



Análisis Comparativo de Redes Neuronales Convolucionales basadas en Transfer Learning para la Detección de Enfermedades Foliares en Banano

Comparative Analysis of Convolutional Neural Networks Based on Transfer Learning for Foliar Disease Detection in Banana

Autores

Vanessa Pamela Briones-Mera



✉ vanessa.briones@espam.edu.ec

Susana Valentina Zambrano-Cedeño



✉ susana.zambrano@espam.edu.ec

Javier Hernán López-Zambrano*



✉ jlopez@espam.edu.ec

Escuela Superior Politécnica Agropecuaria
de Manabí Manuel Félix López, Carrera de
Computación, Calceta, Manabí, Ecuador

*Autor para correspondencia

Comó citar el artículo:

Briones-Mera, V.P., Zambrano-Cedeño, S.V. & López-Zambrano, J.H. (2026). Análisis Comparativo de Redes Neuronales Convolucionales basadas en Transfer Learning para la Detección de Enfermedades Foliares en Banano. *Informática y Sistemas*, 10(2), 142–160. <https://doi.org/10.33936/isrtic.v10i2.8543>

Enviado: 30/06/2026

Aceptado: 20/08/2026

Publicado: 21/08/2026

Resumen

La detección temprana de enfermedades foliares en el cultivo de banano es crucial para minimizar pérdidas económicas y reducir el uso indiscriminado de agroquímicos. Este estudio presenta un análisis comparativo de la eficacia de diversas arquitecturas de redes neuronales convolucionales mediante Transfer Learning. Se evaluaron un total de 14 modelos, algunos clásicos y otros de última generación, agrupados en familias como ResNet, VGG, MobileNet y EfficientNet, sobre un conjunto de datos multiclase que abarca las patologías de mayor impacto en el sector del banano (Sigatoka, Cordana y Pestalotiopsis). Los resultados de los experimentos demuestran que, tras aplicar técnicas de aprendizaje por transferencia (transfer learning) como el ajuste fino (fine-tuning), el modelo EfficientNetV2L alcanzó el desempeño más destacado con una precisión del 99.35%. Este hallazgo supera los resultados de arquitecturas tradicionales y pone en manifiesto la eficiencia del escalado compuesto para capturar rasgos morfológicos complejos en las hojas de banano. Se concluye que la integración de estos modelos de aprendizaje profundo ofrece una herramienta de alta fidelidad para el diagnóstico automatizado en la agricultura de precisión.

Palabras clave: Banano; Transfer Learning; enfermedades foliares; Redes Neuronales Convolucionales; Visión por computador.

Abstract

Early detection of foliar diseases in banana crops is essential to minimize economic losses and reduce the indiscriminate use of agrochemicals. This study presents a comparative analysis of the effectiveness of various convolutional neural network architectures using Transfer Learning. A total of 14 state-of-the-art models were evaluated, including families such as ResNet, VGG, MobileNet, and EfficientNet, using a multiclass dataset that covers the most impactful pathologies in the sector. Experimental results demonstrate that, after applying fine-tuning techniques, the EfficientNetV2L model achieved the most outstanding performance with an accuracy of 99.35%. This finding outperforms traditional architectures and highlights the efficiency of compound scaling in capturing complex morphological features in leaves. It is concluded that the integration of these deep learning models provides a high-fidelity tool for automated diagnosis in precision agriculture.

Keywords: Banana; Transfer Learning; foliar diseases; Convolutional Neural Networks; Computer Vision.





1. Introducción

La agricultura constituye uno de los pilares fundamentales para la seguridad alimentaria mundial y el desarrollo económico de numerosos países. No obstante, los sistemas agrícolas afrontan desafíos crecientes debido al cambio climático, la pérdida de biodiversidad, la degradación de la tierra y la sobreexplotación de recursos naturales, lo que pone en riesgo la seguridad alimentaria ante una población mundial en expansión (FAO, 2021). Entre los cultivos tropicales de mayor relevancia destaca el banano, considerado la fruta más consumida y exportada a nivel global, con un papel estratégico tanto en la economía agrícola como en la dieta de millones de personas (Martínez-Solórzano & Rey-Brina, 2021).

En Ecuador, el sector bananero constituye una de las actividades agrícolas de mayor relevancia económica y social, debido a su aporte significativo al Producto Interno Bruto (PIB), su participación en el comercio exterior y su capacidad de generación de empleo (Cañetaco Barriga & Chalco Siguachi, 2026). Esta actividad productiva genera más de 35.000 empleos, concentrados principalmente en las provincias de El Oro, Guayas y Los Ríos, donde se desarrolla gran parte de la producción bananera del país (CFN, 2024).

No obstante, el rendimiento y la calidad del banano se ven severamente afectados por diversas enfermedades foliares (Tripathi et al., 2023). Como son la Sigatoka, la Cordana y la Pestalotiopsis que son patologías comunes en este cultivo, las mismas que pueden ocasionar pérdidas significativas si no son detectadas a tiempo. Tradicionalmente, su identificación se ha basado en inspecciones visuales realizadas por expertos agrónomos, un proceso que resulta costoso, subjetivo y difícil de escalar a grandes extensiones de cultivo (Villanueva et al., 2025).

En Ecuador no existen estadísticas nacionales oficiales que cuantifiquen de manera conjunta las pérdidas económicas ocasionadas por la Sigatoka, Cordana y Pestalotiopsis en el cultivo de banano. Sin embargo, diversos reportes evidencian el fuerte impacto de la Sigatoka negra sobre la producción bananera. Según datos del Ministerio de Agricultura y Ganadería (MAG), hasta mayo de 2023 aproximadamente el 31 % de las hectáreas de pequeños productores presentaban afectaciones moderadas o severas por esta enfermedad. Provincias como El Oro, Esmeraldas, Cotopaxi, Santo Domingo de los Tsáchilas y Santa Elena registraron incrementos importantes en la superficie afectada, obligando a los productores a incrementar la frecuencia de las fumigaciones y, en consecuencia, los costos de producción. Además, la enfermedad reduce la calidad del fruto destinado a la exportación y contribuye a disminuir la oferta exportable del país, evidenciando su impacto económico sobre el sector

bananero ecuatoriano (Redacción Expreso, 2023).

En concordancia con lo anterior, la Sigatoka Negra (*Pseudocercospora fijiensis*) representa la principal enfermedad foliar que afecta la producción bananera en Ecuador, debido a que disminuye la capacidad fotosintética de las plantas, reduce el peso de los racimos y acelera la maduración de los frutos, comprometiendo su calidad para la exportación. En condiciones de alta incidencia y manejo inadecuado, las pérdidas de rendimiento pueden oscilar entre el 20 % y el 50 %, e incluso superar el 80 % en plantaciones severamente afectadas. Asimismo, el control fitosanitario de esta enfermedad constituye uno de los principales componentes del costo de producción, ya que las aplicaciones frecuentes de fungicidas y otras labores de manejo pueden representar hasta el 48,56 % de los costos totales de producción, afectando la rentabilidad y sostenibilidad económica de los productores bananeros (Galarza Vera, 2024).

Para hacer frente a esta problemática, la integración de inteligencia artificial en la agricultura de precisión ha permitido desarrollar e integrar sistemas automatizados para la detección de enfermedades en cultivos mediante aprendizaje automático y análisis de imágenes digitales, contribuyendo a diagnósticos más precisos y eficientes (Olivares et al., 2021; Vera & Carvajal, 2025). Entre estas técnicas, destaca el aprendizaje por transferencia (Transfer Learning) debido a su capacidad para reutilizar modelos previamente entrenados en grandes bases de datos, beneficiando a la reducción del tiempo de entrenamiento y la mejora del desempeño en tareas de clasificación agrícola (Hosna et al., 2022).

El aprovechamiento de las arquitecturas preentrenadas permite sintetizar el conocimiento de modelos complejos en tareas específicas de fitopatología (ciencia que estudia las enfermedades de las plantas). En experiencias previas llevadas a cabo por investigaciones de la ESPAM MFL, el uso de Inception V3 permitió alcanzar altos niveles de precisión en la clasificación de anomalías en frutos (Montesdeoca Espinoza et al., 2025), lo que fundamenta su inclusión en esta comparativa para el cultivo de banano.

Diversos estudios recientes han aplicado redes neuronales convolucionales preentrenadas, como EfficientNet, ResNet y VGG, para la detección automatizada de enfermedades del banano, alcanzando niveles de precisión superiores al 85% (Jiménez et al., 2025; Sanga et al., 2020; Torres Salamea et al., 2022). Sin embargo, la mayoría de estas investigaciones se han centrado en evaluar una única arquitectura de forma aislada, empleando además conjuntos de datos, particiones y condiciones de entrenamiento distintas entre sí, lo que impide comparar objetivamente su desempeño y limita la posibilidad de extraer

conclusiones generalizables sobre qué familia de modelos resulta más adecuada. Este vacío genera incertidumbre respecto a qué arquitectura ofrece el mejor equilibrio a través de las diferentes métricas de evaluación y requerimiento computacionales en tareas de diagnóstico automatizado de enfermedades foliares (Li et al., 2020).

En este contexto, la presente investigación tiene como objetivo comparar sistemáticamente el desempeño de 14 arquitecturas de redes neuronales convolucionales preentrenadas, agrupadas en las familias VGG, ResNet, MobileNet, DenseNet, EfficientNet, EfficientNetV2, Xception y ConvNeXt, para la detección automática de enfermedades foliares del banano (Sigatoka, Cordana y Pestalotiopsis) mediante transfer learning, empleando el conjunto de datos BananaLSD (Arman et al., 2023). De manera específica, el estudio se plantea las siguientes preguntas de investigación: (i) ¿qué arquitectura ofrece el mejor equilibrio entre precisión, estabilidad y costo computacional para la detección de enfermedades foliares del banano?; y (ii) ¿en qué medida la aplicación de fine-tuning mejora el desempeño respecto a la transferencia de aprendizaje estática? Se plantea como hipótesis que las arquitecturas basadas en escalado compuesto y conexiones residuales o de atención (EfficientNet, EfficientNetV2 y Xception) alcanzarán un desempeño estadísticamente superior al de arquitecturas clásicas de estructura secuencial (como VGG19), debido a su mayor capacidad para capturar patrones morfológicos complejos con menor riesgo de desvanecimiento del gradiente, y que la aplicación de fine-tuning incrementará de forma significativa la precisión y reducirá la brecha entre entrenamiento y prueba respecto a la transferencia de aprendizaje estática.

Aunque el conjunto de datos BananaLSD fue recopilado originalmente en una región geográfica distinta a Sudamérica, su utilización en el contexto ecuatoriano se considera metodológicamente pertinente debido a que las enfermedades foliares incluidas presentan desarrollo y propagación en ambientes tropicales húmedos similares a los existentes en las zonas bananeras del Ecuador. Patologías como la Sigatoka (*Pseudocercospora fijiensis*), la Cordana (*Cordana musae*) y la Pestalotiopsis (*Pestalotiopsis disseminata*) han sido reportadas en cultivos de banano desarrollados bajo condiciones de alta humedad y temperatura (Aviles Amador, 2025; Jiménez et al., 2025). Estos factores climáticos favorecen la aparición de patrones sintomatológicos comparables entre distintas regiones productoras (Ünal, 2025).

Para alcanzar este propósito, se realizó una revisión de modelos de aprendizaje profundo aplicados a la agricultura, un análisis exploratorio del conjunto de datos BananaLSD y el entrenamiento comparativo de catorce arquitecturas de redes neuronales convolucionales. La selección de estas arquitecturas se fundamentó en los resultados de la revisión sistemática de la literatura, en la que se identificaron los modelos más empleados y con mejor desempeño reportado en tareas de clasificación de enfermedades foliares. Para una mejor representación se las

clasificó por familias de redes convolucionales, empezando por modelos clásicos (VGG), redes residuales profundas (ResNet), arquitecturas optimizadas para eficiencia computacional (MobileNet), modelos con conexiones densas (DenseNet) y propuestas de última generación basadas en escalado compuesto y diseño moderno (EfficientNet, EfficientNetV2, Xception y ConvNeXt). Esta revisión sistemática permitió seleccionar arquitecturas con distintos niveles de profundidad, complejidad y costo computacional, con el fin de identificar la más adecuada para aplicaciones reales en agricultura de precisión. Los resultados obtenidos procuran aportar evidencia técnica que sirva como guía para la selección del modelo más adecuado en aplicaciones agrícolas reales, contribuyendo así al fortalecimiento de la agricultura de precisión mediante el uso de tecnologías de inteligencia artificial (Barbedo, 2020).

