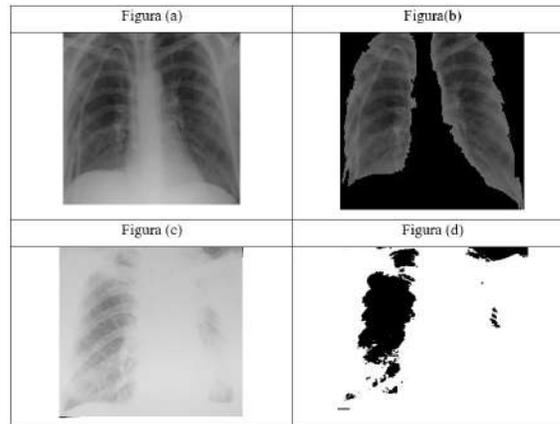


Figura 4.6: Segmentación pulmonar.

Fuente: Elaboración propia

Para resolver ese problema se plantea realizar la segmentación manual como se muestra en la Figura 4.7, para tomar en cuenta ambos pulmones y se haga la identificación pertinente para la clasificación.

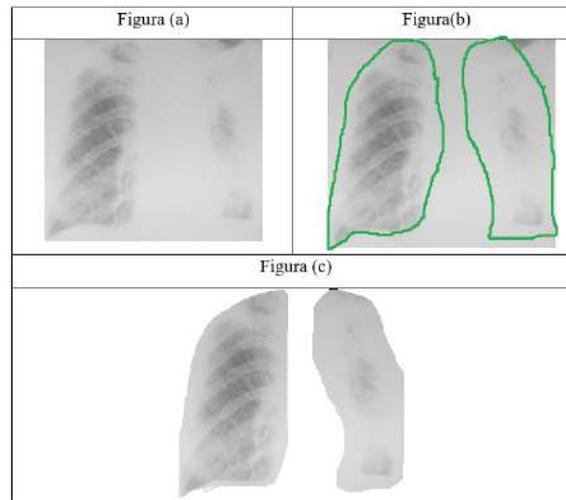


Figura 4.7: Segmentación pulmonar a mano.

Fuente: Elaboración propia

Las imágenes se almacenaron en 6 carpetas de acuerdo a cada clase (Saludable, 2 patrones, 3 patrones, 4 patrones, 5 patrones y 6 patrones).

### **Balanceo de imágenes**

Para el balanceo de imágenes, se usó la función *Data Augmentation*. Todas las clases se ajustaron a 450 imágenes por clase. El aumento de datos de imágenes es una técnica que se utiliza para expandir artificialmente el tamaño de un conjunto de datos de entrenamiento, la cual crea variaciones de las imágenes para mejorar la capacidad de los modelos.

Las funciones que se generan con el balanceo son: Desplazamiento horizontal y vertical, rotación aleatoria, zoom aleatorio como se muestra en la Figura [4.8](#).

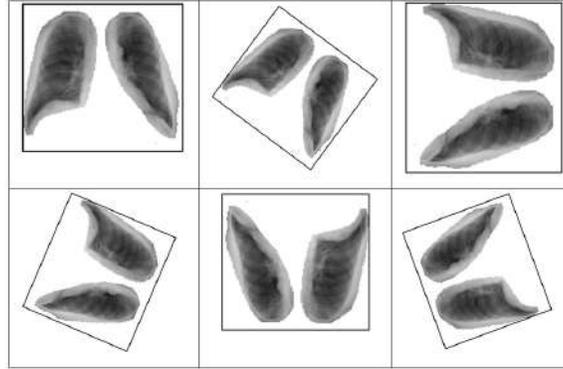


Figura 4.8: Aumento de imágenes.

### Configuración y entrenamiento de la CNN propuesta

El entrenamiento se realizó en una PC, con procesador de *COREi9, 10th*. Generación, con memoria RAM de 16 GB y disco duro solido de 240 GB.

La red se programó con Python, con las librerías de machine learning: tensorflow, Sklearn, numpy y pandas.

Para etiquetar las imágenes se implementó la función *flowfromdirectory* que etiqueto las 6 clases; considerando la selección de cada una de ellas al azar en cada clase. Como se muestra en la Figura [4.9](#). Este proceso es de gran interés ya que hace que el entrenamiento sea supervisado, a la red se le proporcionan imágenes etiqueta con la clase a la cual pertenece.

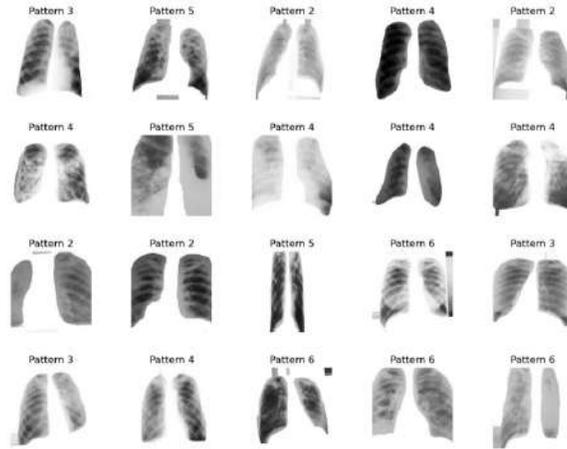


Figura 4.9: Clasificación de las radiografías.

En la Tabla [4.4](#), se muestra la configuración de la red propuesta Mpro2, las imágenes son de 256 x 256 y la primer convolucion es de 80, con un *kernel* de 3 x 3 en total se entrenaron 2,240 parámetros. La segunda convolucion es de 70, y se tomo un *kernel* de 5 x 5 entrenando 50,470 parámetros, la tercer convolucion fue de 60, la cuarta de 50, la quinta de 28 con *kernel* de 3 x 3 hasta llegar al primer *dropout* que se va a encargar de prender y apagar neuronas en este caso se dio un *Dropout* de 30 y posteriormente un *Maxpooling* que tomara los mejores valores y esos pasara a la siguiente capa de convoluciones.

Capas	Salida	Kernel	Parametros
imagen	(256, 256, 80)	3	2240
1	(256,256,70)	5,	50470
1	(256,256,60)	3	37860
1	(254,254,50)	3	27050
1	(254,254,28)	3	12628
Dropout	(254,254,32)		0
Max pooling	(127,127,28)		0
2	(127,127,64)	5	16192
2	(127,127,32)	3	18464
2	(127,127,32)	3	9248
Dropout	(127,127,32)		0
Max pooling	(63,63,32)	3	0
3	(63,63,64)	5	18496
3	(63,63,32)	3	18464
3	(63,63,32)	3, 5	9248
Dropout	(63,63,32)		0
Max pooling	(31,31,32)		0
Dropout	(31,31,32)		0
Flatten	(30752)	3	0
Fully Connected	(5000)		153765000
Dropout	(5000)	0	0
Softmax		6	1005

Tabla 4.4: Arquitectura diseñada Mpro2

Para llegar a esta red propuesta se realizaron mas de 21 entrenamientos a continuación en la Tabla 4.5 y la gráfica 4.10 se muestran 7 entrenamientos con diferencias significativas. Dependiendo de la exactitud se iban modificando los parámetros de la red. En este caso la arquitectura final fue la numero 7 ya que modificando los parámetros de esa red ya no se encontraban mejoras si no todo lo contrario.

Parámetros	1	2	3	4	5	6	7
Convoluciones	12	15	11	16	11	11	10
Capas	3	5	3	6	3	3	3
Capa densa	5,000	7,500	5,000	1,200	5,000	5,000	12,000
Regular del kernel	.0001	.0001	.001	.0001	.0001	.0001	.0001
Optimizador	.003	.01	.001	.0005	.001	.0001	.0003
Pasos por época	5	16	8	6	5	6	6
Épocas	80	10	50	50	60	60	20
Pasos de validación	2	8	4	3	2	3	3
Exactitud	.17	.24	.33	.17	.24	.37	.47

Tabla 4.5: Cambios de los parámetros para entrenamiento

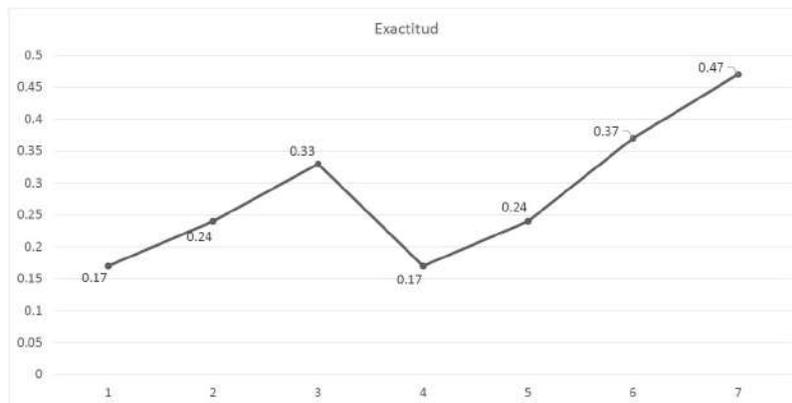


Figura 4.10: Gráfica de la exactitud de los entrenamientos .

Fuente: Elaboración propia

Para evitar el sesgo en la clasificación, se normalizaron los datos. Esto consiste en dividir el valor de los píxeles en 255, para que se tengan valores entre 0 y 1. Las imágenes que se usaron para los entrenamientos de las arquitecturas pre entrenadas se modificaron las resoluciones de acuerdo a las entradas de cada arquitectura.

- ResNet50, entrena con resolución de 224 x 224
- InceptionV3 entrena con resolución de 229 x 229
- VGG16 entrena con resolución de 224 x 224

El conjunto de imágenes para el entrenamiento, validación y prueba, se dividieron en 3 carpetas; la primera de entrenamiento con el 70 % de imágenes; en la segunda de validación con el 20 % y en la tercera de prueba se usó el 10 % (la selección de imágenes se efectuó al azar).

### 4.3. Experimento 3

#### 4.3.1. Transferencia de aprendizaje (TF)

En este experimento se utilizó la transferencia de aprendizaje, que consiste en obtener los pesos de un modelo pre entrenado, que utilizará radiografías del tórax, con la finalidad de ayudar a la red CNN con los pesos previamente entrenados a optimizar tiempos y poder mejorar los resultados, a continuación se muestra un ejemplo en la Figura 4.11.

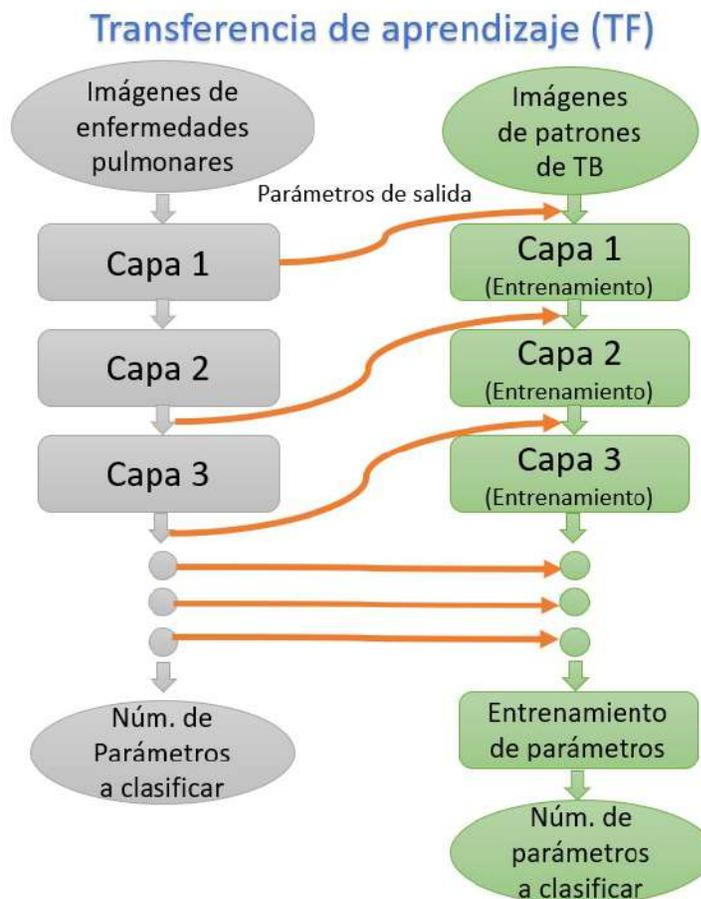


Figura 4.11: Transferencia de aprendizaje

Fuente: Elaboración propia

Para ello se buscaron modelos pre entrenados con diferentes arquitecturas, como la VGG16, ResNet50, InceptionV3. Se encontró el modelo con los pesos de la archi-

itectura VGG16, pero no se contó con los pesos de las demás arquitecturas por lo que se optó por realizar el entrenamiento con radiografías del tórax del experimento 1 y entrenarlas con las 3 arquitecturas antes mencionadas para así obtener los pesos de cada una de ellas y poder hacer la transferencia de aprendizaje.

