

Detección de roya en hojas de café mediante aprendizaje maquina

Sergio Nicolás Siabatto Cleves
Universidad Nacional de Colombia
Bogotá, Colombia
ssiabatto@unal.edu.co

Resumen — La roya del café (*Hemileia vastatrix*) es una de las principales amenazas para la producción de *Coffea Arabica* en Colombia, impactando negativamente la calidad y cantidad de la cosecha. Este proyecto busca desarrollar un software basado en técnicas de deep learning que permita detectar la presencia de la enfermedad a partir de imágenes de las hojas de café.

Se recopilará un conjunto de datos representativo de imágenes de hojas afectadas y saludables en diversas etapas de desarrollo de la enfermedad. Este conjunto será utilizado para entrenar y validar un modelo de redes neuronales profundas, con enfoque en arquitecturas como CNNs (Convolutional Neural Networks).

El software se implementará con el objetivo de mejorar la detección temprana y mitigar el impacto de la roya en la producción cafetera.

Abstract — Coffee leaf rust (*Hemileia vastatrix*) is one of the main threats to the production of *Coffea arabica* in Colombia, negatively affecting both the quality and quantity of coffee yields. This project aims to develop software based on deep learning techniques to detect the presence of this disease using images of coffee leaves.

A representative dataset of images, including both affected and healthy leaves at various stages of disease progression, will be collected. This dataset will be used to train and validate a deep neural network model, focusing on architectures such as Convolutional Neural Networks (CNNs).

The software will be implemented to enhance early detection and mitigate the impact of leaf rust on coffee production.

Palabras Clave — Roya del café, Deep learning, Aprendizaje Maquina, Producción de calidad.

I. INTRODUCCIÓN

La producción de café, específicamente de la variedad *Coffea arabica*, tiene una importancia crítica tanto económica como social en Colombia, donde representa una fuente principal de ingresos para cientos de miles de familias rurales. Sin embargo, la roya del café, causada por el hongo *Hemileia vastatrix*, constituye una de las mayores amenazas para esta industria. Este patógeno afecta el vigor de las plantas y reduce significativamente su productividad, en lugar de matarlas [2].

El ciclo de vida de *H. vastatrix* está estrechamente ligado a factores climáticos como la humedad y la temperatura. La germinación de las esporas y la formación del apresorio, etapas críticas para la infección, requieren agua líquida y temperaturas específicas [2]. Estas condiciones, exacerbadas por los cambios climáticos globales, han favorecido la expansión de la roya a nuevas regiones, afectando tanto a

pequeños como grandes productores [4].

El impacto ecológico de la roya del café es también profundo, ya que su propagación altera los ecosistemas en los que se encuentra, la dinámica ecológica de la roya del café interactúa con los sistemas agrícolas y climáticos, contribuyendo a la propagación de la enfermedad a través de diversas regiones geográficas y afectando la biodiversidad agrícola [3].

En respuesta a este desafío, el presente proyecto propone el desarrollo de un software basado en técnicas de deep learning para la detección temprana de la roya del café a través de imágenes digitales de hojas afectadas. La inteligencia artificial, en particular las arquitecturas de redes neuronales convolucionales (CNNs), ha demostrado ser altamente efectiva para abordar problemas de clasificación y diagnóstico en tiempo real [5].

Por otra parte, se reconoce la importancia de implementar estrategias integrales que combinen tecnología y prácticas sostenibles. Por ejemplo, el uso de métodos naturales para el control de la roya, tales como compuestos bioactivos y biochar, no solo mejora la salud del cultivo, sino que también permite a los productores adoptar prácticas más amigables con el medio ambiente [1].

Este enfoque combina conocimiento científico y tecnológico para abordar uno de los problemas más urgentes en la industria cafetera, fortaleciendo la capacidad de resiliencia de los pequeños productores y mitigando los efectos negativos en la economía cafetera colombiana.

II. REVISIÓN DE LITERATURA / COMPARACIÓN DE METODOLOGÍAS

Se presenta una compilación de 10 artículos que contemplan un tema similar, de estos artículos se realiza la extracción de su metodología, vamos a centrarnos en 3 elementos clave:

- Conjunto de datos y preprocesamiento.
- Arquitectura de CNN
- Entrenamiento y validación.

1. A Coffee Leaf Disease Detection via Deep Learning [6]

- Conjunto de datos y preprocesamiento: El entrenamiento de deep learning para la detección de enfermedades en hojas

implica etiquetar conjuntos de datos de imágenes de hojas y luego usar las redes para reconocer los síntomas y patrones de la enfermedad. Las redes neuronales profundas pueden alcanzar un nivel incomparable de clasificación de enfermedades de hojas y, muchas veces, superan a los algoritmos tradicionales de machine learning debido a su capacidad para aprender características complejas sin supervisión humana. En los estudios comparativos, se utilizaron varios conjuntos de datos, como el Arabica dataset, Leaf image dataset, Plant village dataset, y Image Net dataset, aplicando técnicas de preprocesamiento como filtros gaussianos y de promedio, mini-batch gradient descent, y aumentación de datos mediante espejado y recortes aleatorios.

