

Desarrollo de un modelo de inteligencia artificial integrado en una aplicación móvil para detectar enfermedades en frutos de palta

Development of an artificial intelligence model integrated into a mobile application for detecting diseases in avocado fruits



Luis Fernando Retamozo-Saavedra

Universidad Nacional Micaela Bastidas de Apurímac

<https://orcid.org/0009-0000-8588-8667>

192224@unamba.edu.pe

Erech Ordoñez-Ramos

Universidad Nacional Micaela Bastidas de Apurímac

<https://orcid.org/0000-0002-3058-9683>

eordonhez@unamba.edu.pe

Alejandrina Huaylla-Quispe

Universidad Nacional Micaela Bastidas de Apurímac

<https://orcid.org/0000-0002-0258-8632>

ahuaylla@unamba.edu.pe

Universidad Nacional Micaela Bastidas de Apurímac – Perú

Riqchary, revista de investigación en ciencia y tecnología

[ISSN: 2810-8124 \(en línea\) / ISSN: 2706-543x](https://doi.org/10.57166/riqchary/v8.n1.2026)

Vol. – 8 Núm. 1 - (2026) - Publicado: 17/07/2026

<https://doi.org/10.57166/riqchary/v8.n1.2026>

Páginas: 36 - 44

Recibido 24/05/2026; Aceptado 13/07/2026

<https://doi.org/10.57166/riqchary/v8.n1.2026.5>

Resumen. — Este estudio tuvo como objetivo desarrollar un modelo de inteligencia artificial integrado en una aplicación móvil para la detección de enfermedades en frutos de palta fuerte (*Persea americana* Mill.). Para ello, se emplearon técnicas de visión por computadora utilizando la arquitectura YOLOv8l (large). Se utilizó un conjunto de datos obtenido de la plataforma Kaggle, conformado originalmente por 1 980 imágenes y ampliado mediante técnicas de aumento de datos hasta alcanzar 5 014 imágenes, con el propósito de mejorar la capacidad de generalización del modelo. El sistema logró identificar las clases antracnosis, sarna y frutos sanos, alcanzando un mAP50 de 0.956 y un F1-score de 0.929, lo que evidenció un desempeño satisfactorio durante la evaluación del modelo. Posteriormente, el modelo fue convertido al formato TensorFlow Lite (float32) e integrado en una aplicación móvil desarrollada en Android Studio (Java), permitiendo la inferencia local sin necesidad de conexión a internet. Las pruebas funcionales realizadas en un dispositivo móvil mostraron tiempos de procesamiento promedio de 3 a 4 segundos por imagen, con resultados estables en los modos de análisis por fotografía y galería. En conjunto, la propuesta constituye una herramienta de apoyo para el monitoreo fitosanitario, contribuyendo a la detección temprana de enfermedades en el cultivo de palta.

Palabras clave: Diagnóstico agrícola, inferencia móvil, monitoreo fitosanitario, redes neuronales convolucionales, visión por computadora.

Abstract. — This study aimed to develop an artificial intelligence model integrated into a mobile application for the detection of diseases in Fuerte avocado (*Persea americana* Mill.) fruits. Computer vision techniques based on the YOLOv8l (Large) architecture were employed. The dataset, obtained from the Kaggle platform, originally consisted of 1,980 images and was expanded through data augmentation techniques to 5,014 images in order to improve the model's generalization capability. The system successfully identified the classes anthracnose, scab, and healthy fruits, achieving a mAP50 of 0.956 and an F1-score of 0.929, demonstrating satisfactory performance during model evaluation. Subsequently, the model was converted to the TensorFlow Lite (float32) format and integrated into a mobile application developed in Android Studio (Java), enabling on-device inference without an internet connection. Functional tests performed on an Android mobile device showed average processing times of 3 to 4 seconds per image, with stable performance in both camera and gallery analysis modes. Overall, the proposed system constitutes a supporting tool for phytosanitary monitoring, contributing to the early detection of diseases in avocado cultivation.

Keywords: Agricultural diagnosis, computer vision, convolutional neural networks, mobile inference, phytosanitary monitoring.

1 INTRODUCCIÓN

La variedad palta fuerte (*Persea americana* var. *fuerte*) constituye uno de los cultivos más importantes dentro de la producción agrícola peruana, debido a su alta calidad organoléptica, aceptación comercial y valor nutricional. Sin embargo, su rendimiento se ve afectado por diversas enfermedades fúngicas que reducen significativamente la calidad del fruto, siendo la antracnosis y la sarna las más frecuentes y perjudiciales [1].

La antracnosis, causada por hongos del género *Colletotrichum* (principalmente *C. gloeosporioides* y *C. acutatum*), genera manchas oscuras y hundidas que se expanden con rapidez en ambientes húmedos, ocasionando necrosis y deterioro de la pulpa [2]. Estudios recientes han identificado también a *Colletotrichum siamense* como agente causal en la etapa de postcosecha, lo que evidencia la diversidad de especies involucradas y la complejidad de su control [3]. En condiciones severas, las pérdidas pueden alcanzar entre el 20 % y el 30 % de la producción total [4]. Por su parte, la sarna produce lesiones superficiales y deformaciones en la epidermis del fruto, afectando su apariencia comercial y reduciendo su valor de exportación [5].