Los resultados con las arquitecturas: *Método propuesto (MPro1)*, *InceptionV3*, *VGG16* y *ResNet50* para la *Transferencia de aprendizaje (TF)* se muestran en la tabla 4.6. De este entrenamiento los que nos interesó fue obtener los pesos de cada arquitectura.

Arquitecturas	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
<i>Mpro</i>	0.87	0.89	0.87	0.87
<i>InceptionV3</i>	0.90	0.89	0.90	0.90
<i>VGG16</i>	0.88	0.86	0.86	0.87
<i>ResNet50</i>	0.90	0.89	0.91	0.89

Tabla 4.6: Resultados del experimento 1 con *TF*

## 4.4. Experimento 4

### 4.4.1. Entrenamiento *Backpropagation*

En este experimento se trabajó con la red *Backpropagation*, se realizó la extracción de características para poder realizar los entrenamientos, se utilizó el conjunto de datos segmentado. Las características que se usaron para el análisis de da las radiografías son:

- Altura y ancho
- Área
- Perímetro

- Histogramas
- Detección de bordes
- Rectangularidad
- Inercia
- Solides
- Compacidad
- Centroide
- Momentos invariables de HU
- Momentos normalizados.

Las características fueron guardadas en un archivo en Excel para su posterior análisis. En esta arquitectura al momento de programar las etiquetas a la que correspondía las clases se modificó de la siguiente manera:

- La etiqueta 0 corresponde a la clase de 2 patrones,
- La etiqueta 1 corresponde a la clase de 3 patrones
- La etiqueta 2 corresponde a la clase de 4 patrones
- La etiqueta 3 corresponde a la clase de 5 patrones
- La etiqueta 4 corresponde a la clase de 6 patrones
- La etiqueta 5 corresponde a la clase de las radiografías saludables.

Se diseñó una nueva arquitectura de red Backpropagation, la Tabla [4.7](#). La cual consta de 15 capas secuenciales y de ella se van desglosando diferente número de neuronas en este caso comenzamos con 128 neuronas por lo que obtuvimos 231808 parámetros para entrenamiento, y la función de activación que usamos fue lineal, posteriormente añadimos dos capas con 125 neuronas cada una y de ella los parámetros entrenados

Capas	Salida	Parámetros
1	(128)	231808
1	(125)	16125
1	(125)	15750
1	(128)	16128
1	(125)	16125
2	(125)	15750
2	(128)	16128
2	(125)	16125
Dropout	(64)	8064
Max pooling	(128)	8320
3	(64)	8250
3	(128)	8320
3	(128)	16512
Dropout	(64)	8256
Max pooling	(128)	8320
Softmax	6	774

Tabla 4.7: Arquitectura Backpropagation

fueron de 16125 y 15750 y así sucesivamente hasta llegar a la última capa de clasificación en la cual declaramos que la salida solo tenemos 6 clases y la función de activación fue *Softmax*.

# 5 Resultados de los experimentos

En este apartado se presentan los resultados y la discusión de cada uno de los experimentos por separado.

## 5.1. Resultados y discusiones del experimento 1

Se clasificaron cuatro enfermedades pulmonares (TB, Covid-19, Neumonía viral y bacteriana). Se utilizó la carpeta de test la cual cuenta con una muestra de 50 imágenes por clase, cabe aclarar que estas no se le había mostrado a la red. Para medir el rendimiento del modelo de prueba, se utilizó la matriz de confusión de la cual se obtuvieron los parámetros: TP=Verdadero positivo; TN=Verdadero negativo; FP=Falso positivo; FN=Falso negativo. En la Figura 5.1 se presentan los resultados de la matriz de confusión (a) Mpro1 (b) la ResNet50.

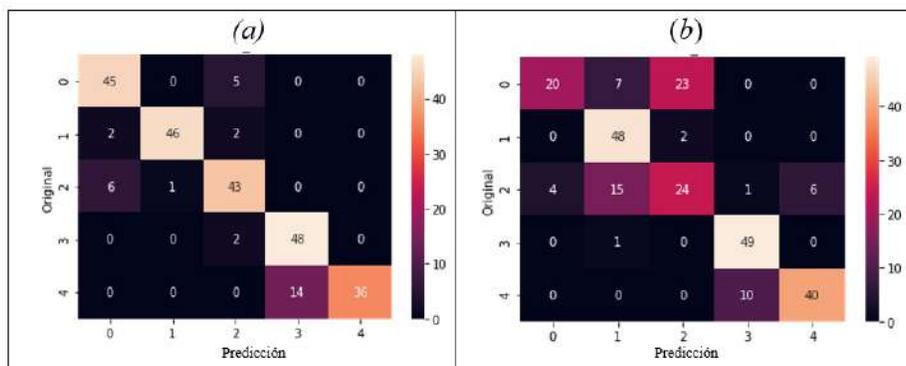


Figura 5.1: Segmentación pulmonar.

En las Tablas 5.1 y 5.2 se pueden observar los resultados obtenidos reflejados de la métrica de precisión, sensibilidad y puntuación F1 del *MPro* y la *ResNet50*

Clases	Precisión	Sensibilidad	Puntuación F1
0= Normal	0.85	0.90	0.87
1= Tuberculosis	0.98	0.92	0.95
2= Covid-19	0.83	0.86	0.84
3= Neumonía viral	0.77	0.96	0.86
4= Neumonía bacteriana	1.00	0.72	0.84
Exactitud			0.87
Macropromedio	0.89	0.87	0.87

Tabla 5.1: Evaluación del Mpro del experimento 1.

Clases	Precisión	Sensibilidad	Puntuación F1
0= Normal	0.83	0.40	0.54
1= Tuberculosis	0.68	0.96	0.79
2= Covid-19	0.49	0.48	0.48
3= Neumonía viral	0.82	0.98	0.89
4= Neumonía bacteriana	0.87	0.80	0.83
Exactitud			0.72
Macropromedio	0.74	0.72	0.71

Tabla 5.2: Evaluación del modelo ResNet50 del experimento 1.

Se puede observar que la exactitud que tiene la *ResNet50* es de 0.72, siendo mejor el Mpro1 con 0.87. La precisión promedio en *ResNet50* fue de 0.74, en contraste con el Mpro1 fue de 0.89. Con el Mpro1 no se usaron imágenes repetidas, pero los autores de la investigación [18] que tuvieron un 91.52% de promedio de precisión, mencionan que si repitieron imágenes para balancear los conjuntos de datos (sabemos que esto puede ocasionar ruido y sesgo en los resultados).

En la investigación [17] se trabajó con neumonía y tuberculosis, con radiografías de tórax (Rayos X) usando como clasificadores una CNN personalizada, VGG-16,

Inception-V3, Inception, ResNet-V2, Xception y DenseNet-211, obteniendo precisiones de hasta 0.94.

También se encontró con trabajo [19], para detectar tuberculosis usaron redes neuronales profundas para obtener las características y sustituyeron la capa de clasificación de la red neuronal por una máquina de soporte vectorial, con ello mejoraron los resultados obtenidos. De este experimento se pueden deducir las siguientes experiencias; se deben usar mas arquitecturas de redes profundas para experimentos subsecuentes, así mismo, se concluye que clasificar con maquinas de soporte vectorial se pueden mejorar los resultados.

También, derivado del entrenamiento de este experimento con redes profundas, se pueden obtener características relevantes especiales de las imágenes de tórax, que se usaran para la transferencia de aprendizaje, destacando que esto es esencial, ya que se utilizaron puras radiografías de tórax.

## 5.2. Resultados y discusiones experimento 2

Derivado de la experiencia adquirida del experimento uno, en este apartado se retomo la experiencia de usar otras arquitecturas de redes profundas y el uso de *MSV*.

Existen arquitecturas que demuestran su alto poder de clasificación, suelen utilizar la base de datos *CIFAR10* o *CIFAR100* donde se pone a prueba el tiempo de entrenamiento y precisión.

En este experimento lo que se desea clasificar no son objetos si no radiografías por lo que se propuso una CNN propia(MPro2). Para demostrar si el MPro2 es óptimo para la clasificación, se comparó contra las arquitecturas ResNet50, VGG16 y InceptionV3 que en el estado del arte demuestran su alto poder de clasificación. En la Figura 5.2 se presentan las gráficas de entrenamiento y de la validación con respecto a las épocas de entrenamiento, las gráficas corresponden: (a) *MPro* y (b) *InceptionV3*, (c) *VGG16* y (d) *ResNet50*. En ellas se puede observar el desempeño durante las épocas y también se puede observar si el rendimiento es el adecuado, ya que podemos llegar a tener *overfitting* y *underfitting*.

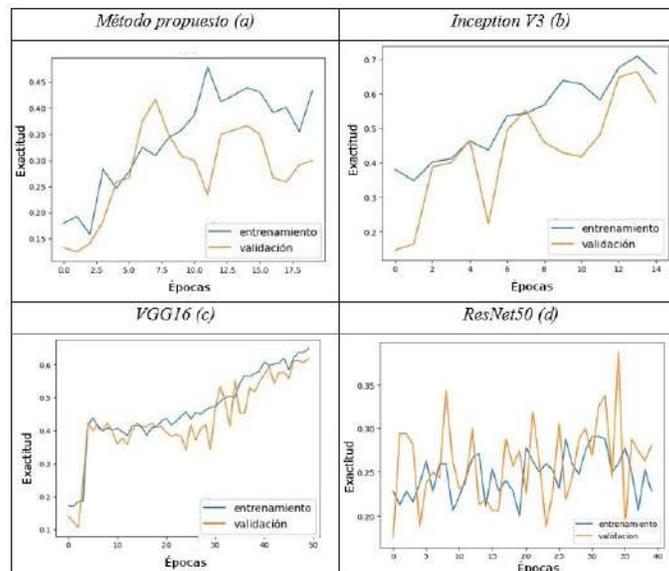


Figura 5.2: Comparación de los entrenamientos.

Posteriormente se otorgó una muestra de imágenes para la obtención de test para medir su rendimiento utilizando la matriz de confusión, obteniendo los parámetros: TP=Verdadero positivo; TN=Verdadero negativo; FP=Falso positivo; FN=Falso negativo. En la Figura 5.3 se presentan los resultados de la matriz de confusión (a) nuestro método propuesto (MPro) y (b) InceptionV3, (c) VGG16 y (d) ResNet50.

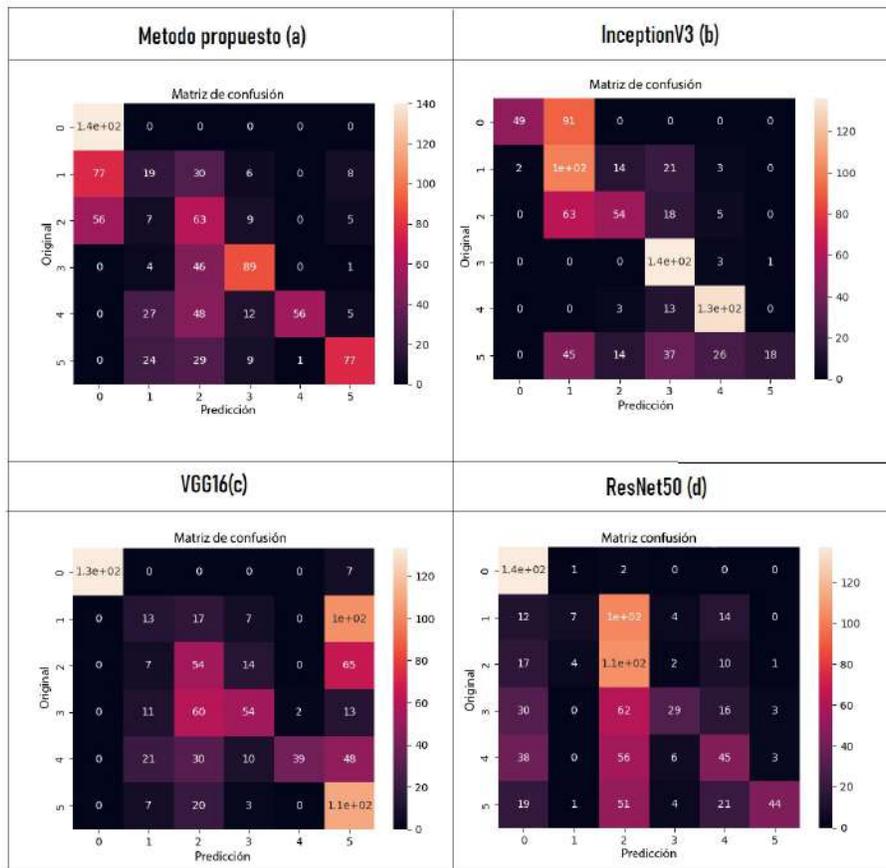


Figura 5.3: Comparación de la Matriz de confusión de cada Arquitectura entrenada del experimento 2.