El presente trabajo se diferencia de estudios previos que se limitan a la evaluación de arquitecturas clásicas o a un número reducido de modelos, mientras que la presente investigación presenta una evaluación sistemática y repetida de 14 arquitecturas de redes neuronales convolucionales (CNN) bajo un pipeline experimental estandarizado. Este enfoque permite incorporar métricas de estabilidad y análisis de generalización, con el objetivo de reducir sesgos inherentes a entrenamientos únicos y fortaleciendo la robustez y confiabilidad de los resultados. Cabe precisar que el principal aporte de este trabajo no radica en un conjunto de datos o un algoritmo inédito -el dataset BananaLSD es de acceso público desde 2023-, sino en establecer un benchmark estandarizado y reproducible que permite comparar de manera objetiva el desempeño de múltiples arquitecturas de aprendizaje profundo bajo condiciones experimentales homogéneas.

El presente estudio integra tanto arquitecturas de última generación con mecanismos de escalado compuesto (la familia EfficientNetV2), como modelos compactos y eficientes, entre ellos ConvNeXt, los cuales han demostrado un desempeño superior en la extracción de características morfológicas complejas en imágenes foliares. A la par, la investigación garantiza la replicabilidad experimental mediante el uso de infraestructura computacional de alto rendimiento (NVIDIA A100) y la descripción explícita de la configuración de hiperparámetros, proporcionando una guía técnica objetiva para la selección de modelos en aplicaciones reales de agricultura de precisión, particularmente para el monitoreo fitosanitario en la región costera ecuatoriana.

2. Materiales y Métodos

La investigación se desarrolló entre octubre de 2025 y febrero de 2026, en la Escuela Superior Agropecuaria de Manabí Manuel Félix López (ESPAM MFL), en el marco del proyecto institucional “Aprendizaje automático y su comportamiento con distintos tipos de conjunto de datos agropecuarios”.

Desde el punto de vista metodológico, la investigación se enmarca en un diseño de tipo experimental comparativo, con





enfoque cuantitativo. La variable independiente corresponde a la arquitectura de red neuronal convolucional empleada (14 niveles, uno por cada arquitectura evaluada) y a la estrategia de entrenamiento aplicada (transferencia de aprendizaje estática frente a ajuste fino o fine-tuning). Como variables dependientes se consideraron la precisión (accuracy), la pérdida (loss), la brecha entre entrenamiento y prueba, y el costo computacional (tiempo de inferencia y número de parámetros). El diseño contempló dos fases experimentales secuenciales -transferencia estática y fine-tuning- aplicadas a las 14 arquitecturas bajo un pipeline estandarizado, con medidas repetidas (5 ejecuciones independientes por modelo) que permitieron estimar la variabilidad y la significancia estadística de los resultados mediante una prueba t de Student pareada.

2.1 Revisión sistemática de la literatura.

Se llevó a cabo una Revisión Sistemática de la Literatura (RSL) orientada a identificar los modelos de aprendizaje profundo más usados en la detección automática de enfermedades foliares del banano. La búsqueda bibliográfica se llevó a cabo mediante el uso de motores de búsqueda académica y bases de datos científicas, entre ellas Google Scholar, ScienceDirect y EBSCOhost. Posteriormente, se aplicaron criterios de inclusión y exclusión con el propósito de asegurar la relevancia y pertinencia de los estudios seleccionados.

Para la búsqueda se emplearon combinaciones de palabras claves centradas en la detección de enfermedades foliares del banano mediante redes neuronales convolucionales (CNN) y aprendizaje por transferencia (Transfer Learning). De un total inicial de 72 artículos, se seleccionaron 20 estudios que cumplieran con los criterios establecidos. Estos constituyeron la base de análisis para identificar las arquitecturas más utilizadas con sus métricas empleadas.

2.2 Conjunto de datos utilizado.

Para el desarrollo de este estudio se seleccionó el conjunto de datos Banana Leaf Spot Diseases (BananaLSD) (Arman et al., 2023), disponible en acceso abierto a través de Mendeley Data <https://doi.org/10.17632/9tb7k297ff.1>, cuyas imágenes fueron capturadas con cámaras de teléfonos inteligentes en los campos de la Universidad Agrícola Bangabandhu Sheikh Mujibur Rahman (Bangladesh) y en zonas agrícolas cercanas, durante junio de 2021. Todas las muestras fueron etiquetadas por un especialista en fitopatología, lo que garantiza la validez de las clases.

El conjunto de datos Banana Leaf Spot Diseases (BananaLSD) incluye imágenes de hojas afectadas por Sigatoka, Cordana y Pestalotiopsis, enfermedades foliares asociadas al cultivo de banano en ambientes tropicales húmedos (Arriaga et al., 2022; Ferrer-Sánchez et al., 2024), lo que se considera pertinente para el entrenamiento y evaluación de los modelos propuestos, tal como se justifica con mayor detalle más adelante en esta sección. En este contexto las redes neuronales extraen automáticamente características relevantes y patrones complejos presentes en las imágenes (Cedeño Campoverde, 2024). Y el uso de transfer learning permite aprovechar las representaciones visuales previamente aprendidas favoreciendo los procesos de generalización en tareas de clasificación agrícola.

Aunque el conjunto de datos BananaLSD fue recopilado en Bangladesh, las provincias ecuatorianas de El Oro, Guayas y Los Ríos presentan condiciones agroclimáticas comparables para el cultivo de banano. Ambas regiones se caracterizan por ambientes tropicales cálidos y húmedos. En Bangladesh, el banano se desarrolla adecuadamente en temperaturas entre 15 y 35 °C, con una humedad relativa de 75–85 %, siendo un cultivo propio de las tierras bajas tropicales húmedas. Por su parte, la región bananera de la costa ecuatoriana presenta temperaturas medias anuales cercanas a 25–27 °C y una humedad relativa aproximada del 83 %, condiciones que favorecen el desarrollo del cultivo y la aparición de enfermedades foliares. En consecuencia, aunque existen diferencias en los patrones de precipitación y estacionalidad, ambas regiones comparten un entorno climático tropical húmedo que favorece la manifestación visual de enfermedades en las hojas del banano, lo que respalda la transferencia de modelos entrenados con BananaLSD al contexto ecuatoriano (Bangladesh Bureau of Statistics, 2024; Enríquez Lapo, 2024).

Cabe señalar que, en el marco de este estudio, no se realizó una validación preliminar con imágenes de campo capturadas en el contexto ecuatoriano. La investigación se concibió como una evaluación comparativa y estandarizada de 14 arquitecturas CNN bajo condiciones experimentales homogéneas, para lo cual se priorizó el uso de un conjunto de datos público, previamente balanceado y etiquetado por un especialista en fitopatología (BananaLSD), que garantiza la reproducibilidad y la comparabilidad objetiva entre modelos. La recolección y el etiquetado experto de un conjunto de imágenes locales ecuatorianas constituyen un proceso de campo independiente que excede el alcance metodológico de este estudio comparativo, por lo que se plantea como una etapa de validación externa para trabajos futuros, tal como se detalla en la sección de limitaciones (3.6).

Este dataset dispone de dos subconjuntos: una versión original

conformada por 937 imágenes y una versión previamente balanceada por los autores del dataset mediante técnicas de aumento de datos, incluyendo desenfoque gaussiano, giro horizontal, recorte, ajuste de contraste lineal, traslación y distorsiones geométricas, alcanzando un total de 1600 imágenes. Cada subconjunto se encuentra distribuido en cuatro categorías: Sigatoka, Cordana, Pestalotiopsis y hojas sanas (healthy). En la presente investigación se utilizó directamente la versión balanceada proporcionada por los autores, por lo que no fue necesario aplicar procedimientos adicionales de balanceo o aumento de datos.

Con el fin de contextualizar visualmente estas categorías antes de describir el pipeline experimental, la Figura 1 presenta muestras representativas de cada clase, sin transformaciones aplicadas: (A) Cordana, con manchas de color pardo-morado y halo amarillento; (B) hoja sana (healthy), sin síntomas visibles; (C) Pestalotiopsis, con áreas necróticas de tonalidad grisácea; y (D) Sigatoka Negra, identificable por estrías necróticas oscuras alineadas paralelamente a las nervaduras foliares, rodeadas de halos cloróticos (amarillentos).

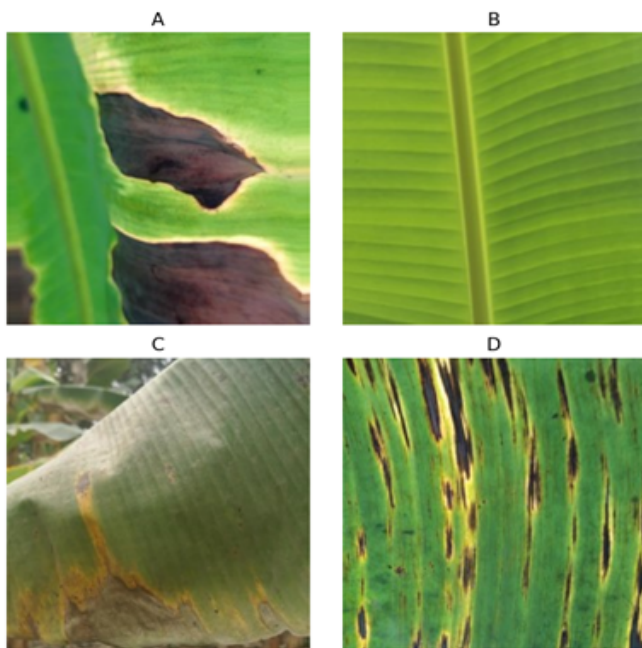


Figura 1. Muestras representativas de las categorías del dataset BananaLSD, sin transformaciones aplicadas. (A) Cordana; (B) Sana (Healthy); (C) Pestalotiopsis; (D) Sigatoka Negra.

Fuente: Arman et al. (2023), BananaLSD dataset

2.3 Entorno de Desarrollo e Infraestructura.

El entrenamiento de los modelos de aprendizaje profundo se realizó en un entorno de cómputo de alto rendimiento provisto por Google Colab Pro, utilizando una GPU NVIDIA A100 como recurso principal de procesamiento. La plataforma de Google en su versión Pro ofrece una disponibilidad de 80 GB de memoria dedicada en la GPU lo que permite entrenar redes

neuronales convolucionales profundas y procesar lotes de datos de tamaño considerable de forma eficiente. Por otro lado, el entorno cuenta con 32 GB de memoria RAM del sistema, lo que posibilita ejecutar sin restricciones las tareas tanto de carga, preprocesamiento y almacenamiento temporal de imágenes y tensores requeridas durante el flujo experimental. Como lenguaje de desarrollo, se empleó Python 3.10 junto con las bibliotecas de código abierto TensorFlow (versión 2.15) y Keras, las cuales facilitaron la implementación de las arquitecturas propuestas y la administración de los procesos de transferencia de aprendizaje. Toda esta configuración permitió desarrollar los experimentos en condiciones estables y con tiempos de entrenamiento adecuados a los objetivos del estudio.

2.4 Análisis exploratorio de datos.

El análisis exploratorio incluyó tres actividades principales:

Análisis de distribución de clases:

Se contabilizó el número de imágenes por cada categoría con el fin de identificar desbalances entre clases. El conjunto original mostró una composición desigual; Sigatoka: 473, Cordana: 162, Pestalotiopsis: 173, Healthy: 129, mientras que el conjunto balanceado presentó una distribución uniforme de 400 imágenes por clase, cabe destacar que en la presente investigación no se realizaron actividades de balanceo, puesto que el conjunto de datos ya ofrece una versión balanceada.

