Detección y Clasificación de Patógenos Foliares mediante Visión por Computador

Gabriel Sánchez Muñoz
gabrielfsm@correo.ugr.es

Germán Rodríguez Vidal
germanrv@correo.ugr.es

Pablo García Bas
pablogarciabas@correo.ugr.es

Miguel Ángel Moreno Castro
miguelangelmc@correo.ugr.es

Abstract

La detección temprana y precisa de enfermedades vegetales es crítica para la agricultura de precisión y la seguridad alimentaria. Los enfoques actuales basados en Deep Learning, a pesar de mostrar resultados prometedores, suelen carecer de información contextual sobre el huésped, lo que limita su precisión en escenarios de múltiples cultivos. Este trabajo propone un enfoque jerárquico de dos etapas para abordar esta limitación. En primer lugar, se realiza una clasificación de la especie arbórea para establecer un contexto biológico y, posteriormente, un detector de objetos localiza y clasifica los patógenos específicos asociados a dicha especie. Los resultados muestran que incorporar la identificación previa del huésped incrementa la precisión media (mAP) en un 10% respecto a los detectores genéricos de una sola etapa ("end-to-end"), demostrando la superioridad de los modelos condicionados biológicamente.

1. Introducción

Las enfermedades de las plantas representan una amenaza significativa para la seguridad alimentaria y la estabilidad económica agrícola, causando pérdidas estimadas de hasta el 30% en el rendimiento de los cultivos cada año [9]. La identificación temprana y precisa de estas patologías es crucial para aplicar medidas de control eficaces y minimizar el uso de químicos. Tradicionalmente, este diagnóstico ha dependido de la inspección visual manual por parte de expertos, un proceso que resulta laborioso, subjetivo y difícil de escalar a grandes explotaciones [1].

En la última década, la visión por computador y, específicamente, el aprendizaje profundo, han emergido como herramientas poderosas para automatizar esta tarea. El uso de Redes Neuronales Convolucionales (*CNNs*) ha permitido grandes avances tanto en la clasificación de imágenes (identificar si una hoja está enferma) como en la detección de objetos (localizar la lesión exacta en la hoja) [6]. Sin embargo, la mayoría de las arquitecturas actuales abordan la detección de enfermedades como un problema monolítico donde los modelos se entrenan para detectar cualquier enfermedad en cualquier tipo de hoja simultáneamente, ignorando a menudo la estructura jerárquica natural de la taxonomía biológica.

Debido a que muchas patologías foliares comparten características visuales similares, independientemente de la especie, los detectores agnósticos al huésped sufren de una alta confusión inter-clase. Esto suele derivar en predicciones erróneas donde se asocian enfermedades a plantas incompatibles, reduciendo la precisión y confiabilidad del sistema en aplicaciones del mundo real.

Para resolver esto, proponemos un enfoque jerárquico que imita el diagnóstico experto donde primero se identifica la especie del árbol para simplificar la búsqueda de la enfermedad y luego se aplica un detector especializado para esa especie.

2. Estado del Arte

2.1. Clasificación

Los dos paradigmas principales en clasificación de imágenes son las arquitecturas basadas en *Vision Transformers* (*ViT*) y las Redes Neuronales Convolucionales (*CNNs*). Nuestro objetivo fue contrastar la capacidad de modelado global de los *Transformers* frente a la eficiencia inductiva de las *CNNs*. Adicionalmente, dada la orientación práctica de este proyecto hacia una futura aplicación móvil, se incluyó en el estudio un modelo diseñado específicamente para entornos de recursos limitados (*edge devices*).

2.1.1 MaxViT

Dentro de la familia de los *Transformers*, *MaxViT* (*Multi-Axis Vision Transformer*) combina tanto mecanismos de atención global como convoluciones locales para capturar

características a múltiples escalas [11]. Su característica distintiva es el mecanismo de *Multi-Axis Self-Attention* (*Max-SA*), que descompone el cálculo de atención en dos operaciones; atención local (*Block Attention*) para capturar texturas finas, y atención global dispersa (*Grid Attention*) para relacionar partes distantes de la imagen.