A continuación, se muestran los resultados que se obtuvieron de las diferentes métricas de la evaluación. Primero se muestra los resultados del MPro2, como se puede apreciar en la Tabla 5.3, se tuvo un precisión de 0.47, el Macropromedio de la precisión fue de 0.42, sensibilidad de 0.47 y puntuación F1 0.39. Se puede observar que el método propuesto no tiene un buen desempeño.

Clases	precisión	sensibilidad	puntuación F1
0	0.99	1.0	0.99
1	0.17	0.01	0.01
2	0.00	0.00	0.00
3	0.34	0.95	0.50
4	0.70	0.30	0.42
5	0.34	0.58	0.43
Exactitud			0.47
Macropromedio	0.42	0.47	0.39

Tabla 5.3: Resultado del MPro2.

Los resultados de la red InceptionV3 se muestran en la Tabla 5.4, en el cual se tuvo una precisión de 0.50, un Macropromedio de la precisión 0.55, sensibilidad 0.50 y puntuación F1 de 0.49. Tomando en cuenta la exactitud del MPro2 con 0.47 e InceptionV3 con 0.50, no varían mucho los resultados.

Clases	Precisión	Especificidad	Puntuación F1
0	0.98	1.0	0.99
1	0.06	0.01	0.01
2	0.24	0.55	0.34
3	0.82	0.39	0.53
4	0.35	0.51	0.41
5	0.88	0.54	0.67
Exactitud			0.50
Macropromedio	0.55	0.50	0.49

Tabla 5.4: Resultado InceptionV3.

Los resultados del entrenamiento de la red VGG16 se muestran en la Tabla 5.5, el cual obtuvo un exactitud de 0.50 y un Macropromedio en precisión de 0.51, sensibilidad de 0.50 y puntuación F1 de 0.45. Los resultados fueron similares a la arquitectura InceptionV3.

Clases	precisión	sensibilidad	puntuación F1
0	1.00	1.0	1.00
1	0.72	0.39	0.51
2	0.94	0.56	0.70
3	0.04	0.01	0.01
4	0.30	1.00	0.47
5	0.05	0.001	0.02
Exactitud			0.50
Macropromedio	0.51	0.50	0.45

Tabla 5.5: Resultado Vgg16.

Los resultados de la red ResNet50 se muestran en la Tabla 5.6, el cual obtuvo una exactitud de 0.38 y un Macropromedio en precisión de 0.36, sensibilidad de 0.31 y puntuación F1 de 0.26. Esta red fue la tuvo un bajo desempeño comparado con las demás arquitecturas.

Clases	Precisión	Sensibilidad	Puntuación F1
0	1.00	0.90	0.95
1	0.03	0.05	0.04
2	0.08	0.61	0.15
3	0.33	0.23	0.27
4	0.33	0.01	0.03
5	0.40	0.06	0.11
Exactitud			0.38
Macropromedio	0.36	0.31	0.26

Tabla 5.6: Resultados ResNet50 experimento 3.

En la Tabla [5.7](#) se muestra la comparación de los resultados, la red *InceptionV3* y *VGG16* que obtuvieron una exactitud de 0.50, mostrando ser las mejores clasificando, posteriormente le siguieron la arquitectura propuesta con 0.47 y al final la red ResNet50 con 0.38

Arquitecturas	Exactitud	Precisión	Sensibilidad	Puntuación F1
MPro	0.47	0.42	0.47	0.39
InceptionV3	0.50	0.55	0.50	0.49
Vgg16	0.50	0.51	0.50	0.45
ResNet50	0.38	0.36	0.31	0.26

Tabla 5.7: Resultados con las 4 arquitecturas con imágenes segmentadas

### 5.2.1. SVM-Métrica ROC

Para poder mejorar la clasificación, se colocó una máquina de soporte vectorial usando diferentes tipos de *Kernel* para saber cuál se ajustaba mejor, los *Kernel* que se

usaron son: *Kernel lineal*, *Gaussiano*, *RBF*, *Sigmoide*, *Polinomial*. En la Figura 5.4 se muestran los resultados con los diferentes tipos del *Kernel* con el método propuesto.

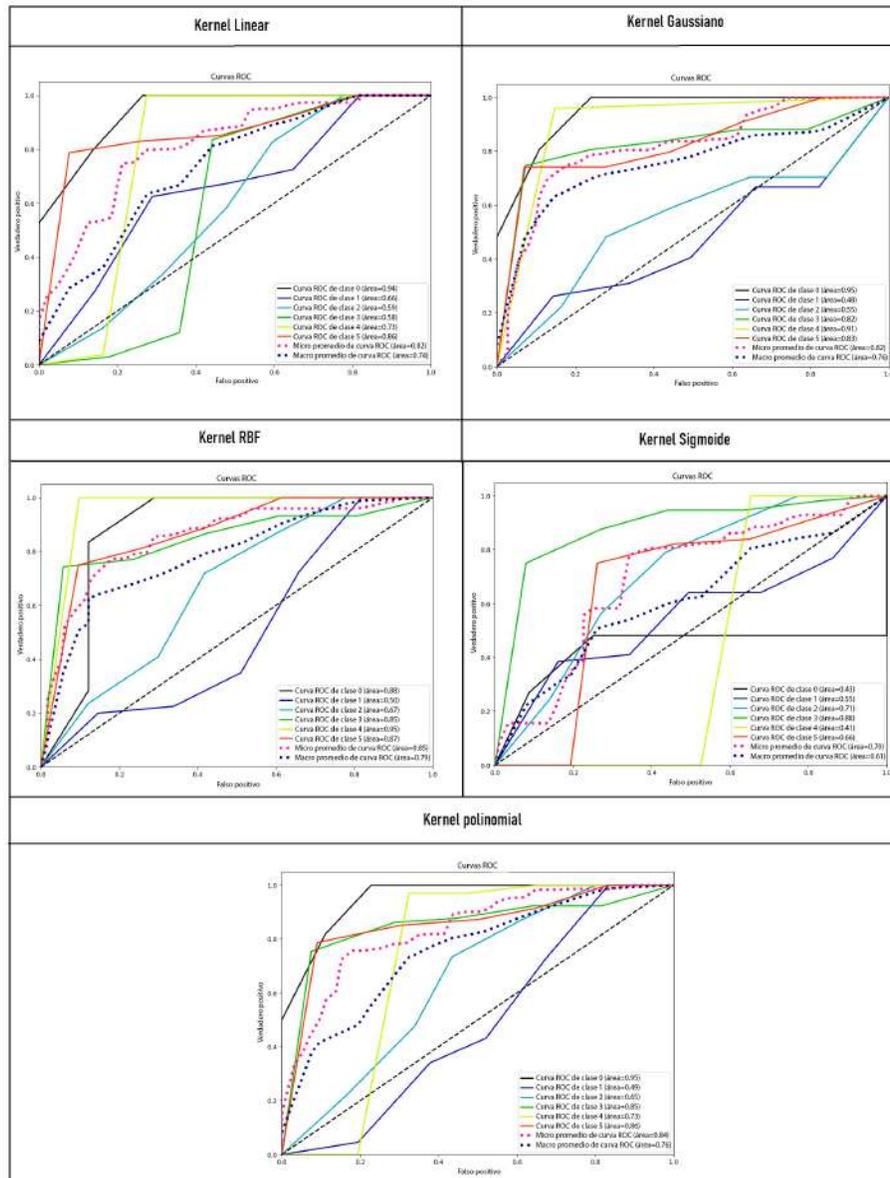


Figura 5.4: Curvas ROC del método propuesto.

En la Figura 5.5 se muestran los resultados de las curvas ROC de la red InceptionV3.

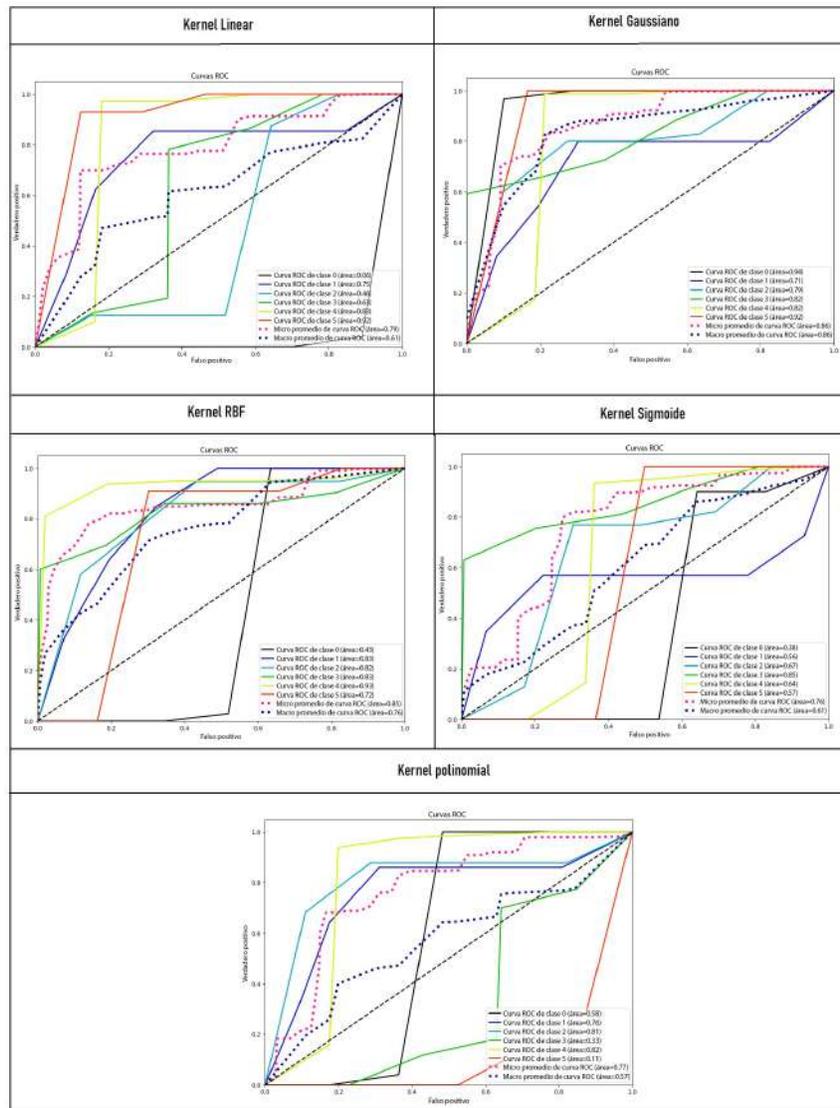


Figura 5.5: Curvas ROC InceptionV3.

En la Figura 5.6 se muestran los resultados de las curvas ROC de la red VGG16.

