



Universidad Autónoma de Querétaro

Facultad de Informática

Desarrollo de un algoritmo basado en *machine learning* para
la identificación de plagas en hojas de frijol

Tesis

Que como parte de los requisitos
para obtener el Grado de

Maestro en Ciencias de la Computación

Presenta

IMT Diana Carmen Rodríguez Lira

Dirigido por:

Dra. Diana Margarita Córdova Esparza

Querétaro, Qro. a 08 de agosto de 2024

La presente obra está bajo la licencia:
<https://creativecommons.org/licenses/by-nc-nd/4.0/deed.es>



CC BY-NC-ND 4.0 DEED

Atribución-NoComercial-SinDerivadas 4.0 Internacional

Usted es libre de:

Compartir — copiar y redistribuir el material en cualquier medio o formato

La licenciante no puede revocar estas libertades en tanto usted siga los términos de la licencia

Bajo los siguientes términos:



Atribución — Usted debe dar [crédito de manera adecuada](#), brindar un enlace a la licencia, e [indicar si se han realizado cambios](#). Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciante.



NoComercial — Usted no puede hacer uso del material con [propósitos comerciales](#).



SinDerivadas — Si [remezcla, transforma o crea a partir](#) del material, no podrá distribuir el material modificado.

No hay restricciones adicionales — No puede aplicar términos legales ni [medidas tecnológicas](#) que restrinjan legalmente a otras a hacer cualquier uso permitido por la licencia.

Avisos:

No tiene que cumplir con la licencia para elementos del material en el dominio público o cuando su uso esté permitido por una [excepción o limitación](#) aplicable.

No se dan garantías. La licencia podría no darle todos los permisos que necesita para el uso que tenga previsto. Por ejemplo, otros derechos como [publicidad, privacidad, o derechos morales](#) pueden limitar la forma en que utilice el material.



Universidad Autónoma de Querétaro
Facultad de Informática
Maestría en Ciencias de la Computación

Desarrollo de un algoritmo basado en *machine learning* para la identificación
de plagas en hojas de frijol

Tesis

Que como parte de los requisitos para obtener el Grado
Maestro en Ciencias de la Computación

Presenta

IMT Diana Carmen Rodríguez Lira

Dirigido por:

Dra. Diana Carmen Rodríguez Lira

Dra. Diana Margarita Córdova Esparza
Presidente

Dr. José Manuel Álvarez Alvarado
Secretario

Dr. Julio Alejandro Romero González
Vocal

Dr. Juan Ramón Terven Salinas
Suplente

M. en C Daniel Cantón Enríquez
Suplente

Centro Universitario, Querétaro, Qro.
México
Agosto 2024

Dedicatorias

Al joven especial, que siempre esta para apoyarme, a mi amigos, a mi familia y al mundo.

Agradecimientos

A los Doctores: Diana Margarita Córdova Esparza, José Manuel Álvarez Alvarado, Juan Ramon Terven Salinas y Julio Alejandro Romero González.

Por guiarme, orientarme y darme las herramientas y aptitudes necesarias para completar este postgrado de manera satisfactoria.

A mis compañeros de estudio Lupita, Adrian, Erick, Felipe, Sergio y Andrea, por hacer de esto una etapa amena y placentera.

A mis docentes por hacer de palabras al aire, conceptos sólidos que se quedarán conmigo para siempre.

Al Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT) por fomentar el estudio y la ciencia en México.

Índice general

	Página
Dedicatorias	3
Agradecimientos	4
Índice de tablas	7
Índice de figuras	8
Resumen	9
Abstract	10
1. Introducción	11
Planteamiento del problema	13
Justificación	15
Plagas de frijol	16
Coleopteras	16
Conchuela y diabrótica	17
Problemáticas de los agricultores por enfermedades o plagas del frijol en México	19
2. Antecedentes	21
3. Fundamentos teóricos	38
Modelos preentrenados	38
YOLO	38
Segmentador SAM	39
Características principales de SAM	39
4. Hipótesis	41
Preguntas de investigación	41
Planteamiento de la Hipótesis	41
5. Objetivos	42
Objetivo General	42
Objetivos Específicos	42

6. Metodología	43
Etapa 1. Creación de un conjunto de datos:	45
Organizar y almacenar los datos de manera estructurada	47
Etapa 2. Procesamiento de la información:	48
Etapa 3. Arquitectura de la red neuronal artificial:	52
Etapa 4. Evaluación de rendimiento:	59
7. Resultados y Discusión	61
Resultados obtenidos	61
Discusión	70
Comparacion de trabajos similares y sus aportaciones a la literatura.	71
Acciones Potenciales.	73
8. Conclusiones	74
Bibliografía	76
A. codigo A	95
B. codigo B	96

Índice de tablas

Tabla	Página
1. Comparación entre técnicas de aprendizaje automático con base en el análisis sistemático.	21
2. Precisión de clasificadores utilizados en análisis sistemático de R. Kumar et al., 2022.	22
3. Revisión del estado de arte de técnicas de machine learning para la detección y/o clasificación de enfermedades en plantas del 2018 al 2020.	25
4. Revisión del estado de arte de técnicas de <i>machine learning</i> para la detección y/o clasificación de enfermedades en plantas de 2022 a la actualidad.	27
5. hiperparámetros estandarizados para la red.	54
6. Hiperparámetros de Aumento de datos.	57
7. Métricas de rendimiento de YOLO.	64
8. Comparación de características en diferentes trabajos.	72

