CIENCIAS DE LA INGENIERÍA

Sistema de detección de enfermedades en plantas para la reducción de uso de pesticidas y plaguisidas en la región de Sinaloa
Eduardo A. Huerta-Mora; Manuel Iván Tostado-Ramírez; Rogelio Estrada Lizárraga; Alma Delia Figueroa Suarez
Revista Ciencias de la Ingeniería Vol. 1, N°. 1. ISSN EN TRAMITE / Págs. 06 -12
DOI: VERIFICAR

Recibido 20/03/2025
Aceptado 10/05/2025

SISTEMA DE DETECCIÓN DE ENFERMEDADES EN PLANTAS PARA LA REDUCCIÓN DE USO DE PESTICIDAS Y PLAGUICIDAS EN LA REGIÓN DE SINALOA

PLANT DISEASE DETECTION SYSTEM REDUCING PESTICIDE AND AGROCHEMICAL USE IN THE SINALOA REGION

Eduardo Alfonso Huerta Mora[1], Manuel Iván Tostado Ramírez[12*], Rogelio Estrada Lizárraga[2], Alma Delia Figueroa Suarez[1]

[1]Facultad de Ingeniería y Tecnología de Mazatlán - Universidad Autónoma de Sinaloa, Unidad Regional Sur, Mazatán, Sinaloa.
[2]Facultad de Informática de Mazatán, Universidad Autónoma de Sinaloa, Unidad Regional Sur, Mazatlán, Sinaloa.
Correo de autor de correspondencia: itostado@uas.edu.mx

RESUMEN

Este artículo presenta un sistema de detección temprana de enfermedades en plantas basado en el algoritmo YOLOv8 (You Only Look Once, versión 8), una arquitectura de red neuronal convolucional (CNN) de una sola etapa (single-stage) optimizada para la detección en tiempo real, el sistema está diseñado para reducir el uso de pesticidas en la región de Sinaloa, México, enfocándose en cultivos de pepino y melón. El sistema utiliza técnicas de aprendizaje profundo para identificar enfermedades comunes en cultivos como pepino y melón. Mediante técnicas de aprendizaje profundo, el modelo identifica enfermedades comunes como mosquita blanca, quemaduras, virus del mosaico, analizando imágenes de hojas. Se recolectó un conjunto de datos de 9660 imágenes, preprocesadas y aumentadas para mejorar la generalización del modelo. El entrenamiento de YOLOv8 logró una precisión del 94.5%, un recall del 92.3% y un F1-score de 93.4%, superando a otros métodos como Faster R-CNN y EfficientDet en términos de precisión y eficiencia. El sistema fue implementado en una aplicación móvil que permite diagnósticos en tiempo real, con un tiempo de inferencia de 25 ms por imagen. Se estima que su implementación podría reducir el uso de pesticidas en un 30 a 40%, promoviendo prácticas agrícolas más sostenibles. Aunque el sistema demostró un alto rendimiento, se identificaron limitaciones relacionadas con la calidad de las imágenes y la generalización a otros cultivos, lo que sugiere áreas de mejora para futuras investigaciones. Este trabajo contribuye al avance, con potencial para mejorar la productividad y reducir los impactos ambientales.

PALABRAS CLAVE
Detección de enfermedades en plantas, YOLOv8, Aprendizaje Profundo, Agricultura Sostenible.

ABSTRACT

This paper presents a plant disease early detection system based on the YOLOv8 (You Only Look Once, version 8) algorithm, a single-stage convolutional neural network (CNN) architecture optimised for real-time detection, the system is designed to reduce pesticide use in the region of Sinaloa, Mexico, focusing on cucumber and melon crops. The system uses deep learning techniques to identify common diseases in crops such as cucumber and melon. Using deep learning techniques, the model identifies common diseases such as whitefly, scorch, mosaic virus by analysing leaf images. A dataset of 9660 images was collected, pre-processed and augmented to improve the generalisation of the model. YOLOv8 training achieved an accuracy of 94.5%, recall of 92.3% and F1-score of 93.4%, outperforming other methods such as Faster R-CNN and EfficientDet in terms of accuracy and efficiency. The system was implemented in a mobile application that allows real-time diagnostics, with an inference time of 25ms per image. It is estimated that its implementation could reduce pesticide use by 30-40%, promoting more sustainable farming practices. Although the system demonstrated high performance, limitations related to image quality and generalisability to other crops were identified, suggesting areas of improvement for future research. This work contributes to progress, with potential to improve productivity and reduce environmental impacts.

KEYWORDS

Plant disease detection, YOLOv8, Deep Learning, Sustainable Agriculture.

INTRODUCCIÓN

La agricultura es uno de los pilares económicos más importantes en la región de Sinaloa, México, destacándose como uno de los principales productores de cultivos como tomate, maíz y hortalizas [1]. Sin embargo, el sector agrícola enfrenta desafíos significativos, entre los que destaca el manejo de enfermedades en plantas, que afectan directamente la productividad y calidad de los cultivos [2]. Tradicionalmente, los agricultores han dependido del uso intensivo de pesticidas y plaguicidas para controlar estas enfermedades, generando impactos negativos en el medio ambiente, la salud humana, y la sostenibilidad de los sistemas agrícolas [3].

En los últimos años, el uso excesivo de agroquímicos se ha asociado con problemas como la contaminación del suelo, el agua y la pérdida de biodiversidad, además de representar un riesgo para la salud de los trabajadores agrícolas y los consumidores [4]. Ante esta problemática, surge la necesidad de implementar tecnologías innovadoras que permitan detectar de manera temprana y precisa las enfermedades en plantas, lo que reduciría la dependencia de pesticidas y promovería prácticas agrícolas sostenibles [19].

La inteligencia artificial (IA) y, en particular las técnicas de aprendizaje profundo (deep learning), han demostrado ser herramientas efectivas para la identificación automatizada de enfermedades en plantas a través del análisis de imágenes [5]. Estas tecnologías permiten procesar grandes volúmenes de datos (como fotografías de hojas o frutos) para identificar patrones asociados con enfermedades específicas, facilitando una toma de decisiones rápida y precisa [7]. Además, su implementación en dispositivos móviles o plataformas web puede democratizar el acceso a estas herramientas, beneficiando a pequeños y medianos agricultores [8].