La identificación de estas enfermedades suele realizarse mediante inspección visual directa, un procedimiento que requiere experiencia técnica y tiempo considerable. Este método presenta limitaciones notorias: la subjetividad del observador, la ausencia de criterios estandarizados y la dificultad para procesar grandes volúmenes de producto [6]. A ello se suma la escasez de personal especializado en zonas rurales, lo cual limita el control fitosanitario oportuno y genera pérdidas económicas significativas, afectando la competitividad de los productores frente a los mercados nacionales e internacionales.

En respuesta a estas limitaciones, las técnicas de visión por computadora y aprendizaje profundo (deep learning) han demostrado alta precisión en la detección de enfermedades vegetales [7], [8]. En particular, los modelos de la familia YOLO (You Only Look Once) han demostrado ser eficientes en tareas de visión por computadora en tiempo real [9], [10]. Han evolucionado progresivamente en precisión y eficiencia computacional desde versiones como YOLOv4 y YOLOv7 [9], [11], lo que permite su aplicación en sistemas en tiempo real. En este contexto, versiones recientes como YOLOv8l incorporan mejoras en el procesamiento multiescala y una mayor robustez frente a variaciones de iluminación, textura y tamaño de los objetos [12].

Diversos estudios reportan precisiones superiores al 90 % utilizando redes neuronales convolucionales (CNN), YOLO y otros modelos de aprendizaje automático en diferentes cultivos [13], [14], [15], [16], [17], [18], [19], [20], [21]. En el contexto peruano, investigaciones previas emplearon modelos basados en SVM, CNN y YOLOv4 para la detección de enfermedades en palta, papa, tomate y aguaymanto, obteniendo resultados satisfactorios y evidenciando el potencial de estas tecnologías para el monitoreo fitosanitario [16], [18], [19], [20], [22]. Sin embargo, la mayoría de estudios se desarrollaron en entornos controlados o con conjuntos de datos limitados, lo que evidencia la necesidad de soluciones móviles aplicables en condiciones reales de campo [23].

En este sentido, a pesar de los avances en la aplicación

de inteligencia artificial en el diagnóstico agrícola, aún persisten limitaciones en el desarrollo de soluciones portátiles que operen eficientemente en condiciones reales de campo, especialmente en entornos con acceso restringido a conectividad y recursos computacionales. Asimismo, la mayoría de investigaciones se ha enfocado en otras variedades o en condiciones controladas, lo que evidencia un vacío en la detección de enfermedades en frutos de palta fuerte mediante dispositivos móviles.

En este contexto, surge la siguiente pregunta de investigación: ¿Es posible desarrollar un modelo de inteligencia artificial basado en la arquitectura YOLOv8l, integrado en una aplicación móvil, que permita detectar automáticamente enfermedades en frutos de palta con alta precisión en condiciones reales de uso?

Por ello, el presente estudio tuvo como objetivo desarrollar un modelo de inteligencia artificial basado en YOLOv8l, integrado en una aplicación móvil, para la detección automática de enfermedades en frutos de palta en condiciones reales de uso.

Para alcanzar el objetivo general, se plantearon los siguientes objetivos específicos:

1. Seleccionar, organizar y preparar un conjunto de imágenes de frutos de palta para el entrenamiento del modelo de inteligencia artificial.
2. Entrenar y evaluar un modelo basado en la arquitectura YOLOv8l utilizando métricas de desempeño como Precision, Recall, F1-score y mAP.
3. Integrar el modelo entrenado en una aplicación móvil desarrollada en Android Studio mediante TensorFlow Lite para realizar inferencias locales.
4. Evaluar el funcionamiento del sistema en condiciones reales de uso mediante pruebas funcionales realizadas en un dispositivo móvil Android.

2 METODOLOGÍA

La investigación se desarrolló en la ciudad de Abancay, región Apurímac, Perú, donde se llevó a cabo la implementación y las pruebas funcionales en dispositivos móviles. El estudio correspondió a una investigación aplicada, con enfoque cuantitativo y metodología de desarrollo tecnológico, orientada al diseño, entrenamiento, implementación y validación experimental de un modelo de inteligencia artificial para la detección de enfermedades en frutos de palta.

La fuente de datos estuvo conformada por imágenes digitales de frutos de palta en diferentes estados fitosanitarios (sanos y con presencia de enfermedades). El entrenamiento y la evaluación del modelo se realizaron utilizando un conjunto de datos público denominado “*Clasificación de enfermedades del aguacatero*”, disponible en la plataforma Kaggle. La muestra estuvo constituida por 1 980 imágenes originales, las cuales fueron seleccionadas y procesadas para el entrenamiento, validación y evaluación del modelo. Posteriormente, la implementación y la prueba funcional de la aplicación móvil se desarrollaron en la ciudad de Abancay con el propósito de verificar la correcta ejecución local del modelo en un dispositivo Android.

2.1 Variables de estudio

El estudio se centró en la evaluación del desempeño

de un modelo de inteligencia artificial basado en la arquitectura YOLOv8l, ampliamente utilizada para la detección de objetos en tiempo real mediante técnicas de visión por computadora [9].

El desempeño del modelo fue evaluado mediante las métricas de **Precision, Recall, F1-score, mAP50 y mAP50-95**, las cuales permitieron cuantificar su capacidad para identificar correctamente las clases antracnosis, sarna y frutos sanos en las imágenes analizadas.