- **Arquitectura de CNN:** Se describen modelos específicos de deep learning para identificar enfermedades en hojas de café y algunas variaciones de las CNN como RegNet, MobileNet, Google Net y Efficient Net. Estas arquitecturas personalizadas cumplen la función importante de desarrollar modelos más precisos y robustos para la detección de enfermedades en hojas de café. En comparación con las redes neuronales feed-forward convencionales, las CNN requieren un menor número de neuronas y parámetros hiper. En la investigación actual, se recomiendan modelos preentrenados como RegNet, MobileNet, Google Net y Efficient Net.
- **Entrenamiento y validación:** El entrenamiento se realizó utilizando modelos preentrenados en ImageNet, aplicando técnicas de fine-tuning y regularización. Se congelaron las capas preentrenadas inicialmente y se entrenaron nuevas capas añadidas, seguidas de un ajuste fino de las capas preentrenadas. Se emplearon early stopping y L2 regularization para prevenir el sobreajuste, y el optimizador Adam con una tasa de aprendizaje de 0.00001. La evaluación se realizó utilizando precisión, recall, F1-score y exactitud. Los resultados de validación mostraron que MobileNet obtuvo un rendimiento superior con una precisión del 99.93%, superando a otros modelos como DenseNet y AlexNet.

2. Coffee Leaf Disease Classification by Using a Hybrid Deep Convolution Neural Network [7]

- **El conjunto de datos utilizado en este estudio es el Plant Village dataset de Kaggle, que contiene 3076 imágenes de hojas de café saludables y enfermas, clasificadas en cuatro grupos: hojas de café saludables, roya del café, podredumbre negra y mancha marrón. Las imágenes se preprocesaron estandarizando su tamaño a 244×244 píxeles. El conjunto de datos se dividió en un 80% para entrenamiento y un 20% para prueba, con un subconjunto de validación separado para ajustar los hiperparámetros y prevenir el sobreajuste.**
- **Arquitectura de CNN:** Se utilizaron las arquitecturas de CNN DenseNet121 e InceptionV3, ambas preentrenadas. DenseNet121 incluye una capa convolucional inicial seguida de cuatro bloques densos y tres capas de transición,

mientras que InceptionV3 emplea módulos de inepción con diversos tamaños de filtro para capturar información en diferentes escalas. Ambas arquitecturas se combinaron en un modelo propuesto que incluye capas totalmente conectadas y una capa de pooling promedio, con salidas concatenadas que pasan por varias capas conectadas y culminan en una capa de clasificación softmax.

- **Entrenamiento y validación:** El entrenamiento se realizó utilizando pesos preentrenados de DenseNet121 e InceptionV3, con capas adicionales totalmente conectadas y una función de pérdida de entropía cruzada jerárquica aplicada a las capas softmax. Se emplearon técnicas de fine-tuning y una capa de preprocesamiento para la extracción de características. El modelo se evaluó utilizando la biblioteca Keras, con un 80% de los datos para entrenamiento y un 20% para prueba, logrando una alta precisión en la clasificación de las enfermedades de las hojas de café.

3. Interactive effects of altitude, microclimate and shading system on coffee leaf rust [8]

- **Conjunto de datos y preprocesamiento:** El estudio se realizó en tres distritos productores de café Arábica en el área del Monte Elgon en el este de Uganda. Se seleccionaron 49 parcelas de café, clasificadas en tres sistemas de sombra: sistema de café-plátano (CB), sistema de café abierto (CO) y sistema de café-árbol (CT). En cada parcela, se seleccionaron sistemáticamente nueve arbustos de café y se evaluaron las hojas sanas y enfermas en intervalos de aproximadamente seis semanas durante la temporada de crecimiento de 2015. Se instalaron registradores de datos de temperatura y humedad relativa en un subconjunto de 27 parcelas.
- **Arquitectura de CNN:** El paper no menciona el uso de una arquitectura de CNN específica para el análisis de los datos. En su lugar, se utilizó un modelo de ecuaciones estructurales (SEM) para inferir los efectos directos e indirectos de la altitud y el sistema de sombra del café sobre la incidencia máxima de la roya del café (CLR) a través de indicadores microclimáticos.

- **Entrenamiento y validación:** El modelo SEM se construyó basándose en un modelo conceptual a priori y en los resultados de la selección de variables microclimáticas potenciales. Cada variable de respuesta, incluidas las variables microclimáticas y CLR, se ajustó como modelos lineales o lineales generalizados. Los mejores modelos se seleccionaron mediante el criterio de información de Akaike (AIC) y se combinaron en una lista de ecuaciones para estimar los efectos directos e indirectos dentro del sistema. El ajuste general del modelo se evaluó utilizando la estadística C de Fisher.

4. Modelling coffee leaf rust risk in Colombia [2]

- **Conjunto de datos y preprocesamiento:** El estudio se centró en la región de interés (ROI) que incluye Colombia y países

vecinos, con un enfoque en las áreas productoras de café. Se utilizaron datos de temperatura y humedad de la reanálisis climático japonés JRA-55, que proporciona estimaciones horarias de temperatura y humedad del dosel desde 1990 hasta 2015. Los datos se interpolaron linealmente a estimaciones horarias y se utilizaron para modelar el riesgo de infección de la roya del café (CLR) en función de la humedad de las hojas y la temperatura.

- **Arquitectura de CNN:** No se utilizó una arquitectura de CNN en este estudio. En su lugar, se empleó un modelo de riesgo de enfermedad basado en la duración de la humedad de las hojas (LWD) y la temperatura, utilizando funciones de respuesta de temperatura y distribuciones de Weibull para modelar la germinación de esporas y la formación de apresorios de *Hemileia vastatrix*.
- **Entrenamiento y validación:** El modelo de riesgo de enfermedad se parametrizó utilizando datos experimentales y se ejecutó en el entorno de programación R. Se calcularon los números relativos de esporas infectantes al final de cada período húmedo y se sumaron para obtener un puntaje de riesgo de infección diario promedio por mes. Se analizaron las variaciones temporales en las variables climáticas y el riesgo de enfermedad mediante la descomposición estacional de series temporales (STL) y se probaron tendencias temporales significativas utilizando un modelo de mínimos cuadrados generalizados con un modelo de error de autocorrelación esférica.