Evaluación de la variabilidad intra e interclase:

La variabilidad intra e interclase se analizó utilizando el modelo VGG16 preentrenado con ImageNet como extractor de características sobre el conjunto de datos aumentado. Para obtener las representaciones vectoriales de las imágenes se empleó la capa fc2, correspondiente a una de las últimas capas densas de la arquitectura. Posteriormente, los vectores obtenidos fueron normalizados mediante normalización L2 para reducir diferencias de escala entre las representaciones. Con base en estos embeddings, se calcularon distancias euclidianas promedio entre imágenes de una misma clase para estimar la variabilidad intra-clase y entre centroides de clases distintas para evaluar la variabilidad inter-clase. Los resultados mostraron una mayor dispersión en las clases Cordana y Pestalotiopsis, mientras que las clases Pestalotiopsis y Healthy presentaron una mayor proximidad entre sus centroides, evidenciando similitudes visuales entre ambas categorías.

Verificación técnica del dataset:

Se verificó la consistencia del formato, las dimensiones y el modo de color de las imágenes mediante scripts desarrollados en Python. Como resultado de esta revisión, se constató que la totalidad de los archivos se encontraba en formato JPEG, con una resolución uniforme de 224×224 píxeles y en modo de color RGB. Asimismo, no se identificaron imágenes corruptas ni archivos que presentaran problemas de lectura.

2.5 Fase de Entrenamiento.

El conjunto de 14 arquitecturas fue entrenado y evaluado sistemáticamente mediante un procedimiento de transferencia de aprendizaje estructurado en dos fases experimentales claramente diferenciadas. Para ello, se utilizó el dataset BananaLSD que fue distribuido en un 80% para entrenar, un 10% para valorar y un 10% para la evaluación final (test set). Todas las arquitecturas evaluadas recibieron imágenes de entrada de 224×224 píxeles en modo de color RGB, dimensión que corresponde a la resolución nativa del dataset BananaLSD y que se mantuvo constante para las 14 arquitecturas con el fin de asegurar la comparabilidad de los resultados entre modelos.

Durante el proceso de entrenamiento se aplicaron transformaciones básicas únicamente sobre los datos de entrenamiento, con el objetivo de incrementar la variabilidad de las muestras y mejorar la capacidad de generalización de los modelos. La figura 2A presenta las transformaciones implementadas, entre las que destacan rotaciones, inversiones, ajustes de brillo y contraste. Por su parte, la figura 2B muestra ejemplos de imágenes resultantes después de aplicar dichas transformaciones a las distintas clases del dataset.

No fue necesario aplicar técnicas adicionales de balanceo de

clases, ya que se utilizó directamente la versión balanceada del dataset BananaLSD, con 400 imágenes por clase y un total de 1600 muestras. Esto disminuye la probabilidad de un sesgo hacia las clases mayoritarias y permite un aprendizaje estable sin la necesidad de métodos de compensación más complejos (como sobremuestreo o ponderación de clases) que pudieran introducir variaciones artificiosas no representativas del dominio verdadero. El aumento de datos (data augmentation) propiamente dicho sí se aplicó durante el entrenamiento, tal como se detalla en la sección 2.5.2.

Fase 1: Transferencia de Aprendizaje Estática

En la etapa inicial, se utilizó cada arquitectura preentrenada con pesos de ImageNet, manteniendo congeladas las capas del modelo base para preservar las características visuales generales. Sobre esta base, se incorporó un top model personalizado diseñado para la adaptación al dominio específico, compuesto por:

- Una capa de Global Average Pooling 2D para reducción de dimensionalidad.
- Una capa de Batch Normalization para estabilizar el aprendizaje.
- Dos capas de Dropout con una tasa de 0.4 para mitigar el sobreajuste (overfitting).
- Una capa densa intermedia de 64 neuronas con activación ReLU.

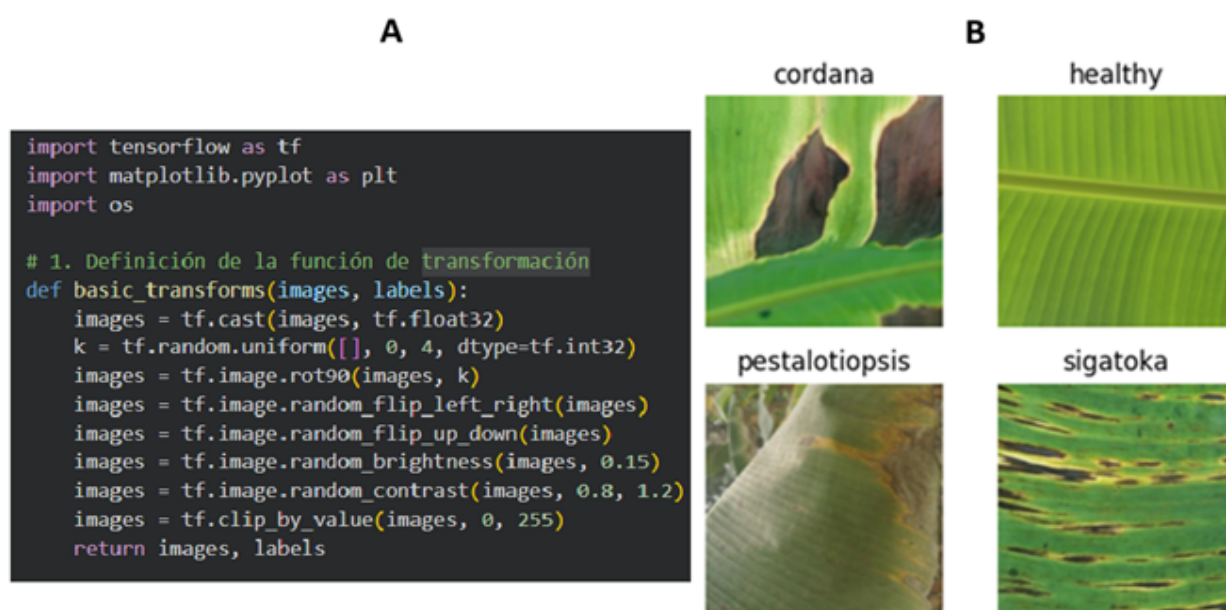


Figura 2. Transformaciones aplicadas. (a) Transformaciones básicas; (b) imágenes después de aplicar las transformaciones
Fuente: Los autores

- Una capa de salida con activación Softmax para la clasificación de las cuatro categorías.

Fase 2: Ajuste Fino (Fine-tuning)

Posteriormente, se implementó una fase de fine-tuning mediante el descongelamiento parcial de las capas superiores del modelo base. Esta estrategia permitió refinar las representaciones de alto nivel, ajustándolas a las particularidades visuales de las enfermedades del banano. Para garantizar una adaptación gradual de los pesos y evitar la pérdida de conocimiento previo (catastrophic forgetting), se empleó una tasa de aprendizaje reducida de 10^{-4} , frente a la tasa inicial de 10^{-3} utilizada en la fase previa.

Configuración de Hiperparámetros y Evaluación

Durante ambos experimentos se mantuvo una configuración común:

- Optimizador: Adam, por su capacidad de adaptar la tasa de aprendizaje individualmente.
- Función de pérdida: Entropía cruzada categórica (Categorical Cross-Entropy).
- Tamaño de lote: 32 imágenes, optimizando el balance entre estabilidad del gradiente y uso de memoria.
- Tasas de aprendizaje: se emplearon dos valores fijos con base en la práctica estándar documentada para transfer learning y fine-tuning con redes convolucionales preentrenadas, sin realizar una búsqueda sistemática (grid search) de hiperparámetros ni pruebas preliminares con valores adicionales: 10^{-3} para la fase de transferencia estática (top model), correspondiente al valor por defecto recomendado en la guía oficial de Keras/TensorFlow para el entrenamiento de las capas superiores con el modelo base congelado, y 10^{-4} para la fase de fine-tuning, siguiendo la práctica de reducir la tasa de aprendizaje entre uno y dos órdenes de magnitud al descongelar capas del modelo base con el fin de evitar la pérdida de las características previamente aprendidas (catastrophic forgetting) (Chollet, 2020; Yosinski et al., 2014). La exploración sistemática de un rango más amplio de tasas de aprendizaje se plantea como línea de trabajo futuro.
- Inicialización de pesos: las capas del top model (Dense, BatchNormalization) se inicializaron con los valores por defecto de Keras/TensorFlow (inicialización Glorot uniforme para las capas densas), sin aplicar un esquema de inicialización personalizado.
- Épocas: En el Experimento 1 (transferencia estática, sin fine-tuning), cada arquitectura se entrenó durante un número fijo de 15 épocas, sin aplicar early stopping. En el Experimento 2 (fine-tuning), el modelo con mejor desempeño se entrenó inicialmente durante 3 épocas con la base convolucional congelada, actualizando únicamente la cabeza de clasificación.

Esta etapa actuó como una fase de estabilización (learning rate warm-up), permitiendo adaptar gradualmente las nuevas capas de clasificación antes de modificar los pesos preentrenados. Posteriormente, se descongelaron parcialmente las capas superiores del modelo base y se continuó el entrenamiento durante un máximo de 15 épocas adicionales, monitoreando la convergencia mediante las curvas de pérdida.

- Early stopping y reducción adaptativa de la tasa de aprendizaje: durante la fase de fine-tuning (Experimento 2) se aplicó una estrategia de detención temprana (EarlyStopping) monitoreando la pérdida de validación, con una paciencia de 4 épocas y restauración de los mejores pesos (restore_best_weights=True), combinada con una reducción adaptativa de la tasa de aprendizaje (ReduceLROnPlateau, paciencia de 2 épocas, factor de reducción de 0.2) para prevenir el sobreajuste y estabilizar la convergencia en las últimas épocas

De las 14 arquitecturas evaluadas en la fase de fine-tuning, 10 completaron las 15 épocas máximas sin activar el criterio de early stopping (ResNet152, InceptionV3, MobileNetV2, Xception, DenseNet121, EfficientNetB3, EfficientNetV2L, EfficientNetV2S, ConvNeXtTiny y VGG19), mientras que los 4 restantes se detuvieron antes de alcanzar dicho límite: ResNet50 (11 épocas), ResNet101 (9 épocas), EfficientNetB0 (11 épocas) y MobileNetV3Large (6 épocas). En promedio, las arquitecturas emplearon 13.4 épocas durante la fase de ajuste fino.

La selección de las capas descongeladas en cada arquitectura se definió mediante un índice de corte específico por modelo (capas anteriores a dicho índice permanecieron congeladas, mientras que las posteriores se habilitaron para reentrenamiento), manteniendo además congeladas todas las capas de Batch Normalization del modelo base durante el fine-tuning para preservar la estabilidad de las estadísticas aprendidas en ImageNet. En arquitecturas profundas (ResNet101, ResNet152, DenseNet121, EfficientNetV2L) se descongeló aproximadamente el último 8-10 % de las capas del modelo base; en arquitecturas más superficiales (VGG19, Xception, MobileNetV2) el porcentaje descongelado fue mayor, entre el 12 % y el 23 %. En dos arquitecturas (ConvNeXtTiny y MobileNetV3Large) el índice de corte definido excedió el número total de capas del modelo base, por lo que, en la práctica, ninguna capa convolucional de estas dos arquitecturas llegó a descongelarse durante la fase de fine-tuning; en ambos casos, la mejora observada tras el fine-tuning (Tabla 3) corresponde exclusivamente al reentrenamiento adicional del top model a una tasa de aprendizaje menor, y no a una adaptación real de las capas convolucionales al dominio de las enfermedades foliares del banano. Este hallazgo se retoma como limitación en la sección 3.6.

