

Comparative evaluation of CNN and SOM in the detection of tuberculosis in chest radiographs using a mobile platform

Asuncion Pomasonco, Alexia Nicoll¹; Barzola Mendoza, Axel Benjamin¹; Huarote Zegarra, Raul Eduardo¹

¹Universidad Nacional Tecnológica de Lima Sur, Peru, 2113110163@untels.edu.pe, 20b3010455@untels.edu.pe,
rhuarote@untels.edu.pe

Abstract– This research presents the development of an automated diagnostic system for the early detection of tuberculosis (TB) in chest radiographs by comparing convolutional neural networks (CNN) and self-organizing maps (SOM), in order to evaluate which one offered better results in medical classification tasks. For the CNN model, pre-trained MobileNetV2 was used as feature extractor being fine-tuned with augmented and normalized images, achieving 98 % accuracy in binary classification (normal or tuberculosis). On the other hand, the SOM model was trained with the vectors generated by the CNN and allowed visualizing the distribution of the data in the feature space, achieving an accuracy of 95 %. Therefore, after comparing the performance of both models, it was chosen to implement the CNN as the core of the system, due to its high accuracy and generalization capability. The final model was integrated into a mobile application that connects to cloud services using Hugging Face for inference and Firestore for results storage. This solution was especially designed for resource-limited contexts, allowing health professionals in rural or hard-to-reach areas to have a support tool for the preliminary diagnosis of TB from a smartphone.

Keywords– tuberculosis, chest radiograph, CNN, SOM, smartphone.

Evaluación comparativa de CNN y SOM en la detección de tuberculosis en radiografías de tórax usando una plataforma móvil

Asuncion Pomasonco, Alexia Nicoll¹ ; Barzola Mendoza, Axel Benjamin¹ ; Huarote Zegarra, Raul Eduardo¹ 

¹Universidad Nacional Tecnológica de Lima Sur, Peru, 2113110163@untels.edu.pe, 20b3010455@untels.edu.pe, rhuarote@untels.edu.pe

Abstract– *La presente investigación presenta el desarrollo de un sistema de diagnóstico automatizado para la detección temprana de tuberculosis (TBC) en radiografías de tórax mediante la comparación de redes neuronales convolucionales (CNN) y mapas autoorganizados (SOM), con el objetivo de evaluar cuál ofrecía mejores resultados en tareas de clasificación médica. Para el modelo CNN, se utilizó MobileNetV2 preentrenado como extractor de características siendo ajustado mediante fine-tuning con imágenes aumentadas y normalizadas, logrando una precisión del 98 % en la clasificación binaria (normal o tuberculosis). Por otro lado, el modelo SOM fue entrenado con los vectores generados por la CNN y permitió visualizar la distribución de los datos en el espacio de características, alcanzando una precisión del 95 %. Por lo tanto, tras comparar el desempeño de ambos modelos, se optó por implementar la CNN como núcleo del sistema, debido a su alta precisión y capacidad de generalización. El modelo final fue integrado en una aplicación móvil que se conecta a servicios en la nube mediante Hugging Face para la inferencia y Firestore para el almacenamiento de resultados. Esta solución fue pensada especialmente para contextos con recursos limitados, permitiendo a profesionales de salud en zonas rurales o de difícil acceso contar con una herramienta de apoyo para el diagnóstico preliminar de la TBC desde un smartphone.*

Palabras clave– *Tuberculosis, radiografía de tórax, SOM, CNN, smartphone.*

I. INTRODUCCIÓN

La tuberculosis (TBC) es una enfermedad infecciosa que continúa siendo un importante problema de salud pública a nivel mundial, especialmente en países de ingresos bajos y medios. La Organización Mundial de la Salud (OMS) estima que en 2019, 1700 millones de personas estaban infectadas con *Mycobacterium tuberculosis* [1]. Según [2] la transmisión ocurre principalmente por vía aérea, afectando mayoritariamente a los pulmones y favoreciendo su rápida diseminación en comunidades densamente pobladas y con recursos sanitarios limitados. A pesar de que existen tratamientos efectivos, J. Polanco en [3] indica que el verdadero reto radica en el diagnóstico preciso, dificultado por la limitada disponibilidad de laboratorios especializados, la necesidad de trasladar a los pacientes a ciudades distantes para realizar pruebas como cultivos y moleculares, y la escasa capacitación del personal de salud, especialmente en las regiones más apartadas.

La detección de casos de TBC según D. Falcón en [4], se basa en la búsqueda activa y permanente de pacientes sintomáticos respiratorios (SR) que presenten tos y flema por

más de 15 días. Los principales métodos diagnósticos incluyen la baciloscopia (BX), con una sensibilidad del 75%, y el cultivo en medio awokado, que alcanza una sensibilidad superior al 90% aunque requiere hasta 8 semanas de incubación. No obstante, en las zonas rurales de difícil acceso, las condiciones de pobreza, malnutrición, hacinamiento y la limitada disponibilidad de personal capacitado dificultan el diagnóstico oportuno de la TBC, lo que conlleva a diagnósticos tardíos o erróneos [5]. Las barreras en el acceso a los servicios de salud, como largas distancias a los centros de atención, falta de infraestructura sanitaria adecuada y recursos diagnósticos limitados, agravan aún más la situación. Estas dificultades afectan directamente el control de la enfermedad, ya que un diagnóstico temprano es esencial para iniciar el tratamiento adecuado y evitar la propagación continua de la infección.

Ante estas limitaciones, las tecnologías basadas en inteligencia artificial han emergido como una alternativa complementaria en el diagnóstico médico, ofreciendo soluciones automatizadas, rápidas y escalables. Inicialmente, los primeros modelos de patología computacional se enfocaron en la identificación de estructuras celulares como el estroma y las células tumorales, utilizando características de textura, orientación y contraste. Posteriormente, el desarrollo de redes neuronales convolucionales permitió analizar grandes volúmenes de imágenes, segmentar tejidos y detectar mitosis con alta precisión, reduciendo el tiempo de análisis y la carga de trabajo de los especialistas [6]. Una alternativa menos explorada, pero eficiente para ciertos contextos, es el uso de Mapas Autoorganizados (Self-Organizing Maps, SOM), una técnica de aprendizaje no supervisado que permite proyectar datos multivariados en espacios bidimensionales conservando las relaciones topológicas entre las muestras. Este enfoque ha demostrado su capacidad para identificar patrones complejos sin requerir grandes volúmenes de datos etiquetados, como se evidenció en la clasificación de calidad de vinos a partir de atributos fisicoquímicos, alcanzando precisiones superiores al 90% en los modelos desarrollados [7].

Los Mapas Autoorganizados han demostrado ser útiles en la clasificación visual y la detección de patrones ocultos en datos médicos. Por ejemplo, el estudio de Manokar [8] empleó Mapas Autoorganizados (SOM) para la clasificación de imágenes médicas, obteniendo resultados comparables a los de modelos supervisados. Se logró una precisión del 72 % en la detección de casos positivos y negativos de demencia,

utilizando la técnica LVQ como complemento para mejorar el rendimiento del modelo. Estos resultados evidencian que el uso de SOM puede ser una alternativa efectiva en las etapas iniciales del desarrollo de sistemas de apoyo al diagnóstico, especialmente cuando se dispone de conjuntos de datos limitados o no etiquetados. Esta técnica no solo permite interpretar de forma visual la estructura de los datos, sino que también reduce el riesgo de sobreajuste en muestras pequeñas, lo cual es una ventaja significativa frente a las redes profundas [9].