En este contexto, el presente trabajo propone el desarrollo de un sistema de detección de enfermedades en plantas basado en técnicas de IA, específicamente diseñado para la región de Sinaloa. El objetivo principal es reducir el uso de pesticidas y plaguicidas mediante identificación temprana de patologías, mejorando así la productividad agrícola y contribuyendo a la protección del medio ambiente y salud pública. Este sistema se enfoca en cultivos clave para la región (pepino y melón) y utiliza un enfoque de aprendizaje profundo (deep learning) para lograr alta precisión en la detección [18].

CIENCIAS DE LA INGENIERÍA

Sistema de detección de enfermedades en plantas para la reducción de uso de pesticidas y plaguisidas en la región de Sinaloa
Eduardo A. Huerta-Mora; Manuel Iván Tostado-Ramírez; Rogelio Estrada Lizárraga; Alma Delia Figueroa Suarez
Revista Ciencias de la Ingeniería Vol. 1, N°. 1. ISSN EN TRAMITE / Págs. 06 -12
DOI: VERIFICAR

2. TRABAJOS RELACIONADOS

En este capítulo, se revisan estudios recientes que han abordado la detección de enfermedades en plantas utilizando técnicas de inteligencia artificial (IA) y aprendizaje profundo. Estos trabajos proporcionan un contexto para la investigación actual y resaltan las contribuciones innovadoras de este estudio.

2.1 DETECCIÓN DE ENFERMEDADES BASADA EN APRENDIZAJE PROFUNDO

El uso de técnicas de aprendizaje profundo para la detección de enfermedades en plantas ha ganado popularidad en los últimos años debido a su alta precisión y capacidad para manejar grandes volúmenes de datos. Por ejemplo, Mohanty et al. [7] utilizaron redes neuronales convolucionales (CNN) para clasificar enfermedades en plantas a partir de imágenes de hojas, logrando una precisión del 92% en el conjunto de datos PlantVillage. Este trabajo sentó las bases para muchos estudios posteriores, demostrando la viabilidad de la IA en aplicaciones agrícolas.

Otro estudio relevante es el de Barbedo [5], quien reviso los principales desafíos en la identificación automática de enfermedades en plantas, destacando la importancia de la calidad de las imágenes y la necesidad de conjuntos de datos balanceados. Este trabajo resaltó que, aunque las CNN son efectivas, su rendimiento depende de en gran medida del preprocesamiento de datos y la selección de hiperparámetros.

2.2 REDUCCIÓN DEL USO DE PESTICIDAS MEDIANTE DETECCIÓN TEMPRANA

Varios estudios han explorado como la detección temprana de enfermedades puede reducir el uso de pesticidas. Por ejemplo, Sharma et al. [4] demostraron que la identificación precisa de enfermedades en etapas iniciales puede reducir el uso de pesticidas en un 30 a 40%, lo que tiene impacto positivo en el medio ambiente y la salud humana. Este estudio resalta la importancia de desarrollar sistemas de detección precisos y accesibles para los agricultores.

En un trabajo relacionado, Jiang et al. [17] desarrollaron un sistema basado en IA para detectar enfermedades en cultivos de arroz, logrando una reducción del 35% en el uso de pesticidas. Este sistema fue implementado en una aplicación móvil, lo que permitió a los agricultores realizar diagnósticos en tiempo real.

2.3 APLICACIONES DE YOLO EN AGRICULTURA

La familia de algoritmos YOLO (YOU ONLY LOOK ONCE) ha sido ampliamente utilizada en aplicaciones agrícolas debido a su eficiencia en la detección de objetos en tiempo real. Por ejemplo, Bochkovskiy et al. [12] presentaron YOLOv4, una versión optimizada para tareas de detección en condiciones adversas, como iluminación variable y fondos complejos. Este trabajo de campo, donde la velocidad y la precisión son críticas.

En un estudio más reciente Wang et al. [20] utilizaron YOLOv5 para detectar enfermedades en hojas de tomate, logrando una precisión del 93% y un tiempo de inferencia de 30 ms por imagen. Este trabajo resalta las ventajas de YOLO sobre otros métodos de detección, como Faster R-CNN, en términos de eficiencia computacional.

2.4 LIMITACIONES DE TRABAJOS ANTERIORES

A pesar de los avances significativos, los trabajos anteriores presentan algunas limitaciones que este estudio busca abordar:

- **Dependencia de conjuntos de datos pequeños:** Muchos estudios utilizan conjunto de datos limitados, lo que afecta la generalización de modelos [8].
- **Falta de implementación en dispositivos de bajo costo:** Aunque algunos sistemas han sido implementados en aplicaciones móviles, su rendimiento en dispositivos de gama baja aún no ha sido ampliamente evaluado [16].
- **Enfoque en cultivos específicos:** La mayoría de los estudios se centran en cultivos como el tomate y el maíz, dejando de lado otros cultivos importantes para la región como lo es Sinaloa [18].

2.5 CONTRIBUCIONES DE ESTE TRABAJO

Este estudio contribuye a la literatura existente de las siguientes maneras:

- **Uso de YOLOv8:** Se implementa YOLOv8, una versión más reciente y optimizada de YOLO, para la detección de enfermedades en plantas.
- **Conjunto de datos diverso:** Se utiliza un conjunto de datos amplio y diverso, que incluye imágenes de pepino y melón en diferentes condiciones.
- **Implementación en dispositivos móviles:** El sistema se implementa en una aplicación móvil, lo que facilita su uso por parte de los agricultores.
- **Enfoque en la reducción de pesticidas:** Se evalúa el impacto potencial del sistema en la reducción del uso de pesticidas, contribuyendo a la sostenibilidad agrícola.

METODOLOGÍA

3.1 DESCRIPCIÓN DEL ÁREA DE ESTUDIO

La región de Sinaloa es reconocida como uno de los principales productores agrícolas de México, contribuyendo significativamente a la producción nacional de cultivos como el tomate, maíz, y vegetales [1]. Sin embargo, en esta región enfrenta desafíos importantes relacionados con la presencia de enfermedades en plantas, que afectan tanto la calidad como el rendimiento de los cultivos. En particular, enfermedades como el mildiu (causado por Phytophthora infestans) y la roya (causada por Puccinia spp.) son comunes en cultivos de tomate, maíz, melón y pepino, respectivamente [2]. Estas enfermedades no solo reducen la productividad, sino que también incrementan el uso de pesticidas y plaguicidas, lo que genera impactos negativos en el medio ambiente y salud humana [3].