El desempeño se cuantificó mediante métricas de precisión (accuracy), pérdida (loss), matrices de confusión y reportes de clasificación. Se analizaron las brechas entre los conjuntos de entrenamiento y prueba para valorar la capacidad de

generalización. Los resultados evidenciaron mejoras consistentes tras el fine-tuning, validando la efectividad de esta estrategia para la optimización de modelos en la detección de patologías foliares.

2.5.1 Arquitecturas evaluadas

El estudio evaluó sistemáticamente 14 arquitecturas representativas de diferentes familias de modelos como se muestra en la tabla 1:

Tabla 1 Lista de arquitecturas evaluadas

Fuente: Los Autores

Familia	Arquitecturas Incluidas	Características Principales
VGG	VGG19	Arquitectura clásica, bloques convolucionales uniformes
ResNet	ResNet50, ResNet101, ResNet152	Conexiones residuales, entrenamiento de redes profundas
Inception	InceptionV3	Convoluciones paralelas a diferentes escalas
MobileNet	MobileNetV2, MobileNetV3Large	Optimizadas para eficiencia en dispositivos móviles
DenseNet	DenseNet121	Conexiones densas entre capas
EfficientNet	EfficientNetB0, B3, V2S, V2L	Escalado compuesto óptimo de profundidad/anchura/resolución
Xception	Xception	Convoluciones separables en profundidad extremas
ConvNeXt	ConvNeXtTiny	Modernización de arquitecturas tipo ResNet

Las arquitecturas evaluadas fueron seleccionadas con el propósito de representar las principales familias de redes convolucionales empleadas en clasificación de imágenes, incluyendo VGG, ResNet, EfficientNet, ConvNeXt, MobileNet y Vision Transformer. Dentro de cada familia se escogieron variantes que abarcan diferentes niveles de complejidad computacional y profundidad; por ejemplo, EfficientNetB0 y EfficientNetV2 (B0 y L) representan la evolución de esta familia desde modelos compactos hasta arquitecturas de alta capacidad, mientras que ConvNeXt Tiny y ConvNeXt Large permiten analizar el comportamiento de configuraciones ligeras y profundas de una arquitectura CNN moderna. Aunque la revisión sistemática identificó otras arquitecturas ampliamente utilizadas, como

DenseNet201 e InceptionResNetV2, estas no fueron incorporadas debido a las limitaciones de recursos computacionales y al elevado costo asociado con el entrenamiento y evaluación de un mayor número de modelos. En consecuencia, se priorizó un conjunto de arquitecturas representativas que permitiera realizar una comparación equilibrada entre distintas filosofías de diseño y generaciones de modelos, manteniendo la viabilidad experimental del estudio.

2.5.2 Desarrollo de los experimentos:

Para el desarrollo de los experimentos se realizaron 2 experimentos dentro de un pipeline; cada flujo fue ejecutado un total de 5 veces, comparando posteriormente los promedios obtenidos de las métricas evaluadas. El conjunto inicial estuvo compuesto por 1600 imágenes, de las cuales el 20% (320 imágenes) fueron destinadas exclusivamente para la fase de prueba (test), dejando un conjunto de 1280 imágenes para el entrenamiento del modelo.

Durante la etapa de entrenamiento se aplicaron operaciones de aumento de datos mediante transformaciones estocásticas en línea (on-the-fly), generadas dinámicamente en cada época durante el proceso de carga y entrenamiento, por lo que las imágenes transformadas no fueron almacenadas físicamente como nuevos archivos dentro del conjunto de datos ni incrementan su tamaño efectivo en disco. Las técnicas de transformación aplicadas incluyeron: imagen original, rotación $+15^\circ$, rotación -15° , flip horizontal, zoom, aumento de brillo, reducción de brillo, cambio de contraste, desplazamiento lateral (shift), adición de ruido leve y combinación de varias transformaciones. Estas operaciones permitieron incrementar la variabilidad del conjunto de entrenamiento, obteniendo aproximadamente 11 variantes generadas dinámicamente por cada imagen original durante el entrenamiento, con el objetivo de mejorar la capacidad de generalización del modelo, como se muestra en la Figura 3.

3. Resultados y Discusión

Mediante la RSL y la selección del estudio que cumplieron con todos los criterios de inclusión, se pudo observar que los modelos preentrenados más utilizados fueron VGG16/VGG19, ResNet50, InceptionV3, DenseNet201, EfficientNetB0 y MobileNetV2, cada uno con métricas de precisión reportadas en un rango aproximado del 85% al 98% según el tamaño y la calidad de los subconjuntos de datos empleados.

3.1 Distribución y balanceo del dataset BananaLSD.

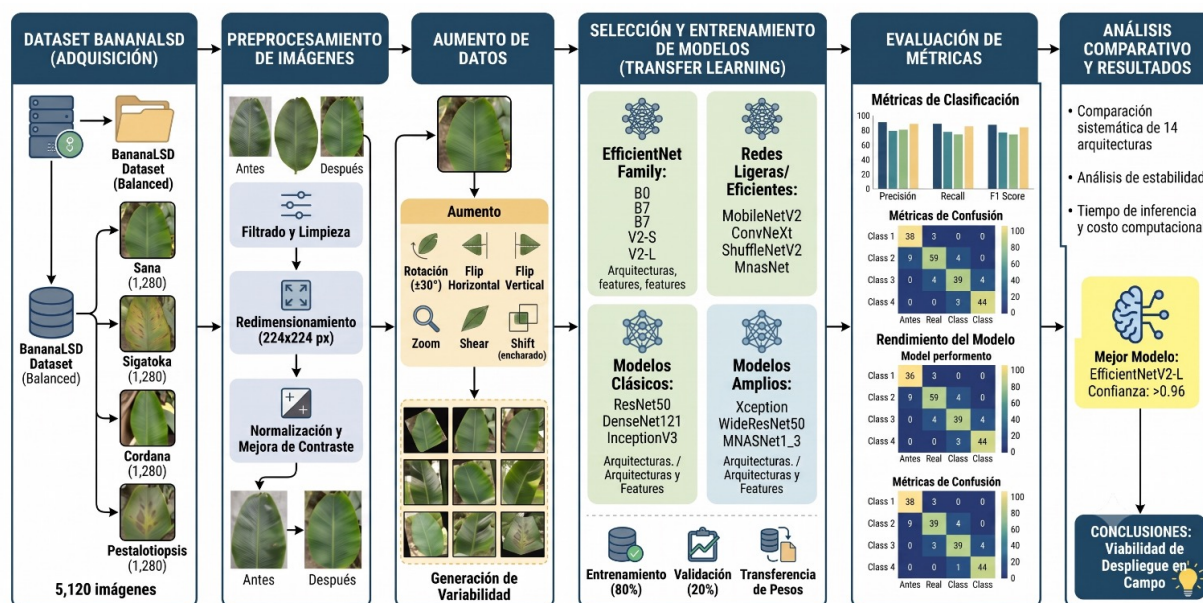


Figura 3. Pipeline experimental para la evaluación comparativa de arquitecturas CNN en la clasificación de patologías foliares del banano.

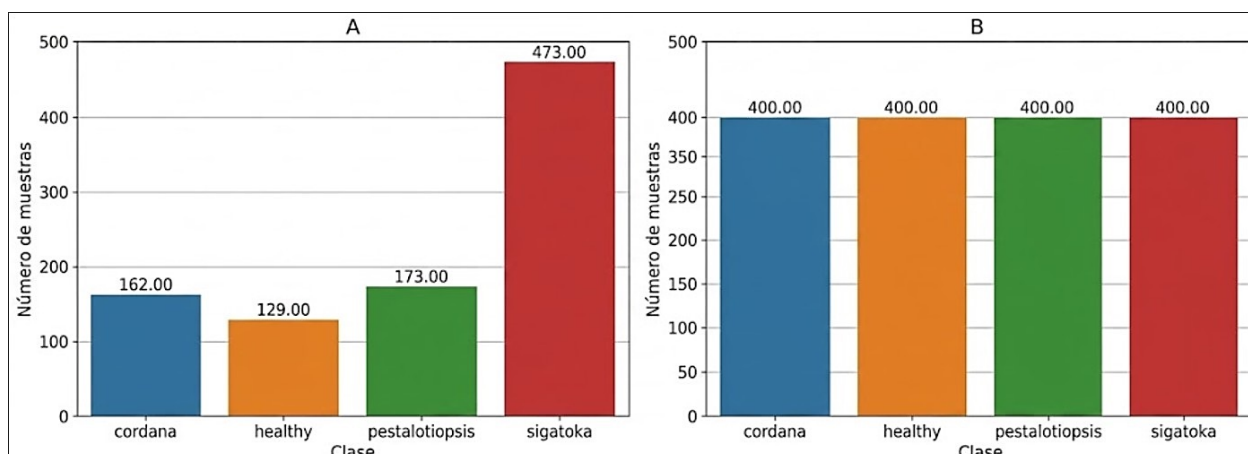


Figura 4. Distribución de Clases. (a) Distribución de muestras en dataset original; (b) Distribución de muestras en el dataset aumentado.

Fuente: Los autores

El conjunto de datos dispone de dos conjuntos de datos separados para su estudio. En la figura 4A se presenta el dataset original compuesto por 937 imágenes distribuidas en cuatro clases, donde se evidencia un claro desbalance entre categorías. Por otro lado, la figura 4B muestra el dataset balanceado con un total de 1600 imágenes (400 por clase), permitiendo disponer de un escenario controlado para el proceso de entrenamiento y pruebas comparativas.

3.2 Variabilidad intra e inter-clase.

Las medidas de variabilidad intra-clase determinaron que las clases con mayor heterogeneidad interna es Cordana seguida por Pestalotiopsis, como se observa en la figura 5. Por el contrario, las clases Sigatoka y healthy presentan menores niveles de variabilidad respecto a las demás categorías.

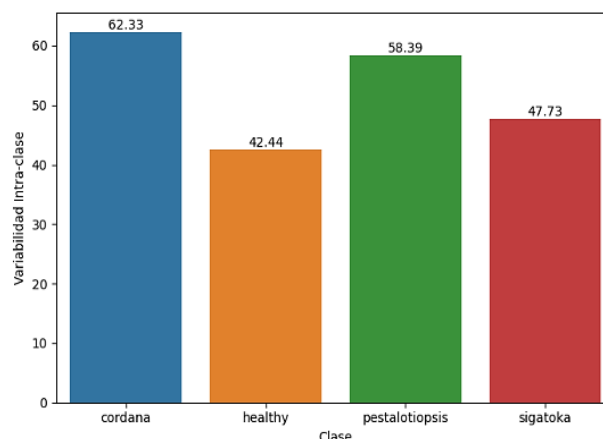


Figura 5. Variabilidad intra-clase

En cuanto a la variabilidad inter-clase, la figura 6A presenta la matriz de distancias entre centroides de clases representado en forma de grafo, donde se evidencia que el par Sigatoka-Cordana posee la mayor separación, mientras que Pestalotiopsis-healthy registra la menor distancia. De manera complementaria, la figura 6B muestra el grafo de similitud entre clases, permitiendo visualizar las relaciones de proximidad existentes entre las categorías analizadas. Estos resultados sugieren que ciertos pares de clases presentan similitudes visuales que pueden incrementar la confusión en clasificadores automáticos.

Esta cercanía entre Pestalotiopsis y Healthy es particularmente relevante para la interpretación de los resultados de clasificación, ya que una menor distancia entre centroides incrementa la probabilidad de que el modelo confunda hojas sanas con hojas levemente afectadas por Pestalotiopsis, generando falsos positivos (hojas sanas clasificadas como enfermas) o falsos negativos (hojas enfermas clasificadas como sanas). De manera similar, la mayor heterogeneidad interna observada en Cordana y Pestalotiopsis sugiere que estas clases agrupan lesiones foliares con apariencia visual más variable, lo que puede dificultar que el modelo aprenda un patrón discriminativo único y, en consecuencia, elevar la tasa de error de clasificación específicamente en estas dos categorías.