En contraste, las redes convolucionales, al ser un modelo supervisado, requieren grandes cantidades de datos etiquetados para alcanzar altos niveles de precisión, lo cual puede ser una limitación en contextos donde la disponibilidad de imágenes médicas clasificadas es escasa [10]. Sin embargo, su capacidad para aprender representaciones jerárquicas y detectar patrones complejos las ha convertido en una herramienta de referencia en tareas de diagnóstico automatizado. Estudios previos han mostrado que, entrenadas adecuadamente, las CNN pueden identificar signos radiológicos de TBC con niveles de precisión cercanos al diagnóstico humano, como es el caso de [11] donde un modelo basado en la arquitectura EfficientNet-B0 alcanzó una precisión del 99.29 % en un conjunto específico de rayos X de tórax y demostró ser altamente efectivo en la detección automática de TBC, reduciendo la subjetividad diagnóstica y el margen de error humano. Otro estudio en [12] logró una precisión del 97.98 % en la detección de características radiológicas clave como infiltrados, cavidades, derrames pleurales y patrones miliares, destacando el potencial de la IA como herramienta complementaria ante la escasez de radiólogos en regiones con recursos limitados; no obstante, también se identificaron limitaciones, como el uso de conjuntos de datos homogéneos y la ausencia de información clínica del paciente, lo que podría afectar la aplicabilidad del modelo en escenarios reales. Estas observaciones demuestran tanto el valor como los desafíos del uso de CNN en el diagnóstico automatizado de la TBC, y abren la puerta a explorar modelos alternativos o complementarios que puedan adaptarse mejor a contextos donde los recursos, tanto humanos como tecnológicos, son limitados.

Este artículo presenta el desarrollo de una aplicación móvil para la detección preliminar de tuberculosis utilizando imágenes de rayos X de tórax. La propuesta se basa en la implementación y evaluación de dos modelos de red neuronal, específicamente un Mapa Autoorganizado (SOM) y una red neuronal convolucional (CNN), ambos diseñados para identificar patrones radiológicos compatibles con indicios de TBC, con el objetivo de comparar su rendimiento diagnóstico y determinar su viabilidad en contextos de recursos limitados. El sistema permite al usuario cargar una imagen de tórax, la cual es procesada por ambos modelos para identificar si existen patrones compatibles con indicios de TBC. Esta solución busca aportar una herramienta accesible para profesionales de salud en campo, brigadas médicas y centros

de atención primaria, especialmente en regiones donde el diagnóstico oportuno aún es una meta lejana.

II. MATERIALES Y MÉTODOS

A. Base de datos de imágenes

El insumo principal para esta investigación es un conjunto de datos público obtenido del repositorio Kaggle, titulado "Tuberculosis (TB) Chest X-ray Database". Este dataset contiene una colección de imágenes radiográficas divididas en dos categorías fundamentales para el problema de diagnóstico:

1) *Normal*: Imágenes correspondientes a individuos sanos que no presentan signos visuales de patología pulmonar, cuyas radiografías presentan una estructura pulmonar clara y bien definida.

2) *Tuberculosis*: Imágenes de pacientes con diagnóstico clínico de tuberculosis, las cuales exhiben las anomalías estructurales asociadas a dicha enfermedad teniendo algunas especificaciones comunes.

Para evitar el sesgo del modelo hacia la categoría mayoritaria y garantizar un entrenamiento equitativo, se realizó un proceso de balanceo de categoría. Se redujo el número de imágenes de la categoría 'Normal' mediante submuestreo aleatorio (random down-sampling) para igualar el número de imágenes de la categoría 'Tuberculosis'.

El conjunto de datos final y balanceado utilizado para el entrenamiento y la evaluación consta de 1370 imágenes, distribuidas equitativamente en 685 imágenes para cada categoría.

Para la validación del modelo, este conjunto se dividió de manera estratificada para mantener la proporción de clases en ambos subconjuntos:

1) *Conjunto de entrenamiento (80%)*: Corresponde a la mayor parte de los datos y se utiliza exclusivamente para el proceso de entrenamiento de las redes neuronales SOM y CNN, es decir, para la organización del mapa.

2) *Conjunto de prueba (20%)*: Es una porción menor de los datos que se mantiene completamente separada durante el entrenamiento. Se utiliza al final del proceso para evaluar la capacidad del modelo para clasificar correctamente imágenes que nunca ha visto, proporcionando una medida imparcial de su poder de generalización.

B. Reducción de dimensionalidad (PCA)

Es una técnica estadística ampliamente utilizada para la reducción de dimensionalidad en conjuntos de datos de alta dimensión. Su objetivo principal es transformar las variables originales en un nuevo conjunto de variables no correlacionadas llamadas componentes principales, ordenadas según la cantidad de varianza que explican del conjunto de datos. Al conservar solo los primeros componentes, se puede mantener la información más relevante mientras se eliminan redundancias y ruido. PCA es especialmente útil en el procesamiento de imágenes, donde cada píxel representa una variable y se requiere simplificar la información para su análisis computacional eficiente.

Un ejemplo representativo del uso de PCA en imágenes puede encontrarse en la investigación de [13], donde se aplica la técnica a un conjunto de imágenes para visualizar el efecto de la reducción de dimensionalidad en la reconstrucción visual. En la Fig. 1, se observa cómo la imagen original puede ser representada de forma aproximada utilizando solo un subconjunto de componentes principales, preservando la estructura general a pesar de la pérdida de detalles.

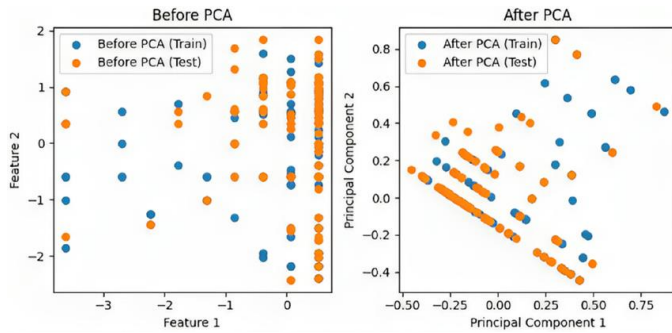


Fig. 1 Ejemplo visual del efecto de PCA: a la izquierda, imagen original; a la derecha, reconstrucción a partir de componentes principales. Imagen tomada de [13].

C. Mapas autoorganizados (SOM)

Los Mapas Autoorganizados o Self-Organizing Maps (SOM) fueron introducidos por Teuvo Kohonen en [14] como una arquitectura de red neuronal no supervisada capaz de proyectar datos de alta dimensión en un espacio bidimensional preservando la topología de los datos originales. Esta red neuronal utiliza un mecanismo de aprendizaje competitivo, donde las neuronas compiten para representar patrones de entrada, y aquella más cercana (neurona ganadora) es la que ajusta sus pesos junto con sus vecinas. Este proceso permite que, tras un número suficiente de iteraciones, las neuronas se especialicen en representar regiones particulares del espacio de entrada.