2.2 Conjunto de datos

El conjunto de datos utilizado en esta investigación se obtuvo del repositorio "Clasificación de enfermedades del aguacatero" disponible en la plataforma Kaggle, publicado por Campos Ferreira y Gonzales Camacho [1]. A partir de este repositorio, se seleccionaron imágenes correspondientes a frutos de palta fuerte (*Persea americana* Mill.), clasificadas en tres categorías: antracnosis, sarna y sano. Las imágenes originales fueron importadas a la plataforma Roboflow, donde se realizó la anotación manual de las regiones de interés mediante etiquetas en formato YOLO, procedimiento comúnmente empleado en tareas de detección de objetos en agricultura mediante técnicas de aprendizaje profundo [24]. El conjunto inicial estuvo conformado por 1 980 imágenes originales balanceadas, distribuidas equitativamente en 660 imágenes por cada clase.

Posteriormente, el conjunto de datos fue dividido en tres subconjuntos: **entrenamiento, validación y prueba**, manteniendo la proporción por clase. La distribución de estas imágenes originales se presenta en la tabla 1, donde se observa que el subconjunto de entrenamiento concentra la mayor cantidad de datos, seguido de validación y prueba, garantizando un adecuado equilibrio entre aprendizaje y evaluación del modelo.

TABLA 1
Distribución del conjunto de datos original (sin aumento de datos)

Clase	Total por clase	Entrenamiento	Validación	Prueba
Antracnosis	660	528	100	32
Sano	660	528	100	32
Sarna	660	528	100	32
Total	1 980	1 584	300	96

Con el objetivo de mejorar la capacidad de generalización del modelo y reducir el riesgo de sobreajuste, se aplicó un proceso de aumento de datos (data augmentation) sobre el subconjunto de entrenamiento, técnica ampliamente utilizada en modelos de aprendizaje profundo para incrementar la diversidad del conjunto de datos y mejorar su desempeño [25]. Las transformaciones incluyeron rotaciones, volteos horizontales y verticales, así como variaciones geométricas, generando nuevas muestras a partir de las imágenes originales.

Como resultado de este proceso, el número de imágenes de entrenamiento se incrementó significativamente, ampliando el conjunto total a 5 014 imágenes. Es importante destacar que los subconjuntos de validación y prueba

no fueron modificados, con el fin de garantizar una evaluación objetiva del modelo y evitar sesgos derivados del uso de datos artificialmente generados. La distribución final del conjunto de datos se resume en la tabla 2.

TABLA 2

Distribución final del conjunto de datos (con aumento de datos)

Finalmente, para evitar posibles sesgos en la evaluación, las imágenes generadas mediante aumento de datos no fueron incluidas en los subconjuntos de validación ni prueba, asegurando así que las métricas obtenidas reflejen

Conjunto	Imágenes originales	Imágenes aumentadas	Total	Porcentaje
Entrenamiento	1584	3034	4618	92%
Validación	300	0	300	6%
Prueba	96	0	96	2%
Total	1980	3034	5014	100%

el desempeño real del modelo frente a datos no vistos.

2.3 Modelo de inteligencia artificial basado en YOLOv8l

En este estudio se empleó un modelo de detección de objetos basado en la arquitectura YOLOv8l (Large), perteneciente a la familia de detectores de una sola etapa (one-stage detectors), capaz de realizar simultáneamente la localización y clasificación de objetos dentro de una imagen, lo que facilita su aplicación en tiempo real [10]. La elección de esta arquitectura se fundamentó en su capacidad para lograr un equilibrio entre precisión y velocidad de inferencia, características ampliamente reportadas en la literatura sobre detección de objetos mediante aprendizaje profundo [26].

La variante YOLOv8l fue seleccionada por su mayor capacidad de representación y extracción de características en comparación con versiones más ligeras, favoreciendo la detección de patrones visuales complejos asociados a enfermedades en frutos agrícolas [23]. No obstante, esta elección implica un mayor costo computacional, aspecto que influye en los tiempos de inferencia en dispositivos móviles y que ha sido reportado en estudios relacionados con implementación de modelos de aprendizaje profundo en hardware con recursos limitados [27].

La configuración de hiperparámetros se realizó considerando las capacidades del hardware disponible y la necesidad de lograr un entrenamiento estable, cuyos valores se presentan en la Tabla 3.

TABLA 3
Hiperparámetros definidos

Parámetro	Valor
Modelo base	YOLOv8l
Epochs	200
Image Size	640
Batch Size	64
Learning Rate inicial (lr0)	0.01
Early Stopping (patience)	20
Tipo de entrenamiento	Fine-tuning

Una vez finalizado el entrenamiento, el modelo con mejor desempeño fue exportado al formato TensorFlow Lite (TFLite) utilizando representación en precisión float32, con el propósito de facilitar su integración en la aplicación móvil desarrollada para Android. Debido a que no se aplicó cuantización a formatos de menor precisión, como *float16* o *int8*, la conversión permitió conservar prácticamente el mismo desempeño obtenido durante el entrenamiento. No obstante, esta decisión incrementó el tamaño del modelo y el tiempo de inferencia, priorizando la exactitud de las predicciones sobre la ejecución en tiempo real en dispositivos móviles.

2.4 Configuración de hardware y software

La fase de entrenamiento del modelo se ejecutó en Google Colaboratory, una plataforma de computación en la nube que permite el desarrollo de experimentos de aprendizaje automático sin necesidad de instalar software localmente y que ofrece acceso a recursos de hardware especializados como GPU y TPU. Este tipo de entornos facilita el desarrollo y la reproducibilidad experimental, además de simplificar la integración de bibliotecas y el almacenamiento en la nube, lo que resulta especialmente útil en proyectos de aprendizaje profundo que requieren alto poder de cómputo [28]. Para el entrenamiento se utilizó una GPU NVIDIA A100, basada en arquitectura Ampere, diseñada para cargas de trabajo intensivas de inteligencia artificial, proporcionando alto rendimiento en entrenamiento e inferencia mediante Tensor Cores y memoria HBM de alto ancho de banda, lo cual permite acelerar significativamente el procesamiento de modelos de visión computacional [29].