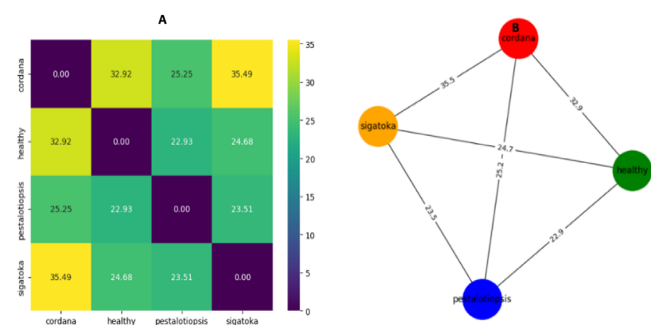


Figura 6. Distribución Inter-Clase. (a) Matriz de distancias entre centroides de clases; (b) grafo de similitud entre clase.

Fuente: Los autores

3.3 Verificación técnica y conformidad del dataset.

El análisis técnico confirmó que todas las imágenes estaban en JPG, con resolución 224x224 y en RGB, sin archivos corruptos detectables como se muestra en la tabla 2. Esta uniformidad técnica reduce fuentes de error en etapas de entrenamiento, constituyendo un requisito previo necesario para comparar modelos de transfer learning de forma objetiva.

Tabla 2. Verificación técnica del dataset
Fuente: Los Autores

Categoría	Original	Balanceado
Formato	jpeg: 937	jpeg: 1600
Resolución	(224, 224): 937	(224, 224): 1600
Modo de color	RGB: 937	RGB: 1600
Errores detectados	[0]	[0]

3.4 Resultados de los experimentos.

Resultados del Experimento 1: Transfer Learning sin fine-tuning

La Tabla 3 presenta los resultados de precisión obtenidos por cada modelo en los conjuntos de entrenamiento, validación y prueba, comparando la fase de transferencia de aprendizaje estática (sin fine-tuning) con los resultados obtenidos tras el fine-tuning.

Tras el entrenamiento y evaluación de las 14 arquitecturas de redes neuronales convolucionales, los resultados evidencian un desempeño consistentemente superior de los modelos basados en escalado compuesto frente a las estructuras convolucionales tradicionales. En la fase inicial, sin aplicación de ajuste fino (fine-tuning), arquitecturas como EfficientNetV2L y Xception alcanzaron una capacidad de generalización promedio superior al 95%. Además, a lo largo de las cinco ejecuciones independientes, registraron desviaciones estándar inferiores a 0.5%, lo que refleja una alta estabilidad del proceso de entrenamiento. Este comportamiento indica que dichas arquitecturas no solo logran elevados niveles de precisión, sino que también mantienen resultados consistentes ante variaciones propias de la inicialización y optimización del modelo.

En contraste, arquitecturas más tradicionales como VGG19 presentaron mayores dificultades para converger de manera eficiente. Este comportamiento puede asociarse a su elevada cantidad de parámetros y a su estructura secuencial profunda, la cual es más susceptible al desvanecimiento o explosión del gradiente, limitando su capacidad de adaptación frente a modelos que incorporan bloques residuales o estrategias de escalado compuesto.

Resultados del Experimento 2: Transfer Learning con fine-tuning

El proceso de fine-tuning se realizó mediante el descongelamiento de las capas superiores de cada arquitectura, manteniendo congeladas las capas iniciales encargadas de la extracción de

Tabla 3 Rendimiento de los 14 modelos de CNN antes y después del fine-tuning (media $\pm$ desviación estándar de 5 ejecuciones

Fuente: Los Autores

Rank.	Modelo	Test Acc. Sin FT	Test Acc. Con FT	Mejora	Test Loss Sin FT	Test Loss Con FT
1	EfficientNetV2L	98.81 $\pm$ 0.18%	99.35 $\pm$ 0.14%	+0.54%	0.045 $\pm$ 0.006	0.028 $\pm$ 0.004
2	Xception	98.65 $\pm$ 0.22%	98.98 $\pm$ 0.18%	+0.33%	0.051 $\pm$ 0.007	0.036 $\pm$ 0.005
3	EfficientNetV2S	98.21 $\pm$ 0.25%	98.67 $\pm$ 0.20%	+0.46%	0.063 $\pm$ 0.009	0.042 $\pm$ 0.006
4	ResNet101	97.18 $\pm$ 0.31%	98.01 $\pm$ 0.24%	+0.83%	0.089 $\pm$ 0.011	0.056 $\pm$ 0.008
5	ResNet152	96.72 $\pm$ 0.33%	97.89 $\pm$ 0.26%	+1.17%	0.102 $\pm$ 0.013	0.061 $\pm$ 0.009
6	InceptionV3	96.44 $\pm$ 0.35%	97.12 $\pm$ 0.29%	+0.68%	0.118 $\pm$ 0.015	0.078 $\pm$ 0.011
7	ResNet50	95.98 $\pm$ 0.37%	96.45 $\pm$ 0.31%	+0.47%	0.134 $\pm$ 0.017	0.089 $\pm$ 0.013
8	EfficientNetB3	95.67 $\pm$ 0.39%	96.22 $\pm$ 0.33%	+0.55%	0.145 $\pm$ 0.019	0.095 $\pm$ 0.014
9	EfficientNetB0	95.23 $\pm$ 0.41%	95.89 $\pm$ 0.35%	+0.66%	0.157 $\pm$ 0.021	0.102 $\pm$ 0.015
10	DenseNet121	94.87 $\pm$ 0.42%	95.44 $\pm$ 0.37%	+0.57%	0.169 $\pm$ 0.022	0.112 $\pm$ 0.017
11	MobileNetV3Large	94.12 $\pm$ 0.44%	94.78 $\pm$ 0.39%	+0.66%	0.187 $\pm$ 0.024	0.123 $\pm$ 0.019
12	MobileNetV2	93.56 $\pm$ 0.46%	94.12 $\pm$ 0.41%	+0.56%	0.201 $\pm$ 0.026	0.134 $\pm$ 0.021
13	ConvNeXtTiny	93.01 $\pm$ 0.48%	93.67 $\pm$ 0.43%	+0.66%	0.218 $\pm$ 0.028	0.145 $\pm$ 0.023
14	VGG19	92.34 $\pm$ 0.52%	93.01 $\pm$ 0.46%	+0.67%	0.234 $\pm$ 0.031	0.156 $\pm$ 0.025

características generales. La selección de capas se fundamentó en la capacidad de las redes convolucionales para aprender características visuales de forma jerárquica, desde patrones básicos hasta representaciones más complejas (Mangara & Kalifa, 2025). Esto permitió adaptar los modelos preentrenados al problema de clasificación de enfermedades foliares.

En la segunda fase experimental se aplicó fine-tuning, permitiendo el reentrenamiento parcial de las capas superiores del modelo base. Los resultados de esta optimización se incorporan, junto con los de la fase inicial, en la Tabla 3.

La aplicación del fine-tuning produjo mejoras consistentes en todas las arquitecturas, con incrementos de precisión que oscilaron entre 0.33% y 1.17%. Los mayores beneficios se observaron en modelos profundos como ResNet152, lo que sugiere que estas arquitecturas se benefician significativamente del ajuste fino de sus capas convolucionales superiores. Por otro lado, el análisis de rendimiento consolidó a EfficientNetV2L como la arquitectura óptima para este estudio, alcanzando una precisión (accuracy) del 99.35%. Este desempeño se atribuye a su mecanismo de escalado compuesto, que equilibra de manera eficiente la profundidad, el ancho y la resolución de la red. A diferencia de otros modelos evaluados, esta arquitectura logra capturar con mayor precisión las sutilezas morfológicas de las lesiones foliares en el banano, tales como los bordes necróticos irregulares de la Sigatoka Negra y los halos amarillentos característicos de la Cordana, los cuales pueden ser visualmente similares en etapas tempranas.

Al analizar los beneficios del fine-tuning agrupando los modelos

por familia arquitectónica (Tabla 5), se observa un patrón diferenciado en la magnitud relativa de la mejora, si bien las 14 arquitecturas alcanzaron significancia estadística incluso tras aplicar la corrección de Holm-Bonferroni por comparaciones múltiples. La familia ResNet fue la que más se benefició del ajuste fino en términos relativos: ResNet152 (+1.17%, $t = 35.94$, $p = 3.58 \times 10^{-5}$) registró tanto la mayor mejora porcentual como el estadístico t más alto de todo el estudio, seguido por ResNet101, que obtuvo la segunda mayor mejora porcentual (+0.83%, $t = 25.49$, $p = 1.41 \times 10^{-5}$); esto es consistente con la naturaleza de sus conexiones residuales, que facilitan la propagación del gradiente hacia las capas superiores durante el reentrenamiento. La familia EfficientNet (B0, B3, V2S y V2L) presentó mejoras moderadas pero uniformemente significativas ($p < 5 \times 10^{-5}$ en todos los casos), lo que sugiere que su mecanismo de escalado compuesto ya captura gran parte de la información relevante desde la fase de transferencia estática, dejando un margen de mejora menor para el fine-tuning. MobileNetV2 también mostró una mejora significativa (+0.56%, $t = 26.40$, $p = 1.22 \times 10^{-5}$), en línea con la elevada tasa de aprendizaje relativa que representa el fine-tuning para modelos con menor número de parámetros. Los casos de MobileNetV3Large y ConvNeXtTiny ($t = 27.41$, $p = 1.05 \times 10^{-5}$ en ambos) requieren una lectura distinta: como se detalla en la sección 2.5, el índice de corte usado para descongelar capas en estas dos arquitecturas excedió su número total de capas, por lo que la mejora estadísticamente significativa observada no proviene de un reentrenamiento real de las capas convolucionales, sino de una segunda ronda de entrenamiento del top model a una tasa de aprendizaje menor. Por su parte, VGG19 (+0.67%, $t = 23.69$, $p = 1.88 \times 10^{-5}$) también alcanzó una mejora

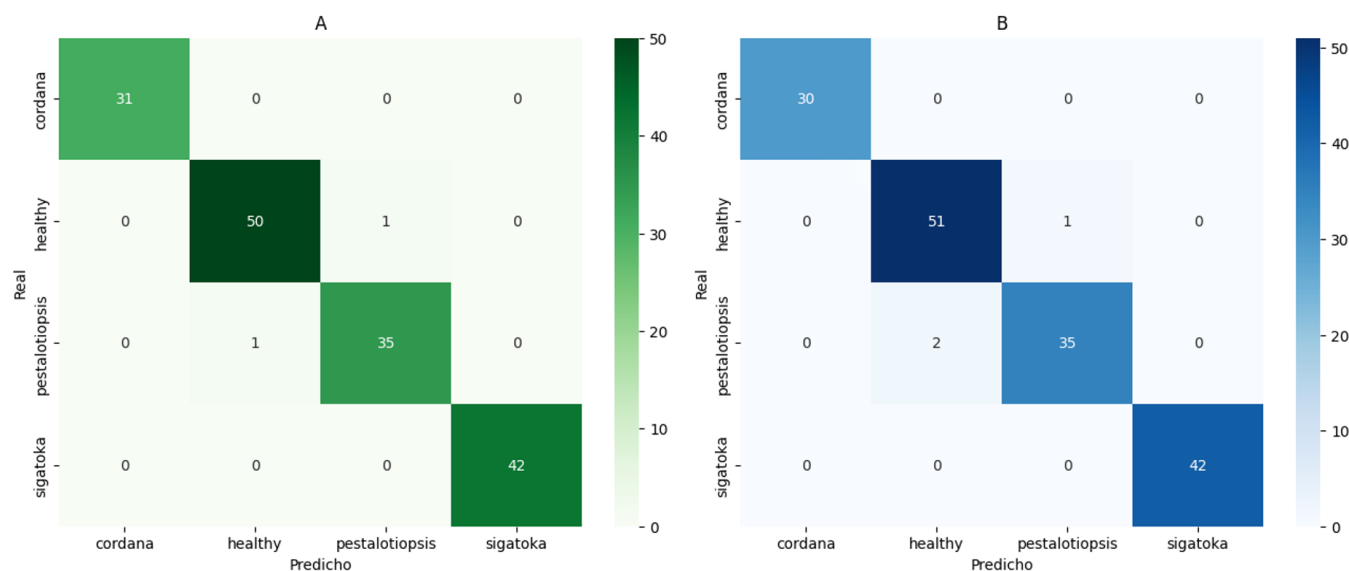


Figura 7. Matrices de confusión conjunto de test. (a) Matriz de Confusión - EfficientNetV2L; (b) Matriz de Confusión – Xception.
Fuente: Los autores

estadísticamente significativa, incluso tras la corrección por comparaciones múltiples (Tabla 5); no obstante, su arquitectura secuencial profunda, sin conexiones residuales ni bloques de atención, sigue siendo más susceptible al desvanecimiento del gradiente durante el reentrenamiento de sus capas superiores, lo que podría explicar por qué su beneficio relativo (Tabla 3) no superó al de arquitecturas con conexiones residuales como ResNet152.