A diferencia de otras redes, el SOM no requiere etiquetas durante el proceso de aprendizaje, lo que lo hace especialmente útil en tareas de exploración, visualización y agrupamiento. Su aplicación se ha extendido a múltiples dominios, como la clasificación de imágenes, reconocimiento de patrones, minería de datos y análisis biomédico. El mapa resultante puede visualizarse como una cuadrícula donde cada nodo representa un clúster de datos con características similares, facilitando la interpretación del agrupamiento incluso en casos complejos.

D. Redes neuronales convolucionales (CNN)

Las redes neuronales convolucionales (CNN, por sus siglas en inglés) surgieron como una arquitectura especializada dentro del aprendizaje profundo, diseñada específicamente para el procesamiento de datos con estructura espacial, como las imágenes. Inspiradas en el funcionamiento del córtex visual de los animales, las CNN están compuestas por capas que aplican filtros convolucionales capaces de extraer características jerárquicas de los datos de entrada.

Estas capas detectan patrones simples, como bordes o texturas, en las primeras etapas, y progresivamente identifican estructuras más complejas, como formas u objetos específicos, conforme se profundiza en la red [15].

A diferencia de las redes neuronales tradicionales, las CNN aprovechan propiedades como la compartición de pesos y la reducción de dimensionalidad mediante operaciones de *pooling*, lo que disminuye significativamente la cantidad de parámetros y mejora la eficiencia computacional. Su naturaleza supervisada implica que requieren conjuntos de datos etiquetados para aprender a clasificar o detectar patrones con precisión, pero a cambio ofrecen una gran capacidad de generalización cuando se entrenan adecuadamente. Las CNN se han convertido en la arquitectura predominante para tareas de visión por computadora, incluyendo el diagnóstico médico asistido por imágenes, debido a su alta precisión, adaptabilidad y capacidad para automatizar el reconocimiento de patrones complejos con un rendimiento comparable al de especialistas humanos.

E. MobileNetV2

Es un modelo CNN profundo y ligero de 53 capas con un número reducido de parámetros y un tamaño de entrada de 224×224 [16]. Emplea la técnica de convoluciones separables en profundidad (DSC) para facilitar la portabilidad, y no solo aborda el problema de pérdida de información en capas no lineales dentro de bloques convolucionales mediante cuellos de botella lineales, sino que además incorpora una novedosa estructura llamada Residuos invertidos, diseñada para conservar la información [17]. Este modelo se ha utilizado ampliamente en el análisis de imágenes médicas.

III. ESTRATEGIA

Las estrategias existentes en la literatura varían ampliamente. Algunos enfoques se centran en un preprocesamiento exhaustivo de la imagen, utilizando técnicas como realce y normalización para mejorar la extracción de características con arquitecturas como VGG16 [18, 21, 23, 24]. Otros priorizan la optimización de modelos mediante transferencia de aprendizaje, ajustando capas finales e incorporando regularización [19]. También, [22] plantea una estrategia que consiste en comparar el rendimiento del modelo al procesar imágenes en rayos X en diferentes tamaños y modos de color (RGB vs. escala de grises), e implementar un mecanismo de *early stopping* para reducir el tiempo de entrenamiento. Asimismo, enfoques más avanzados integran módulos de atención para focalizar el análisis en regiones de interés [20] o emplean arquitecturas como ResNet50 para procesar las radiografías de tórax y generar un vector de 2048 características por imagen. Este vector se utiliza como entrada para tres clasificadores distintos: Máquinas de Soporte Vectorial (SVM), Regresión Logística y K-Vecinos más Cercanos (KNN) [25].

La estrategia actual, presentada en la Fig. 2, adopta un enfoque avanzado de aprendizaje por transferencia para la clasificación de imágenes de tórax mediante una red neuronal

convolucional (CNN). El proceso se inicia con las imágenes de rayos X originales, organizadas por su diagnóstico (normal o tuberculosis). En lugar de un preprocesamiento manual, se emplea un generador de datos que estandariza cada imagen redimensionándola a 224x224 píxeles y aplica un robusto aumento de datos en tiempo real, incluyendo rotaciones, zoom y desplazamientos para mejorar la generalización del modelo. El núcleo de esta metodología es la red MobileNetV2, pre-entrenada en ImageNet, que actúa como un extractor de características congelado durante una fase inicial donde solo se entrena un nuevo clasificador personalizado. Posteriormente, se realiza un "ajuste fino" (fine-tuning), descongelando las capas superiores de MobileNetV2 para re-entrenarlas sutilmente con una tasa de aprendizaje baja, adaptando así el modelo a los patrones específicos de las radiografías. Todas las imágenes son normalizadas al rango [0, 1] y divididas automáticamente en conjuntos de entrenamiento y validación para guiar el proceso y evaluar el rendimiento final.

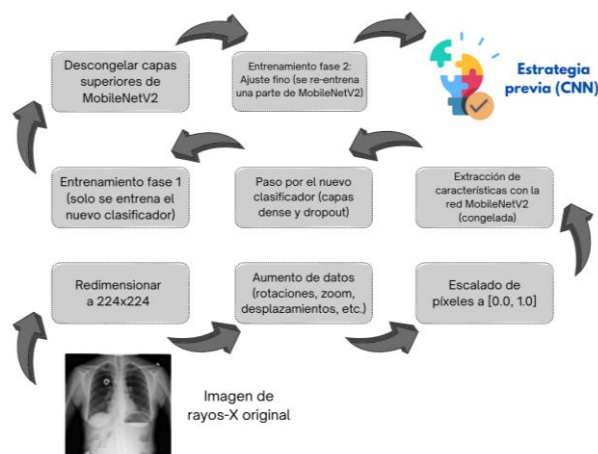


Fig. 2 Estrategia previa a la aplicación de la CNN.

Posteriormente, se implementó la arquitectura de la red neuronal convolucional (CNN) utilizando un enfoque de aprendizaje por transferencia (*transfer learning*). Se adoptó el modelo MobileNetV2, pre-entrenado en la base de datos ImageNet, como un sofisticado extractor de características. La arquitectura se cargó sin su cabezal de clasificación original (*include_top=False*) y sus capas convolucionales se mantuvieron "congeladas" durante la fase inicial del entrenamiento. Esta técnica permite aprovechar el conocimiento profundo del modelo para identificar patrones visuales complejos, como texturas y formas, directamente de las imágenes de tórax, las cuales fueron estandarizadas a un tamaño de entrada de 224x224 píxeles para ser compatibles con la red pre-entrenada.