2.1.2 EfficientNetV2

EfficientNetV2 es una evolución de la familia de *EfficientNet* (*CNN*) que optimiza tanto la arquitectura como el proceso de entrenamiento para mejorar la velocidad y precisión [10]. Esta arquitectura mejora a su predecesora mediante la introducción de bloques *Fused-MBConv* [10]. Estos bloques reemplazan las convoluciones *depthwise* separables tradicionales por convoluciones estándar 3×3 fusionadas en las primeras capas.

2.1.3 MobileNet

MobileNet es una arquitectura basada en *CNN* diseñada para dispositivos móviles y aplicaciones de *edge computing* [4]. La base de su eficiencia radica en las convoluciones *depthwise* separables, que factorizan la operación de convolución estándar en dos capas más ligeras (de tamaños 3×3 y 1×1), reduciendo drásticamente la cantidad de parámetros y operaciones [4]. Aunque su capacidad de representación es menor que los modelos anteriores, su inclusión es crítica para evaluar el *trade-off* entre precisión y latencia en una aplicación real para agricultores.

2.2. Detección de Objetos

El objetivo fundamental en detección de objetos es localizar y clasificar regiones de interés (*bounding boxes*). Históricamente, métodos dominantes como *Deformable Parts Model* (*DPM*) abordaban este problema mediante un enfoque de ventana deslizante (*sliding window*) que resultaban inviables computacionalmente al procesar todas las posibles sub-ventanas [2].

La evolución comenzó con **R-CNN**, un modelo de 3 etapas donde una búsqueda selectiva genera regiones potencialmente interesantes, una *CNN* extrae características y un *SVM* las clasifica [3]. Sus limitaciones principales son el alto coste de procesar regiones por separado y la falta de aprendizaje en la búsqueda selectiva.

2.2.1 Faster R-CNN

Faster R-CNN integra la propuesta de regiones dentro de la red neuronal, reemplazando el algoritmo fijo por una *Region Proposal Network* (*RPN*) entrenable y más rápida [8].

En primer lugar se definen *anchor points* sobre el mapa de características de la imagen, actuando como centros de posibles regiones con diferentes escalas y *ratios*. Seguidamente, la *RPN* predice simultáneamente la probabilidad de objeto y las coordenadas del *bounding box*.

A pesar de su alta precisión, el hecho de ser un modelo de dos etapas penaliza su velocidad de inferencia.

2.2.2 YOLO

Como respuesta a la latencia, *You Only Look Once* (*YOLO*) redefinió la detección no como una clasificación de regiones, sino como un problema único de regresión, descartando por completo el *pipeline* disjunto de *DPM* y *R-CNN*.

YOLO utiliza una única red neuronal convolucional que procesa la imagen completa de una sola vez. Divide la imagen en una cuadrícula ($S \times S$) donde cada celda predice B *bounding boxes* y sus respectivas puntuaciones de confianza.

La arquitectura unificada de *YOLO* permite una inferencia en tiempo real inalcanzable para los métodos de dos etapas [7]. Mientras que *Faster R-CNN* procesa aproximadamente 0.5 *FPS*, *YOLO* alcanza los 45 *FPS*, ofreciendo la inmediatez necesaria para aplicaciones agrícolas prácticas.

3. Metodología Propuesta

3.1. Dataset

Para validar nuestro enfoque jerárquico, hemos confeccionado un *dataset* que integra imágenes de tres cultivos. Las imágenes provienen de la plataforma *Roboflow Universe*, que contiene una gran cantidad de *datasets* de código abierto, y han sido seleccionadas por su calidad y variedad de condiciones de iluminación.

- *Rose Dataset*: Consta de 2,725 imágenes y abarca 4 clases (*Black Spot*, *Downy Mildew*, *Powdery Mildew* y *Healthy*).
- *Potato Dataset*: Comprende 812 imágenes distribuidas en 3 clases (*Early Blight*, *Late Blight*, *Healthy*).
- *Apple Dataset*: Incluye 1582 imágenes que cubren 5 clases (*Alternaria Spot*, *Brown Spot*, *Gray Spot*, *Rust*, *Healthy*).