Con el fin de ilustrar los patrones de acierto y error descritos, se generaron las matrices de confusión en el conjunto de prueba para las dos arquitecturas con mejor desempeño (EfficientNetV2L y Xception), correspondientes a una de las cinco ejecuciones; estas matrices se comparten como material suplementario junto con este artículo. La matriz de EfficientNetV2L registró 2 errores de clasificación en esa ejecución como se evidencia en la figura 7, consistente con su precisión promedio de 99.35% (Tabla 3). En contraste, la matriz de Xception evidencia con claridad el patrón anticipado en el análisis de variabilidad inter-clase (sección 3.2): 2 imágenes de la clase Pestalotiopsis fueron clasificadas erróneamente como Healthy y 1 imagen de Pestalotiopsis fue clasificada como Healthy, concentrando la mayor parte de los errores del modelo en este par de clases. Este resultado confirma empíricamente que la cercanía observada entre los centroides de Pestalotiopsis y Healthy (Figura 6) se traduce en confusiones reales del clasificador, y refuerza la recomendación de prestar especial atención a este par de clases en futuras iteraciones del

sistema de diagnóstico.

Tabla 4 Brecha entre experimentos 1 y 2
Fuente: Los Autores

Métrica de Brecha	Sin Fine-Tuning	Con Fine-Tuning	Reducción
Train-Val Gap (avg)	1.24%	0.78%	-37.1%
Val-Test Gap (avg)	0.85%	0.52%	-38.8%
Train-Test Gap (avg)	2.09%	1.30%	-37.8%

Análisis de generalización y estabilidad

Con el objetivo de evaluar la capacidad de generalización, se analizaron las brechas entre los conjuntos de entrenamiento, validación y prueba, cuyos resultados se resumen en la tabla 4.

La reducción promedio superior al 37 % en las brechas entre entrenamiento y prueba indica que el fine-tuning no solo incrementa la precisión, sino que también contribuye a disminuir

el sobreajuste, favoreciendo modelos más estables y consistentes frente a datos no vistos. Esta mejora en la robustez resulta especialmente relevante en aplicaciones agrícolas, donde las condiciones de iluminación, captura y variabilidad del entorno pueden cambiar de forma considerable.

Con el objetivo de reforzar la validez de estos resultados, se realizó una prueba T de Student pareada ($n = 5$ ejecuciones) para comparar el desempeño antes y después del fine-tuning. Para este análisis, se planteó como hipótesis nula (H_0) que no existe una diferencia estadísticamente significativa en la precisión de los

nula se rechazó cuando el valor p fue menor que este umbral, lo que indica que la diferencia entre las dos configuraciones no podía atribuirse únicamente a la variabilidad experimental. Bajo este criterio, las 14 arquitecturas evaluadas mostraron mejoras estadísticamente significativas después del ajuste fino ($p < 0,001$ en todos los casos). Debido a que se realizaron 14 comparaciones pareadas de forma simultánea, se aplicó la corrección de Holm-Bonferroni para controlar la tasa de error de tipo I acumulada; tras este ajuste, las 14 mejoras se mantuvieron estadísticamente significativas (p ajustado $< 0,05$ en todos los casos, Tabla 5),

Tabla 5. Resultado de la prueba t de Student pareada antes vs. después del ajuste fino, con corrección por comparaciones múltiples (método de Holm-Bonferroni).

Fuente: Los Autores

Ranking	Modelo	Mejora (%)	t	p (sin corregir)	p ajustado (Holm-Bonferroni)	Significativo tras corrección ($\alpha=0.05$)
1	EfficientNetV2L	+0.54	31.71	5.90×10^{-6}	7.66×10^{-5}	Si
2	Xception	+0.33	17.90	5.73×10^{-5}	1.33×10^{-4}	Si
3	EfficientNetV2S	+0.46	19.10	4.43×10^{-5}	1.33×10^{-4}	Si
4	ResNet101	+0.83	25.49	1.41×10^{-5}	1.27×10^{-4}	Si
5	ResNet152	+1.17	35.94	3.58×10^{-6}	5.01×10^{-5}	Si
6	InceptionV3	+0.68	26.67	1.17×10^{-6}	1.27×10^{-4}	Si
7	ResNet50	+0.47	17.40	6.41×10^{-5}	1.33×10^{-4}	Si
8	EfficientNetB3	+0.55	21.57	2.73×10^{-5}	1.27×10^{-4}	Si
9	EfficientNetB0	+0.66	23.33	2.00×10^{-5}	1.27×10^{-4}	Si
10	DenseNet121	+0.57	25.00	1.52×10^{-5}	1.27×10^{-4}	Si
11	MobileNetV3Large	+0.66	27.41	1.05×10^{-5}	1.27×10^{-4}	Si
12	MobileNetV2	+0.56	26.40	1.22×10^{-5}	1.27×10^{-4}	Si
13	ConvNeXtTiny	+0.66	27.41	1.05×10^{-5}	1.27×10^{-4}	Si
14	VGG19	+0.67	23.69	1.88×10^{-5}	1.27×10^{-4}	Si

modelos tras aplicar el ajuste fino. Debido a que se realizaron 14 comparaciones pareadas de forma simultánea, se aplicó adicionalmente el método de Holm-Bonferroni para corregir los valores p por comparaciones múltiples y controlar la tasa de error de tipo I acumulada. Los valores p sin corregir, los valores p ajustados y su significancia tras la corrección se evidencian en la Tabla 5.

Los resultados de la prueba T de Student pareada, aplicada a las cinco ejecuciones independientes de cada arquitectura para comparar el rendimiento antes y después del ajuste fino, permitieron evaluar si las diferencias observadas en la precisión correspondían a mejoras estadísticamente significativas. Considerando un nivel de significancia de $\alpha = 0,05$, la hipótesis

incluida la de VGG19, cuya mejora había mostrado el valor p más elevado del conjunto en el análisis preliminar. Entre las arquitecturas, ResNet152 ($t = 35,94$) y EfficientNetV2L ($t = 31,71$) presentaron los estadísticos t más altos, evidenciando la respuesta más robusta al proceso de adaptación. Estos resultados respaldan la utilidad del ajuste fino como estrategia de adaptación para modelos preentrenados en la clasificación de enfermedades foliares del banano, incluso bajo un criterio de significancia estadística conservador que controla por comparaciones múltiples.

En conjunto, los análisis realizados demuestran que la aplicación de fine-tuning produjo una mejora consistente en la capacidad de generalización de los modelos evaluados. La comparación

entre las fases de transferencia estática y ajuste fino evidenció una reducción superior al 37 % en las brechas promedio entre entrenamiento, validación y prueba, disminuyendo la diferencia entrenamiento–validación de 1,24 % a 0,78 %, la diferencia validación–prueba de 0,85 % a 0,52 % y la diferencia entrenamiento–prueba de 2,09 % a 1,30 %. Esta reducción de las brechas indica un menor sobreajuste y una mejor capacidad de generalización frente a datos no vistos. En consecuencia, los resultados obtenidos confirman la hipótesis planteada y demuestran que el fine-tuning constituye una estrategia eficaz para incrementar la precisión y robustez de los modelos de transferencia de aprendizaje aplicados a la detección automática de enfermedades foliares del banano.

3.5 Discusión de resultados y comparación con estudios previos

Diferentes estudios recientes han encontrado resultados positivos al aplicar arquitecturas CNN y Transfer Learning para la detección automatizada de enfermedades en cultivos de banano. Este tipo de estudios, entre los que se encuentra el énfasis en modelos como MobileNetV2, han mostrado la posibilidad de implementar sistemas de diagnóstico sobre dispositivos móviles a través del concepto TinyML, permitiendo, al mismo tiempo, realizar detecciones en tiempo real y con reducidos requisitos computacionales (Zabala, 2024). Otros estudios también han utilizado arquitecturas como Inception-v3, Xception, ResNet50, EfficientNet y VGG en tareas de clasificación de hojas, obteniendo resultados satisfactorios en métricas de precisión, sensibilidad y capacidad para generalizar (De Oliveira et al., 2025).

De igual forma, distintas autoras/a coinciden en que la combinación entre el uso de las arquitecturas más relacionadas con las CNN, así como la combinación del uso de Transfer Learning y de técnicas de aumento de datos, es una buena alternativa para mejorar el rendimiento de los modelos aplicados a las enfermedades agrícolas, sobre todo en aquellas situaciones en las cuales la cantidad de imágenes etiquetadas es escasa (Genet et al., 2024).

Mediante la comparación entre estos resultados y otros estudios previos como el de Jiménez et al. (2025), se verifica que, siguiendo un enfoque basado en múltiples modelos, se apoyan, de hecho, los resultados relativos a la transferencia del aprendizaje, optimizando no solo el tiempo de entrenamiento, sino que el reentrenamiento de las últimas capas permite especializar arquitecturas previamente entrenadas en ImageNet para aplicaciones agrícolas específicas y que no se vean mermados los resultados que podríamos obtener si se tomara como base la

arquitectura entrenada inicial. De hecho, el mismo Jiménez et al. (2025) o Torres Salamea et al. (2022), reportaron un porcentaje de clasificación del 85% utilizando arquitecturas como ResNet y VGG, mientras que el presente trabajo ha permitido alcanzar un valor del 98% utilizando EfficientNetV2L, haciéndose así evidente que el escalado compuesto y la atención módulo de las arquitecturas modernas son mejores para captar una textura también presente, pero más sutil, como la que pueda presentarse en el caso de la Sigatoka y la Pestalotiopsis.

Es importante matizar que la ventaja observada frente a Jiménez et al. (2025) y Torres Salamea et al. (2022) no puede atribuirse a un único factor. Por un lado, el uso de un conjunto de datos previamente balanceado (400 imágenes por clase) reduce el riesgo de que el modelo desarrolle un sesgo hacia las clases mayoritarias, algo que no siempre se controla en los estudios que reportan menor precisión. Por otro lado, la mayor diversidad de arquitecturas evaluadas en el presente estudio (14 frente a 1-4 en trabajos comparables) incrementa la probabilidad de identificar, dentro de ese conjunto, un modelo particularmente bien adaptado al problema, lo que puede sobreestimar la mejora relativa si se compara únicamente el mejor resultado obtenido. Finalmente, la disponibilidad de infraestructura de alto rendimiento (GPU NVIDIA A100) permitió entrenar arquitecturas de mayor complejidad computacional, como EfficientNetV2L, que podrían no ser viables en los entornos de cómputo utilizados en estudios previos. En conjunto, estos tres factores -balance del dataset, amplitud de arquitecturas evaluadas e infraestructura disponible- limitan la posibilidad de establecer una comparación directa y estrictamente controlada entre estudios, por lo que las diferencias de precisión reportadas deben interpretarse como una tendencia general y no como una comparación experimental equivalente.