Para el desarrollo de la aplicación móvil se utilizó Android Studio ejecutándose en una computadora portátil Acer Nitro 5 con GPU NVIDIA GTX 1650, 24 GB de memoria RAM y procesador Intel de 11ª generación, mientras que las pruebas funcionales se realizaron en un dispositivo móvil Poco X7 Pro con sistema operativo Android 16. La inferencia en dispositivos móviles constituye un enfoque relevante en sistemas de inteligencia artificial modernos debido a que reduce la latencia, mejora la privacidad de los datos y permite ejecutar modelos en tiempo real utilizando aceleradores gráficos móviles o NPUs, lo cual es ampliamente soportado por frameworks como TensorFlow Lite [30]. Este enfoque permite trasladar modelos entrenados en infraestructuras de alto rendimiento hacia entornos de ejecución en el borde (edge computing).

2.5 Aplicaciones móviles basadas en IA

La integración de modelos de inteligencia artificial en dispositivos móviles permite aplicaciones prácticas en entornos agrícolas [27]. El desarrollo en Android Studio con Java permitió integrar el modelo entrenado para ejecutar inferencias locales a partir de imágenes capturadas por fotografía o seleccionadas desde la galería del dispositivo [31].

Para garantizar eficiencia operativa, se aplicaron técnicas de reducción de resolución, ejecución en segundo

plano y uso de modelos compactos, lo que posibilitó la inferencia sin conexión a internet ni hardware adicional. Estas características convierten al teléfono móvil en una herramienta de diagnóstico portátil, accesible y de bajo costo, especialmente útil en entornos rurales [32].

2.6 Métricas de evaluación

El rendimiento del modelo se evaluó mediante las métricas Precisión (P), Exhaustividad o Recall (R), F1-Score, y Precisión media promedio (mAP), definidas en las ecuaciones (1)–(4).

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

$$F1 = 2 \times \frac{P \times R}{P+R} \quad (3)$$

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (4)$$

Donde **TP** (True Positives) representa los casos detectados correctamente como enfermos, **FP** los falsos positivos y **FN** los falsos negativos. En el contexto de detección de objetos, no se considera la métrica de verdaderos negativos (**TN**), debido a la naturaleza del problema, donde el espacio de no objetos es indefinido.

Las métricas empleadas permitieron evaluar cuantitativamente el desempeño del modelo en tareas de detección de objetos. Se utilizaron Precision, Recall, F1-Score y mAP (mean Average Precision), métricas ampliamente empleadas en modelos de visión por computadora. En particular, se consideraron mAP50 y mAP50-95 como se observa en la Figura 1, donde la primera utiliza un umbral IoU de 0.50 y la segunda evalúa un rango entre 0.50 y 0.95 con incrementos de 0.05, proporcionando una evaluación más rigurosa de la calidad de las predicciones.

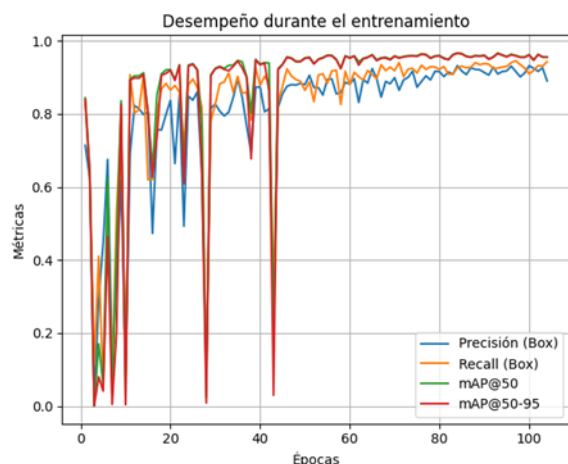


Fig. 1. Desempeño del modelo durante el entrenamiento

3 RESULTADOS

El modelo propuesto mostró un desempeño equilibrado en la detección de las tres clases evaluadas: antracnosis, sano y sarna. El F1-score promedio alcanzado fue de 0.929, lo que evidenció una adecuada relación entre precisión y exhaustividad. La clase sarna presentó el mejor desempeño, seguida de sano y antracnosis.

Los valores de mAP50 (0.956) y mAP50-95 (0.955) confirmaron la capacidad del modelo para localizar correctamente los objetos en las imágenes, validando su idoneidad para aplicaciones en entornos reales. Las métricas detalladas por clase se presentan en la tabla 4.

TABLA 4
Métricas de evaluación por clase

Clase	Precisión (P)	Recall (R)	F1-Score	mAP50	mAP50-95
Antracnosis	0.944	0.892	0.917	0.939	0.937
Sano	0.921	0.938	0.929	0.951	0.951
Sarna	0.932	0.950	0.941	0.978	0.977
Promedio general	0.932	0.927	0.929	0.956	0.955

3.1 Matriz de confusión

La Figura 2 presenta el mapa de calor correspondiente a la matriz de confusión obtenida durante la fase de validación, mientras que la Figura 3 muestra un gráfico de barras comparativo de los valores de *Precision*, *Recall* y *F1-score* para cada una de las clases evaluadas. En conjunto, estos resultados evidencian un desempeño consistente del modelo en la clasificación de las enfermedades de la palta. La clase *sano* presentó el mejor rendimiento, con 103 imágenes correctamente clasificadas, seguida por *antracnosis* (91) y *sarna* (80).