El *dataset* combinado se dividió aleatoriamente en subconjuntos de Entrenamiento (70%), Validación (15%) y Prueba (15%), preservando la estratificación de clases para garantizar una evaluación equilibrada.

El *dataset* de manzanas requirió un preprocesamiento adicional, ya que sus anotaciones originales estaban en formato de máscaras de segmentación semántica en lugar de coordenadas de *bounding boxes*. Utilizando *OpenCV* (`cv2.findContours` y `cv2.boundingRect`), se convirtieron automáticamente las máscaras de las lesiones en *bounding boxes*, generando las coordenadas necesarias para su integración en el flujo de trabajo de *YOLO*.

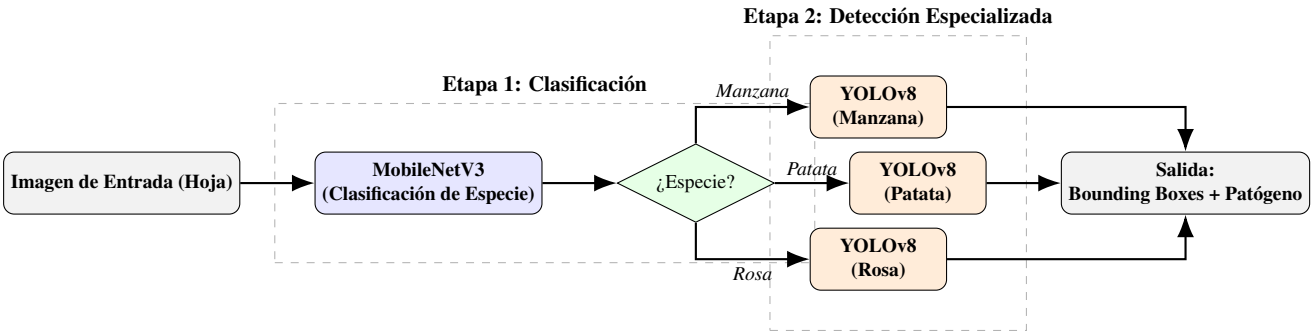

Figure 1. Arquitectura del Pipeline Especializado

3.2. Arquitecturas

Para evaluar la eficacia del enfoque jerárquico propuesto, se implementaron y compararon tres variantes arquitectónicas del *pipeline* de detección. Todas comparten la etapa inicial de clasificación de especies mediante MobileNet, pero difieren en la estrategia de especialización durante la detección:

- ***Pipeline* Especializado**: La predicción del clasificador actúa como un enrutador dinámico. Como se ilustra en la Figura 1, dependiendo de la especie identificada, la imagen se procesa con un detector YOLOv8 independiente entrenado exclusivamente con el subconjunto de datos correspondiente (Rosa, Patata o Manzana). Esto maximiza la especialización del modelo y elimina la confusión inter-especie a nivel de detector, alineándose con el diagnóstico experto donde cada patología se evalúa en su contexto biológico específico.

- ***Pipeline* Unificado**: El clasificador identifica la especie, pero la detección se realiza mediante un único modelo YOLOv8 global entrenado con el *dataset* combinado de todas las especies y patógenos (Figura 4). Sin embargo, la predicción del clasificador se utiliza como mecanismo de *masking* lógico, restringiendo el espacio de búsqueda exclusivamente a las enfermedades biológicamente plausibles para la especie detectada. Este enfoque evalúa si el condicionamiento post-procesado es suficiente para reducir la ambigüedad sin incrementar el coste computacional de múltiples detectores.

- **Monolítico**: Un único detector YOLOv8 global entrenado sin información de especie, procesando todas las clases de patógenos simultáneamente en un enfoque completamente agnóstico al huésped. Esta configuración sirve como línea base para cuantificar el impacto de la información contextual biológica en la precisión y la generalización.

4. Experimentos

4.1. Selección de Modelos

La definición de la arquitectura final se fundamentó en un análisis exhaustivo del compromiso entre precisión predictiva y coste computacional, priorizando la viabilidad del despliegue en dispositivos móviles.