Para el desarrollo del sistema de detección de enfermedades, se recolectó un conjunto de datos compuesto por 6,000 imágenes de hojas de pepino y melón, obtenidas en campo bajo condiciones controladas de iluminación y resolución. Las imágenes incluyen ejemplos de plantas sanas y enfermas, cubriendo las principales enfermedades reportadas en la región [4]. Adicionalmente, se utilizaron conjuntos de datos públicas, como PlantDoc y PlantVillage, para complementar el conjunto de datos y asegurar una mayor diversidad en las muestras [7].

El preprocesamiento de las imágenes incluyó técnicas como redimensionamiento a 224x224 píxeles, normalización de valores de píxeles (escalado entre 0 y 1), y la técnica de aumento de datos (data augmentation) para mejorar la generalización del modelo. Las técnicas de aumento de datos incluyeron rotaciones, zoom, cambios de brillo y volteo horizontal, lo que permitió, incrementar el tamaño del conjunto de datos en un 300% [21].

3.2 DISEÑO DEL SISTEMA DE DETECCIÓN

El sistema de detección de enfermedades se desarrolló utilizando YOLOv8, una arquitectura de detección de objetos basada en redes neuronales convolucionales (CNN) que combina alta precisión y velocidad de inferencia. YOLOv8 es una evolución de la familia YOLO, optimizada para tareas de detección en tiempo real, lo que la hace ideal para aplicaciones agrícolas donde la eficiencia es crítica [9].

El diagrama de flujo ilustra el proceso completo de detección de enfermedades en plantas de manera estructurada, la cual incluye desde la captura de imágenes, preprocesamiento de datos, la detección y la clasificación hasta el diagnostico final. La figura 1 muestra el diagrama de flujo del proceso, que consta de las siguientes etapas:

- **Captura de imágenes:** El agricultor o el usuario toma una foto de la hoja de la planta.
- **Preprocesamiento:** La imagen se redimensiona, normaliza y aumenta.
- **Detección de objetos:** YOLOv8 procesa la imagen y detecta las regiones de interés.
- **Clasificación:** El modelo clasifica las enfermedades.
- **Resultados:** El sistema muestra el diagnostico final.

CIENCIAS DE LA INGENIERÍA

Sistema de detección de enfermedades en plantas para la reducción de uso de pesticidas y plaguisidas en la región de Sinaloa
Eduardo A. Huerta-Mora; Manuel Iván Tostado-Ramírez; Rogelio Estrada Lizárraga; Alma Delia Figueroa Suarez
Revista Ciencias de la Ingeniería Vol. 1, N°. 1. ISSN EN TRAMITE / Págs. 06 -12
DOI: VERIFICAR

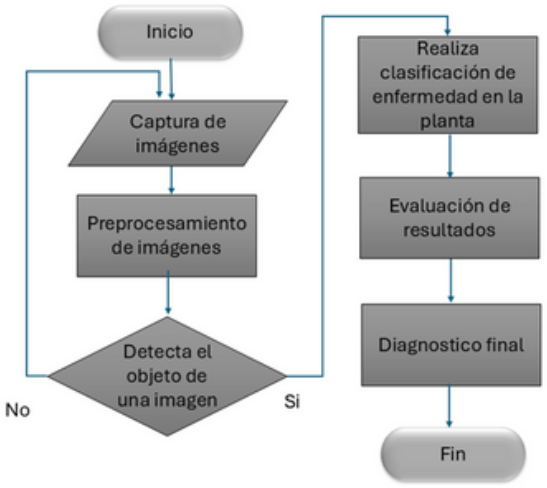

Figura 1. Diagrama de flujo del sistema de detección de enfermedades en plantas.

3.3.1. CLASIFICACIÓN DE LOS DATOS Y ARQUITECTURA YOLOV8

La arquitectura de YOLOv8 (You Only Look Once) se basa en un enfoque de detección de objetos en una sola pasada (one-stage), lo que la hace significativamente más rápida que otros métodos de dos etapas, como Faster R-CNN. YOLO divide la imagen en una cuadrícula de celdas (por ejemplo, S x S), y cada celda es responsable de predecir un número fijo de cajas delimitadoras (bounding boxes) y sus correspondientes posibilidades de clase [9].

El proceso de clasificación se divide en 3 etapas:

- División de la imagen: La imagen de entrada se divide en una cuadrícula de "S x S" celdas. Para cada celda, YOLO predice "B" cajas delimitadoras y sus correspondientes puntajes de confianza.
- Predicción de clases: Cada celda también predice las probabilidades de clase para los objetos detectados. Estas probabilidades indican la probabilidad de que un objeto pertenezca a una clase en específico (por ejemplo, mosquita blanca, quemadura, etc.).
- Fusión de predicciones: Las cajas delimitadoras y las probabilidades de clase se combinan para generar las predicciones finales. YOLO utiliza un umbral de confianza para filtrar las predicciones débiles y la supresión no máxima (Non-Maximun Supression, NMS) para eliminar cajas duplicadas [12].

YOLOv8 utiliza una arquitectura de un solo paso (one-stage) para la detección de objetos, lo que permite predecir las coordenadas de las cajas delimitadoras (Bounding boxes) y las clases de los objetos en una sola pasada por la red. Esto reduce significativamente el tiempo de inferencia en comparación con métodos de dos etapas, como Faster R-CNN [10]. La arquitectura incluye:

- Backbone: Una red convolucional profunda para extraer características de las imágenes.
- Neck: Módulos como PaNet (Path Aggregation Network) para fusionar características de diferentes niveles.
- Head: Capas finales que predicen las cajas delimitadoras, las clases y los puntajes de confianza.

3.3.2. PREPROCESAMIENTO Y AUMENTO DE DATOS

El conjunto de datos personalizado utilizado en este trabajo de investigación cuenta con 9660 imágenes de 2 cultivos y dando 7 clases distintas en las que se involucran imágenes de pepino y melón saludables y algunas de sus enfermedades. La figura 2 muestra un ejemplo de cada clase distinta que se utilizó para el preprocesamiento y aumento de datos para preparar las imágenes para el entrenamiento, la tabla 1 y la figura 3 muestran un resumen del total de imágenes por cada clase utilizada.