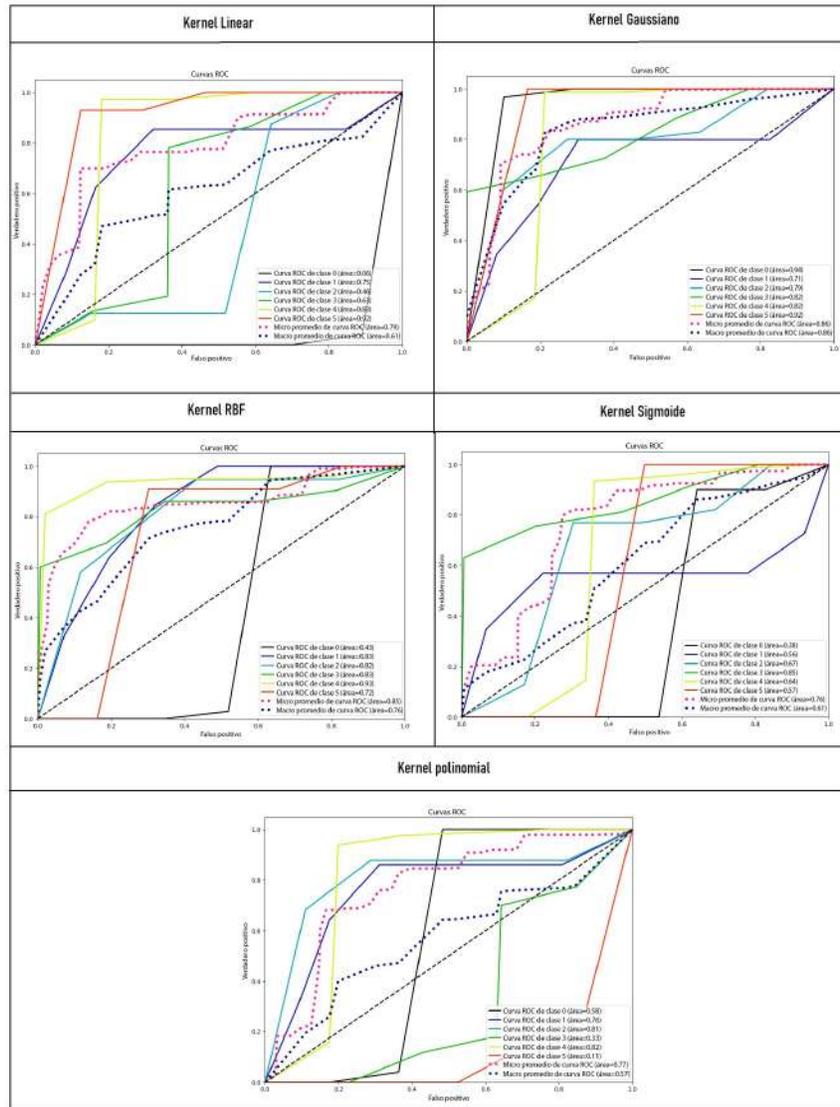


Figura 5.6: Curvas ROC VGG16.

En la Figura 5.7 se muestran los resultados de las curvas ROC de la red ResNet50.

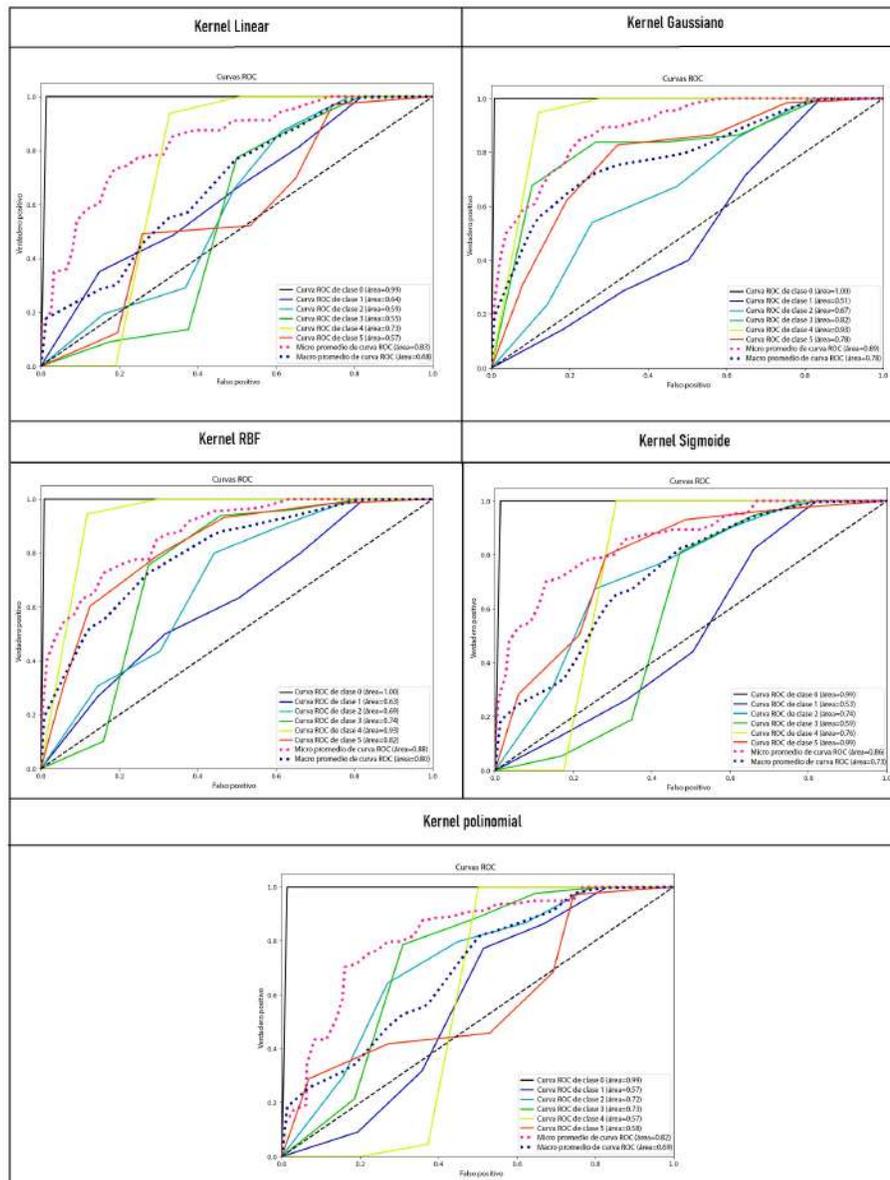


Figura 5.7: Curvas ROC ResNet50.

La comparación de los resultados de los diferentes Kernel utilizados se muestra en la Figura 5.8, donde se puede apreciar los resultados de las redes entrenadas con los perspectivas kernel utilizados.

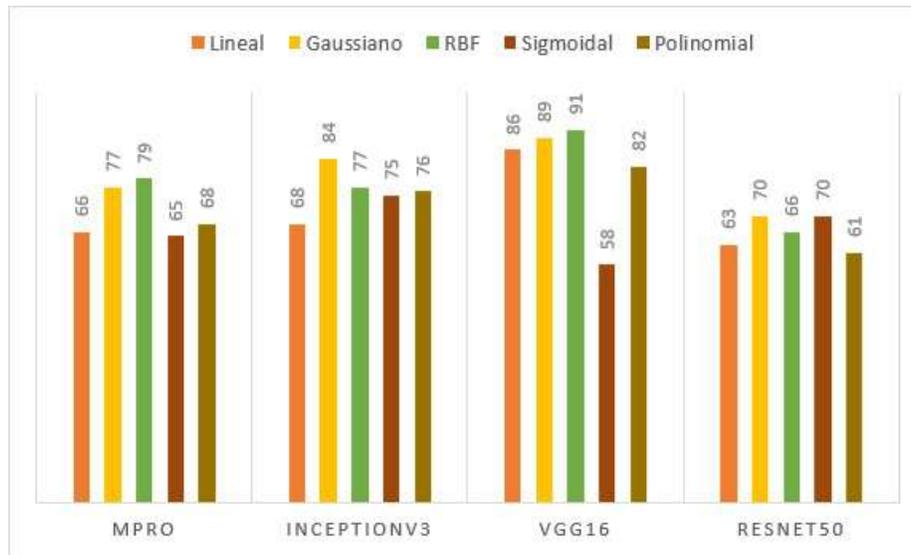


Figura 5.8: Comparación de resultados de curva ROC, experimento 2.

En la Tabla 5.8 se muestran los mejores resultados de las curvas ROC de cada arquitectura. Y como se puede apreciar la que mostró mejor resultado es VGG16 con 0.91, con el *kernel RBF*, la segunda mejor red es *InceptionV3* con 0.84 y el *kernel Gaussiano*, la tercera fue MPro con el *kernel RBF* 0.79 y la cuarta fue *ResNet50* con *kernel sigmoidal* 0.70.

Arquitecturas	Mejor Kernel	AUC
MPro	<i>RBF</i>	0.79
InceptionV3	<i>Gaussiano</i>	0.84
VGG16	<i>RBF</i>	<b>0.91</b>
ResNet50	<i>Sigmoidal</i>	0.70

Tabla 5.8: Mejores resultados de las curvas ROC Experimento 2

### 5.3. Resultados y discusiones experimento 3

En consecuencia, de los resultados del experimento 2 y con la perspectiva de mejorar los resultados, se decidió usar transferencia de aprendizaje. Usando los pesos derivados del primer experimento, donde se uso una red neuronal convolucional con imágenes del tórax de cuatro enfermedades pulmonares; con ello se obtuvieron características relevantes de 1870 radiografías de tórax en las primeras capas de entrenamiento de la red convolucional.

#### 5.3.1. Transferencia de aprendizaje

Los resultados obtenidos se muestran a continuación en la Figura 5.9, en ellas se puede observar su desempeño durante las épocas y también podemos observar su rendimiento.

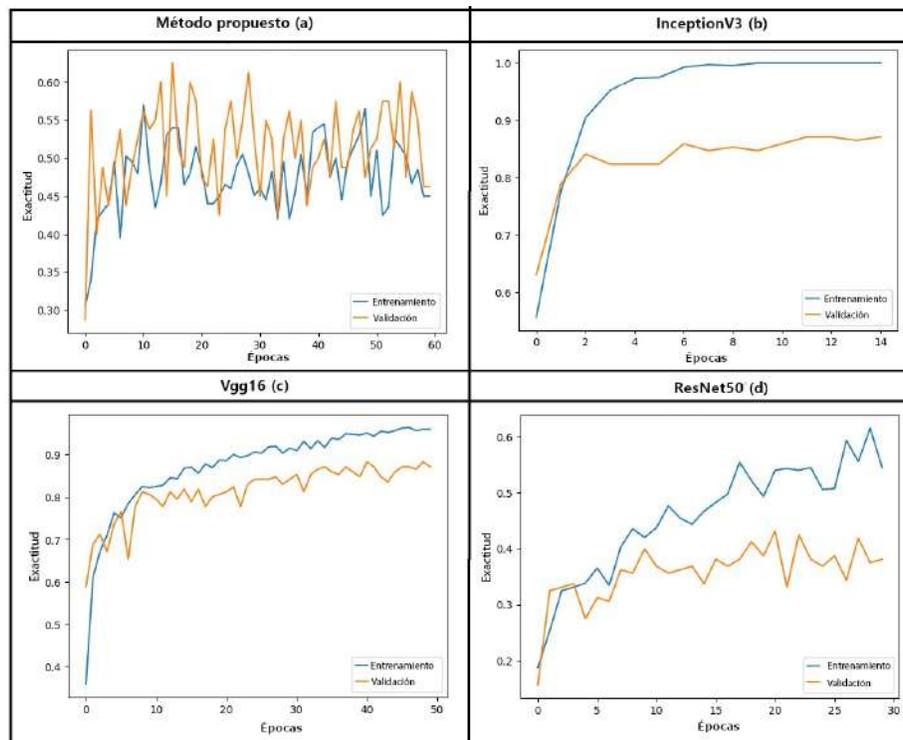


Figura 5.9: Gráficas de entrenamiento experimento 3.

Posteriormente se otorgó una muestra desbalanceada de imágenes para la obtención de test para medir su rendimiento utilizando la matriz de confusión, obteniendo los parámetros: TP=Verdadero positivo; TN=Verdadero negativo; FP=Falso positivo; FN=Falso negativo. En la Figura 5.10 se presentan los resultados de la matriz de confusión (a) *MPro*, (b) *InceptionV3*, (c) *VGG16* y (d) *ResNet50*.