4.1.1 Clasificación

Para la etapa de clasificación de especies se seleccionó **MobileNet** [4] tras compararlo con modelos de mayor densidad como MaxViT y EfficientNetV2. Los experimentos preliminares revelaron que, dada la baja cardinalidad del problema de clasificación (limitado a tres clases), los modelos de alta capacidad sufrían una saturación de rendimiento, no aportando ganancias significativas de precisión que justificaran la latencia adicional introducida.

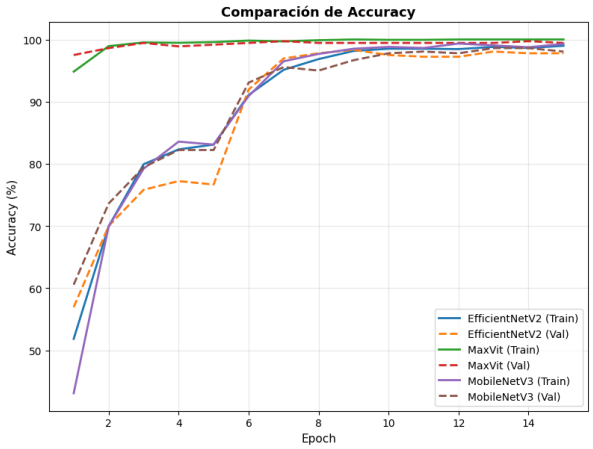

Figure 2. Comparativa de precisión entre modelos de clasificación.

Como se observa en las Figuras 2 y 3, MobileNet demostró ofrecer un rendimiento competitivo con una fracción de los parámetros y, crucialmente, una latencia de inferencia significativamente menor que los otros, alineándose con

los requisitos de eficiencia del sistema y viabilidad del despliegue en dispositivos móviles.

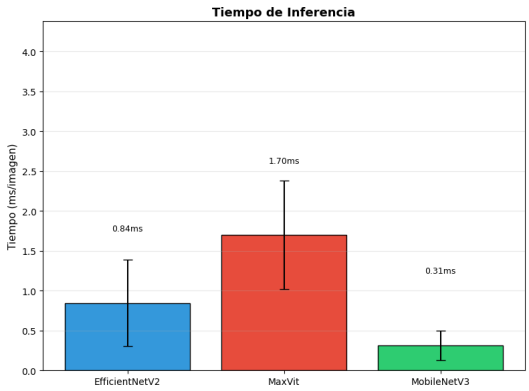

Figure 3. Comparativa de latencia de inferencia entre modelos de clasificación.

4.1.2 Detección de Objetos

Para la etapa de localización de patógenos, el estudio comparativo descartó inicialmente los detectores de dos etapas como **Faster R-CNN**. A pesar de su teórica precisión, esta arquitectura mostró una tendencia temprana al sobreajuste y una velocidad de inferencia insuficiente para la experiencia de usuario en tiempo real, debido a la sobrecarga computacional de RPN. Por consiguiente, se optó por el paradigma de una sola etapa, acotando la selección final a las variantes *small* de **YOLOv8** y la reciente **YOLO11**.

Modelo	Params (M)	mAP@50	mAP@50-95
YOLOv8s	11.2	**0.8741**	**0.5899**
YOLOv11s	9.4	0.8709	0.5856
YOLOv11n	**2.6**	0.8652	0.5815
YOLOv8n	8.7	0.8649	0.5797
YOLOv5s	9.1	0.8691	0.5792
YOLOv5n	**2.6**	0.8655	0.5775
YOLOv3-tiny	8.7	0.8365	0.5043
Faster R-CNN	> 40	0.7473	0.4321

Table 1. Comparativa de eficiencia (Parámetros) y precisión promedio (mAP) incluyendo Faster R-CNN.

Aunque ambas versiones de YOLO mostraron un desempeño sobresaliente, la balanza se inclinó finalmente a favor de **YOLOv8s** por su superior estabilidad y eficiencia. En términos de robustez inter-dominio, la versión 8 ofreció una mayor consistencia, superando a la v11 en el *dataset* de rosas (+1.5% mAP) y manteniendo un empate técnico en patatas, mientras que las mejoras de la v11 en manzanas fueron marginales y no compensaron su inestabilidad en otros escenarios. Críticamente, YOLOv8s demostró ser computacionalmente más eficiente, requiriendo menores tiempos de convergencia durante el entrenamiento y ofreciendo una latencia de inferencia un 6.4% menor (2.66 img/s frente a 2.50 img/s).