Las principales confusiones se observaron entre las clases antracnosis y sano, registrándose siete clasificaciones cruzadas en cada sentido. Este comportamiento

puede explicarse porque, en las etapas iniciales de la antracnosis, las lesiones presentan cambios cromáticos leves y alteraciones poco perceptibles en la textura superficial del fruto, características que pueden asemejarse a variaciones naturales asociadas al proceso de maduración o a reflejos producidos por las condiciones de iluminación. Asimismo, la diversidad de fondos presentes en el conjunto de datos probablemente incrementó la similitud visual entre ambas clases, favoreciendo algunas clasificaciones erróneas. No obstante, el valor promedio de F1-score (0.929), resultado comparable con estudios previos que emplean redes neuronales profundas para la detección de enfermedades [7], [24].

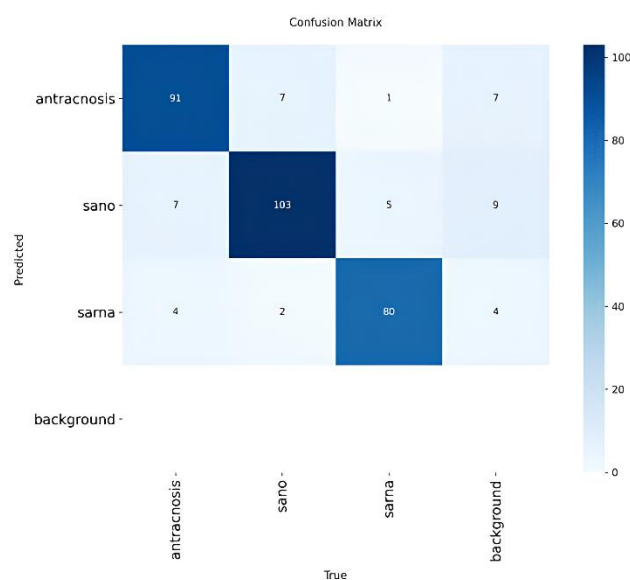


Fig. 2. Mapa de calor de la matriz de confusión del modelo resultante

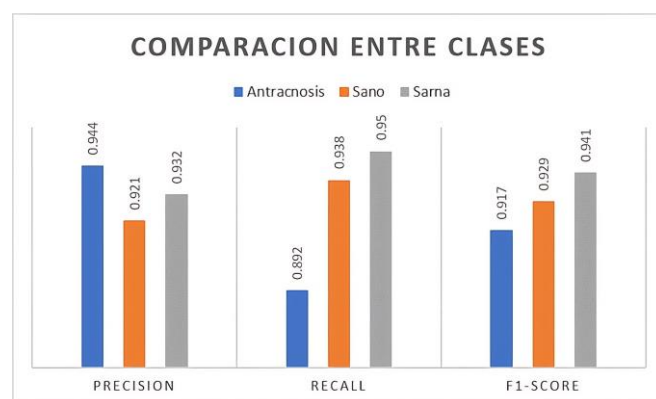


Fig. 3. Comparación de las métricas Precision, Recall y F1-score obtenidas por el modelo para cada clase evaluada.

3.2 Desempeño en la aplicación móvil

La aplicación móvil desarrollada en Android Studio (Java) integró satisfactoriamente el modelo YOLOv8l convertido al formato TensorFlow Lite (float32), optimizado

para dispositivos de gama media. Esta conversión permitió conservar los pesos del modelo con una mínima pérdida de precisión, garantizando la ejecución de inferencias locales sin conexión a internet.

Durante las pruebas funcionales, el tiempo promedio de procesamiento fue de 3 a 4 segundos por imagen, considerando todo el flujo de ejecución: carga de la imagen, redimensionamiento (640×640 píxeles), inferencia y visualización del resultado tal como muestra la tabla 5. Dicho rendimiento se consideró adecuado para diagnósticos puntuales en entornos agrícolas con recursos limitados.

Sin embargo, el procesamiento en tiempo real mediante la cámara no resultó viable en la mayoría de dispositivos debido a la alta demanda computacional. Por ello, se implementaron únicamente los modos de captura por fotografía y selección desde galería, priorizando estabilidad y eficiencia energética.

La aplicación cuyo diseño se muestra en la Figura 4, incorporó además un módulo de historial de diagnósticos, que almacena imágenes analizadas junto con sus resultados, facilitando el seguimiento de evaluaciones previas y mejorando la trazabilidad del proceso.