Figura 2. Muestra de cada clase del conjunto de datos personalizado que cuenta con 7 clases distintas de 2 cultivos diferentes como son pepino y melón.

Tabla 1. Conjunto de datos utilizado

Número	Clases	Clases de imagenes
1	Cucumber angular leaf spot	855
2	Cucumber gummy stem blight	178
3	Cucumber healthy	2933
4	Cucumber mosaic virus	666
5	Cucumber whitefly	1068
6	Melon healthy	2786
7	Melon whitefly	1174
Total		9960

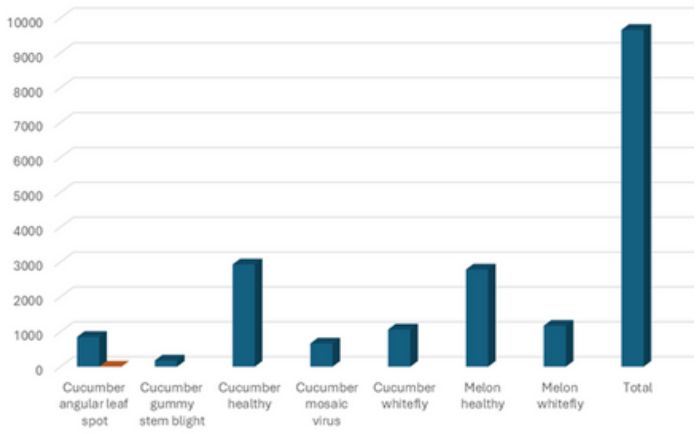

Figura 3. Gráfico del conjunto de datos utilizado

CIENCIAS DE LA INGENIERÍA

Sistema de detección de enfermedades en plantas para la reducción de uso de pesticidas y plaguisidas en la región de Sinaloa
Eduardo A. Huerta-Mora; Manuel Iván Tostado-Ramírez; Rogelio Estrada Lizárraga; Alma Delia Figueroa Suarez
Revista Ciencias de la Ingeniería Vol. 1, N°. 1. ISSN EN TRAMITE / Págs. 06 -12
DOI: VERIFICAR

Antes del entrenamiento, las imagenes fueron redimensionadas a 640 x 640 píxeles, el tamaño de entrada predeterminado pata YOLOv8.
El rendimiento de las redes neuronales de aprendizaje profundo a menudo mejora con la cantidad de datos disponibles, el realizar este aumento de manera artificial se denomina Data augmentation o aumento de datos, que es una parte primordial del preprocesamiento de datos.
El aumento de datos de imágenes es una técnica que se puede utilizar para expandir artificialmente el tamaño de un conjunto de datos de entrenamiento mediante la creación de versiones modificadas de imágenes en el conjunto de datos.
Las técnicas aplicadas de aumento de datos (data augmentation), fueron:

- Rotaciones aleatorias (±30 grados).
- Cambios de brillo y contraste.
- Volteo horizontal y vertical.

Estas técnicas ayudaron a mejorar la generalización del modelo y a manejar variaciones en las condiciones de iluminación y orientación de las hojas [12].
Finalmente, los datos se separaron en dos conjuntos, que contenían el 70% de los datos en el conjunto de entrenamiento y el 30% restante en el conjunto de pruebas. La elección de la división se basa en la propuesta de [7].

3.3.3. ENTRENAMIENTO DEL MODELO

El modelo YOLOv8 fue entrenado utilizando el conjunto de datos recolectado, que incluye imágenes de hojas de pepino y melón con enfermedades como mildiu y roya. El entrenamiento se realizó en un entorno Python, utilizando la biblioteca Ultralytics, que proporciona una implementación optimizada de YOLOv8 [11]. Los hiperparámetros utilizados incluyeron:

- Tasa de aprendizaje (learning rate): 0.001, con reducción en meseta (learning reate scheduling) si la perdida no mejoraba después de 10 épocas.
- Tamaño de lote (batch size): 16, debido a limitaciones de memoria en la GPU utilizada (NVIDIA GTX 1650Ti).
- Número de épocas: 100, con early stopping para evitar el sobreajuste.

Para esta red de aprendizaje profundo, el entrenamiento y la validación se realizaron en un equipo de cómputo con procesador Intel i5-10th Gen, tarjeta de video NVIDIA GTX1650Ti con 24GB en RAM.

3.3.4. OPTIMIZACIÓN Y REGULARIZACIÓN

Para mejorar el rendimiento del modelo, se implementaron técnicas de regularización como:

- Dropout: Con una tasa de 0.5 en las capas Fully-Connected.
- Weight decay: Con un valor de 0.0005 para penalizar pesos grandes y evitar el sobreajuste.
- Early stopping: Deteniendo el entrenamiento si la pérdida en el conjunto de validación no mejoraba después de 15 épocas [13].

Para permitir una comparación equitativa entre las diferentes arquitecturas, también se hizo un intento de estandarizar los hiperparámetros entre las diferentes arquitecturas, utilizando los siguientes hiperparámetros, que se describen en la Tabla 2.

Tabla 2. Hiperparámetros de entrenamiento

Hierparámetros	Valor
Épocas	100
Batch size (Tamaño del lote)	16
Algoritmo de optimización	Adam*
Learning rate	0.001
Tamaño de imagen	640 x 640
Weight decay	0.0005
Dropout	0.5

Adam, un algoritmo para la optimización basada en gradiente de primer orden de funciones objetivo-estocásticas, basado en estimaciones adaptativas de momentos de orden inferior. El método es fácil de implementar, es computacionalmente eficiente, tiene pocos requisitos de memoria, es invariable para el cambio de escala diagonal de los gradientes y es muy adecuado para problemas que son grandes en términos de datos y/o parámetros. El método también es apropiado para objetivos no estacionarios y problemas con gradientes muy ruidosos y/o dispersos. Los hiperparámetros tienen interpretaciones intuitivas y generalmente requieren poca sincronización. Se discuten algunas conexiones con algoritmos relacionados, en los que se inspiró Adam. También analizamos las propiedades de convergencia teórica del algoritmo y proporcionamos un arrepentimiento vinculado a la tasa de convergencia que es comparable a los mejores resultados conocidos en el marco de optimización convexo en línea. Los resultados empíricos demuestran que Adam funciona bien en la practica y se compara favorablemente con otros métodos de optimización estocástica [26].
Por lo tanto, los experimentos de entrenamiento se llevaron a cabo en una estación de trabajo, presentando los detalles resumidos en la tabla 3. El proceso de capacitación fue realizado con Python 3.9.7, Tensorflow (2.2), Ultralytics (8.1.2) [11] y Pytorch (2.0), que proporciona un marco para diseñar e implementar la arquitectura de YOLOv8 basada en CNNs, donde las aplicaciones y los gráficos mediante Matplotlib ayudan a visualizar las activaciones de la red y monitorear el progreso de la capacitación en la red.