5. The coffee rust crises in Colombia and Central America, 2008–2013, impacts, plausible causes and proposed solutions [9]

- **Conjunto de datos y preprocesamiento:** El estudio se centró en las epidemias de roya del café en Colombia y Mesoamérica entre 2008 y 2013, analizando factores económicos y meteorológicos. Se recopilaron datos sobre la distribución de la lluvia, la duración de la luz solar y la variabilidad de la temperatura diurna, que se consideraron cruciales para el desarrollo de la epidemia. Además, se observaron las prácticas de manejo del cultivo, como la aplicación de fertilizantes y fungicidas, y su impacto en la severidad de la enfermedad.
- **Arquitectura de CNN:** No se utilizó una arquitectura de CNN en este estudio. En su lugar, se emplearon análisis estadísticos y modelos de riesgo de enfermedad para evaluar la influencia de factores meteorológicos y de manejo del cultivo en la propagación de la roya del café. Se consideraron variables como la carga de frutos, la altitud y la gestión de la sombra en las plantaciones de café.
- **Entrenamiento y validación:** El estudio utilizó datos históricos y observaciones de campo para modelar la incidencia de la roya del café. Se aplicaron modelos estadísticos para identificar las relaciones entre las variables meteorológicas y la severidad de la enfermedad. Además, se implementaron sistemas de monitoreo y advertencia temprana para detectar aumentos inusuales en

la incidencia de la roya del café y promover medidas de control. Se realizaron encuestas y análisis de datos para validar los hallazgos y ajustar las estrategias de manejo de la enfermedad.

6. Coffee Leaf Rust Disease Detection and Implementation of an Edge Device for Pruning Infected Leaves via Deep Learning Algorithms [10]

- **Conjunto de datos y preprocesamiento:** El conjunto de datos fue recopilado en una finca cafetera de Majalengka, Indonesia, utilizando dos dispositivos: una cámara digital mirrorless (Canon EOS M3) y una cámara de teléfono móvil (Xiaomi Redmi Note 10). Se recopilaron 2024 imágenes, que se dividieron en un 70% para entrenamiento (1417 imágenes), 20% para validación (405 imágenes) y 10% para prueba (202 imágenes). Se realizó aumento de datos (rotación, volteo, escalado) para mejorar la diversidad del conjunto. Las imágenes etiquetadas como “CLR” (coffee leaf rust) se anotaron usando la herramienta LabelImg, enfocándose en las hojas más afectadas por roya. diversidad del conjunto de datos y reducir el sobreajuste.
- **Arquitectura de CNN:** Se utilizó la arquitectura YOLO (You Only Look Once) en dos versiones, YOLOv5 y YOLOv8, para detección de objetos. YOLOv8 mostró mejoras significativas sobre YOLOv5 en métricas como mAP (precisión media promedio) y recall. El modelo procesa imágenes en cuadrículas $S \times S$, prediciendo múltiples cajas delimitadoras y puntuaciones de confianza.
- **Entrenamiento y validación:** Se entrenaron ambos modelos con 100 épocas, tamaño de lote 16 y tamaño de entrada de 416×416 píxeles. El proceso de entrenamiento tomó 5 horas para YOLOv5 y 6 horas para YOLOv8. Métricas principales:
- YOLOv5: mAP = 69.9%, recall = 63.4%.
- YOLOv8: mAP = 73.2%, recall = 65.9%.

El modelo entrenado se probó también con un dispositivo edge (NVIDIA Jetson Nano) para detección en tiempo real con imágenes de prueba adicionales capturadas en el campo.

7. Enhancing crop productivity with fined-tuned deep convolution neural network for Potato leaf disease detection [11]

- **Conjunto de datos y preprocesamiento:** El conjunto de datos utilizado en este estudio es el Potato Leaf Disease Dataset, que contiene 3076 imágenes organizadas en siete clases: Bacteria, Fungi, Nematode, Pest, Phytophthora, Virus y Healthy. Las imágenes fueron capturadas en un entorno no controlado en varias granjas de papa en Java Central, Indonesia, utilizando múltiples cámaras de teléfonos inteligentes. Las imágenes se dividieron en subconjuntos de entrenamiento (70%), validación (20%) y prueba (10%). Las imágenes se redimensionaron y

normalizaron para mantener la consistencia y facilitar la convergencia del modelo. Se aplicaron técnicas de aumento de datos para abordar el desequilibrio de clases y mejorar la generalización del modelo.

- **Arquitectura de CNN:** En esta investigación se emplearon tres arquitecturas de CNN: DenseNet201, ResNet152V2 y NasNetMobile. DenseNet201 se seleccionó por su patrón de conectividad densa que facilita la reutilización de características, ResNet152V2 por su arquitectura profunda con conexiones residuales que permiten capturar detalles intrincados, y NasNetMobile por su eficiencia y adaptabilidad, ideal para dispositivos móviles y entornos con recursos limitados. Estas arquitecturas fueron elegidas por su eficacia probada en tareas similares y su capacidad para manejar la variabilidad inherente del conjunto de datos. Estas características son particularmente ventajosas en conjuntos de datos con imágenes complejas y diversas como el nuestro. Las conexiones densas del modelo ayudan a reducir el número de parámetros mientras mantienen un alto rendimiento, lo que lo convierte en una excelente opción para mejorar los resultados en conjuntos de datos más nuevos y amplios.
- **Entrenamiento y validación:** El entrenamiento se realizó utilizando aprendizaje por transferencia, donde los modelos preentrenados en el conjunto de datos ImageNet fueron ajustados con el conjunto de datos de enfermedades de hojas de papa. Se aplicaron técnicas de regularización como el early stopping y la regularización L2 para prevenir el sobreajuste. El optimizador Adam con una tasa de aprendizaje de 0.00001 y la función de pérdida de entropía cruzada categórica se utilizaron para gestionar el problema de clasificación multiclase. El proceso de ajuste fino permitió una convergencia más rápida y un mejor rendimiento en el conjunto de datos específico, mejorando la capacidad del modelo para generalizar a datos nuevos y no vistos.