Esta combinación de estabilidad predictiva y agilidad de procesamiento consolidó a YOLOv8s como el detector idóneo para la arquitectura jerárquica propuesta.

4.2. Configuración Experimental

La ejecución de los experimentos y el entrenamiento de los modelos se llevaron a cabo utilizando el entorno de computación de Google Colab, aprovechando la aceleración por hardware mediante GPUs NVIDIA A100.

Para la etapa inicial de clasificación de especies, se instanció la arquitectura **MobileNetV3 Small** con pesos preentrenados en ImageNet-1k, implementando una estrategia de aprendizaje progresivo en dos fases.

Inicialmente, se aplicó *freezing* a la red troncal para entrenar exclusivamente la cabecera de clasificación durante 5 épocas, empleando el optimizador AdamW con una tasa de aprendizaje de 1×10^{-3}. Posteriormente, se procedió al *unfreezing* de los parámetros para una fase de *fine-tuning* durante 10 épocas, donde el *learning rate* se redujo a 8×10^{-6}.

En lo referente a la etapa de detección, la configuración del modelo **YOLOv8s** también se fundamentó en *transfer learning* desde el *dataset* COCO. El entrenamiento se estandarizó con una resolución de entrada de 640×640 píxeles y un *batch size* de 16, extendiéndose a lo largo de 30 épocas completas. Este proceso mantuvo las configuraciones de optimización predeterminadas del *framework* Ultralytics [5] e integró técnicas de *data augmentation* en tiempo de ejecución, destacando el uso de *mosaic augmentation*, para incrementar la variabilidad de las muestras y mejorar la capacidad de generalización del modelo.

4.3. Métricas de Evaluación

En tareas de detección de objetos, la métrica estándar es la *mean average precision* (mAP), la cual se fundamenta en el coeficiente de *Intersection over Union* (IoU).

$$\text{IoU} = \frac{|B_p \cap B_{gt}|}{|B_p \cup B_{gt}|} \tag{1}$$

La IoU cuantifica la superposición entre la predicción del modelo (B_p) y la anotación real (B_{gt}) de la *bounding box*, sirviendo como criterio para determinar el éxito de una detección.

En la práctica se reportan los valores de **mAP50** y **mAP50-95** para ofrecer una visión completa del desempeño del detector. El **mAP50** establece un umbral de tolerancia relajado (IoU ≥ 0.50), considerando válida cualquier predicción que se solape al menos en un 50% con la referencia real, lo cual es indicativo de la capacidad del modelo para una localización general del objeto. Por el contrario, la

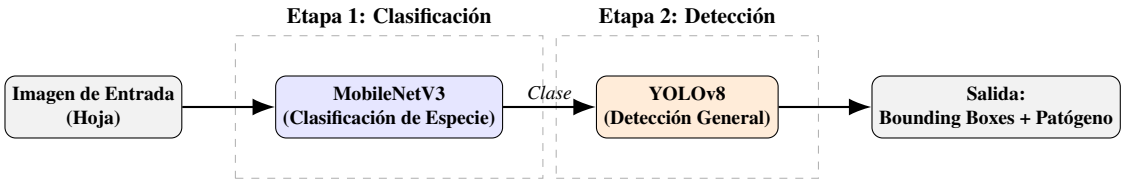

Figure 4. Arquitectura del Pipeline Unificado

métrica **mAP50-95** es mucho más exigente, ya que calcula el promedio de la precisión sobre diez umbrales progresivos de IoU (desde 0.50 hasta 0.95 en intervalos de 0.05).