Sobre la base congelada de MobileNetV2, se construyó un nuevo cabezal de clasificación. Se empleó una capa *GlobalAveragePooling2D* que transforma eficientemente los mapas de características en un vector, seguida de una capa densa de 128 neuronas con activación ReLU y una capa dropout con una tasa de 0.5 para mejorar la generalización. La

clasificación final se realiza mediante una capa de salida con una sola neurona y función de activación sigmoide, optimizada para la tarea binaria (normal o tuberculosis). El entrenamiento se dividió en dos etapas: primero se entrenó únicamente el nuevo cabezal y luego se realizó un ajuste fino (*fine-tuning*), descongelando las capas superiores de MobileNetV2 y continuando el entrenamiento con una tasa de aprendizaje muy baja. Se utilizó el optimizador Adam, la función de pérdida *binary_crossentropy* y se emplearon callbacks como *EarlyStopping* y *ModelCheckpoint* para detener el entrenamiento de forma óptima y guardar el mejor modelo basado en la precisión de validación. En la Fig. 3 se detallan los parámetros de compilación y el proceso de entrenamiento.

```
# Definir los callbacks
# Guardará el mejor modelo basado en la precisión de validación
checkpoint = ModelCheckpoint(
    'modelo_cnn_tbc_mejorado.h5',
    monitor='val_accuracy',
    save_best_only=True,
    mode='max'
)

# Detendrá el entrenamiento si la pérdida de validación no mejora en 3 épocas
early_stopping = EarlyStopping(
    monitor='val_loss',
    patience=5, # Aumentado a 5 para dar más margen
    restore_best_weights=True # Se asegura de que el modelo final tenga los mejores pesos
)

# Definir el número de épocas
epochs = 25 # Con transfer learning, a menudo se necesitan menos épocas

# Entrenar el modelo
history = model.fit(
    train_generator,
    epochs=epochs,
    validation_data=validation_generator,
    callbacks=[checkpoint, early_stopping]
)
```

Fig. 3 Proceso de compilación y entrenamiento del modelo CNN.

La Figura 4 presenta las curvas de aprendizaje del modelo CNN, las cuales son fundamentales para evaluar su robustez y capacidad de generalización. Un indicador clave de un modelo bien entrenado es la convergencia de las métricas de entrenamiento y validación, mientras que una brecha significativa entre ambas señalaría un sobreajuste, donde el modelo memoriza los datos de entrenamiento, pero falla al generalizar a datos nuevos.

Como se observa en el gráfico de precisión (izquierda), la curva de validación (naranja) sigue muy de cerca la trayectoria ascendente de la curva de entrenamiento (azul), estabilizándose ambas en valores superiores al 97% con una mínima diferencia. De forma análoga, el gráfico de pérdida (derecha) muestra que ambas curvas descienden y se mantienen en valores bajos y próximos. Este comportamiento es la evidencia visual de que las estrategias de regularización y el aumento de datos fueron efectivas para prevenir el sobreajuste. En lugar de memorizar, el modelo aprendió patrones visuales generalizables, lo que confirma su robustez y lo hace un candidato fiable para la clasificación de radiografías no vistas previamente.



Fig. 4 Curvas de aprendizaje de precisión y pérdida durante el entrenamiento.

Ahora, para el entrenamiento de la red neuronal SOM se usa una metodología híbrida que está ilustrada en la Fig. 5, y se diseñó para analizar la estructura de los datos mediante la combinación del aprendizaje supervisado de la CNN con el aprendizaje no supervisado del Mapa Autoorganizado (SOM). El flujo de trabajo comienza utilizando la red MobileNetV2, previamente entrenada y ajustada, no como un clasificador final, sino como un potente extractor de características. Cada imagen de rayos X es procesada por la red hasta la capa GlobalAveragePooling2D, lo que resulta en la generación de un vector de características de 1280 dimensiones que encapsula la información visual más relevante. Estos vectores se convierten en la entrada para el entrenamiento de la red neuronal SOM, el cual se realiza en dos fases: primero, una no supervisada donde el mapa aprende a organizar sus neuronas según la similitud de los vectores de entrada, y luego, una supervisada para el etiquetado de las neuronas, asignando a cada una la etiqueta de clase (normal o tuberculosis) que la activó con mayor frecuencia. El resultado final es un mapa SOM etiquetado, una representación visual en 2D que permite analizar la separabilidad de las clases.

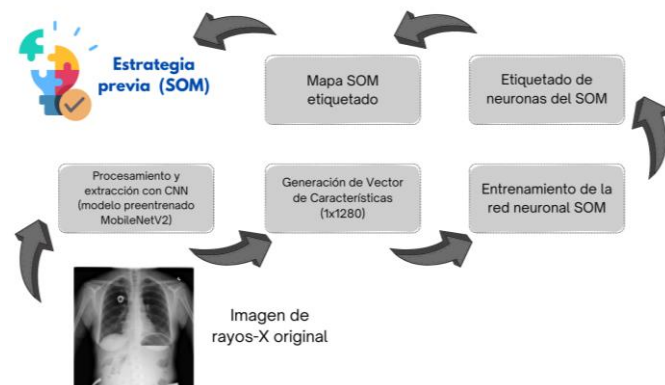


Fig. 5 Estrategia de aprendizaje de imágenes de tórax para SOM.

En la red neuronal SOM utilizada en este trabajo, se configuraron cuidadosamente los parámetros para optimizar el proceso de aprendizaje sobre las características extraídas por la CNN. El entrenamiento se llevó a cabo durante un total de 20,000 iteraciones, un número significativamente mayor para permitir una organización más detallada del mapa. Se

estableció una tasa de aprendizaje inicial (alfa) de 0.5, la cual disminuye progresivamente a lo largo del entrenamiento. Esta tasa de aprendizaje, junto con un radio de vecindad inicial (sigma) de 1.5, controla la magnitud y el alcance de los ajustes en los pesos de las neuronas, permitiendo una convergencia desde una organización global a ajustes locales finos.

El mapa SOM se definió como una rejilla bidimensional de 20x20 neuronas, creando un espacio de 400 unidades para la visualización y clusterización de los datos. Cada imagen, representada como un vector de 1280 características generado por el modelo MobileNetV2, sirvió como entrada a la red. Para actualizar los pesos de las neuronas cercanas a la ganadora (BMU), se empleó una función de vecindad gaussiana, que garantiza una transición suave en el mapa. La distancia euclidiana fue la métrica subyacente utilizada para determinar la neurona ganadora para cada vector de entrada. Finalmente, se fijó una semilla aleatoria (random_seed=42) para asegurar la reproducibilidad total del proceso de entrenamiento y la consistencia de los resultados. La Fig. 6 resume estos hiperparámetros clave que definen la arquitectura y el comportamiento de la red SOM implementada.

```
# 1. Definir los parámetros de la SOM
som_grid_rows = 20
som_grid_cols = 20
input_len = X_train_features.shape[1] # número de características (1280)
iterations = 20000

# 2. Inicializar la SOM
som = MiniSom(x=som_grid_cols, y=som_grid_rows,
              input_len=input_len,
              sigma=1.5,
              learning_rate=0.5,
              neighborhood_function='gaussian',
              random_seed=42)

# 3. Entrenar la SOM (fase no supervisada)
print(f"\n--- Entrenando la SOM con {iterations} iteraciones... ---")
som.random_weights_init(X_train_features)
som.train_random(data=X_train_features, num_iteration=iterations)
print("Entrenamiento de la SOM completado.")

# 4. Etiquetar las neuronas de la SOM (fase supervisada)
print(f"\n--- Etiquetando las neuronas de la SOM... ---")
label_map = som.labels_map(X_train_features, y_train)
class_names = list(train_generator.class_indices.keys())
```

Fig. 6 Parámetros de la red SOM para el proceso de aprendizaje.