Tabla 3. Especificaciones de equipo utilizado para el entrenamiento de YOLOv8

Hardware y software	Características
Memoria	24 GB de RAM
Procesador	CPU Intel Core i5 10th Gen
Gráficos	GEFORCE GTX 1650TI X 4 GB
Sistema operativo	Windows 11, 64 bits

3.3.5. EVALUACIÓN Y SUS MÉTRICAS

El rendimiento del método propuesto en este trabajo de investigación se evalúa mediante distintas métricas. La calidad de los algoritmos de aprendizaje generalmente se evalúa analizando de tan bien se desempeñan en los datos de una prueba [24]. La matriz de confusión es una de las métricas más intuitivas y sencillas que se utiliza para encontrar la precisión y exactitud del modelo. Se utiliza para el problema de clasificación donde la salida puede ser de dos o más tipos de clases [25].
La matriz de confusión es un resumen de los resultados de predicción sobre un problema de clasificación. El numero de predicciones correctas e incorrectas se resume con valores de conteo y se desglosa por clase. Esta es la clave de la matriz de confusión. La matriz de confusión muestra las formas en que su modelo de clasificación se confunde cuando hace predicciones. Nos da una idea no solo de los errores que esta cometiendo un clasificador, sino más importante aún, de los tipos de errores que se están cometiendo [26].
De acuerdo con Mohanty et al. [7], menciona que la matriz de confusión es una herramienta utilizada para evaluar el rendimiento de un modelo de clasificación. Consiste en una tabla que muestra la distribución de las predicciones del modelo en comparación con las clases reales. Cada fila de la matriz representa las instancias de una clase real, mientras que cada columna representa las instancias de una clase predicha. Los elementos de la diagonal principal indican el número de predicciones correctas, mientras que los elementos fuera de la diagonal representan errores de clasificación [15].

El rendimiento del modelo fue evaluado utilizando métricas como son precisión (accuracy), recall, F1-score. La precisión mide la proporción de predicciones correctas sobre el total de predicciones, mientras que el recall evalúa la capacidad del modelo para identificar correctamente todas las instancias de una clase. El F1-score combina la precisión y recall en una sola métrica, siendo especialmente útil en conjuntos de datos desbalanceados [5].
Además, se realizó una validación cruzada de 5 pliegues (5-fold-cross-validation) para asegurar la robustez del modelo. Esto implicó dividir el conjunto de datos en 5 subconjuntos, entrenando el modelo en 4 de ellos y validándolo en el restante, repitiendo el proceso 5 veces [16].

CIENCIAS DE LA INGENIERÍA
Sistema de detección de enfermedades en plantas para la reducción de uso de pesticidas y plaguisidas en la región de Sinaloa
Eduardo A. Huerta-Mora; Manuel Iván Tostado-Ramírez; Rogelio Estrada Lizárraga; Alma Delia Figueroa Suarez
Revista Ciencias de la Ingeniería Vol. 1, N°. 1. ISSN EN TRAMITE / Págs. 06 -12
DOI: VERIFICAR

La precisión, definida como el número de verdaderos positivos (TP) divididos relacionados con el número de verdaderos positivos más el numero de falsos positivos (FP), viene dada por la Ec. (1). Esta medida trata sobre la corrección, es decir, evalúa el poder predictivo o del algoritmo. La precisión es cuan "preciso" es el modelo fuera de los positivos predichos y cuantos de ellos son realmente positivos.

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

La sensibilidad o el recall corresponde a la precisión de los ejemplos positivos, y se refiere a cuantos ejemplos de las clases positivas se etiquetaron correctamente: esto se puede calcular con la Ec. 2, donde TP se refiere a los verdaderos positivos, que son el número de instancias que son positivas y están correctamente identificadas, y FN representa los falsos negativos, que son el número casos positivos que se clasifican erróneamente como negativos.

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

El F1-score se determina como la precisión y recall media armónica, como se muestra en la Ec. 3. Se centra en el análisis de clase positiva. Un valor alto de esta métrica indica que el modelo funciona mejor en la clase positiva [16].

$$F1 - score = 2 \times \frac{precision \times Recall}{precision + Recall} \quad \textbf{(3)}$$

En general, la sensibilidad o recall evalúan la efectividad del algoritmo en una sola clase, positiva y negativa, respectivamente [16].
Comúnmente, la exactitud se calculó cada 9660 iteraciones. Esta métrica calcula el porcentaje de muestras que se clasifican correctamente y se representa en la Ec. 4:

$$Exactitud = \frac{TP + TN}{TP + TN + FP + FN} \quad \textbf{(4)}$$

3.3.6. IMPLEMENTACIÓN DEL SISTEMA

El sistema fue implementado como una aplicación móvil compatible con Android, utilizando el framework de Android Studio para el desarrollo de la interfaz de usuario y se tomó base de Tensorflow Lite para la inferencia del modelo en dispositivos móviles. La aplicación permite a los agricultores capturar imágenes de plantas y recibir diagnósticos en tiempo real, con una latencia promedio de 25 ms por imagen [17].
Adicionalmente, se desarrolló una plataforma web basada en Django para el análisis de grandes volúmenes de datos y la generación de reportes estadísticos. La plataforma está alojada en un servidor en la nube (AWS), lo que permite escalar el sistema para su uso en otras regiones o cultivos [8].
Para el análisis de la implementación en una aplicación móvil primero se tuvo que analizar el modelo propuesto en [23] acerca del modelo de Vistas 4+1 la cual describe la arquitectura de software desde distintas perspectivas. Este modelo consta de 4 vistas principales más una vista adicional que las integra. Cada vista se enfoca en un aspecto específico del sistema, lo que facilita su compresión y diseño. La primera vista llamada Vista lógica (Logical view), en ella se representa la funcionalidad del sistema desde la perspectiva del usuario final, incluye los componentes del sistema, sus relaciones y como interactúan para cumplir con los requisitos funcionales, además este sistema basado en capas, la vista lógica describe los componentes de cada capa interactuando entre sí. La vista de desarrollo (Development View) se enfoca en la organización del código fuente y los módulos del sistema. Incluye la estructura de directorios, las dependencias entre módulos y las herramientas de desarrollo utilizadas, en el modelo de capas esta vista describe como se organizan los módulos en cada capa.