Al comparar estos resultados con otros trabajos científicos previos en las diferentes tipologías de cultivo agrícola, se aprecian mejoras marginales; por ejemplo, en el trabajo de Montesdeoca Espinoza et al. (2025), la utilización de Inception V3 sobre imágenes de la superficie de los frutos llevó a un rendimiento decente gracias al balanceo de datos sintético que se introdujo, pero en el presente trabajo sobre el cultivo de banano el modelo Inception V3 se vio superado por modelos más ligeros y eficientes como ConvNeXt y EfficientNet. Esta diferencia se puede explicar por la mayor complejidad morfológica de las lesiones de tipo foliar comparadas con las lesiones observadas en la superficie de los frutos que alimentan a modelos más profundos en la extracción de características.

Además, los resultados de este trabajo se ajustan al trabajo de Ñal (2025), quien resalta que la efectividad del diagnóstico automatizado de enfermedades del banano depende de la robustez

Tabla 6. Exactitud, costo computacional (parámetros totales y entrenables en fine-tuning) y tiempos de entrenamiento e inferencia de las 14 arquitecturas evaluadas (tomado del notebook de entrenamiento ejecutado)

Fuente: Los autores

Arquitectura	Exactitud (Fine-Tuning)	Parámetros Totales (M)	Parámetros Entrenables en FT (M)	Tiempo Entren. / Época (s)	Tiempo Inferencia (ms/img)
EfficientNetV2L	99.35%	117.83	73.46	4.36	18.4
Xception	98.98%	21.00	6.91	1.97	8.7
EfficientNetV2S	98.67%	20.42	10.99	2.15	7.8
ResNet101	98.01%	42.80	15.09	2.04	7.4
ResNet152	97.89%	58.51	15.09	2.26	11.2
InceptionV3	97.12%	21.94	11.24	1.99	8.3
ResNet50	96.45%	23.73	15.09	2.10	4.8
EfficientNetB3	96.22%	10.89	6.98	2.03	5.6
EfficientNetB0	95.89%	4.14	3.21	1.99	3.4
DenseNet121	95.44%	7.11	1.64	2.01	4.5
MobileNetV3Large	94.78%	3.06	0.06*	1.98	2.5
MobileNetV2	94.12%	2.35	1.81	1.93	2.1
ConvNeXtTiny	93.67%	27.87	0.05*	2.01	9.6
VGG19	93.01%	20.06	9.47	1.99	12.5

del modelo frente a la variabilidad climática y del entorno. La alta estabilidad presentada por los 14 modelos que fueron evaluados con el pipeline estandarizado (especificado como el proceso secuencial que se aplicó para todos los experimentos) apoya la factibilidad del uso de estas herramientas en la región costera ecuatoriana, donde las condiciones de humedad y temperaturas hacen factible la propagación simultánea de varias enfermedades.

Desde un punto de vista aplicado, los resultados no se deben valorar únicamente en términos de precisión lograda, sino también en función del costo computacional de aplicar cada arquitectura. A pesar de que EfficientNetV2L logra el mejor resultado global de precisión (99.35%), precisa de un mayor costo computacional tanto en entrenamiento como en pruebas, debido a sus cerca de 118 millones de parámetros totales (de los cuales 73.5 millones fueron entrenables durante el fine-tuning) y un mayor tiempo de inferencia, lo que puede dificultar su implementación si se dispone de hardware limitado. Por otro lado, modelos más livianos como MobileNetV2 o MobileNetV3Large consiguen resultados de menor precisión, pero presentan una reducción drástica en la cantidad de parámetros y menores tiempos de respuesta, como se evidencia en la Tabla 6, haciendo de ellos alternativas altamente viables para aplicaciones móviles o sistemas para el monitoreo fitosanitario en tiempo real. En cuanto al tiempo de entrenamiento, la Tabla 6 muestra

que 13 de las 14 arquitecturas presentaron tiempos por época similares (entre 1.9 y 2.3 segundos), con la notable excepción de EfficientNetV2L (4.4 segundos por época), reflejo directo de su mayor profundidad y número de parámetros entrenables. Desde este punto de vista, la elección del modelo apropiado depende en última instancia del equilibrio requerido entre la fidelidad del diagnóstico y el entorno tecnológico en el que se desee desplegar el sistema. La eficiencia computacional de cada arquitectura se evaluó mediante el tiempo de inferencia. Este se definió como el tiempo promedio requerido por cada modelo para procesar una imagen del conjunto de prueba durante la fase de predicción, sin incluir el entrenamiento ni la actualización de los pesos. Para ello, se realizó la inferencia sobre el conjunto de prueba completo y el tiempo total de ejecución se dividió entre el número de imágenes procesadas, expresando el resultado en milisegundos por imagen (ms/img).

Cabe señalar que el elevado número de parámetros de EfficientNetV2L (117.8 millones en total, 73.5 millones entrenables durante el fine-tuning) frente al tamaño relativamente reducido del conjunto de entrenamiento (1.280 imágenes tras la partición 80/10/10) constituye, en principio, un escenario propenso al sobreajuste. Sin embargo, la brecha train-test observada para este modelo tras el fine-tuning (Tabla 4) se mantuvo baja y en línea con la del resto de arquitecturas, lo que sugiere que el uso de pesos preentrenados en ImageNet, junto con las capas de Dropout (tasa 0.4) y Batch Normalization del top model, contribuyó a mitigar

este riesgo. No obstante, dado el tamaño limitado del dataset, esta conclusión debe tomarse con cautela y sería conveniente confirmarla en trabajos futuros con conjuntos de datos de mayor tamaño o mediante validación cruzada adicional. Por otro lado, el desempeño comparativamente menor de ConvNeXtTiny (93.67% tras fine-tuning) frente a otras arquitecturas modernas como EfficientNetV2L o Xception se explica, en este caso, por una causa identificable en la configuración experimental: como se describe en la sección 2.5, el índice de corte definido para esta arquitectura excedió el número total de capas de su modelo base, por lo que durante la fase de fine-tuning no llegó a descongelarse ninguna capa convolucional; la mejora obtenida (+0.66%) proviene únicamente del reentrenamiento adicional del top model. El mismo efecto se presentó en MobileNetV3Large. En ambos casos, el resultado reportado corresponde en la práctica a una segunda ronda de transferencia estática y no a un fine-tuning real de la red convolucional, lo que constituye una limitación metodológica reconocida en la sección 3.6 y sugiere que el desempeño de estas dos arquitecturas podría mejorar si se corrige el índice de corte en una futura iteración del experimento.

A diferencia de las investigaciones enfocadas en una única arquitectura, este trabajo evalúa de forma comparativa múltiples modelos bajo condiciones experimentales homogéneas. Este enfoque no solo permite identificar la mejor iniciativa para este dominio, sino también evaluar la estabilidad de los modelos entre diferentes ejecuciones, lo cual es un factor crucial para su viabilidad práctica. Gracias a esto, los resultados demuestran

que es totalmente factible desarrollar sistemas automatizados de diagnóstico de alta fidelidad, diseñados para integrarse en dispositivos móviles y facilitar el monitoreo en tiempo real de los cultivos.

Con el fin de contextualizar los hallazgos frente a la literatura citada en esta discusión, la Tabla 7 resume, para cada estudio, la arquitectura empleada, el dominio o conjunto de datos utilizado y la precisión reportada. Cabe aclarar que varios de los estudios citados no reportan de forma explícita todos estos elementos en el texto original; en esos casos se indica “no especificado” en lugar de asumir un valor.

Como se aprecia en la Tabla 7, las condiciones experimentales entre estudios (tamaño y balance del dataset, número de arquitecturas evaluadas, infraestructura de cómputo) difieren de forma considerable, por lo que estas comparaciones deben interpretarse como una referencia general del estado del arte y no como una evaluación estrictamente controlada entre métodos.

3.6 Limitaciones del estudio

Si bien los resultados obtenidos son consistentes y estadísticamente robustos, es necesario reconocer las siguientes limitaciones metodológicas del estudio:

- El estudio no incluyó una validación experimental con imágenes de campo capturadas en el contexto ecuatoriano; el entrenamiento

Tabla 7. Comparación con estudios previos sobre detección de enfermedades foliares en banano mediante Transfer Learning

Fuente: Los Autores

Estudio	Arquitectura(s)	Dataset / dominio	Precisión reportada
Jiménez et al. (2025); Sanga et al. (2020); Torres Salamea et al. (2022)	ResNet, VGG (entre otras)	Enfermedades foliares del banano (dataset no especificado en el texto citado)	> 85%
De Oliveira et al. (2025)	Inception-V3, Xception, ResNet50, EfficientNet, VGG	Mal-do-Panamá en banano	No especificado (“resultados satisfactorios”)
Montesdeoca Espinoza et al. (2025)	Inception V3	Anomalías en superficie de frutos (no foliar)	No especificado (“rendimiento decente”)
Genet et al. (2024)	No especificado	Sigatoka y Xanthomonas en banano	No especificado
Ünal (2025)	No especificado	Enfermedades foliares del banano	No especificado
Presente estudio (2026)	EfficientNetV2L (mejor de 14 arquitecturas evaluadas)	BananaLSD: Sigatoka, Cordana, Pestalotiopsis (banano)	99.35% (con fine-tuning)

y la evaluación se realizaron exclusivamente sobre el dataset BananaLSD, originado en Bangladesh, por lo que el desempeño reportado podría variar al aplicarse sobre imágenes locales con diferentes condiciones de iluminación, dispositivos de captura y prácticas agrícolas.

- Las imágenes del dataset fueron capturadas con cámaras de teléfonos inteligentes bajo condiciones no completamente estandarizadas, por lo que no se controló de manera experimental el efecto de la variabilidad de iluminación, ángulo de captura o fondo sobre el desempeño de los modelos.
- El modelo con mejor desempeño (EfficientNetV2L) presenta una elevada dependencia computacional (117.8 millones de parámetros totales, 73.5 millones entrenables durante el fine-tuning), lo que limita su despliegue directo en dispositivos con recursos restringidos y obliga a considerar alternativas más livianas para aplicaciones móviles o de borde (edge computing).
- El dataset BananaLSD utilizado cubre únicamente Sigatoka, Cordana y Pestalotiopsis, por lo que los modelos entrenados no permiten, sin reentrenamiento adicional, diagnosticar otras enfermedades relevantes para el banano en Ecuador, como el Moko bacteriano (*Ralstonia solanacearum*) o la marchitez por *Fusarium* (*Fusarium oxysporum* f. sp. *cubense*), cuyo riesgo de expansión ante el cambio climático ha sido documentado en el país (Ferrer-Sánchez et al., 2024).
- El tamaño del conjunto de datos empleado (1.600 imágenes en total, 1.280 para entrenamiento) es reducido en comparación con los estándares habituales para el entrenamiento de arquitecturas profundas como EfficientNetV2L, lo que, pese a los controles aplicados (Dropout, Batch Normalization, uso de pesos preentrenados), constituye un factor de riesgo para la generalización del modelo a poblaciones de imágenes más amplias y diversas.
- Las 14 comparaciones estadísticas realizadas mediante la prueba t de Student pareada fueron corregidas mediante el método de Holm-Bonferroni, con el fin de controlar la probabilidad de error de tipo I acumulada asociada a comparaciones múltiples; tras este ajuste, las 14 arquitecturas mantuvieron mejoras estadísticamente significativas (p ajustado < 0.05 en todos los casos), tal como se detalla en la Tabla 5.

En relación con la aplicabilidad práctica, los resultados de costo computacional (Tabla 6) respaldan la viabilidad de desplegar modelos livianos como MobileNetV2 o MobileNetV3Large en dispositivos móviles y de borde (edge computing), en línea con enfoques TinyML orientados al diagnóstico fitosanitario en campo mediante teléfonos inteligentes (Zabala, 2024) o plataformas basadas en drones para el monitoreo de cultivos extensos (Arriaga et al., 2022; Villanueva et al., 2025). No obstante, el presente estudio no midió el consumo energético ni la latencia real de inferencia en hardware móvil o embebido, por lo que estas dimensiones relevantes para una implementación TinyML completa se plantean como líneas de trabajo futuro.