En la Fig. 7 se visualiza el Mapa Autoorganizado (SOM) resultante, que proyecta las características de alta dimensión extraídas por la CNN en un mapa topológico 2D. El fondo del gráfico corresponde a la Matriz-U (*Unified Distance Matrix*), donde los colores más oscuros indican regiones de alta similitud (*clusters*) y los más claros representan las fronteras o distancias entre ellos. Sobre este mapa se superponen los marcadores de las clases: círculos rojos para 'Tuberculosis' y cuadrados verdes para 'Normal'. Se observa una clara segregación topológica de las clases: los casos de 'Normal' se agrupan predominantemente en las regiones laterales e inferior del mapa, mientras que los casos de 'Tuberculosis' forman un cluster muy denso en la esquina superior izquierda, con otras agrupaciones en la zona central. Notablemente, las fronteras claras de la Matriz-U coinciden con las zonas de transición entre las dos clases, validando visualmente la capacidad del

modelo para encontrar patrones distintivos y separar los dos diagnósticos.

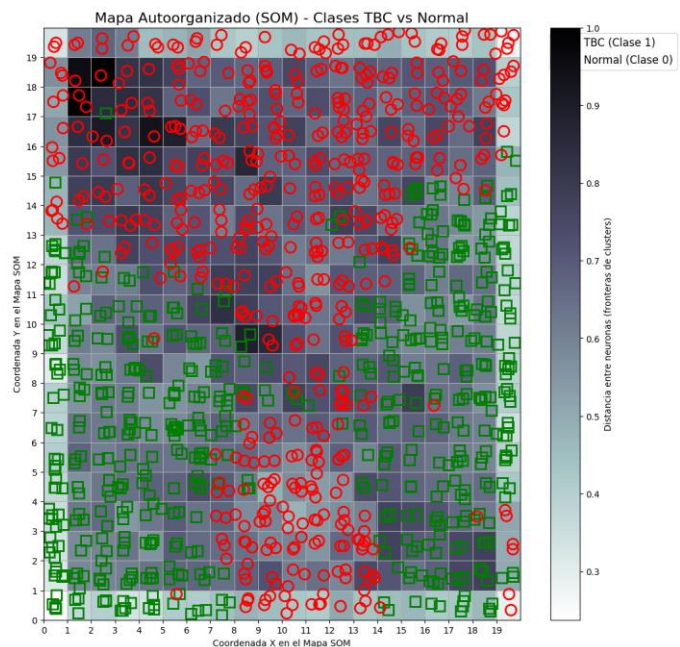


Fig. 7 Visualización de agrupamientos utilizando la matriz de distancia unificada de la red SOM.

IV. DESARROLLO

A. Diseño del sistema

La arquitectura del sistema se ha implementado utilizando tres servicios principales: Google Colab para el entrenamiento de los modelos, Hugging Face para el despliegue y la inferencia, y Google Firestore como base de datos en la nube. El desarrollo del modelo de red neuronal se lleva a cabo en un entorno de Google Colab, aprovechando las librerías TensorFlow y Keras. Como resultado del proceso de entrenamiento y ajuste fino, se genera el archivo .h5 o .pkl, el cual contiene la arquitectura y los pesos aprendidos por el modelo. Este archivo .h5 o .pkl se despliega en la plataforma de Hugging Face para crear un servicio web (API) accesible desde cualquier lugar.

El flujo de operación del sistema se inicia en la aplicación móvil, desarrollada en Android Studio. Cuando un usuario selecciona una radiografía para su análisis, la aplicación realiza dos acciones principales. Primero, envía la imagen en un formato adecuado al servicio desplegado en Hugging Face. Este servicio procesa la imagen con el modelo .h5 o .pkl y devuelve un diagnóstico en formato JSON. Segundo, la aplicación utiliza la librería Retrofit2 para comunicarse con el servicio de Firestore DB, donde almacena tanto la imagen original como el resultado del diagnóstico (predicción, confianza, fecha). Finalmente, la aplicación puede consultar esta base de datos para mostrar al usuario una "Lista de presentación", que sirve como un historial de todos los análisis realizados.

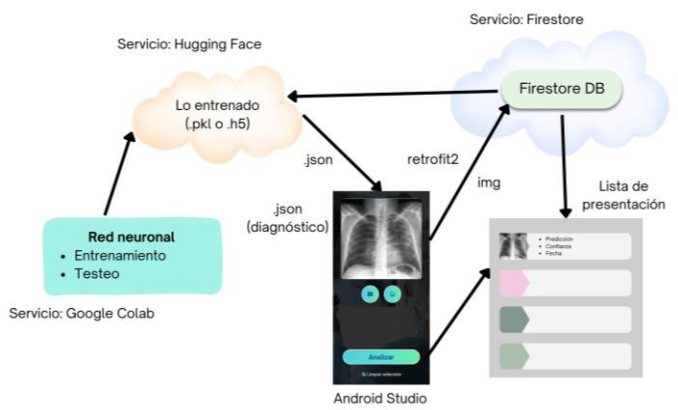


Fig. 8 Diseño backend del sistema.

B. Desarrollo del diseño de requisitos

De acuerdo con la Fig. 9, se muestran los siguientes casos de uso para el aplicativo móvil:

- 1) *Usuario*: Es el personal de salud o el paciente que utilizará la aplicación para obtener un pre-diagnóstico a partir de una radiografía de tórax.
- 2) *Toma fotos*: El aplicativo permite al usuario seleccionar una imagen de una radiografía de tórax desde la galería del dispositivo móvil o capturarla directamente usando la cámara.
- 3) *Realiza diagnóstico*: Esta funcionalidad es el núcleo de la aplicación. Una vez seleccionada la imagen, el usuario inicia el proceso de análisis, que envía la radiografía a un servicio en la nube para su procesamiento y espera la recepción del resultado.
- 4) *Diagnóstico con la RN*: El modelo de red neuronal (en formato .h5), desplegado en un servicio de Hugging Face, recibe la imagen enviada por el aplicativo. El modelo procesa la imagen, realiza la inferencia para determinar si es un caso 'Normal' o de 'Tuberculosis' y devuelve el diagnóstico a la aplicación móvil.