TABLA 5
Resumen del rendimiento

Parámetro	Valor	Descripción
Formato del modelo	.tflite (float32)	Modelo exportado en formato TensorFlow Lite (float32), sin aplicar cuantización
Tamaño del modelo	~160 MB	Conversión desde YOLOv8l original
Tiempo promedio de inferencia	3–4 s por imagen	Modo fotografía y galería
FPS en tiempo real	<1 FPS (cuadros por segundo)	No implementado por limitaciones de hardware
Precisión promedio	93.2 %	Consistente con el modelo original
Conectividad	No requerida	Inferencia completamente local
Funcionalidades adicionales	Historial de análisis	Registro visual de resultados previos

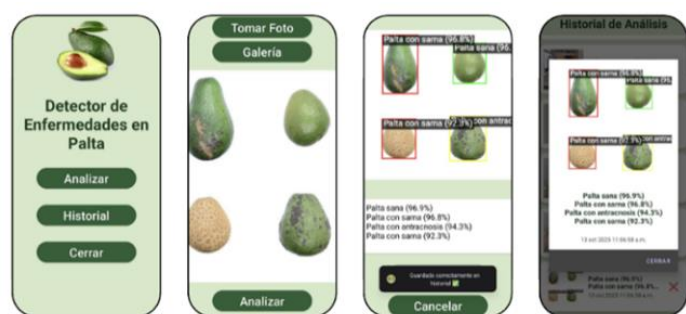


Fig. 4. Diseño final del aplicativo móvil Android para detección de enfermedades en palta

3.3 Discusión de resultados

Los resultados obtenidos muestran que el modelo YOLOv8l alcanzó un desempeño competitivo en la detección de enfermedades en frutos de palta, obteniendo un F1-score promedio de 0.929 y un mAP50 de 0.956. Estos resultados son consistentes con investigaciones previas que reportan precisiones superiores al 90 % en tareas de detección agrícola mediante modelos basados en deep learning [17], [18], [19].

En comparación con variantes anteriores como YOLOv4 y enfoques basados en Faster R-CNN, el modelo propuesto presentó un desempeño superior en términos de precisión y estabilidad, lo que puede atribuirse a las mejoras introducidas en YOLOv8l, particularmente en la extracción multiescala de características y la optimización del backbone [9]. Asimismo, a diferencia de métodos tradicionales basados en SVM o CNN orientados únicamente a clasificación [16], [17], el modelo implementado permitió realizar simultáneamente detección y localización de enfermedades en el fruto.

El uso de técnicas de aumento de datos contribuyó a mejorar la capacidad de generalización del modelo frente a variaciones de iluminación, textura y posición de los objetos. Del mismo modo, la integración mediante TensorFlow Lite permitió conservar un nivel de precisión cercano al obtenido en entorno de escritorio, aunque se evidenciaron limitaciones relacionadas con el costo computacional del modelo durante la inferencia en dispositivos móviles. Estas restricciones impidieron una ejecución fluida en tiempo real, siendo necesario utilizar modos de análisis por fotografía y galería.

El principal aporte científico de esta investigación radica en la adaptación, entrenamiento, validación e implementación de un modelo basado en YOLOv8l para la detección automática de antracnosis y sarna en frutos de palta, integrado en una aplicación móvil Android mediante TensorFlow Lite con ejecución local. A diferencia de estudios centrados únicamente en el entrenamiento de modelos de inteligencia artificial, esta propuesta demuestra la viabilidad de trasladar el modelo a un entorno móvil funcional, permitiendo realizar inferencias sin conexión a internet. Asimismo, el trabajo evidencia el potencial de esta tecnología como herramienta de apoyo para el monitoreo fitosanitario en contextos agrícolas, especialmente en zonas donde el acceso a especialistas o a infraestructura tecnológica es limitado, contribuyendo así a la aplicación práctica de la inteligencia artificial en el sector agrícola.

3.4 Limitaciones del estudio

El presente estudio presenta algunas limitaciones que deben considerarse al interpretar los resultados. En primer lugar, el modelo fue entrenado utilizando imágenes correspondientes únicamente a frutos de palta de la variedad Fuerte, por lo que su desempeño en otras variedades requiere ser evaluado antes de su aplicación en diferentes contextos agrícolas. Asimismo, aunque el conjunto de datos fue ampliado mediante técnicas de aumento de datos para mejorar la capacidad de generalización, aún es con-

veniente incorporar imágenes obtenidas en una mayor diversidad de condiciones ambientales, niveles de iluminación y escenarios de campo.

Por otra parte, la implementación del modelo en formato TensorFlow Lite (float32) priorizó la precisión sobre la velocidad de procesamiento, lo que impidió la ejecución de inferencias en tiempo real mediante video continuo en dispositivos móviles de gama media. En consecuencia, la aplicación fue diseñada para operar mediante captura de fotografías o selección de imágenes desde la galería. Asimismo, las métricas de desempeño fueron obtenidas a partir de una única evaluación sobre el conjunto de prueba, por lo que no se calcularon intervalos de confianza ni se aplicaron procedimientos de validación cruzada. En futuras investigaciones se recomienda incorporar técnicas como la validación cruzada o métodos de remuestreo (bootstrap) para estimar la variabilidad estadística de las métricas obtenidas y fortalecer la evaluación del modelo. Del mismo modo, sería conveniente evaluar estrategias de optimización, como la cuantización del modelo o el empleo de arquitecturas más ligeras, así como validar el sistema en otras variedades de palta y en diferentes regiones productoras para fortalecer su capacidad de generalización.