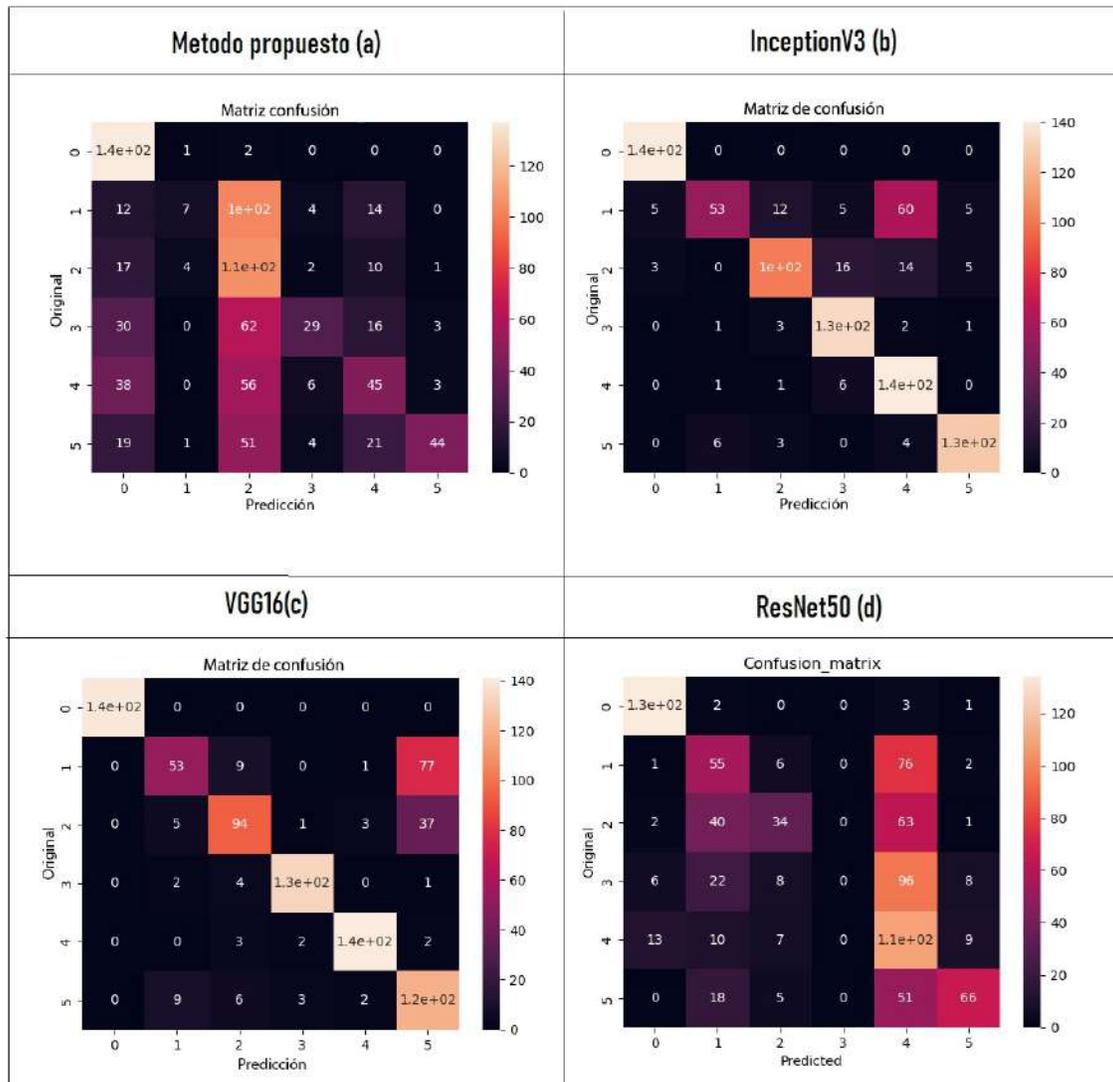


Figura 5.10: Matriz de confusión experimento 3

A continuación, se detalla los resultados del matiz de confusión de cada arquitectura. Primero se muestra los resultados del método propuesto, como se puede apreciar

en la Tabla [5.9](#), se tuvo un *Precisión* de 0.58, el *Macropromedio de la precisión* fue de 0.63, *Sensibilidad* de 0.58 y *Puntuación F1* 0.56.

Clases	<i>Precisión</i>	<i>Sensibilidad</i>	<i>Puntuación F1</i>
0	0.99	1.00	1.00
1	0.53	0.19	0.22
2	0.62	0.25	0.36
3	0.39	0.87	0.54
4	0.74	0.62	0.68
5	0.49	0.61	0.54
Precisión			0.58
Macropromedio	0.63	0.58	0.56

Tabla 5.9: Resultado del método propuesto

Los resultados de la matriz de confusión de la red InceptionV3 se muestran en la Tabla [5.10](#), en el cual se tuvo una exactitud de 0.83, un Macropromedio de la precisión 0.86, sensibilidad 0.83 y puntuación F1 de 0.81.

Clases	<i>Precisión</i>	Sensibilidad	Puntuación F1
0	0.91	1.00	0.98
1	0.96	0.38	0.55
2	0.92	0.69	0.79
3	0.84	0.96	0.89
4	0.65	0.97	0.78
5	0.82	0.95	0.88
Exactitud			0.83
Macropromedio	0.86	0.83	0.81

Tabla 5.10: Resultados InceptionV3 experimento 3.

Los resultados del entrenamiento de la red VGG16 se muestran en la tabla [5.11](#), el cual obtuvo una exactitud de 0.80 y un macropromedio en precisión de 0.82, sensibilidad de 0.49 y Valor-F de 0.79.

Clases	Precisión	Sensibilidad	Puntuacion F1
0	0.99	1.00	1.00
1	0.72	0.41	0.55
2	0.88	0.56	0.68
3	0.87	0.99	0.93
4	0.91	0.97	0.94
5	0.56	0.89	0.69
Exactitud			0.80
Macropromedio	0.82	0.49	0.79

Tabla 5.11: Resultados VGG16 experimento 3.

Los resultados de la red *ResNet50* se muestran en la Tabla 5.12, el cual obtuvo una exactitud de 0.48 y un promedio en precisión de 0.61, sensibilidad de 0.48 y puntuación F1 de 0.44.

Clases	Precisión	Sensibilidad	Puntuación F1
0	0.61	0.99	0.75
1	0.30	0.79	0.43
2	0.76	0.14	0.23
3	0.65	0.14	0.23
4	0.50	0.40	0.44
5	0.86	0.42	0.56
Exactitud			0.48
Macropromedio	0.61	0.48	0.44

Tabla 5.12: Resultado ResNet50.

En la Tabla 5.13 se muestra la comparación de los resultados obtenidos del *MPro* y de las arquitecturas *ResNet50*, *VGG16* e *InceptionV3* de las cuales la arquitectura

*InceptionV3* fue el que obtuvo una mejor exactitud de 0.83, posteriormente le siguieron los modelos *VGG16* con 0.80, método propuesto con 0.58 y por último la red *ResNet50* con 0.48.

Arquitecturas	Exactitud	Precisión	Sensibilidad	Puntuación F1
MPro	0.58	0.51	0.47	0.51
InceptionV3	<b>0.83</b>	0.86	0.83	0.81
Vgg16	0.80	0.82	0.49	0.79
ResNet50	0.48	0.61	0.48	0.44

Tabla 5.13: Resultados con las 4 arquitecturas TF

### 5.3.2. SVM-Métrica ROC y Kernel

Otro método para medir el rendimiento de la clasificación, es mediante la curva ROC, el cual mide el AUC. Se utilizó una máquina de soporte vectorial para la clasificación y se usaron diferentes tipos de Kernel para saber cuál se ajustaba mejor, los Kernel que se usaron son: Kernel lineal, Gaussiano, RBF, Sigmoidee, Polinomial. En la Figura [5.11](#) se muestran los resultados con los diferentes tipos del kernel con el método propuesto.

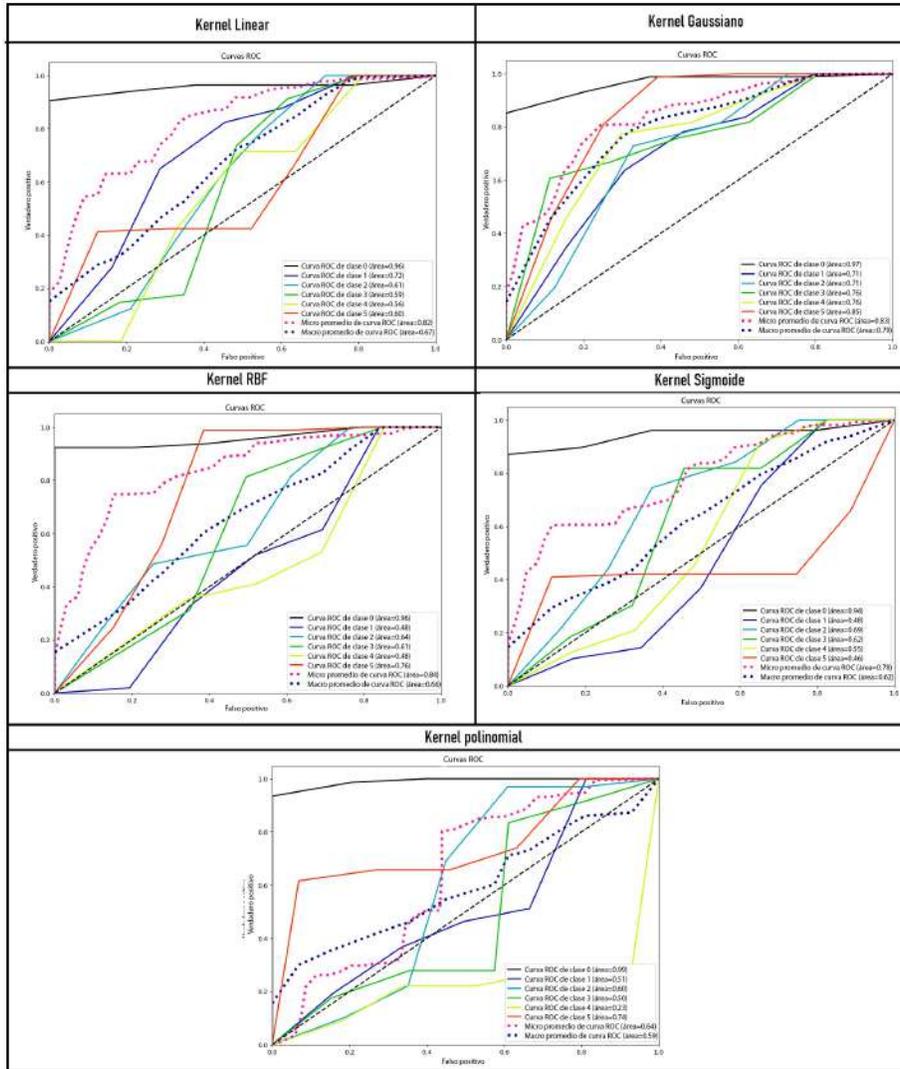


Figura 5.11: Curvas ROC del método propuesto.

En la Figura 5.12 se muestran los resultados de las curvas ROC de la red InceptionV3.

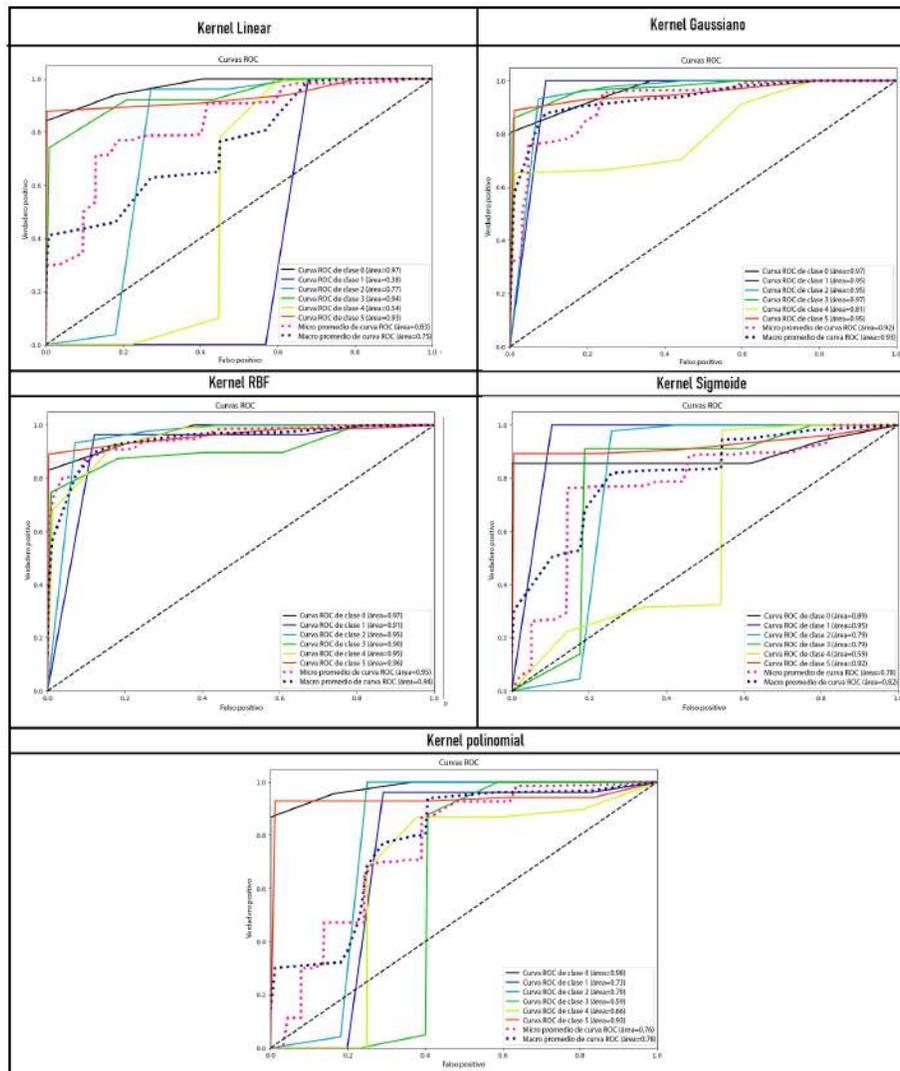


Figura 5.12: Curva ROC Red InceptionV3.