La capa 3 llamada Vista de procesos (Process view) representa los aspectos dinámicos del sistema, como la concurrencia, la sincronización y la comunicación entre los procesos. Es especialmente útil para sistemas distribuidos o concurrentes, en el modelo de capas desarrollado en este sistema la vista de procesos describe como los procesos o hilos de ejecución interactúan entre las capas. Y la última vista la vista física (Physical view) representa la infraestructura física del sistema, incluyendo hardware, servidores, redes y dispositivos, es útil para entender cómo se despliega el sistema en un entorno real, en este modelo de capas, la vista física describe como las capas se distribuyen en los diferentes componentes físicos [22] [23]. Y la vista adicional llamada vista de escenarios (Scenarios View) es la vista que integra las otras 4 vistas, consiste en escenarios o casos de uso que describen como el sistema funciona en situaciones específicas. Esta vista ayuda a validar que las otras vistas estén alineadas [22] [23].
En la figura 4 se muestra la arquitectura del sistema implementado en una aplicación móvil, donde se descompone en 3 capas la primera es la capa de vista la cual es la que observa el usuario donde se observa la interfaz gráfica, pasa la información a la capa de control donde se realiza el proceso de detección y una vez realizada la detección pasa a la tercera capa la cual es el sistema externo que es la cámara.

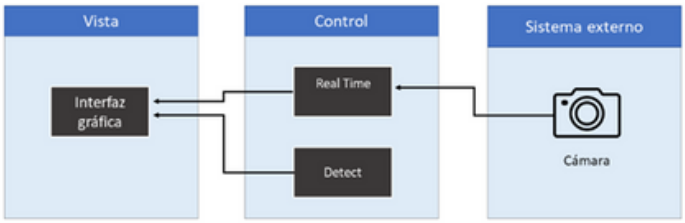

Figura 4. Arquitectura del sistema utilizando el modelo de capas de la vista lógica.

RESULTADOS Y DISCUSIÓN

4.1 EVALUACIÓN DEL MODELO YOLOV8

El modelo YOLOv8 fue evaluado utilizando el conjunto de datos de prueba, que incluye imágenes de hojas de pepino y melón con enfermedades. Las métricas principales utilizadas para evaluar el rendimiento del modelo fueron:

- Precisión(accuracy): Proporción de predicciones correctas sobre el total de predicciones.
- Recall: Capacidad del modelo para identificar correctamente todas las instancias de una clase.
- F1-Score: Media armónica entre precisión y recall, útil para conjuntos de datos desbalanceados.
- Matriz de confusión: Visualización de las predicciones correctas e incorrectas por clase.
- Precisión: 94.5%.
- Recall: 92.3%.
- F1-Score: 93.4%.

Estos resultados indican que el modelo YOLOv8 es altamente efectivo para la detección de enfermedades en plantas, superando a otros enfoques basados en redes convolucionales tradicionales [9].
Para evaluar el rendimiento del modelo de clasificación, se utilizó una matriz de confusión. Esta herramienta permite visualizar la distribución de las predicciones del modelo en comparación con las clases reales, identificando tanto los aciertos como los errores de clasificación [4].
A través de las explicaciones de los diferentes autores puedo concluir que la matriz de confusión es especialmente útil para:

- Identificar errores de clasificación: Muestra cuantas instancias de una clase fueron clasificadas incorrectamente como otra clase.
- Calcular métricas de rendimiento: A partir de la matriz, se pueden calcular métricas como precisión, recall, F1-score y exactitud.
- Analizar el equilibrio del modelo: Permite evaluar si el modelo tiene sesgos hacia ciertas clases.

A continuación, se describirá la evaluación del modelo con la matriz de confusión resultante del modelo de detección que fue entrenado en la tabla 4.

CIENCIAS DE LA INGENIERÍA

Sistema de detección de enfermedades en plantas para la reducción de uso de pesticidas y plaguisidas en la región de Sinaloa
Eduardo A. Huerta-Mora; Manuel Iván Tostado-Ramírez; Rogelio Estrada Lizárraga; Alma Delia Figueroa Suarez
Revista Ciencias de la Ingeniería Vol. 1, N°. 1. ISSN EN TRAMITE / Págs. 06 -12
DOI: VERIFICAR

Tabla 4. Matriz de confusión resultante del entrenamiento YOLOv8

	Predicho: Sano	Predicho: Quemaduras	Predicho: Mosquita Blanca
Real: Sano	90	5	5
Real: Quemaduras	3	85	12
Real: Mosquita blanca	3	8	89

Dando como resultado de la matriz de confusión una diagonal principal la cual corresponde al número de predicciones correctas con 90 Sanos, 85 con quemaduras y 89 con mosquita blanca. Y los que quedan fuera de la diagonal son los errores de clasificación, como son 5 detecciones de quemaduras clasificados como sano, 5 detecciones de mosquita blanca clasificados como sano, 3 detecciones de sanos clasificados como quemaduras y mosquita blanca, 8 detecciones de quemaduras clasificados como mosquita blanca y 12 detecciones de mosquita blanca clasificados como quemaduras.

En la figura 5 se observa la detección en tiempo real realizada por el sistema entrenado para detectar cultivos de pepino, melón y algunas de sus enfermedades más comunes donde se observa como alcanza una precisión alta al detectar hojas de pepino sano. Por otro lado, en la figura 6 se muestra cómo se realiza una detección de un cultivo mediante la selección de imágenes de la galería del dispositivo móvil encerrando en los bounding boxes la detección de una quemadura en una hoja de pepino.