8. Sustainable Coffee Production A Federated Learning Framework with CNN for Disease Detection and Classification [12]

- **Conjunto de datos y preprocesamiento:** Para cada clase de enfermedad de la hoja de café, se recopilaron datos de cinco clientes distintos. Se obtuvieron imágenes de alta resolución de hojas de café enfermas y sanas utilizando métodos de imagen estandarizados. Las imágenes se alinearon y se eliminó el ruido. Se utilizaron enfoques de estandarización, normalización y aumento de datos para preparar el conjunto de datos para el entrenamiento del modelo.
- **Arquitectura de CNN:** La arquitectura de la CNN incluye varias capas: capas de convolución, de pooling, totalmente conectadas y una capa de salida. La arquitectura se optimizó para la extracción de características de las fotos de hojas de café. Se utilizaron funciones de activación ReLU y una función de pérdida adecuada para reducir el error durante el entrenamiento. Se ajustaron hiperparámetros

como la tasa de aprendizaje, el tamaño del lote y el número de épocas. La arquitectura específica incluye una capa de entrada de 584×584 , tres capas de convolución con filtros de 5×5 y funciones de activación ReLU, tres capas de max-pooling con filtros de 2×2 , dos capas totalmente conectadas con 120 y 60 unidades, y una capa de salida Softmax para clasificar en 15 clases de enfermedades de la hoja de café.

- **Entrenamiento y validación:** Se empleó el aprendizaje federado, donde cinco clientes con datos representativos de diversas geografías y condiciones climáticas entrenaron localmente un modelo CNN básico. Los parámetros del modelo se enviaron a un servidor central y se promediaron utilizando métodos de agregación segura. Este proceso se repitió hasta alcanzar la convergencia. Para evaluar el rendimiento del modelo en datos no observados, se utilizó la validación cruzada k-fold y métricas estándar como precisión, exactitud, recall, F1-score y curva ROC. Se garantizó la privacidad de los datos mediante el aprendizaje federado y se adoptaron prácticas sostenibles para minimizar el impacto ambiental.

9. Tea leaf disease detection using segment anything model and deep convolutional neural networks [13]

- **Conjunto de datos y preprocesamiento:** El conjunto de datos utilizado en este trabajo fue subido a Kaggle y consiste en imágenes de hojas de té de las plantaciones de té del distrito de Unakoti en el estado de Tripura, India. El conjunto de datos contiene 5867 imágenes de hojas con 6 clases: Algal Spot, Brown Blight, Gray Blight, Healthy, Helopeltis y Red Spot. Las imágenes fueron capturadas con una cámara móvil de 12MP a una resolución de 3000×3000 píxeles y luego aumentadas a 5867 imágenes y redimensionadas a 256×256 píxeles. El 64% de las imágenes se utilizó para el entrenamiento, el 20% para pruebas y el 16% para validación. Durante el preprocesamiento, se realizó el recorte automático de las imágenes de las hojas utilizando bibliotecas de OpenCV para minimizar el efecto de los detalles de fondo.
- **Arquitectura de CNN:** La arquitectura de la CNN propuesta incluye cuatro bloques de capas de convolución, cada uno seguido de una capa de MaxPooling y una capa de Normalización por Lotes. La entrada es una imagen RGB de 128×128 píxeles. Las capas de convolución extraen jerarquías espaciales de características de las imágenes de entrada, y la salida de estas capas se aplanan en un vector unidimensional para las capas densas. Se utiliza una capa de Dropout con una tasa de 0.5 para evitar el sobreajuste, seguida de una capa totalmente conectada con 6 unidades y una activación Softmax para la clasificación de las clases. La función de activación ReLU se utiliza para introducir no linealidad y acelerar el entrenamiento.
- **Entrenamiento y validación:** El modelo CNN+MLP se entrenó en dos tipos de conjuntos de datos: uno con imágenes de hojas recortadas y otro con imágenes recortadas superpuestas con máscaras generadas por SAM. El modelo logró una precisión de validación superior al

94% en el conjunto de datos de imágenes recortadas y una precisión del 92% en el conjunto de datos con máscaras SAM. Se utilizó la técnica ReduceLROnPlateau para reducir la tasa de aprendizaje si la pérdida de validación no mejoraba durante varias épocas, lo que mejoró la precisión del modelo en un 3%. Todos los entrenamientos y evaluaciones se realizaron utilizando la GPU NVIDIA Tesla P100 en el entorno de Kaggle Notebooks.