En la Figura 5.13 se muestran los resultados de las curvas ROC de la red VGG16.

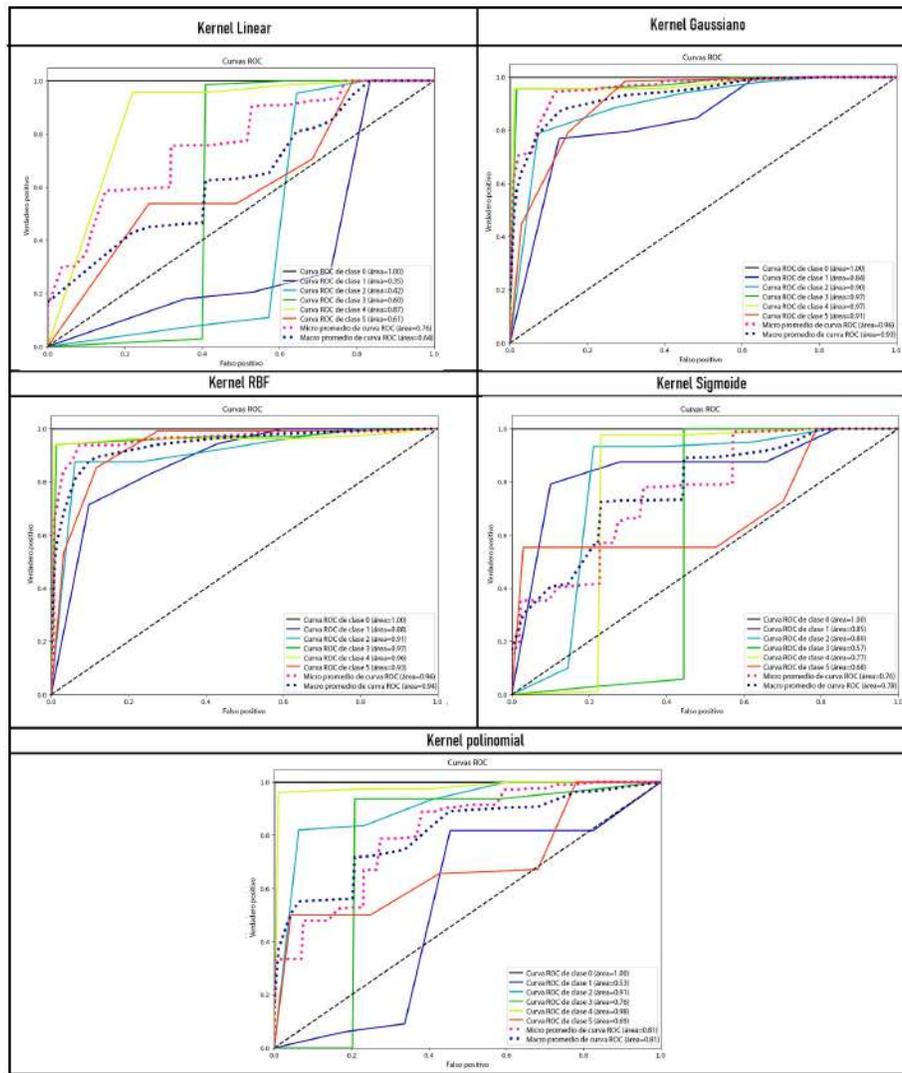


Figura 5.13: Curva ROC Red VGG16.

En la Figura 5.14 se muestran los resultados de las curvas ROC de la red ResNet50.

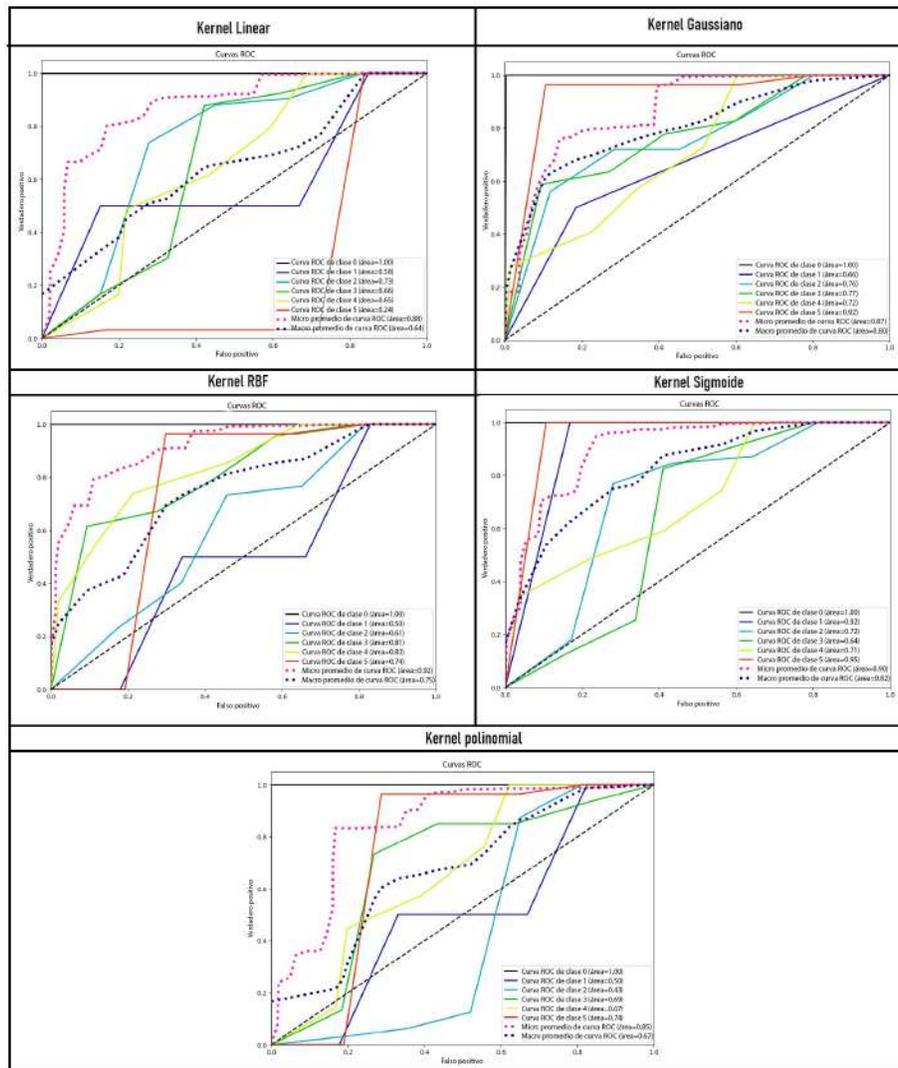


Figura 5.14: Curva ROC Red ResNet50.

La comparación de los resultados de los diferentes Kernel utilizados se muestra en la Figura 5.15, donde se puede apreciar los resultados de las redes entrenadas con los perspectivas kernel utilizado.

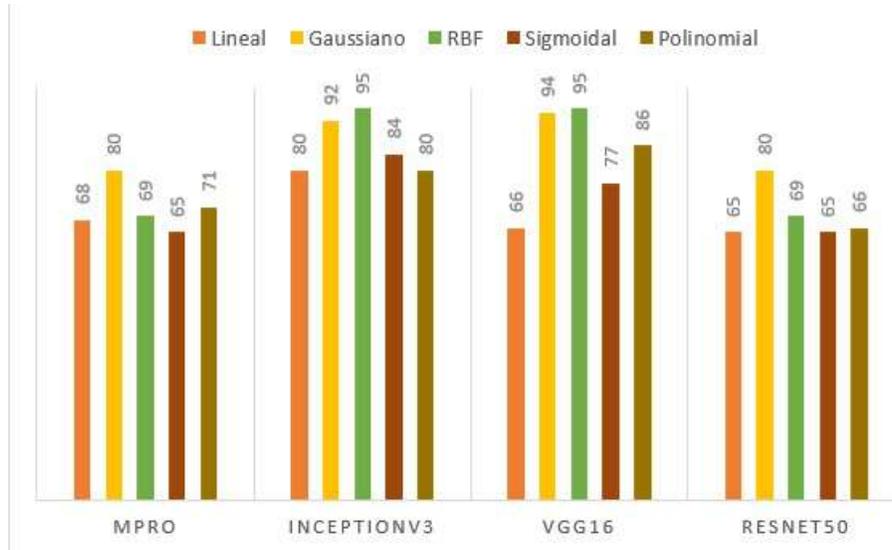


Figura 5.15: Comparación de Curvas ROC, experimento 3.

En la Tabla 5.14 se muestra los mejores resultados de las curvas ROC de cada una de las arquitecturas. A pesar de que las arquitecturas InceptionV3 y VGG16 tienen el mismo resultado de AUC con 0.95. Considerando los resultados del experimento dos

Arquitecturas	Mejor Kernel	AUC
MPro	Gaussiano	0.80
<b>InceptionV3</b>	<i>RBF</i>	<i>0.95</i>
VGG16	RBF	0.95
ResNet50	<i>Gaussiano</i>	0.80

Tabla 5.14: Mejores resultados de las curvas ROC

y tres, se puede observar que cuando se usa transferencia de aprendizaje se mejoran los resultados del AUC. Subiendo la AUC de la arquitectura VGG16 kernel RBF de 0.91 a 0.95, subiendo también InceptionV3 de 0.84 con el kernel gaussiano a 0.95 con el kernel RBF. Entonces se puede deducir que utilizar transferencia de aprendizaje mejora en

4 puntos los resultados de la AUC. También se puede considerar que sustituyendo la clasificación de la red neuronal por una *SVM*, mejora sustancialmente la clasificación.

## 5.4. Resultados y discusiones del experimento 4

El entrenamiento fue de 2000 épocas, Figura 5.16, a continuación, se muestra la gráfica del entrenamiento la cual se puede ver que ambas redes se siguen y que la gráfica de entrenamiento llega al 80 % de exactitud y la gráfica de validación llega a entre el 73 y 78 % de exactitud.

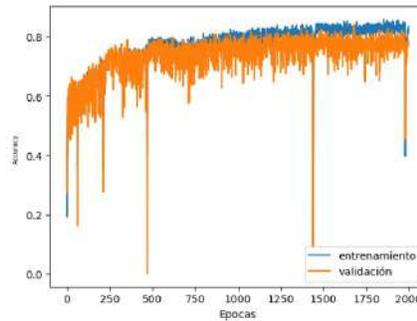


Figura 5.16: Grafica de entrenamiento de la red Backpropagation.

Los resultados de la matriz de confusión se muestran en la Figura 5.17, en este ejemplo se dio una muestra de testing desbalanceada ya que programo para que tomara y guardara ejemplos a lazar para su posterior análisis.