4.4. Resultados

Al analizar la Tabla 2, nos encontramos con un escenario muy interesante. A diferencia de lo que podría esperarse, el modelo **Monolítico** demostró ser bastante competente, alcanzando un mAP50 promedio de 0.8742. Esto indica que la arquitectura YOLOv8 es robusta y capaz de aprender múltiples clases simultáneamente sin desmoronarse. Sin embargo, los datos también revelan que este enfoque generalista tiene una cota superior, aunque detecta bien, le cuesta perfeccionar los resultados en las clases más difíciles o cuando se requiere una precisión geométrica muy alta.

La comparación con el método **Unificado** es reveladora. Los resultados son prácticamente idénticos a los del Monolítico (por ejemplo, en patatas ambos obtienen un 0.7780 y en manzanas un 0.9302). Esto nos indica que el modelo base no estaba cometiendo errores de confusión de especies, por lo que aplicar un filtro lógico a posteriori no aportó ninguna mejora real. El problema no era la confusión taxonómica, sino la falta de capacidad de la red para especializarse en los detalles finos de cada plaga cuando tiene que atender a todas a la vez.

Aquí es donde brilla el enfoque **Especializado** puesto que al entrenar modelos dedicados logramos romper ese techo de rendimiento que limitaba a los otros dos enfoques. La mejora más notable se dio en la clase patatas, la más difícil del conjunto, donde la precisión saltó de un 0.77 a un 0.84. De igual forma, en las rosas, aunque la detección ya era buena, el modelo especializado mejoró drásticamente la calidad del ajuste de las cajas (*bounding boxes*), subiendo el mAP50-95 de 0.54 a casi 0.65. Como se puede apreciar en la Figura 5, el modelo es capaz de localizar con precisión múltiples lesiones en una misma hoja, incluso en casos de solapamiento parcial. Esto confirma que, para obtener un rendimiento de excelencia (con un promedio global de 0.9170), es necesario que la red neuronal se centre exclusivamente en un solo cultivo, captando matices que un modelo generalista pasa por alto.

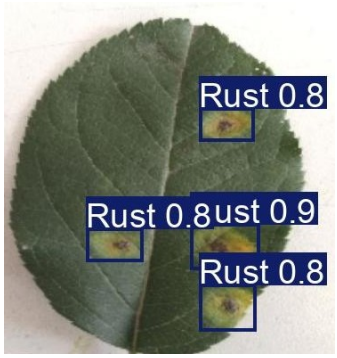

Figure 5. Ejemplo de detección y clasificación de patógenos en una hoja de manzana utilizando el *pipeline* especializado.

5. Conclusiones

El presente trabajo ha explorado cómo la integración del contexto biológico puede refinar la detección de enfermedades vegetales. Nuestros experimentos demuestran que, si bien las arquitecturas modernas como YOLOv8 tienen una gran capacidad de generalización (logrando un buen desempeño base), la estrategia jerárquica de especialización sigue siendo superior para alcanzar los niveles de precisión necesarios en una aplicación real de agricultura de precisión.

Los datos validan que la segregación del problema (usar un modelo experto por cada especie) aporta un valor añadido significativo. Hemos logrado elevar la precisión media (mAP50) de un 0.8742 en el modelo base a un 0.9170 en el sistema propuesto. Más importante aún es la ganancia en la calidad de la localización (mAP50-95), que subió de 0.61 a casi 0.68. Este incremento demuestra que los modelos especializados no solo encuentran más enfermedades (especialmente en casos difíciles como la patata), sino que delimitan el daño con mucha mayor exactitud, algo fundamental si en el futuro se quiere estimar la severidad de la infección.

Como parte fundamental de este trabajo, se ha desarrollado y publicado todo el código fuente del proyecto en un repositorio de acceso abierto. En él se incluyen los *notebooks* utilizados para el entrenamiento de los modelos, los *scripts* de validación y las instrucciones necesarias para reproducir los experimentos presentados. Todo

Especie	Monolítico		Unificado		Especializado	
	mAP50	**mAP50-95**	**mAP50**	**mAP50-95**	**mAP50**	**mAP50-95**
Manzana	0.9302	0.6932	0.9302	0.6931	0.9530	0.7250
Patatas	0.7780	0.6153	0.7780	0.6153	0.8411	0.6654
Rosas	0.9144	0.5445	0.9142	0.5441	0.9569	0.6490
Avg	**0.8742**	**0.6177**	**0.8741**	**0.6175**	**0.9170**	**0.6798**