10. *Deep learning-based crop health enhancement through early disease prediction [14]*

- **Conjunto de datos y preprocesamiento:** El conjunto de datos utilizado en este trabajo incluye imágenes de plantas sanas y afectadas por diversas enfermedades. Se pueden utilizar conjuntos de datos públicos como Plant Village o Fungi dB, o crear un conjunto de datos propio capturando imágenes de plantas con una cámara de alta calidad en condiciones bien iluminadas. Las imágenes se anotaron manualmente para indicar si la planta está sana o enferma. El conjunto de datos se dividió en 70% para entrenamiento, 15% para validación y 15% para pruebas. Se aplicaron técnicas de aumento de datos como rotación, volteo y zoom para mejorar la diversidad del conjunto de datos y la generalización del modelo.
- **Arquitectura de CNN:** La arquitectura de la CNN diseñada para este proyecto incluye capas de convolución para aprender patrones locales de las imágenes, seguidas de capas de Max-Pooling para reducir las dimensiones espaciales. Después de varias capas de convolución y Max-Pooling, las características se aplanan en un vector unidimensional que se alimenta a capas totalmente conectadas. La capa de salida es una capa densa con neuronas igual al número de clases en la tarea de clasificación, utilizando activación Softmax para problemas de clasificación multiclase. La arquitectura específica utilizada es Resnet, conocida por su capacidad para extraer características complejas de las imágenes.
- **Entrenamiento y validación:** El modelo CNN se inició con la arquitectura diseñada y se compiló con una función de pérdida (entropía cruzada categórica) y un optimizador (Adam). El modelo se entrenó utilizando los datos preprocesados, ajustando los pesos iterativamente para minimizar la pérdida a lo largo de múltiples épocas. Se monitoreó el rendimiento del modelo en los conjuntos de entrenamiento y validación, ajustando hiperparámetros como la tasa de aprendizaje y el tamaño del lote para optimizar el rendimiento. Se aplicaron técnicas de regularización como Dropout y normalización por lotes para mejorar la generalización y evitar el sobreajuste.

Después de esta comparativa se realizará la elección de las 3 metodologías mas similares y a partir de ellas se creará una metodología que recoja lo mejor de cada una de ellas.

1. *Metodología Resultante*

- **Conjunto de datos y preprocesamiento:** Se pueden utilizar

conjuntos de datos públicos como Plant Village o crear un conjunto de datos propio capturando imágenes de plantas con una cámara de alta calidad en condiciones bien iluminadas. Las imágenes se anotaron manualmente para indicar si la planta está sana o enferma. El conjunto de datos se dividió en 70% para entrenamiento, 15% para validación y 15% para pruebas. Se aplicaron técnicas de aumento de datos como rotación, volteo y zoom para mejorar la diversidad del conjunto de datos y la generalización del modelo. Durante el preprocesamiento, se realizará el recorte automático de las imágenes de las hojas utilizando bibliotecas de OpenCV para minimizar el efecto de los detalles de fondo.

- **Arquitectura de CNN:** La arquitectura de la CNN diseñada incluye capas de convolución para aprender patrones locales de las imágenes, seguidas de capas de Max-Pooling para reducir las dimensiones espaciales. Después de varias capas de convolución y Max-Pooling, las características se aplanan en un vector unidimensional que se alimenta a capas totalmente conectadas. La capa de salida es una capa densa con neuronas igual al número de clases en la tarea de clasificación, utilizando activación Softmax para problemas de clasificación multiclase. Se emplearán arquitecturas preentrenadas como DenseNet201, ResNet152V2 y NasNetMobile, conocidas por su capacidad para extraer características complejas de las imágenes y su eficiencia en entornos con recursos limitados.

- **Entrenamiento y validación:** El entrenamiento se realizará utilizando aprendizaje por transferencia, donde los modelos preentrenados en el conjunto de datos ImageNet serán ajustados con el conjunto de datos específico de enfermedades de hojas. Se aplicarán técnicas de regularización como el early stopping y la regularización L2 para prevenir el sobreajuste. El optimizador Adam con una tasa de aprendizaje de 0.00001 y la función de pérdida de entropía cruzada categórica se utilizarán para gestionar el problema de clasificación multiclase. El proceso de ajuste fino permite una convergencia más rápida y un mejor rendimiento en el conjunto de datos específico, mejorando la capacidad del modelo para generalizar a datos nuevos y no vistos. Se utilizará la técnica ReduceLROnPlateau para reducir la tasa de aprendizaje si la pérdida de validación no mejoraba durante varias épocas, lo que mejoró la precisión del modelo.

III. METODOLOGÍA

El primer paso en la metodología fue la búsqueda y selección de un dataset adecuado para la detección de enfermedades en hojas de café. Se seleccionó un dataset que contenía imágenes de hojas de café clasificadas en dos categorías: saludables e infectadas [15]. Este dataset fue crucial para entrenar y evaluar el modelo de detección de enfermedades.

Una vez obtenido el dataset, se procedió al preprocesamiento de los datos. Este proceso incluyó las siguientes etapas:

- **Carga de Imágenes:**
 - Se desarrolló una función para cargar imágenes desde las carpetas de hojas saludables e infectadas. Las imágenes se redimensionaron a 224x224 píxeles para que coincidieran con el tamaño de entrada del modelo MobileNetV2.
 - Se estableció un límite máximo de imágenes a cargar para cada clase (max_images) para controlar el tamaño del dataset y facilitar el entrenamiento.
- **Etiquetado de Imágenes:**
 - Se asignaron etiquetas a las imágenes: 0 para hojas saludables y 1 para hojas infectadas.
 - Las imágenes y etiquetas se combinaron en un solo conjunto de datos.
- **Normalización de Imágenes:** Las imágenes se normalizaron dividiendo los valores de los píxeles por 255 para que los valores estuvieran en el rango [0, 1].
- **División del Dataset:** El conjunto de datos se dividió en conjuntos de entrenamiento, validación y prueba utilizando la función train_test_split de sklearn. Se utilizó un 15% de los datos para validación y un 15% para prueba.