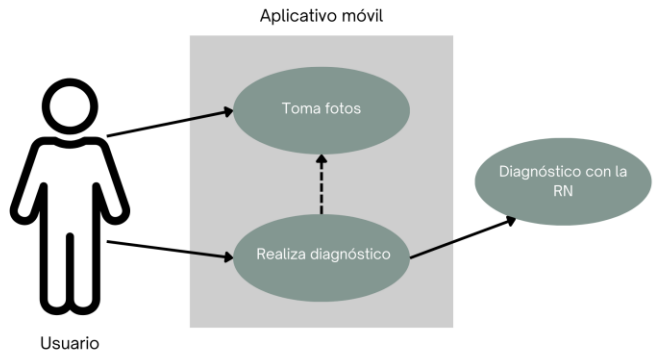


Fig. 9 Funcionalidad del sistema con el usuario

C. Consideraciones de despliegue, escalabilidad y privacidad

El diseño de la arquitectura del sistema (Fig. 8) fue concebido para un despliegue práctico y escalable en entornos

con recursos limitados, con especial atención a la privacidad de los datos.

1) *Despliegue y escalabilidad*: La arquitectura cliente-servidor, que desacopla la aplicación móvil del modelo de IA, ofrece ventajas significativas. El modelo, alojado en Hugging Face, puede ser actualizado o reemplazado de forma centralizada sin necesidad de que los usuarios actualicen la aplicación. Tanto Hugging Face como Google Firestore son servicios en la nube gestionados que escalan automáticamente, garantizando la disponibilidad del servicio incluso con un aumento en el número de usuarios.

2) *Privacidad y seguridad de los datos*: La aplicación móvil no recopila ni transmite ninguna información de identificación personal (IIP) junto con la radiografía. La imagen se envía a la API para su análisis y el resultado se almacena en Firestore asociado a un identificador de usuario anónimo generado por la propia aplicación. Para una implementación clínica real, el sistema debería integrarse con los protocolos de seguridad existentes y cumplir con las regulaciones locales de protección de datos de salud. Las imágenes y diagnósticos se manejarían dentro de una red segura y con el consentimiento explícito del paciente.

V. RESULTADOS

A. Análisis comparativo y selección del modelo final

Para determinar el modelo más efectivo para la implementación final en el aplicativo móvil, se realizó una evaluación comparativa del rendimiento entre la Red Neuronal Convolutacional (CNN) con ajuste fino y el Mapa Autoorganizado (SOM). La selección se basó en un análisis detallado de los reportes de clasificación y las matrices de confusión generadas a partir del mismo conjunto de datos de validación.

El modelo CNN con ajuste fino (Fig. 10) demostró una superioridad notable, alcanzando una precisión (*accuracy*) general del 98%. Este resultado indica una alta fiabilidad y sugiere que el rendimiento del modelo es consistentemente alto en datos no vistos. El análisis detallado por clase muestra un excelente balance, con un F1-score de 0.98 tanto para la clase 'normal' como para 'tbc', reforzando su capacidad para identificar correctamente ambas condiciones. La matriz de confusión (Fig. 10) corrobora este rendimiento, mostrando solo 6 errores de clasificación en total, de los cuales únicamente 5 fueron falsos negativos, un número críticamente bajo en un contexto de diagnóstico. En contraste, el modelo SOM (Fig. 11), aunque competente, obtuvo una precisión general inferior del 95%, cometiendo un total de 14 errores. De manera crítica, el SOM produjo 9 falsos negativos, un número significativamente mayor que el de la CNN. En un contexto de diagnóstico médico, minimizar los falsos negativos es de suma importancia, ya que un diagnóstico erróneo de "normal" para un paciente con tuberculosis puede tener graves consecuencias. Por lo tanto, debido a su mayor precisión, su menor tasa de error general y, fundamentalmente, su mayor fiabilidad para detectar correctamente los casos

positivos, se seleccionó el modelo CNN con ajuste fino como la red neuronal definitiva para ser desplegada en el servicio de diagnóstico.

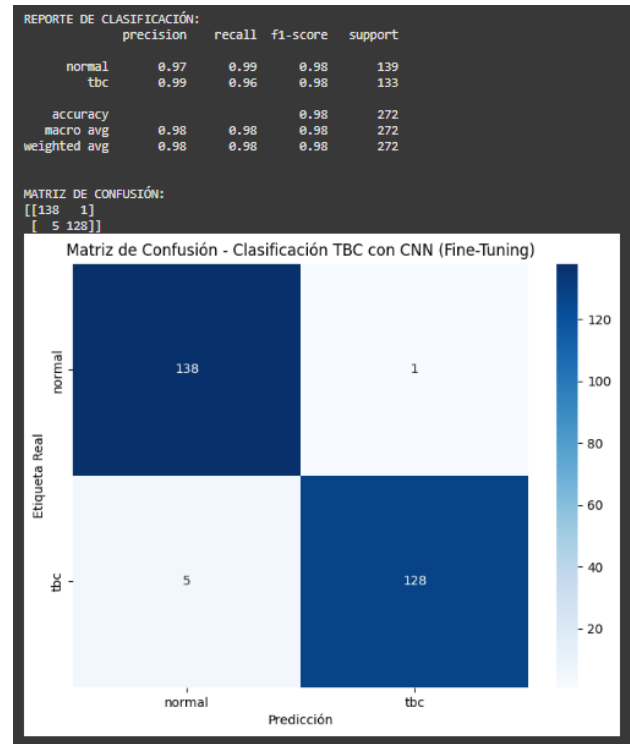


Fig. 10 Reporte de clasificación de la CNN.

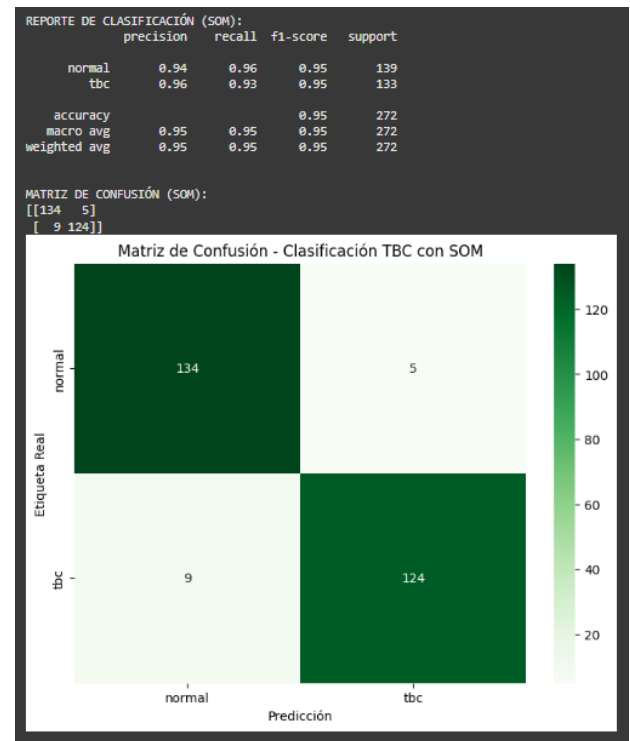


Fig. 11 Reporte de clasificación de SOM.