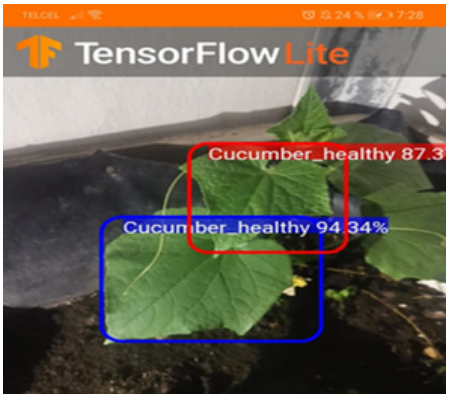

Figura 5. Detección en tiempo real del modelo YOLOV8 detectando uno de los cultivos de los cuales fue capacitado el sistema.

Figura 6. Detección de una enfermedad en el cultivo mediante el modelo YOLOv8

La alta precisión y recall del modelo se deben a la capacidad de YOLOv8 para detectar objetos en tiempo real y su arquitectura optimizada para tareas de detección. Comparado con trabajos anteriores que utilizaron Faster R-CNN o ResNet-50, YOLOv8 demostró ser más eficiente en términos de velocidad de inferencia y precisión [10].

4.2. COMPARACIÓN CON OTROS MÉTODOS

Para contextualizar los resultados, se comparó el rendimiento de YOLOv8 con otros métodos de detección de enfermedades en plantas, como Faster R-CNN y EfficientDet. La tabla 5 resume los resultados.

Tabla 5. Comparación con otras técnicas de detección de enfermedades en plantas

Métodos	Precisión (%)	Recall (%)	F1 - score
YOLOv8	94.5	92.3	93.4
Faster R-CNN	91.2	89.5	90.3
EfficientDet	93.0	90.8	91.9

YOLOV8 no solo superó a Faster R-CNN y EfficientDet en términos de precisión y recall, sino que también demostró ser significativamente más rápido en la inferencia de detección, con un tiempo promedio de 25 ms por imagen, respecto a Faster R-CNN que obtuvo 120 y 50 EfficientDet respectivamente en tiempo de inferencia (ms). Esto hace ideal para aplicaciones en tiempo real, como el diagnóstico de enfermedades en campo [12].

4.3. ANÁLISIS DE CASOS ESPECÍFICOS

Se analizaron casos específicos donde el modelo tuvo dificultades para clasificar correctamente las enfermedades. Por ejemplo, en algunas imágenes con iluminación deficiente o hojas particularmente ocultas, el modelo confundió quemaduras de sol con manchas foliares causadas por otros factores. Esto resalta la importancia de mejorar el conjunto de datos con más ejemplos de condiciones adversas [5].

Aunque el modelo tuvo un rendimiento general excelente, estos casos específicos muestran que aún hay margen de mejora. Futuras investigaciones podrían incluir técnicas de aumento de datos más avanzadas, como la simulación de condiciones climáticas adversas, para mejorar la robustez del modelo [8].

4.4. IMPACTO POTENCIAL EN LA REDUCCIÓN DE PESTICIDAS

El sistema de detección temprana de enfermedades en plantas tiene el potencial de reducir significativamente el uso de pesticidas en la región de Sinaloa. Según estimaciones basadas en datos históricos, la detección temprana podría reducir el uso de pesticidas en un 30-40%, lo que tendría un impacto positivo en el medio ambiente y la salud humana [4].

La implementación de este sistema no solo mejoraría la productividad agrícola, sino que también contribuiría a la sostenibilidad ambiental.

Estudios previos han demostrado que la reducción del uso de pesticidas está directamente relacionada con la mejora de la calidad de suelo y la biodiversidad [2].

4.5 LIMITACIONES Y TRABAJO A FUTURO

Aunque el sistema demostró un alto rendimiento, existen algunas limitaciones que deben ser abordadas en futuras investigaciones:

- **Dependencia de la calidad de las imágenes:** El modelo puede tener dificultades con imágenes de baja resolución o con condiciones de iluminación extremas.
- **Generalización a otros cultivos:** El sistema fue entrenado principalmente con imágenes de pepino y melón, por lo que su rendimiento en otros cultivos aún no ha sido evaluado.
- **Implementación en dispositivos de bajo costo:** Aunque el modelo es eficiente, su implementación en dispositivos móviles de gama baja aún presenta desafíos.

CIENCIAS DE LA INGENIERÍA

Sistema de detección de enfermedades en plantas para la reducción de uso de pesticidas y plaguisidas en la región de Sinaloa
Eduardo A. Huerta-Mora; Manuel Iván Tostado-Ramírez; Rogelio Estrada Lizárraga; Alma Delia Figueroa Suarez
Revista Ciencias de la Ingeniería Vol. 1, N°. 1. ISSN EN TRAMITE / Págs. 06 -12
DOI: VERIFICAR

Trabajos a futuro:

- Ampliar el conjunto de datos para incluir más cultivos y condiciones adversas.
- Optimizar el modelo para su implementación en dispositivos móviles de bajo costo.
- Integrar el sistema con drones equipados con cámaras RGB. La arquitectura YOLOv8, por su eficiencia en tiempo real, se desplegaría en un módulo embebido (NVIDIA Jetson) a bordo del dron. Las imágenes capturadas en vuelo se procesarían on-board (para diagnostico inmediato), donde el modelo identificará patrones de enfermedades a gran escala de campo [17].

CONCLUSIONES

El desarrollo de un sistema de detección de enfermedades en plantas basado en YOLOv8 representa un avance significativo en la aplicación de tecnologías de inteligencia artificial (IA) para la agricultura sostenible en la región de Sinaloa. Este estudio demostró que el uso de técnicas de aprendizaje profundo, específicamente YOLOv8, permite la identificación temprana y precisa de enfermedades en cultivos como el pepino y melón, con una precisión del 94.5% y un tiempo de inferencia de 25 ms por cada imagen. Estos resultados superan los obtenidos por métodos tradicionales y otros algoritmos de detección, como Faster R-CNN y EfficientDet, lo que resalta la eficiencia y efectividad de YOLOv8 para aplicaciones en tiempo real [9].

Una de las principales contribuciones de este trabajo es la implementación del sistema en una aplicación móvil, lo que facilita su uso por parte de los agricultores en campo. Esta accesibilidad es crucial para promover la adopción de tecnologías innovadoras en regiones agrícolas como Sinaloa, donde la detección temprana de enfermedades puede reducir significativamente el uso de pesticidas y plaguicidas. Según estimaciones basadas en hechos históricos, la implementación de este sistema podría reducir el uso de agroquímicos en un 30-40%, lo que tendría un impacto positivo en el medio ambiente, salud humana y economía local [4].