Table 2. Comparativa de rendimiento (mAP50 y mAP50-95) entre las tres estrategias evaluadas.

el material está disponible en https://github.com/GabrielFranciscoSM/Hojas_con_resfriado, con el objetivo de que estos resultados puedan servir de base para futuros estudios o mejoras por parte de la comunidad universitaria

En conclusión, aunque el filtrado lógico (método Unificado) resultó innecesario debido a la robustez del detector base, la especialización completa demostró ser el camino correcto. Al combinar la eficiencia de MobileNetV3 para identificar el cultivo con la precisión de detectores YOLOv8s dedicados, hemos diseñado un sistema que ofrece lo mejor de dos mundos: la agilidad necesaria para funcionar en dispositivos móviles y la fiabilidad de un diagnóstico experto.

5.1. Líneas Futuras

Una de las extensiones más inmediatas para este trabajo sería evaluar la escalabilidad de la arquitectura jerárquica ante un aumento significativo en la diversidad de cultivos. Actualmente, el sistema ha validado su eficacia con tres especies, pero un escenario agrícola real implica gestionar docenas de variedades. Sería interesante investigar si el clasificador MobileNetV3 mantiene su precisión al actuar como *router* para 20 o 30 especies distintas, o si el aumento de clases comenzaría a introducir cuellos de botella que requieran una red más profunda. Ampliar el *dataset* permitiría confirmar si la estrategia de dividir el problema sigue siendo superior a los enfoques monolíticos a gran escala.

Desde una perspectiva de *deployment*, el siguiente paso lógico es la implementación física de estos modelos en una aplicación móvil nativa para realizar pruebas de campo en tiempo real. Aunque hemos simulado las restricciones de hardware, el despliegue real implica retos adicionales como la variabilidad extrema de la iluminación exterior o la estabilidad de la cámara en mano. En esta línea, se propone explorar técnicas de *TinyML*, como la cuantización de pesos (pasar de precisión de 32 bits a 8 bits) y la poda neuronal (*pruning*), para reducir aún más el tamaño de los modelos YOLOv8. El objetivo sería lograr que el sistema funcione fluidamente incluso en teléfonos de gama baja sin conexión a internet, democratizando el acceso a esta tecnología.

Finalmente, desde el punto de vista agronómico, el sistema podría evolucionar de la simple detección a la estimación de la severidad del daño. Actualmente, el modelo nos dice "dónde" está la enfermedad y "qué" es, pero no "cuánto" daño ha sufrido la planta. Incorporar técnicas de segmentación semántica o algoritmos de post-procesado que calculen el porcentaje de área foliar afectada sería un valor añadido enorme. Esto permitiría al sistema no solo diagnosticar, sino recomendar la urgencia del tratamiento o la dosis exacta de producto fitosanitario necesaria, convirtiendo la herramienta en un asistente integral para la toma de decisiones en el cultivo.

References

[1] Jayme Garcia Arnal Barbedo. Digital image processing techniques for detecting, quantifying and classifying plant diseases. *SpringerPlus*, 2(1):660, 2013. 1

[2] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010. 2

[3] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv preprint arXiv:1311.2524*, 2014. 2

[4] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 2, 3

[5] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolo. https://github.com/ultralytics/ultralytics, 2023. 4

[6] Sharada P Mohanty, David P Hughes, and Marcel Salathé. Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7:1419, 2016. 1

[7] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2016. 2

[8] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016. 2

[9] Serge Savary, Laetitia Willocquet, Sarah Jane Pethybridge, Paul Esker, Neil McRoberts, and Andy Nelson. The global

burden of pathogens and pests on major food crops. *Nature Ecology & Evolution*, 3(3):430–439, 2019. 1

[10] Mingxing Tan and Quoc V. Le. Efficientnetv2: Smaller models and faster training, 2021. 2

[11] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yandong Li. Maxvit: Multi-axis vision transformer. *arXiv preprint arXiv:2204.01697*, 2022. 2