4 CONCLUSIONES

- I. Se desarrolló e implementó satisfactoriamente un modelo de inteligencia artificial basado en la arquitectura YOLOv8l, integrado en una aplicación móvil Android, capaz de detectar automáticamente enfermedades en frutos de palta mediante inferencia local.
- II. El modelo obtuvo un desempeño satisfactorio durante la evaluación, alcanzando un F1-score promedio de 0.929 y un mAP50 de 0.956, lo que evidencia su capacidad para identificar las clases antracnosis, sarna y frutos sanos con un elevado desempeño de clasificación.
- III. La integración mediante TensorFlow Lite permitió ejecutar el modelo sin conexión a internet, facilitando su utilización en entornos agrícolas con acceso limitado a infraestructura tecnológica y constituyendo una alternativa práctica para apoyar el monitoreo fitosanitario.
- IV. Aunque el modelo presentó limitaciones para la inferencia en tiempo real debido a su demanda computacional, demostró un funcionamiento estable mediante los modos de captura fotográfica y selección de imágenes desde la galería, ofreciendo una solución viable para el diagnóstico asistido en campo.
- V. Como línea futura de investigación, se recomienda ampliar el conjunto de datos con imágenes de otras variedades de palta y diferentes condiciones ambientales, evaluar técnicas de optimización como la cuantización del modelo o arquitecturas más ligeras, e integrar la aplicación con plataformas de gestión agrícola que permitan centralizar el seguimiento fitosanitario y facilitar su escalabilidad en

entornos productivos.

5 REFERENCIAS

- [1] U. E. Campos-Ferreira, J. M. González-Camacho, and A. Carrillo-Salazar, "Automatic identification of avocado fruit diseases based on machine learning and chromatic descriptors," *Rev. Chapingo Ser. Hortic.*, vol. 29, no. 3, pp. 115-130, Sep. 2023. <https://doi.org/10.5154/r.rchsh.2023.04.002>
- [2] A. Tapia Rodríguez, J. F. Ramírez Dávila, M. L. Salgado Siclán, Á. Castañeda Vildózola, F. I. Maldonado Zamora, and A. V. Lara Díaz, "Spatial distribution of anthracnose (*Colletotrichum gloeosporioides* Penz) in avocado in the State of Mexico, Mexico," *Rev. Argent. Microbiol.*, vol. 52, no. 1, pp. 72-81, Jan. 2020. <https://doi.org/10.1016/j.ram.2019.07.004>
- [3] J. A. Herrera-González, S. Bautista-Baños, M. Serrano, S. Ramos-Bell, and P. Gutiérrez-Martínez, "Colletotrichum siamense causante de antracnosis en poscosecha de aguacate 'Hass'," *Rev. Mex. Cienc. Agric.*, vol. 15, no. 5, Jul. 2024. <https://doi.org/10.29312/remexca.v15i5.3434>
- [4] INTAGRI, "Antracnosis en el cultivo de aguacate," *Artículos Técnicos de Intagri*, vol. 4, no. 81, 2017. [Online]. Available: <https://www.intagri.com/articulos/fitosanidad/antracnosis-en-el-cultivo-de-aguacate>
- [5] E. Trinidad Angel, F. de J. Ascencio Valle, J. Armando Ulloa, J. C. Ramirez Ramirez, and J. A. Ragazzo Sanchez, "Identificación y caracterización de Colletotrichum spp. causante de antracnosis en aguacate" Nayarit, México," vol. 1, no. 19, pp. 3953-3964, Nov. 2017. <https://doi.org/10.29312/remexca.v0i19.664>
- [6] S. Mishra, T. H. Ayane, V. Ellappan, D. S. Rathee, and H. Kalla, "Avocado fruit disease detection and classification using modified SCA-PSO algorithm-based MobileNetV2 convolutional neural network," *Iran J. Comput. Sci.*, vol. 5, no. 4, pp. 345-358, Dec. 2022. <https://doi.org/10.1007/s42044-022-00116-7>
- [7] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Comput. Electron. Agric.*, vol. 145, pp. 311-318, Feb. 2018. <https://doi.org/10.1016/j.compag.2018.01.009>
- [8] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," *Comput. Electron. Agric.*, vol. 161, pp. 272-279, Jun. 2019. <https://doi.org/10.1016/j.compag.2018.03.032>
- [9] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, Apr. 2020. [Online]. Available: <http://arxiv.org/abs/2004.10934>
- [10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, real-time object detection," *arXiv preprint arXiv:1506.02640*, May