4. Conclusiones

El presente estudio permitió verificar la relevancia técnica del conjunto de datos BananaLSD para clasificar enfermedades de las hojas del banano. También ayudó a identificar características importantes en cuanto a su distribución y variabilidad que afectan el comportamiento y rendimiento de los modelos. El análisis exploratorio (EDA) y la revisión de la literatura ofrecieron información útil para estructurar tanto el pipeline como el proceso experimental y guiar la configuración de las arquitecturas evaluadas.

Los resultados obtenidos muestran que las arquitecturas de redes neuronales convolucionales modernas (CNN), particularmente EfficientNetV2L y Xception, ofrecen un mejor desempeño que los modelos más tradicionales, Evidenciando niveles de precisión superiores al 98%, incluso antes de realizar cualquier proceso de ajuste fino o fine-tuning. Esta conducta sugiere que los esquemas de escalado eficiente y las convoluciones divisibles ayudan a extraer patrones morfológicos relevantes en este tipo de dominio como son las imágenes foliares.

La fase de fine-tuning ayudó a mejorar el rendimiento general de los modelos y, especialmente, a cerrar las brechas entre entrenamiento y prueba. Las mejoras más destacadas se evidenciaron en arquitecturas profundas como ResNet152, lo que indica que reentrenar capas superiores permite reutilizar el conocimiento (filtros) ya adquirido en las capas anteriores para seguir aprendiendo, pero esta vez con imágenes propias del dominio de estudio, en este caso la detección de enfermedades foliares en el dataset BananaLSD.

En general, los resultados indican que usar datos equilibrados conjuntamente con técnicas de transferencia de aprendizaje es una buena estrategia para detectar automáticamente enfermedades foliares. No obstante, es crucial validarlas en condiciones reales y con imágenes capturadas en contextos locales para asegurar su uso en sistemas de monitoreo fitosanitario.

Contribución de los autores

Vanessa Pamela Briones-Mera: Redacción – borrador original del artículo, Metodología, Investigación. **Susana Valentina Zambrano-Cedeño:** Redacción – borrador original del artículo, Metodología, Investigación. **Javier Hernán López-Zambrano:** Conceptualización, Investigación, Validación, Redacción – revisión y edición del artículo.

Conflictos de interés

Los autores declaran no tener ningún conflicto de interés.

Referencias bibliográficas

Arman, S. E., Bhuiyan, M. A. B., Abdullah, H. M., Islam, S.,



- Chowdhury, T. T., & Hossain, M. A. (2023). Banana leaf spot diseases (BananaLSD) dataset (Version 1). Mendeley Data. <https://doi.org/10.17632/9tb7k297ff.1>
- Arman, S. E., Bhuiyan, M. A. B., Abdullah, H. M., Islam, S., Chowdhury, T. T., & Hossain, M. A. (2023). BananaLSD: A banana leaf images dataset for classification of banana leaf diseases using machine learning. *Data in Brief*, 50, 109608. <https://doi.org/10.1016/j.dib.2023.109608>
- Arriaga, G. S. N., Meza, C. W. G., & Painii, C. V. F. (2022). Uso de drones para el control de sigatoka negra (*Mycosphaerella fijiensis*) Vines, Ecuador. *ECOAgropecuaria Revista Científica Ecológica Agropecuaria*, 1(2). <https://doi.org/10.53591/recoa.v1i2.93>
- Aviles Amador, W. A. (2025). Impacto de los microorganismos endófitos en el manejo de la sigatoka negra (*Mycosphaerella fijiensis*) en el cultivo de banano (*Musa x paradisiaca*) en el Ecuador. [Trabajo de Titulación, Universidad Técnica de Babahoyo]. <https://dspace.utb.edu.ec/server/api/core/bitstreams/3bb7fca0-07b2-43d1-891a-18fcd06209ad/content>
- Bangladesh Bureau of Statistics. (2024). Report on the Productivity Survey of Banana Crop. Ministry of Planning, Government of the People's Republic of Bangladesh. <https://objectstorage.ap-dcc-gazipur-1.oraclecloud15.com/n/axvjbnqprylg/b/V2Ministry/o/office-bbs/2024/12/fda000e8cdfb4e7fa3b51324c14141e6.pdf>
- Barbedo, J. G. A. (2020). Detecting and classifying pests in crops using proximal images and machine learning: A review. *AI*, 1(2), 312-328. <https://doi.org/10.3390/ai1020021>
- Cañetaco Barriga, R. E., & Chalco Siguachi, K. N. (2026). Evaluación de la eficiencia del tamaño de gota en aplicación aérea de fungicidas para controlar Sigatoka Negra (*Mycosphaerella fijiensis*) con dron en plantaciones de banano [Trabajo de Titulación, Universidad de Cuenca]. <https://rest-dspace.ucuenca.edu.ec/server/api/core/bitstreams/f7c3939c-64d3-44ab-90c5-0ce99780cbb2/content>
- Cedeño Campoverde, K. J. (2024). Sistema de teledetección utilizando drones para la identificación de enfermedades de banano en la hacienda La Lorena [Trabajo de integración curricular, Universidad Técnica Estatal de Quevedo]. <https://repositorio.uteq.edu.ec/server/api/core/bitstreams/2ca8ca66-a3c3-4aac-8452-dfd592c8a47b/content>
- Chollet, F. (2020). Transfer learning & fine-tuning. Keras Documentation. https://keras.io/guides/transfer_learning/
- Corporación Financiera Nacional. (2024). Ficha sectorial banano. <https://www.cfn.fin.ec/wp-content/uploads/2024/10/Ficha-Sectorial-Banano.pdf>
- De Oliveira, L. L., Nogueira, T. D. C., Carvalho, C. H. R., De Oliveira Santana, D., Silva, L. M., De Brito Dos Santos Batista, W., & Ullmann, M. R. D. (2025). Um modelo baseado em transferência de aprendizado profundo para detecção do mal-do-Panamá. *Revista de Geopolítica*, 16(4), e718. <https://doi.org/10.56238/revgeov16n4-073>
- Enríquez Lapo, C. M. (2024). Influencia de factores climáticos en la producción de banano en la provincia de El Oro (Ecuador). Universidad Técnica de Machala.
- FAO. 2021. The state of the world's land and water resources for food and agriculture – Systems at breaking point. Synthesis report 2021. Rome.
- Ferrer-Sánchez, Y., Barahona-Manzaba, D. G., Plasencia, V. A. H., & Abasolo-Pacheco, F. (2024). Riesgo de expansión de *Fusarium oxysporum* f. *cubense* (Nectriaceae) ante el cambio climático en Ecuador continental. *Acta Botanica Mexicana*, 131. <https://doi.org/10.21829/abm131.2024.2207>
- Galarza Vera, J. J. (2024). Evaluación de alternativas fitosanitarias para el manejo de Sigatoka negra (*Pseudocercospora fijiensis* Morelet) en el cultivo de banano (*Musa AAA*) [Tesis de grado, Universidad Agraria del Ecuador]. Repositorio Digital de la Universidad Agraria del Ecuador. <https://cia.uagraria.edu.ec/Archivos/GALARZA%20VERA%20JONATHAN%20JORDANO.pdf>
- Genet, Y. H., Sinshaw, N. T., Assefa, B. G., & Mohapatra, S. K. (2024). Sigatoka and *Xanthomonas* banana leaf disease detection via transfer learning. *Scientia Iranica*, 0(0), 0. <https://doi.org/10.24200/sci.2024.62306.7766>
- Hosna, A., Merry, E., Gyalmo, J., Alom, Z., Aung, Z., & Azim, M. A. (2022). Transfer learning: a friendly introduction. *Journal Of Big Data*, 9(1). <https://doi.org/10.1186/s40537-022-00652-w>
- Jiménez, N., Orellana, S., Mazon-Olivo, B., Rivas-Asanza, W., & Ramírez-Morales, I. (2025). Detection of leaf diseases in banana crops using deep learning techniques. *AI*, 6(3), 61. <https://doi.org/10.3390/ai6030061>
- Li, Y., Nie, J., & Chao, X. (2020). Do we really need deep CNN for plant diseases identification? *Computers And Electronics In Agriculture*, 178, 105803. <https://doi.org/10.1016/j.compelecres.2020.105803>

org/10.1016/j.compag.2020.105803

- Manggara, M. G., & Kalifia, A. D. (2025). Komparasi model ResNet50 dan EfficientNetV2M dengan penerapan transfer learning dan fine tuning pada klasifikasi penyakit bercak daun tanaman pisang. *www.hostjournals.com*. <https://doi.org/10.47065/bulletincsr.v6i1.855>
- Martínez-Solórzano, G. E., & Rey-Brina, J. C. (2021). Bananos (Musa AAA): Importancia, producción y comercio en tiempos de Covid-19. *Agron. Mesoam*, 1034-1046. <https://doi.org/10.15517/am.v32i3.43610>
- Montesdeoca Espinoza, L.J., Zambrano Rojas, S.J., Pinargote Bravo, V.J. & Cedeño Valarezo, L.C. (2025). Balanceo de conjuntos de datos basado en redes generativas aplicado a imágenes del sector agrícola. *Informática y Sistemas* 9(2), pp. 164-176 <https://doi.org/10.33936/isrtic.v9i2.7782>
- Olivares, B. O., Vega, A., Rueda Calderón, M. Á., Rey, J. C., & Lobo, D. (2021). Clasificación de zonas afectadas por la marchitez en banano: Una aplicación con algoritmos de machine learning en Venezuela. *Revista Especializada de Ingeniería y Ciencias de la Tierra*, 1(1), 1–15. <https://revistas.up.ac.pa/index.php/REICIT/article/view/2440/2230>
- Redacción Expreso. (2023, 26 de agosto). Se dispara la afectación de Sigatoka. *Diario Expreso*. <https://www.expreso.ec/actualidad/economia/dispara-afectacion-sigatoka-171205.html>
- Sanga, S., Mero, V., Machuve, D., & Mwanganda, D. (2020, 7 abril). Mobile-based deep learning models for banana diseases detection. *arXiv.org*. https://arxiv.org/abs/2004.03718?utm_source=202
- Torres Salamea, H. M., Amores Romero, K. S., & Trelles Muñoz, K. G. (2022). Implementación de un sistema para detectar la enfermedad de la Sigatoka Negra en una plantación de banano empleando técnicas de visión artificial. *Universidad del Azuay*. <https://dspace.uazuay.edu.ec/handle/datos/11682>
- Tripathi, J. N., Ntui, V. O., & Tripathi, L. (2023). Precision genetics tools for genetic improvement of banana. *The Plant Genome*, 17(2). <https://doi.org/10.1002/tpg2.20416>
- Ünal, C. (2025). Incorporating deep learning into the diagnosis of banana leaf spot diseases for the protection of banana crops. *Tarım Bilimleri Dergisi*, 31(3), 780-794. <https://doi.org/10.15832/ankutbd.1579442>
- Vera, Ch. J. D., & Carvajal, R. H. R. (2025). Avances en la inteligencia artificial para incrementar el rendimiento en los cultivos. *AgroEcosistemas Revista Para la Transformación Agraria Sostenible*, 13, e748. <https://aes.ucf.edu.cu/index.php/aes/article/view/748>
- Villanueva, J. R. E., Ustate, L. M. T., Estrada, M. A. G., & Martínez, M. L. C. (2025). Segmentación foliar en sistemas de cultivo de banano: aplicaciones del flujo de trabajo fotogramétrico y teledetección mediante imágenes multiespectrales de aeronaves no tripuladas (UAS). *Ciencia E Ingeniería Neogranadina*, 35(1), 25-45. <https://doi.org/10.18359/rcin.7365>
- Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems*, 27, 3320-3328. <https://proceedings.neurips.cc/paper/2014/hash/375c71349b295fbc2dcdca9206f20a06-Abstract.html>
- Zabala, C. J. J. (2024). Banana Scan: Aplicación Android para la detección de enfermedades en cultivos de banano mediante TinyML. <https://hdl.handle.net/20.500.12495/14049>