Para la construcción del modelo, se utilizó la arquitectura MobileNetV2 preentrenada en el dataset ImageNet. Los pasos específicos fueron los siguientes:

- **Carga del Modelo Base:** Se cargó el modelo MobileNetV2 sin la capa superior (include_top=False) y con pesos preentrenados en ImageNet.
- **Adición de Capas Personalizadas:**
 - Se añadió una capa de GlobalAveragePooling2D para reducir las dimensiones espaciales.
 - Se añadió una capa densa con 1024 unidades y activación ReLU.
 - Se añadió una capa de salida con 2 unidades (correspondientes a las clases saludable e infectada) y activación sigmoide.
- **Compilación del Modelo:** El modelo se compiló utilizando el optimizador Adam, la función de pérdida binary_crossentropy y la métrica de precisión.

El modelo se entrenó utilizando el conjunto de datos preprocesado. Los pasos específicos fueron los siguientes:

- **Evaluación en el Conjunto de Validación:**
 - Se evaluó el modelo en el conjunto de validación para obtener la precisión y la pérdida.
 - Se generó un informe de clasificación y una matriz de confusión para analizar el rendimiento del modelo.
- **Evaluación en un Subconjunto Aleatorio:**
 - Se seleccionó un subconjunto aleatorio de imágenes para una evaluación más detallada.

- Se mostraron 20 imágenes con sus predicciones y etiquetas reales.
- Se mostraron 5 imágenes de manera individual, una a una, con sus predicciones y etiquetas reales.
- **Predicción de una Imagen Individual:** Se implementó una funcionalidad para predecir la clase de una imagen individual seleccionada por el usuario.

El código del proyecto se organizó en varios archivos para facilitar su mantenimiento y comprensión, este código está almacenado en el siguiente repositorio de GitHub: <https://github.com/Ssiabatto/Coffee-Leaf-Diseases-Detection.git>

IV. RESULTADOS

Los resultados se presentan en dos secciones, un resultado acorde a la fase de entrenamiento del modelo y un resultado acorde a la fase de prueba de este.

- **Fase de Entrenamiento:** Durante la fase de entrenamiento, el modelo fue entrenado utilizando un conjunto de datos balanceado de imágenes de hojas de café saludables e infectadas. Se utilizaron 500 imágenes de cada clase, y el modelo fue entrenado durante 20 épocas. A continuación, se presentan los resultados de cada época:
 - **Época 1:**
 - Precisión: 78.74%
 - Pérdida: 1.6106
 - Precisión de validación: 94.00%
 - Pérdida de validación: 0.6874
 - **Época 2**
 - Precisión: 85.77%
 - Pérdida: 1.0525
 - Precisión de validación: 86.67%
 - Pérdida de validación: 0.7323
 - **Época 3:**
 - Precisión: 87.13%
 - Pérdida: 0.7793
 - Precisión de validación: 98.00%
 - Pérdida de validación: 0.2329
 - **Época 4:**
 - Precisión: 87.07%
 - Pérdida: 0.4745
 - Precisión de validación: 98.00%
 - Pérdida de validación: 0.1451
 - **Época 5:**
 - Precisión: 94.66%
 - Pérdida: 0.2457
 - Precisión de validación: 98.67%
 - Pérdida de validación: 0.0948
 - **Época 6:**
 - Precisión: 97.56%

- Pérdida: 0.1022
- Precisión de validación: 99.33%
- Pérdida de validación: 0.0794

○ **Época 7:**

- Precisión: 98.33%
- Pérdida: 0.0900
- Precisión de validación: 99.33%
- Pérdida de validación: 0.0542

○ **Época 8:**

- Precisión: 97.35%
- Pérdida: 0.0998
- Precisión de validación: 99.33%
- Pérdida de validación: 0.0589

○ **Época 9:**

- Precisión: 96.85%
- Pérdida: 0.0920
- Precisión de validación: 92.67%
- Pérdida de validación: 0.2100

○ **Época 10:**

- Precisión: 97.42%
- Pérdida: 0.1212
- Precisión de validación: 99.33%
- Pérdida de validación: 0.0529

○ **Época 11:**

- Precisión: 98.92%
- Pérdida: 0.0515
- Precisión de validación: 99.33%
- Pérdida de validación: 0.0520

○ **Época 12:**

- Precisión: 99.33%
- Pérdida: 0.0426
- Precisión de validación: 99.33%
- Pérdida de validación: 0.0669

○ **Época 13:**

- Precisión: 99.31%
- Pérdida: 0.0599
- Precisión de validación: 99.33%
- Pérdida de validación: 0.0501

○ **Época 14:**

- Precisión: 98.44%
- Pérdida: 0.0523
- Precisión de validación: 99.33%
- Pérdida de validación: 0.0474

○ **Época 15:**

- Precisión: 99.02%
- Pérdida: 0.0533
- Precisión de validación: 99.33%
- Pérdida de validación: 0.0487

○ **Época 16:**

- Precisión: 98.58%

- Pérdida: 0.0560
- Precisión de validación: 99.33%
- Pérdida de validación: 0.0477

○ **Época 17:**

- Precisión: 99.20%
- Pérdida: 0.0532
- Precisión de validación: 99.33%
- Pérdida de validación: 0.0527

○ **Época 18:**

- Precisión: 98.61%
- Pérdida: 0.0578
- Precisión de validación: 99.33%
- Pérdida de validación: 0.0540

○ **Época 19:**

- Precisión: 99.02%
- Pérdida: 0.0379
- Precisión de validación: 99.33%
- Pérdida de validación: 0.0486

○ **Época 20:**

- Precisión: 98.81%
- Pérdida: 0.0599
- Precisión de validación: 99.33%
- Pérdida de validación: 0.0606

El modelo mostró una mejora continua en la precisión y una disminución en la pérdida durante las primeras épocas. La precisión de validación alcanzó un 99.33% en varias épocas, lo que indica un buen rendimiento del modelo en el conjunto de validación. Esto se puede evidenciar en la figura 1 y en la figura 2 respectivamente.