B. Frontend del aplicativo móvil

El diseño de la interfaz de usuario, desarrollado en Android Studio con Kotlin, guía al usuario a través de un flujo intuitivo en tres pasos. Primero, una pantalla de bienvenida introduce la misión de la aplicación de forma concisa. La única acción disponible es un botón (Button) "Explorar imagen", que actúa como el punto de entrada principal, invitando al usuario a comenzar el proceso de análisis sin distracciones. Al pulsar el botón de inicio, el usuario accede a la pantalla de selección (Fig. 12a). Aquí puede elegir una radiografía de dos maneras: utilizando el icono de cámara para tomar una foto en el momento o el icono de galería para seleccionar una imagen existente en el dispositivo. La imagen seleccionada se muestra en un visor central para su confirmación. Una vez que el usuario está satisfecho con su selección, el botón "Analizar" envía la imagen al servicio en la nube donde reside el modelo de red neuronal para su evaluación. Un botón secundario de "Limpiar selección" ofrece la flexibilidad de cambiar la imagen antes del análisis.

Tras el procesamiento, la aplicación presenta el "Informe de Diagnóstico" (Fig. 12b), una pantalla diseñada para comunicar el resultado de forma inmediata y comprensible. Muestra la imagen analizada, el diagnóstico final ("NORMAL" o "TBC") con una tipografía grande y clara, y el "Nivel de Confianza" del modelo en un gráfico circular prominente. Esta visualización simplifica la interpretación del resultado. En la parte inferior, un enlace de "Ver historial de predicciones" permite al usuario navegar a un registro de todos los análisis realizados.

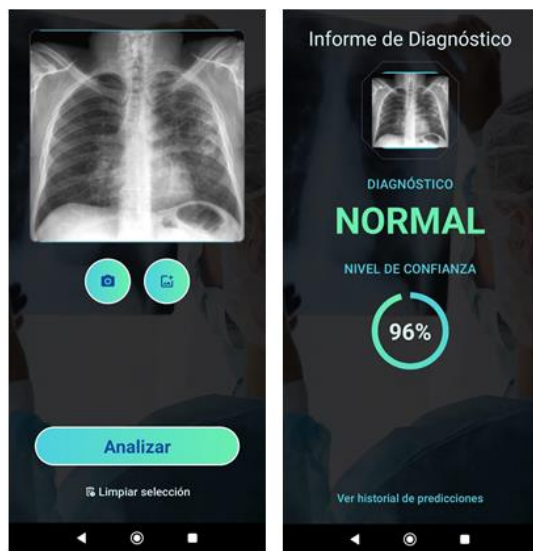


Fig. 12. Interfaz de la aplicación móvil: (a) Pantalla de análisis de radiografía; (b) Pantalla de diagnóstico de radiografía.

Luego, una posterior pantalla funciona como un registro digital de los diagnósticos. Utiliza una lista vertical donde cada elemento representa un análisis previo. Cada entrada en la lista muestra de forma compacta la miniatura de la radiografía, el resultado obtenido, el porcentaje de confianza y

la fecha y hora exactas del análisis. Esto permite al usuario llevar un seguimiento de sus pruebas y consultar resultados anteriores de manera organizada y eficiente.

C. Funcionamiento de los servicios del backend

Para hacer que el modelo de red neuronal sea accesible a la aplicación móvil, se utilizó la plataforma de Hugging Face. Los archivos del modelo de la red convolucional entrenado (modelo_cnn_tbc_mejorado.h5 y modelo_cnn_tbc_mejorado.json) se subió a este servicio para crear un punto de acceso API (Application Programming Interface). Hugging Face se encarga de gestionar la infraestructura del servidor, permitiendo que la aplicación envíe una imagen y reciba una predicción en formato JSON. Este enfoque desacopla el modelo de la aplicación, facilitando futuras actualizaciones del modelo sin necesidad de modificar el código del aplicativo móvil.

La persistencia de los datos del sistema se gestiona a través de Google Firebase Firestore, una base de datos NoSQL en la nube. Cada vez que se realiza un diagnóstico, la aplicación móvil almacena un registro en Firestore que incluye la imagen analizada, el resultado de la predicción (normal o TBC), el nivel de confianza y la fecha del análisis. Este servicio no solo actúa como un backend robusto y escalable, sino que también permite la sincronización de datos en tiempo real, asegurando que el historial de predicciones del usuario esté siempre actualizado y accesible desde el aplicativo.

VI. CONCLUSIONES

El presente estudio propuso e implementó un sistema de diagnóstico asistido por inteligencia artificial para la detección de tuberculosis en radiografías de tórax, integrando técnicas de aprendizaje profundo y aprendizaje no supervisado. Se evaluaron dos enfoques principales: una red neuronal convolucional (CNN) con ajuste fino, basada en la arquitectura MobileNetV2 y un Mapa Autoorganizado (SOM) entrenado con vectores de características extraídos por la misma red.

Los resultados experimentales evidenciaron que el modelo CNN con ajuste fino alcanzó un rendimiento significativamente superior, logrando una precisión del 98 % y una tasa de falsos negativos considerablemente menor en comparación con el SOM. Debido a su mayor capacidad de generalización y fiabilidad diagnóstica, dicho modelo fue seleccionado como la opción final para ser integrada en el sistema de clasificación automatizada. En contraste, el SOM, que si bien mostró una precisión general aceptable, demostró limitaciones en la detección de casos positivos, lo cual representa un riesgo en contextos médicos.

Finalmente, se implementó un sistema funcional que integra el modelo CNN en una aplicación móvil respaldada por servicios en la nube, utilizando Hugging Face para la inferencia y Google Firestore para el almacenamiento persistente de resultados. Esta arquitectura demuestra la viabilidad de desplegar soluciones de diagnóstico inteligente accesibles con potencial de aplicación en entornos clínicos de

bajo recurso para apoyar el diagnóstico temprano de la tuberculosis.

VII. DISCUSIÓN

Los hallazgos obtenidos en esta investigación evidencian la alta eficacia del modelo CNN basado en MobileNetV2 para la detección de tuberculosis en radiografías de tórax, alcanzando una precisión del 98 %. Esta cifra supera los resultados reportados en estudios previos, donde se han utilizado tanto modelos entrenados desde cero como arquitecturas profundas más complejas. Por ejemplo, Sarawagi et al. en [26] desarrollaron una red neuronal convolucional sin utilizar pesos preentrenados, alcanzando una precisión del 96.57 %. Aunque este enfoque es eficiente en recursos, la transferencia de aprendizaje con ajuste fino mejora el rendimiento diagnóstico y reduce los falsos negativos.

De manera similar, Hossain et al. en [27] realizaron una comparación entre diversas arquitecturas profundas para el mismo fin de detección de TBC, incluyendo InceptionV3, Xception, DenseNet y una CNN personalizada, identificando al modelo Xception como el de mayor precisión general (96.95 %), con métricas de precisión, recall y F1 de 97 %, 96 % y 97 % respectivamente. En contraste, la arquitectura MobileNetV2 utilizada en este estudio no solo alcanza una precisión superior, sino que lo hace mediante un diseño optimizado para dispositivos móviles, lo que amplía su aplicabilidad en entornos clínicos con recursos limitados.

AGRADECIMIENTO

Agradecimiento a todos los participantes de la presente investigación, cuyo aporte permitió desarrollar un sistema de diagnóstico automatizado para la detección de tuberculosis en radiografías de tórax basado en inteligencia artificial.