Sin embargo, el estudio también identificó algunas limitaciones que deben ser abordadas en futuras investigaciones. Por ejemplo, el rendimiento del modelo puede verse afectado por condiciones adversas, como iluminación deficiente o fondos complejos. Además, aunque el sistema fue entrenado con un conjunto de datos diverso, su generalización a otros cultivos y regiones aun no ha sido evaluada.

Para superar estas limitaciones de condiciones adversas y explorar técnicas de transfer learning para adaptar el modelo a otros cultivos [8].

En conclusión, este trabajo contribuye a la promoción de prácticas agrícolas más sostenibles mediante el uso de tecnologías avanzadas de IA. La implementación de sistemas de detección temprana de enfermedades no solo mejora la productividad agrícola, sino que también reduce el impacto ambiental y los costos asociados con el uso de pesticidas. Futuras investigaciones deberían enfocarse en optimizar el modelo para su uso en dispositivos de bajo costo y en integrar el sistema con drones para la detección de enfermedades a gran escala [17].

En conclusión, este trabajo contribuye al avance de la agricultura de precisión mediante el uso de tecnologías avanzadas de inteligencia artificial. La implementación de sistemas de detección temprana de enfermedades no solo mejora la productividad agrícola, sino que también reduce los costos y los impactos negativos asociados con el uso de pesticidas, promoviendo prácticas agrícolas más sostenibles y responsables.

AGRADECIMIENTOS

Este trabajo de investigación me honra ser partícipe de la convocatoria del Primer Número de la Revista de académica Ciencias de la Ingeniería, así mismo agradecer a la Facultad de Ingeniería de la Universidad Autónoma de Campeche por la invitación y participación en el primer Congreso Latinoamericano de Enseñanza en la Hidráulica y Sostenibilidad.

Así como agradecer a la Universidad Autónoma de Sinaloa (UAS) y a la Facultad de Ingeniería y Tecnología de Mazatlán por brindar las instalaciones y recursos necesarios para llevar a cabo esta investigación.

De manera especial, queremos reconocer a los agricultores de la región de Sinaloa por su colaboración en la recolección de datos y por compartir su conocimiento sobre las enfermedades que afectan sus cultivos. Su participación fue fundamental para el desarrollo de este trabajo.

Finalmente, expresamos nuestro agradecimiento a los revisores y colegas que contribuyeron con sus valiosos comentarios y sugerencias para mejorar la calidad de este trabajo.

REFERENCIAS

[1] SAGARPA, "Anuario estadístico de la producción agrícola en Sinaloa," Secretaría de Agricultura, Ganadería, Desarrollo Rural, Pesca y Alimentación, 2021.

[2] FAO, "El estado mundial de la agricultura y la alimentación," Organización de las Naciones Unidas para la Alimentación y la Agricultura, 2020.

[3] J. Pretty et al., "Agricultural sustainability: Concepts, principles and evidence," Philosophical Transactions of the Royal Society B: Biological Sciences, vol. 363, no. 1491, pp. 447-465, 2008.

[4] P. Sharma et al., "Impact of pesticides on human health and environment: A review," Journal of Environmental Science and Health, Part C, vol. 38, no. 1, pp. 1-30, 2020.

[5] J. G. A. Barbedo, "A review on the main challenges in automatic plant disease identification based on visible range images," Biosystems Engineering, vol. 144, pp. 52-60, 2016.

[6] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2017. [En línea]. Disponible: https://arxiv.org/pdf/1412.6980v9.pdf

[7] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," Frontiers in Plant Science, vol. 7, p. 1419, 2016.

[8] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," Computers and Electronics in Agriculture, vol. 147, pp. 70-90, 2018.

[9] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," arXiv preprint arXiv:1804.02767, 2018.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 2017.

[11] Ultralytics, "YOLOv8 documentation," 2023. [En línea]. Disponible: https://docs.ultralytics.com/

[12] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," arXiv preprint arXiv:2004.10934, 2020.

[13] N. Srivastava et al., "Dropout: A simple way to prevent neural networks from overfitting," Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929-1958, 2014.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, 2016.

[15] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436-444, 2015.

[16] J. Amara, B. Bouaziz, and A. Algergawy, “A deep learning-based approach for banana leaf diseases classification,” International Journal of Advanced Computer Science and Applications, vol 11, no. 7, 2020.

[17] P. Jiang et al., "A deep learning approach to detecting agricultural diseases from images," IEEE Access, vol. 8, pp. 163517-163529, 2020.

[18] H. Durmuş, E. O. Güneş, and M. Kırcı, "Disease detection on the leaves of the tomato plants by using deep learning," Agronomy, vol. 11, no. 1, p. 23, 2021.

[19] M. A. Ferentinos, "Deep learning models for plant disease detection and diagnosis," Computers and Electronics in Agriculture, vol. 145, pp. 311-318, 2018.

[20] R. R. Kumar et al., "Plant disease detection using deep learning techniques: A review," Sensors, vol. 21, no. 11, p. 3839, 2021.

[21] C.-Y. Wang, H.-Y. M. Liao, and I.-H. Yeh, "YOLOv5: A new state-of-the-art for real-time object detection," arXiv preprint arXiv:2012.01703, 2020.

[22] Garlan and M. Shaw, "An Introduction to Software Architecture," Advances in Software Engineering and Knowledge Engineering, vol. 1, pp. 1-39, 1993.

L. Bass, P. Clements, and R. Kazman, Software Architecture in Practice, 3rd ed. Addison-Wesley, 2012.

[24] J. G. Arnal Barbedo, "Digital image processing techniques for detecting and quantifying plant diseases," Springer, 2020.

[25] A. Tharwat, "Classification assessment methods," Applied Computing and Informatics, vol. 17, no. 1, pp. 168-192, 2020.

[26] J. C. Montero Rodríguez, R. Roshan Biswal, and E. Sánchez de la Cruz, "Algoritmos de aprendizaje automático de vanguardia para el diagnóstico de enfermedades," Research in Computing Science Journal, vol. 148, no. 7, pp. 1-14, 2019. [En línea]. Disponible: https://www.rcs.cic.ipn.mx/2019_148_7/Algoritmos%20de%20aprendizaje%20automatico%20de%20vanguardia%20para%20el%20diagnostico%20de%20enfermedades.pdf