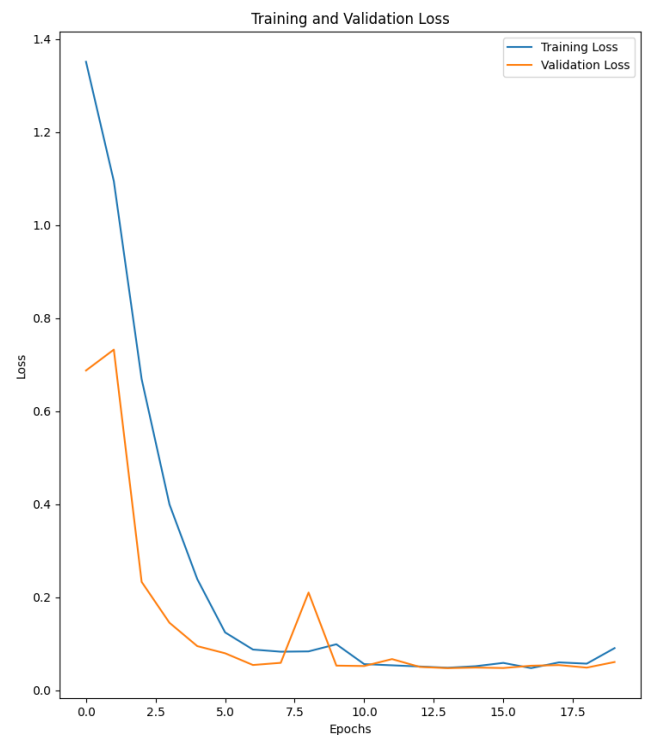


Figura 1. Pérdida de entrenamiento y validación.

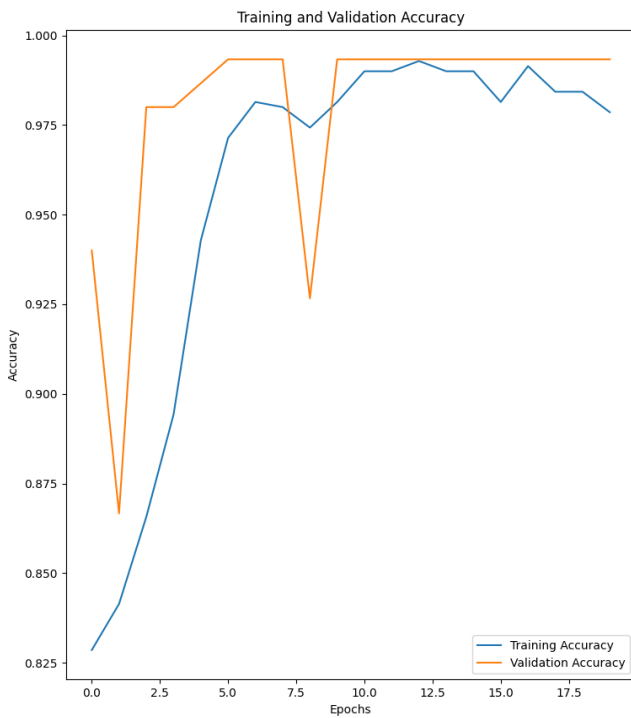


Figura 2. Precisión de entrenamiento y validación.

- **Fase de Prueba:** Para evaluar el rendimiento del modelo en la fase de prueba, se utilizó un conjunto de datos de prueba separado. A continuación, se presentan los resultados de la evaluación del modelo en el conjunto de prueba:

- **Precisión de Prueba:** 99.10%
- **Pérdida de Prueba:** 0.0285

Estos resultados se obtuvieron al procesar 4098 imágenes del dataset, con un aleatorio entre la cantidad de imágenes saludables e infectadas, posteriormente se obtuvo la matriz de confusión presentada en la figura 3.

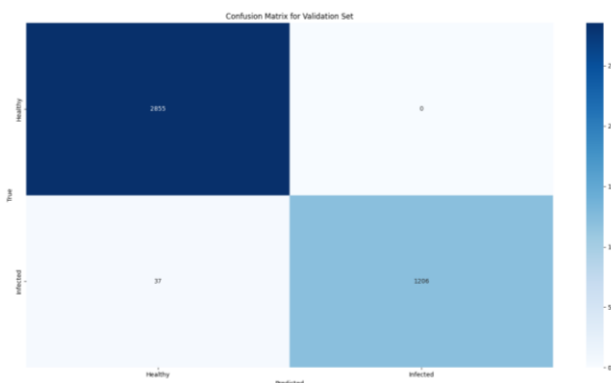


Figura 3. Matriz de confusión.

El modelo mostró una alta precisión en el conjunto de prueba, lo que indica que es capaz de generalizar bien a datos no vistos previamente.

Los resultados se mostraron de la forma que se observa en la figura 4, figura 5 y figura 6, donde se ve el siguiente formato, primero un valor en True (Corresponde al estado real de la hoja) y posteriormente un valor Pred

(Corresponde a la respuesta del modelo al analizar la imagen.

True: Healthy
Pred: Healthy



Figura 4. Muestra 1.