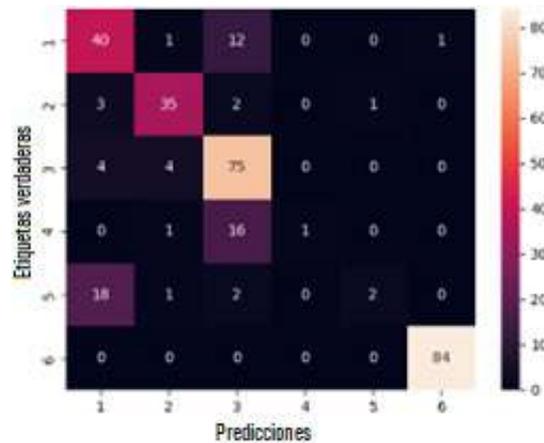


Figura 5.17: Matriz de confusión de la red Backpropagation.

En la Tabla 5.15 se muestra los resultados de la *precisión*, *sensibilidad*, *puntuacion F1* y *exactitud* de las clases, obteniendo un *Exactitud* general de 0.80.

Clases	<i>Precisión</i>	<i>Sensibilidad</i>	<i>Puntuacion F1</i>
0	0.62	0.75	0.67
1	0.83	0.85	0.84
2	0.70	0.90	0.79
3	1.00	0.06	0.11
4	0.67	0.09	0.15
5	0.99	1.00	0.99
Precisión			0.80
Macropromedio	0.80	0.80	0.80

Tabla 5.15: Resultados Backpropagation .

## 6 Discusiones generales

No es posible hacer una comparación de los resultados obtenidos con otros autores porque no se usaron los mismos datos. Así mismo, se debe destacar que, aunque se trata de clasificar TB, en el estado del arte se encontraron perspectivas diferentes; como clasificar tubérculos de entre otras enfermedades llegando a tener una presión de 98 % [18], otra investigación similar que clasifica enfermedades introdujo una máquina de soporte vectorial como clasificador, usando las características que se obtienen de la red neuronal convolucional llegando de hasta 99 % de AUC [19]. Con este enfoque, se trabajó en el primer experimento, la aportación que se tuvo fue obtener los parámetros para realizar transferencia de aprendizaje.

Otra perspectiva para clasificar TB que se encontró en el estado del arte fue identificar diferentes patrones de TB, sin considerar otras enfermedades. Es muy importante destacar que solo se identifica un patrón por radiografía de tórax, con el uso de una red neuronal [23], identifican los siguientes patrones en una sola radiografía de un paciente (no asociados): Exudación, Calcificación, Nódulos, Miliar, Derrame pleural libre y Derrame pleural encapsulado, obteniendo unas precisiones de 22.7 hasta 84.62. Otro investigador [25] con esta misma perspectiva usó una red profunda y una SVM con un AUC de 0.82 %.

La investigación que se realizó en este trabajo pudo haber seguido esta perspectiva, sin embargo, después de haber obtenido la base del el Instituto Nacional de Enfermedades Infecciosas <https://depot.tbportals.niaid.nih.gov/>, se encontró que en una sola radiografía se podían encontrar hasta seis patrones., esto dio un giro a la investigación ya que ahora se debería identificar en una misma radiografía de dos patrones y hasta

seis, es de destacar que en el estado del arte, no se encontró ninguna publicación que abordara la identificación de tuberculosis desde esta perspectiva. Por lo que se considera una contribución importante para la identificación de la tuberculosis en radiografías de tórax. Es satisfactorio señalar que las precisiones que se obtuvieron son buenas ya que se llega a un AUC de hasta 0.95, usando redes neuronales profundas, maquinas de soporte vectorial y el kernel RBF.

## 7 Conclusiones

La TB es una enfermedad preocupante porque a pesar de que hay tratamiento siguen habiendo deseos. Por dicha razón es de suma importancia generar buenos diagnósticos para que la enfermedad no se siga propagando. Como contribución de este trabajo se propone un modelo para identificar entre dos y seis patrones de tuberculosis en una radiografía de tórax. Y con ello se cumple el objetivo del presente trabajo.

Los patrones que se detectaron son: *fibrosis, opacidad, nódulos, cavitaciones, pleural y consolidación*. Con un *AUC* de 0.95. Se concluye que las redes neuronales profundas son una buena opción para obtener características de imágenes donde no se identifican específicamente características.

Para mejorar los resultados es necesario utilizar una maquina de soporte vectorial como clasificador, considerando las características que se obtienen de una red profunda, en este trabajo se encontró que *InceptionV3* y *VGG16* con el *kernel RBF* tienen un buen desempeño. Así mismo, se recomienda usar transferencia de aprendizaje ya que ayudo a subir la *AUC*. Para obtener los parámetros que se usan en la transferencia de aprendizaje, es recomendable usar radiografías de tórax de enfermedades que se pueden diagnosticar con rayos X; como Covid-19, neumonía y tuberculosis.

Es imprescindible que se segmenten los pulmones de las radiografías de rayos X, así mismo, eliminar radiografías donde la enfermedad ya es muy avanzada, ya que no se puede hacer una segmentación de los pulmones para evitar ruido.

Con respecto al experimento donde se entreno una red Backpropagation se obtuvo una precisión de 0.80, pero sería recomendable hacer más pruebas definiendo otras características inherentes a los patrones de la tuberculosis.

La investigación aporta la clasificación de varios patrones de TB en una radiografía, ya que en otras investigaciones hablan de patrones radiológicos unitarios. Al realizar este trabajo se tuvieron conflictos al poder realizar las clasificaciones; porque a la clasificación se le iban agregando más patrones cada vez. Como trabajo a futuro se plantea la posibilidad ampliar el conjunto de datos para continuar con nuevas investigaciones.

## 8 Artículos publicados

En la figura [8.1](#), se muestra la primera página del primer artículo redactado, el cual fue publicado en el congreso del ICIC (International Conference on Intelligent Computing) 2021, generando un capítulo de libro publicado en la editorial Springer. A continuación, se anexa el *link*: [https://link.springer.com/chapter/10.1007/978-3-030-84532-2\\_25](https://link.springer.com/chapter/10.1007/978-3-030-84532-2_25).



Figura 8.1: Artículo publicado

En la figura [8.2](#), se muestra la primera página del segundo artículo redactado, el cual fue publicado en la revista ELCVIA. A continuación, se anexa el *link*: <https://elcvia.cvc.uab.cat/article/view/1561/628>

*Electronic Letters on Computer Vision and Image Analysis 23(1):47-59, 2024*

## Classification of radiological patterns of tuberculosis with a Convolutional neural network in x-ray images

Jessica Sanchez <sup>\*</sup>, Adrian Trueba <sup>\*</sup>, Jair Cervantes <sup>\*</sup>, Farid Garcia-Lamont, José Sergio Ruiz Castilla<sup>\*</sup>, and Karthik Kantipudi<sup>†</sup>

<sup>\*</sup> UAEMEX (Autonomous University of Mexico State), Texcoco, 56259, Mexico

<sup>†</sup> National Institutes of Health, Bethesda, MD, USA

Received 9th of March, 2023; accepted 8th of June 2024

### Abstract

Tuberculosis is a disease that causes more deaths worldwide. To detect it, laboratory studies are performed so it can take more than 30 days, the quickest diagnosis is made by X-ray radiographs. Tuberculosis presents different radiological patterns (consolidation, fibrosis, opacity, pleural, nodules and cavitations), depending on the pattern found, the specialist prescribes the most appropriate treatment. According to the diagnosis and radiographs provided by the National Institute of Allergy and Infectious Disease laboratory, there is not only one single pattern in a radiograph, but there can also be from one to six patterns. The aim of this article is to detect the presence of two to six tuberculosis patterns in the same X-ray radiograph. No research was found that addresses the classification of more than one pattern of tuberculosis, this is what is relevant to this work. For tuberculosis pattern classification, a convolutional neural network with its own architecture was proposed, and the VGG16, InceptionV3 and ResNet-50 architectures were tested. A support vector machine was used for multi-classification. The results were validated with the area under the curve, with the Macro-average metric with the following results: in the proposed architecture and the InceptionV3 it was 0.80, and in the VGG16 it was 0.75, as for the ResNet-50 network 0.79 was obtained. The proposed architecture has better classification results, as does InceptionV3.

**Key Words:** Tuberculosis patterns, Chest X-rays, convolutional neural networks.

## 1 Introduction

One disease that has not been eradicated in the world is tuberculosis (TB), an infectious bacterial infection caused by the bacterium *Mycobacterium* or Koch's bacillus [1]. Although it is curable and preventable, it still causes deaths because it is spread through the air; when a person with pulmonary TB coughs or sneezes, they expel tubercle bacilli into the air and spread the disease [2].

The World Health Organization (WHO) estimates that one third of the world's population has latent TB, meaning that there are people infected with the bacillus who have no symptoms of the disease, but are carriers of the bacillus and can transmit the infection [3]. People infected with the TB bacillus have a 5-10 % lifetime

Correspondence to: [jsancheza560@alumno.uaemex.mx](mailto:jsancheza560@alumno.uaemex.mx)

Recommended for acceptance by Angel D. Sappa

<https://doi.org/10.5565/rev/elcvia.1822>

ELCVIA ISSN: 1577-5097

Published by Computer Vision Center / Universitat Autònoma de Barcelona, Barcelona, Spain

Figura 8.2: Artículo en revisión

# Bibliografía

- [1] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” dec 2015.
- [2] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” sep 2014.
- [3] A. Castiñeira Estévez, M. López Pedreira, M. Pena Rodríguez, and M. Liñares Iglesias, “Medicina integral : medicina preventiva y asistencial en el medio rural.” pp. 192–206, mar 1980. [Online]. Available: <http://www.elsevier.es/es-revista-medicina-integral-63-articulo-manifestaciones-radiologicas-tuberculosis-pulmonar-13029>
- [4] R. Garza-Velasco, J. Ávila-de Jesús, and L. M. Perea-Mejía, “Tuberculosis pulmonar: la epidemia mundial continúa y la enseñanza de este tema resulta crucial y compleja,” *Educación Química*, vol. 28, no. 1, jan 2017.
- [5] Geneva: World Health Organization, “Global tuberculosis report 2021,” World Health Organization, Tech. Rep., oct 2021. [Online]. Available: <https://www.who.int/teams/global-tuberculosis-programme/tb-reports/global-tuberculosis-report-2021>
- [6] Newsroom World Health Organization, “Tuberculosis,” oct 2021. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/tuberculosis>
- [7] Geneva: World Health Organization, “Global strategy for tuberculosis research and innovation.” pp. 1–44, oct 2020. [Online]. Available: <https://www.who.int/es/publications/i/item/9789240010017>
- [8] A. M. Ismael and A. Şengür, “Deep learning approaches for COVID-19 detection based on chest X-ray images,” *Expert Systems with Applications*, vol. 164, feb 2021.
- [9] S. Stirenko, Y. Kochura, O. Alienin, O. Rokovyi, Y. Gordienko, P. Gang, and W. Zeng, “Chest X-Ray Analysis of Tuberculosis by Deep Learning with Segmentation and Augmentation,” in *2018 IEEE 38th International Conference on Electronics and Nanotechnology (ELNANO)*. IEEE, apr 2018.
- [10] T. Karnkawinpong and Y. Limpiyakorn, “Classification of pulmonary tuberculosis lesion with convolutional neural networks,” *Journal of Physics: Conference Series*, vol. 1195, apr 2019.