2016. [Online]. Available: <http://arxiv.org/abs/1506.02640>
<https://doi.org/10.1109/CVPR.2016.91>
- [11] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," arXiv preprint arXiv:2207.02696, Jul. 2022. [Online]. Available: <http://arxiv.org/abs/2207.02696>
<https://doi.org/10.1109/CVPR52729.2023.00721>
- [12] Y. Wang et al., "Enhanced multiscale plant disease detection with the PYOLO model innovations," *Sci. Rep.*, vol. 15, 2025, art. 5179. <https://doi.org/10.1038/s41598-025-89034-9>
- [13] L. Guadalima-Inga, R. Lojano-Chavez, and M. Cabrera-Sarango, "Inteligencia artificial en sanidad vegetal," 2020. Accessed: May 20, 2026. [Online]. Available: <https://virtual.cuautilan.unam.mx/intar/ceiaait/wp-content/uploads/sites/14/2021/03/7-Art%C3%ADculos-editados-55-61.pdf>
- [14] D. Pineda Medina et al., "Sistema para el diagnóstico de plagas de *Solanum tuberosum* L. mediante técnicas de inteligencia artificial," *Rev. Prot. Veg.*, vol. 38, no. 3, pp. 1–6, 2023. [Online]. Available: <https://cu-id.com/2247/v38e16>
- [15] B. Roldan Ortega, R. Roshan Biswal, and E. Sanchez de la Cruz, "Detección de enfermedades en el sector agrícola utilizando inteligencia artificial," *Res. Comput. Sci.*, vol. 148, no. 7, pp. 419–427, 2019. [Online]. Available: https://rcs.cic.ipn.mx/2019_148_7/Deteccion%20de%20enfermedades%20en%20el%20sector%20agr%C3%ADcola%20utilizando%20Inteligencia%20Artificial.pdf
<https://doi.org/10.13053/rcs-148-7-31>
- [16] G. T. Castro Alvarez, "Aplicación de algoritmos inteligentes para reconocimiento automático de enfermedades foliares de cultivo de palta," tesis de grado, Ilo, Perú, 2019. [Online]. Available: <https://repositorio.unam.edu.pe/items/e3e92514-188c-4972-9921-afabe6236a03>
- [17] E. D. Cabrera Herrera, "Aplicación web usando deep learning aplicado al reconocimiento de plagas y enfermedades para mejorar el tratamiento de los cultivos de papa Yungay en Cutervo-Angurra," tesis de grado, Chiclayo, Perú, 2025. [Online]. Available: <https://orcid.org/0000-0003-1178-0519>
- [18] B. R. J. Huaman Caceres, "Machine learning para la detección de plagas en las hojas del tomate Abancay 2022," tesis de grado, Univ. Nac. Micaela Bastidas de Apurímac, Abancay, Perú, 2025. [Online]. Available: <https://repositorio.unamba.edu.pe/items/379b774c-4385-4756-a6e1-2ab7fd74adcf>
- [19] J. M. Medina Cercado and J. A. Urteaga Montoya, "Impacto de la aplicación móvil 'Healthy Plant' para detectar enfermedades foliares en cultivos de aguaymanto haciendo uso de inteligencia artificial con Custom Vision en la ciudad de Cajamarca 2021," tesis de grado, Univ. Privada del Norte, Cajamarca, Perú, 2021. [Online]. Available: <https://repositorio.upn.edu.pe/item/d8ea424d-3c96-4d84-acc8-475c1d12ddb3>
- [20] N. A. Alvarez Vargas, "Detección de enfermedades en el cultivo de papa mediante el uso de machine learning en Abancay, 2022," tesis de grado, Univ. Nac. Micaela Bastidas de Apurímac, Abancay, Perú, 2025. [Online]. Available: <https://repositorio.unamba.edu.pe/items/3b6581d1-8b23-427a-a25d-316aaf09554>
- [21] P. Jeet et al., "Disease detection in fruits and vegetables using machine learning with OpenVINO technology," vol. 1, no. 1, 2025. <https://doi.org/10.62762/DA.2025.743124>
- [22] J. Navarro Cruz, "Diseño de un sistema automático de reconocimiento de patrones para la clasificación de palta en la comunidad de Mollebamba-Chincheros," tesis de grado, Univ. Nac. José María Arguedas, Andahuaylas, Perú, 2023. [Online]. Available: <https://repositorio.unajma.edu.pe/item/b7426bc2-48a5-4559-bf50-5c580dc9f4c7>
- [23] A. Koirala, K. B. Walsh, Z. Wang, and C. McCarthy, "Deep learning - Method overview and review of use for fruit detection and yield estimation," *Comput. Electron. Agric.*, vol. 162, pp. 219–234, Jul. 2019. <https://doi.org/10.1016/j.compag.2019.04.017>
- [24] A. Fuentes, S. Yoon, S. C. Kim, and D. S. Park, "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition," *Sensors*, vol. 17, no. 9, Sep. 2017. <https://doi.org/10.3390/s17092022>
- [25] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, Dec. 2019. <https://doi.org/10.1186/s40537-019-0197-0>
- [26] L. Liu et al., "Deep learning for generic object detection: A survey," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 261–318, Feb. 2020. <https://doi.org/10.1007/s11263-019-01247-4>
- [27] A. Picon, A. Alvarez-Gila, M. Seitz, A. Ortiz-Barredo, J. Echazarra, and A. Johannes, "Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild," *Comput. Electron. Agric.*, vol. 161, pp. 280–290, Jun. 2019. <https://doi.org/10.1016/j.compag.2018.04.002>
- [28] F. P. Lara Galicia, "Guía sobre Google Colab: Aprendizaje y soluciones en programación Python." 2024. [Online]. Available: <https://www.godaddy.com/resources/latam/desarrollo/google-colab-que-es-como-utilizarlo>
- [29] V. Nguyen, A. Tao, O. Gallo, J. Lisiecki, and A. Badki, "Improving computer vision with NVIDIA A100 GPUs." 2020. [Online]. Available: <https://developer.nvidia.com/blog/improving-computer-vision-with-nvidia-a100-gpus/>
- [30] J. Lee et al., "On-device neural net inference with mobile GPUs," arXiv preprint arXiv:1907.01989,

- Jul. 2019. [Online]. Available:
<http://arxiv.org/abs/1907.01989>
- [31] R. Lopez Lucia, "Aplicación Android y servicio web Spring para la detección y registro de señales de tráfico de velocidad usando deep learning con tiny-YOLOv3 y OpenCV," tesis de grado, 2020. [Online]. Available:
<https://idus.us.es/items/dcbf6484-cef0-4b70-a438-a365c74b0c4a>
- [32] R. A. Solis Villanueva, "Modelo inteligente basado en técnicas de inteligencia artificial y su impacto en el desarrollo de aplicaciones en los dispositivos móviles," Lima, Perú, 2019. [Online]. Available:
<https://repositorio.unfv.edu.pe/items/929c2d8c-a7ca-4e25-8414-d05afbcf45fe>