True: Infected
Pred: Infected



Figura 4. Muestra 2.

True: Infected
Pred: Infected



Figura 4. Muestra 3.

V. CONCLUSIONES

Los resultados se presentan en dos secciones, un resultado acorde a la fase de entrenamiento del modelo y un resultado acorde a la fase de prueba de este.

- **Rendimiento del Modelo:** El modelo entrenado mostró una alta precisión tanto en el conjunto de validación como en el conjunto de prueba, alcanzando una precisión del 99.10%. Esto sugiere que el modelo es efectivo para clasificar hojas de café como saludables o infectadas.
- **Mejora Continua:** Durante el entrenamiento, el modelo mostró una mejora continua en la precisión y una disminución en la pérdida, lo que indica que el modelo estaba aprendiendo de manera efectiva.
- **Generalización:** La alta precisión en el conjunto de prueba sugiere que el modelo es capaz de generalizar bien a datos no vistos previamente, lo que es crucial para su aplicación en el mundo real.
- **Recomendaciones Futuras:** Para mejorar aún más el rendimiento del modelo, se podrían considerar las siguientes acciones:
 - Aumentar el tamaño del conjunto de datos de entrenamiento.
 - Experimentar con diferentes arquitecturas de modelos y parámetros de entrenamiento.
 - Aplicar técnicas de aumento de datos para mejorar la robustez del modelo.

REFERENCIAS

- [1] E. Melchor-Martínez, et al., "Towards an Eco-Friendly Coffee Rust Control," MDPI, 2022.
- [2] D. P. Bebber, A. Delgado Castillo, and S. J. Gurr, "Modelling coffee leaf rust risk in Colombia with climate reanalysis data," *Phil. Trans. R. Soc. B*, vol. 368, no. 1620, pp. 1-12, 2013.
- [3] S. Roy, et al., "Ecological dynamics of coffee leaf rust," International Coffee Organization, 2015.
- [4] L. Castro-Llanos, et al., "Assessment of Coffee Leaf Rust in Andean Regions," *Frontiers in Agronomy*, vol. 15, pp. 1-12, 2023.
- [5] Y. Dadi, et al., "Applications of Deep Learning in Agricultural Diagnosis," Springer AI Advances, 2023.
- [6] Bidarakundi, P. M., & Kumar, B. M. (2024). A Coffee Leaf Disease Detection via Deep Learning Algorithms. In 2024 International Conference on Trends in Quantum Computing and Emerging Business Technologies (TQCEBT) (pp. 1-6). CHRIST (Deemed to be University), Pune Lavasa Campus, India. IEEE. DOI: 10.1109/TQCEBT59414.2024.10545110.
- [7] Singh, M. K., & Kumar, A. (2024). Coffee Leaf Disease Classification by Using a Hybrid Deep Convolution Neural Network. Springer Nature Singapore Pte Ltd. Received: 8 November 2023 / Accepted: 10 May 2024.
- [8] Theresa Liebig, Fabienne Ribeyre, Peter Läderach, Hans-Michael Poehling, Piet van Asten & Jacques Avelino (2019) Interactive effects of altitude, microclimate and shading system on coffee leaf rust, *Journal of Plant Interactions*, 14:1, 407-415, <https://doi.org/10.1080/17429145.2019.1643934>
- [9] Avelino, J., Cristancho, M., Georgiou, S., Imbach, P., Aguilar, L., Bornemann, G., Läderach, P., Anzueto, F., Hruska, A. J., & Morales, C. (2015). The coffee rust crises in Colombia and Central America (2008–2013): impacts, plausible causes and proposed solutions. *Food Security*, 7(2), 303-321. <https://doi.org/10.1007/s12571-015-0446-9>
- [10] Araaf, R.T.; Minn, A.; Ahamed, T. Coffee Leaf Rust Disease Detection and Implementation of an Edge Device for Pruning Infected Leaves via Deep Learning Algorithms. *Sensors* 2024, 24, 801. <https://doi.org/10.3390/s24248018>
- [11] Mhala, P., Bilandani, A., & Sharma, S. (2025). Enhancing crop productivity with fined-tuned deep convolution neural network for Potato leaf disease detection. *Expert Systems With Applications*, 267, 126066. <https://doi.org/10.1016/j.eswa.2024.126066>
- [12] Jindal, V., Garg, U., Kukreja, V., Mehta, S., & Bhattacharjee, A. (2023). Sustainable Coffee Production: A Federated Learning Framework with CNN for Disease Detection and Classification. In 2023 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS) (pp. 776-783). IEEE. <https://doi.org/10.1109/ICCCIS60361.2023.10425772>
- [13] Balasundaram, A., Sundaresan, P., Bhavsar, A., Mattu, M., Kavitha, M. S., & Shaik, A. (2025). Tea leaf disease detection using segment anything model and deep convolutional neural networks. *Results in Engineering*, 25, 103784. <https://doi.org/10.1016/j.rineng.2024.103784>
- [14] Venkata Santhosh Yakkala, Krishna Vamsi Nusimala, Badisa Gayathri, Sriya Kanamarlapudi, S. S. Aravindh, Ayodeji Olalekan Salau & S. Srihar (2025) Deep learning-based crop health enhancement through early disease prediction, *Cogent Food & Agriculture*, 11:1, 2423244, <https://doi.org/10.1080/23311932.2024.2423244>
- [15] Jepkoech, Jennifer; Mugo, David; Kenduiywo, Benson; Chebet, Edna (2021), "JMuBEN2", Mendeley Data, V1, doi: 10.17632/tgv3zb82nd.1