- [11] T. Evgeniou and M. Pontil, “Support Vector Machines: Theory and Applications,” pp. 249–257, 2001.
- [12] Y. Tian, Y. Shi, X. Chen, and W. Chen, “AUC Maximizing Support Vector Machines with Feature Selection,” *Procedia Computer Science*, vol. 4, 2011.
- [13] National Institute of Health, “NIAID TB PORTALS DEPOT.” [Online]. Available: <https://depot.tbportals.niaid.nih.gov/>
- [14] M. O. Omisore, O. W. Samuel, and E. J. Atajeromavwo, “A Genetic-Neuro-Fuzzy inferential model for diagnosis of tuberculosis,” *Applied Computing and Informatics*, vol. 13, no. 1, jan 2017.
- [15] S. Jaeger, A. Karargyris, S. Candemir, L. Folio, J. Siegelman, F. Callaghan, Zhiyun Xue, K. Palaniappan, R. K. Singh, S. Antani, G. Thoma, Yi-Xiang Wang, Pu-Xuan Lu, and C. J. McDonald, “Automatic Tuberculosis Screening Using Chest Radiographs,” *IEEE Transactions on Medical Imaging*, vol. 33, no. 2, feb 2014.
- [16] S. Hwang, H.-E. Kim, J. Jeong, and H.-J. Kim, “A novel approach for tuberculosis screening based on deep convolutional neural networks,” mar 2016.
- [17] U. Lopes and J. Valiati, “Pre-trained convolutional neural networks as feature extractors for tuberculosis detection,” *Computers in Biology and Medicine*, vol. 89, oct 2017.
- [18] A. H. Al-Timemy, R. N. Khushaba, Z. M. Mosa, and J. Escudero, “An Efficient Mixture of Deep and Machine Learning Models for COVID-19 and Tuberculosis Detection Using X-Ray Images in Resource Limited Settings,” pp. 77–100, jul 2021. [Online]. Available: <https://link.springer.com/10.1007/978-3-030-69744-0{ }6>
- [19] S. Rajaraman and S. K. Antani, “Modality-Specific Deep Learning Model Ensembles Toward Improving TB Detection in Chest Radiographs,” *IEEE Access*, vol. 8, 2020.
- [20] S. Khobragade, A. Tiwari, C. Patil, and V. Narke, “Automatic detection of major lung diseases using Chest Radiographs and classification by feed-forward artificial neural network,” in *2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*. IEEE, jul 2016.
- [21] P. Lakhani and B. Sundaram, “Deep Learning at Chest Radiography: Automated Classification of Pulmonary Tuberculosis by Using Convolutional Neural Networks,” *Radiology*, vol. 284, no. 2, aug 2017.
- [22] C. Quezada and M. E. Guarda, “LAS FACETAS RADIOLÓGICAS DE LA TUBERCULOSIS PULMONAR,” *Neumología Pediátrica*, vol. 10, no. 4, pp. 179–2015, 2015.

- [23] Y. Xie, Z. Wu, X. Han, H. Wang, Y. Wu, L. Cui, J. Feng, Z. Zhu, and Z. Chen, "Computer-Aided System for the Detection of Multicategory Pulmonary Tuberculosis in Radiographs," *Journal of Healthcare Engineering*, vol. 2020, pp. 1–12, aug 2020. [Online]. Available: <https://www.hindawi.com/journals/jhe/2020/9205082/>
- [24] W. H. Curioso and M. J. Brunette, "Inteligencia artificial e innovación para optimizar el proceso de diagnóstico de la tuberculosis," *Revista Peruana de Medicina Experimental y Salud Pública*, vol. 37, no. 3, sep 2020.
- [25] A. S. Becker, C. Blüthgen, V. D. Phi van, C. Sekaggya-Wiltshire, B. Castelnuovo, A. Kambugu, J. Fehr, and T. Frauenfelder, "Detection of tuberculosis patterns in digital photographs of chest X-ray images using Deep Learning: feasibility study," *The International Journal of Tuberculosis and Lung Disease*, vol. 22, no. 3, mar 2018.
- [26] K. Ramana and S. Khader Basha, "Neural image recognition system with application to tuberculosis detection," in *International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004*. IEEE, 2004.
- [27] E. López and Y. Amador, "Tuberculosis," *Revista Cubana de Estomatología*, vol. 38, pp. 33–51, nov 2001. [Online]. Available: [http://scielo.sld.cu/scielo.php?script=sci\\_abstract&pid=S0034-75072001000100005&lng=en&nrm=iso](http://scielo.sld.cu/scielo.php?script=sci_abstract&pid=S0034-75072001000100005&lng=en&nrm=iso)
- [28] OMS, "Tuberculosis," 2023. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/tuberculosishttps://www.who.int/es/news-room/fact-sheets/detail/tuberculosis>
- [29] SSA México, "Estandares para la atención de la Tuberculosis en México," Tech. Rep., 2008.
- [30] D. Juan Carlos Rodríguez, "Tuberculosis," *Revista Médica Clínica Las Condes*, vol. 25, no. 3, pp. 547–552, may 2014.
- [31] J. González-Martín, J. M. García-García, L. Anibarro, R. Vidal, J. Esteban, R. Blanquer, S. Moreno, and J. Ruiz-Manzano, "Documento de consenso sobre diagnóstico, tratamiento y prevención de la tuberculosis," *Archivos de Bronconeumología*, vol. 46, no. 5, pp. 255–274, may 2010.
- [32] Z. Ul Abideen, M. Ghafoor, K. Munir, M. Saqib, A. Ullah, T. Zia, S. A. Tariq, G. Ahmed, and A. Zahra, "Uncertainty Assisted Robust Tuberculosis Identification With Bayesian Convolutional Neural Networks," *IEEE Access*, vol. 8, pp. 22 812–22 825, 2020. [Online]. Available: <https://ieeexplore.ieee.org/document/8972440/>

- [33] J. R. Barba Evia, “Tuberculosis. ¿Es la pandemia ignorada?” *Revista Mexicana de Patología Clínica y Medicina de Laboratorio*, vol. 67, no. 2, pp. 93–112, 2020.
- [34] “Rayos X,” 2022. [Online]. Available: <https://www.nibib.nih.gov/espanol/temas-cientificos/rayos-x>
- [35] F. Domínguez Del Valle, B. Fernández, M. Pérez de Las Casas, B. Marín, and C. Bermejo, *[Clinical manifestations and radiology of thoracic tuberculosis]*., 2007, vol. 30 Suppl 2.
- [36] A. Molina Pablo AND Espinoza, “Tuberculosis pleural,” *Revista Costarricense de Ciencias MÃ*, vol. 26, pp. 45 – 50, nov 2005. [Online]. Available: [http://www.scielo.sa.cr/scielo.php?script=sci\\_arttext&pid=S0253-29482005000300005&nrm=iso](http://www.scielo.sa.cr/scielo.php?script=sci_arttext&pid=S0253-29482005000300005&nrm=iso)
- [37] J. J. E. Elizondo and L. E. P. Maestre, *Fundamentos para el procesamiento de imágenes*, uabc ed.
- [38] E. A. Gutiérrez, L. S. González, R. Á. F. Díaz, and J. C. M. Antolín, *PROCESAMIENTO digital de imagen: fundamentos y prácticas con MATLAB*, león: univ ed., 2003.
- [39] N. L. S. Palomino and U. R. Concha, “Técnicas de Segmentación en Procesamiento Digital de Imágenes,” 2014.
- [40] J. J. Romero Cardalda, C. Dafonte Vázquez, Á. Gómez García, and F. J. Penousal Martins Machado, “Inteligencia Artificial y Computación Avanzada,” p. 400, 2007. [Online]. Available: <https://cdv.dei.uc.pt/wp-content/uploads/2014/03/ms07.pdf>
- [41] R. Florez Lopez and J. M. Fernández Fernández, *Las redes neuronales artificiales fundamentos teóricos y aplicaciones prácticas*, . Oleiros (La Coruña) : Netbiblo, Ed., España, 2008.
- [42] A. García Serrano, *INTELIGENCIA ARTIFICIAL Fundamentos, práctica y aplicaciones* , 2nd ed., Alfaomega-RC Libros, Ed., España, oct 2012.
- [43] W. Rivas Asanza and B. Mazón Olivo, “Redes neuronales artificiales aplicadas al reconocimiento de patronesRedes neuronales artificiales aplicadas al reconocimiento de patrones,” 2018. [Online]. Available: <http://repositorio.utmachala.edu.ec/handle/48000/12499>
- [44] I. P. Borrero and Manuel E. Gegúndez Arias, *Deep learning Fundamentos, teoría y aplicación*, primera ed ed., Uhu.es, Ed. Universidad de Huelva, 2021.
- [45] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, may 2015.

- [46] R. Sathya and A. Abraham, "Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification," *International Journal of Advanced Research in Artificial Intelligence*, vol. 2, no. 2, 2013.
- [47] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic Routing Between Capsules," oct 2017.
- [48] C. Quintero, F. Merchán, A. Cornejo, and J. S. Galán, "Uso de Redes Neuronales Convolucionales para el Reconocimiento Automático de Imágenes de Macroinvertebrados para el Biomonitorio Participativo," *KnE Engineering*, vol. 3, no. 1, p. 585, feb 2018.
- [49] B. Triwijoyo, B. Sabarguna, W. Budiharto, and E. Abdurachman, "ICIC Express Letters ICIC International 2020 ISSN," *ICIC Express Letters*, vol. 14, pp. 635–641, oct 2020.
- [50] J. Cowley, "Redes neuronales convolucionales," dec 2018. [Online]. Available: <https://developer.ibm.com/es/articles/cc-convolutional-neural-network-vision-recognition/>
- [51] J. Fang, Y. Zhou, Y. Yu, and S. Du, "Fine-Grained Vehicle Model Recognition Using A Coarse-to-Fine Convolutional Neural Network Architecture," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1782–1792, jul 2017.
- [52] S. Lawrence, C. Giles, Ah Chung Tsoi, and A. Back, "Face recognition: a convolutional neural-network approach," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98–113, jan 1997.
- [53] S. Indolia, A. K. Goswami, S. Mishra, and P. Asopa, "Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach," *Procedia Computer Science*, vol. 132, pp. 679–688, 2018.
- [54] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," dec 2014.
- [55] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," dec 2015.
- [56] S. Akter, F. M. J. M. Shamrat, S. Chakraborty, A. Karim, and S. Azam, "COVID-19 Detection Using Deep Learning Algorithm on Chest X-ray Images," *Biology*, vol. 10, no. 11, p. 1174, nov 2021. [Online]. Available: <https://www.mdpi.com/2079-7737/10/11/1174>
- [57] L. Torrey and J. Shavlik, "Transfer Learning," *Handbook of Research on Machine Learning Applications and Trends*, pp. 242–264, 2010. [Online]. Available: <http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-60566-766-9.ch011>

- [58] S. Raschka, *Machine Learning con PyTorch y Scikit-Learn*. Marcombo, 2023. [Online]. Available: <https://books.google.com.mx/books?id=NumwEAAAQBAJ>
- [59] S. Visa, B. Ramsay, A. Ralescu, and E. Knaap, “Confusion Matrix-based Feature Selection.” in *CEUR Workshop Proceedings*, vol. 710, oct 2011, pp. 120–127. [Online]. Available: <https://www.researchgate.net/publication/220833270{ }Confusion{ }Matrix-based{ }Feature{ }Selection>
- [60] A. K. Santra and J. Christy, “Genetic Algorithm and Confusion Matrix for Document Clustering,” *International Journal of Computer Science Issues*, vol. 9, no. 1, 2012.
- [61] A. K. Baszczyńska, “Empirical and Kernel Estimation of the ROC Curve,” *Acta Universitatis Lodzianensis. Folia Oeconomica*, vol. 1, no. 311, apr 2015.
- [62] D. Belete and M. D H, “Grid search in hyperparameter optimization of machine learning models for prediction of hiv/aids test results,” *International Journal of Computers and Applications*, vol. 44, pp. 1–12, 09 2021.
- [63] T. Rahman, A. Khandakar, M. A. Kadir, K. R. Islam, K. F. Islam, R. Mazhar, T. Hamid, M. T. Islam, S. Kashem, Z. B. Mahbub, M. A. Ayari, and M. E. H. Chowdhury, “Reliable Tuberculosis Detection Using Chest X-Ray With Deep Learning, Segmentation and Visualization,” *IEEE Access*, vol. 8, 2020.
- [64] L. L. Wang, K. Lo, Y. Chandrasekhar, R. Reas, J. Yang, D. Burdick, D. Eide, K. Funk, Y. Katsis, R. Kinney, Y. Li, Z. Liu, W. Merrill, P. Mooney, D. Murdick, D. Rishi, J. Sheehan, Z. Shen, B. Stilson, A. Wade, K. Wang, N. X. R. Wang, C. Wilhelm, B. Xie, D. Raymond, D. S. Weld, O. Etzioni, and S. Kohlmeier, “CORD-19: The COVID-19 Open Research Dataset,” apr 2020.
- [65] D. S. Kermany, M. Goldbaum, W. Cai, C. C. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan, J. Dong, M. K. Prasadha, J. Pei, M. Y. Ting, J. Zhu, C. Li, S. Hewett, J. Dong, I. Ziyar, A. Shi, R. Zhang, L. Zheng, R. Hou, W. Shi, X. Fu, Y. Duan, V. A. Huu, C. Wen, E. D. Zhang, C. L. Zhang, O. Li, X. Wang, M. A. Singer, X. Sun, J. Xu, A. Tafreshi, M. A. Lewis, H. Xia, and K. Zhang, “Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning,” *Cell*, vol. 172, no. 5, feb 2018.
- [66] L. G. Morales, *Tratamiento Digital de la Imagen*, Spanien, Ed., 2021.