REFERENCIAS

- [1] W. H. Curioso and M. J. Brunette, "Inteligencia artificial e innovación para optimizar el proceso de diagnóstico de la tuberculosis," *Rev. Peru. Med. Exp. Salud Pública*, vol. 37, pp. 554–558, 2020.
- [2] O. F. B. Leiva, H. S. G. Aristizábal, and J. Z. A. Delgado, "Aplicación de técnicas de inteligencia artificial para la detección de tuberculosis pulmonar en Colombia," *Rev. EIA*, vol. 20, no. 39, pp. 3909–pp., 2023.
- [3] J. E. Polanco-Pasaje et al., "Cascada de atención de la tuberculosis para la población indígena en Colombia: una investigación operativa," *Rev. Panam. Salud Pública*, vol. 44, p. e150, 2020.
- [4] D. C. Falcon Córdova, *Tuberculosis en el mundo y en el Ecuador, en la actualidad (2021)*. 2024.
- [5] J. A. Bedón Echeverría, Impacto de la pobreza en la incidencia de TB en zonas rurales de Ecuador: un estudio en la provincia de Cotopaxi, cantón Pujilí, parroquia Pilalo en las zonas rurales de Apagua, Corralpungo, Redrovan, Conucto, Milin y Chilca, Master's thesis, Universidad de las Américas, Quito, Ecuador, 2025.
- [6] A. G. Caba et al., "Inteligencia artificial en el diagnóstico histopatológico de microorganismos," *Rev. Repert. Med. Cir.*, vol. 33, no. 3, pp. 230–237, 2024.
- [7] A. A. C. Sánchez, M. A. S. Chavez, and E. I. P. Ojeda, "Wine quality classification strategy using the SOM neural network based on chemical analysis," in *Proc. 21st LACCEI Int. Multi-Conf. Eng., Educ. Technol.*, Buenos Aires, Argentina, Jul. 2023.
- [8] V. Manokar, R. Balaji, V. M. Nandhini, S. Shankar, and L. M. Patnaik, "Wavelet Decomposition and Classification of Diseased fMRI Brain Images Using Self Organized Maps," in *Proc. 6th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Coimbatore, India, Mar. 2020, pp. 1342–1348.
- [9] K. C. Santosh, S. Vajda, S. Antani, and G. Thoma, "Automatic Pulmonary Abnormality Screening Using Thoracic Edge Map," in *Proceedings of the 28th IEEE International Symposium on Computer-Based Medical Systems (CBMS)*, Sao Carlos, Brazil, Jun. 2015, pp. 360–361.
- [10] A. R. Ismail, A. S. M. Khalili, N. F. A. Rahim, and S. Q. Nisa, "Deep convolutional generative adversarial networks for imbalance medical image classification," *Int. J. Pers. Comput. Commun. (IJPC)*, vol. 9, no. 2, pp. 98–103, Jul. 2023.
- [11] T. M. Zihan, A. Zami, M. Ahmed, R. Islam, and F. Ahmed, "From pixels to diagnosis: convolutional neural networks in tuberculosis screening," in *Proc. 6th Int. Conf. Electr. Eng. Inf. Commun. Technol. (ICEEICT)*, Dhaka, Bangladesh, May 2024, pp. 1187–1192.
- [12] K. Agarwal, "Training AI to detect Tuberculosis in chest X-rays," *Am. J. Sci. Res. (AJSR)*, vol. 2, no. 4, pp. 175–182, 2024.
- [13] M. Shahid, M. A. Hassan, F. Iqbal, A. Altaf, S. W. H. Shah, A. V. Elizaincin, and I. Ashraf, "Enhancing movie recommendations using quantum support vector machine (QSVM)," *The Journal of Supercomputing*, vol. 81, no. 1, pp. 78, 2025.
- [14] T. Kohonen, *Self-Organizing Maps*. Berlin: Springer, 1989.
- [15] J. Wu, *Introduction to Convolutional Neural Networks*, National Key Lab for Novel Software Technology, Nanjing University, China, vol. 5, no. 23, p. 495, 2017.
- [16] Y. Chandola, J. Virmani, H. S. Bhadauria, and P. Kumar, "Lightweight end-to-end pre-trained CNN-based computer-aided classification system design for chest radiographs," in *Deep Learning for Chest Radiographs*. Elsevier, 2021, pp. 167–183.
- [17] K. Dong, C. Zhou, Y. Ruan, and Y. Li, "MobileNetV2 model for image classification," in *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*, Guangzhou, China: IEEE, dic. 2020, pp. 476–480.
- [18] S. Chandrasekaran, T. R. Mahesh, S. B. Khan, S. Palaiahnakote, and S. Alzahrani, "A Hybrid Model for Classification of Tuberculosis Chest X-Rays Images," *Malaysian Journal of Computer Science*, vol. 37, no. 3, pp. 193–204, 2024.
- [19] S. Soam, R. Singh, M. Manglik and S. Sharma, "Comparative Study of Keras CNNs for Tuberculosis Detection from Chest X-rays," *International Journal of Scientific Research in Engineering and Management (IJSREM)*, vol. 8, no. 5, pp. 1–7, May 2024.
- [20] T. Xu and Z. Yuan, "Convolution neural network with coordinate attention for the automatic detection of pulmonary tuberculosis images on chest x-rays," *IEEE Access*, vol. 10, pp. 86710–86717, 2022.
- [21] K. Sarawagi, A. Pagrotra, H. Dhiman, and N. Singh, "Self-trained convolutional neural network (CNN) for tuberculosis diagnosis in medical imaging," *Cureus*, vol. 16, no. 6, 2024.
- [22] R. Kurniawan, T. Badriyah, and I. Syarif, "Tuberculosis Detection based on Lung X-Ray Images Using Convolutional Neural Networks (CNN)," *Applied Technology and Computing Science Journal*, vol. 7, no. 1, pp. 14–22, 2024.
- [23] F. N. Promy et al., "Chest X-ray Image Classification Using Convolutional Neural Network to Identify Tuberculosis," in *Int. Conf. on Intelligent Computing & Optimization*, Cham, Switzerland: Springer Nature, 2023, pp. 127–139.
- [24] M. Norval, Z. Wang, and Y. Sun, "Pulmonary tuberculosis detection using deep learning convolutional neural networks," in *Proc. 3rd Int. Conf. Video and Image Processing*, Dec. 2019, pp. 47–51.
- [25] M. Alsaffar et al., "Detection of tuberculosis disease using image processing technique," *Mobile Information Systems*, vol. 2021, no. 1, Art. no. 7424836, 2021.
- [26] K. Sarawagi, A. Pagrotra, H. Dhiman, and N. Singh, "Self-Trained Convolutional Neural Network (CNN) for Tuberculosis Diagnosis in Medical Imaging," *Cureus*, Jun. 2024.
- [27] E. Polisetty, S. Sayeed, O. R. Duttala, and N. Senthamarai, "Detection and Classification of Tuberculosis Disease Using Hybrid Deep Learning Method," in *2024 International Conference on Electronic Systems and Intelligent Computing (ICESIC)*, Chennai, India: IEEE, nov. 2024, pp. 162–167.