Índice de figuras

Figura	Página
1. Hectáreas destinadas a cultivo por producto.	15
2. Daños en el cultivo de frijol común.	18
3. Esqueletización de hoja del frijol a causa del daño ocasionado por la <i>Epilachna varivestis</i> Mulsant.	18
4. Daños causados en el frijol por diabrotica balteata.	19
5. Plantas mencionadas en más de un artículo y su frecuencia.	35
6. Técnicas de aumento de datos.	36
7. Arquitectura de YOLO.	39
8. Metodología propuesta.	44
9. Ejemplo de imágenes descartadas.	46
10. Ejemplo de imágenes aceptadas.	46
11. Ejemplo de imágenes aceptadas.	47
12. Definición de las clases en la sección categorías del programa de etiquetas.	49
13. Segmentador.	50
14. Etiqueta de segmentado.	51
15. Ejemplos de imagen etiquetada.	52
16. Imágenes y su respectivo archivo de etiquetas.	53
17. Pseudo código del algoritmo.	56
18. Ejemplo de imágenes creadas por el modelo.	58
19. Curva de precisión contra <i>recall</i> para los 12 modelos de YOLO comparados bajo diferentes umbrales.	61
20. Curva de confianza contra F1 para 12 versiones de modelos YOLO.	62
21. Curva de confianza contra <i>recall</i> para los 12 modelos de YOLO.	62
22. Curva de confianza contra precisión para 12 versiones de modelos YOLO.	63
23. Predicción con YOLO5n.	65
24. Predicción con YOLO5s.	65
25. Predicción con YOLO5m.	66
26. Predicción con YOLO5l.	66
27. Predicción con YOLO5x.	67
28. Predicción con YOLO8n.	67
29. Predicción con YOLO8s.	68
30. Predicción con YOLO8m.	68
31. Predicción con YOLO8l.	69
32. Predicción con YOLO8x.	69
33. Predicción con YOLO9c.	70
34. Predicción con YOLO9e.	70

Resumen

La agricultura en México desempeña un papel crucial tanto a nivel económico como alimentario y social, contribuyendo significativamente a la economía global y la seguridad alimentaria. México, con su diversidad de climas y suelos, es un productor importante de una amplia variedad de cultivos, abasteciendo tanto al mercado interno como el global. Sin embargo, el mantenimiento de cultivos saludables es esencial para evitar daños por enfermedades y plagas, que pueden afectar la calidad y rendimiento de los productos. Las técnicas tradicionales de manejo agrícola son ineficientes y consumen muchos recursos, por lo que se buscan soluciones innovadoras.

El aprendizaje automático surge como una opción viable para detectar plagas y enfermedades en cultivos, aunque enfrenta desafíos en el procesamiento de imágenes debido a factores como sombras y cambios de iluminación. Este trabajo se enfoca en desarrollar un algoritmo basado en técnicas de procesamiento de imágenes y redes neuronales artificiales para identificar daños en las hojas del frijol que pueden ser causados por alguna plaga del orden Coleoptera: Coccinellidae y/o Chrysomelidae. Los resultados muestran una precisión de 0.792 para la clase "sana" y 0.658 para la clase "infectada". A pesar de que el algoritmo demuestra un desempeño satisfactorio, se identifican áreas de mejora como el manejo del desequilibrio de clases y la prevención del sobre ajuste.

Palabras clave: Aprendizaje automático, Procesamiento de imágenes, Redes neuronales artificiales, Identificación de plagas en cultivos

Abstract

Agriculture in Mexico plays a crucial role economically, nutritionally, and socially, significantly contributing to the global economy and food security. With its diverse climates and soils, Mexico produces a wide variety of crops, supplying both domestic and global markets. However, maintaining healthy crops is essential to prevent damage from diseases and pests, which can affect product quality and yield. Traditional agricultural techniques are inefficient and resource-intensive, prompting the search for innovative solutions.

Machine learning emerges as a viable option for detecting pests and diseases in crops, though it faces challenges in image processing due to factors such as shadows and lighting changes. This work focuses on developing a machine learning algorithm to identify damage in bean leaves caused by Coleoptera: Coccinellidae and/or Chrysomelidae, using image processing techniques and artificial neural networks. Results show accuracies of 0.792 for the "healthy" class and 0.658 for the "infected" class. Despite its overall good performance, areas for improvement include handling class imbalance and preventing overfitting.

Key words: Machine Learning, Image Processing, Artificial, Neural Networks, Pest Identification in Crops.

Capítulo 1

Introducción

La industria agrícola está constantemente bajo presión para satisfacer las demandas de una población en crecimiento, los avances tecnológicos recientes en este sector han permitido un aumento en la productividad de los cultivos. Sin también es necesario destacar que existe una degradación en el medio ambiente y se han registrado pérdidas significativas debido a plagas, patógenos y malezas (S. Sharma et al., 2017). Según la Organización de las Naciones Unidas para la Alimentación y la Agricultura (FAO por sus siglas en inglés), hasta el 40 % de la producción agrícola mundial se pierde cada año debido a las plagas que afectan a diversos cultivos (Boliko, 2019). Uno de los desafíos más significativos que enfrenta el sector agrícola es la detección temprana de plagas y enfermedades de las plantas. Las plagas y enfermedades pueden causar estragos en los cultivos, dado que muchos países dependen en gran medida de la productividad agrícola (Martin, 2019). Una de las tareas más desafiantes en la industria agrícola es identificar y clasificar correctamente las plagas y enfermedades de las plantas. El daño causado por insectos a los cultivos afecta significativamente el rendimiento agrícola, y clasificar este tipo de plaga es una tarea difícil debido a su estructura compleja y las complejas conexiones entre varias especies (Amiri et al., 2023; Rehman et al., 2022).

Históricamente, la detección manual de enfermedades se ha basado en la experiencia de los agricultores o en directrices. Cada enfermedad de las plantas progresa a través de diversas fases, y cuando un cultivo es afectado por una enfermedad o plaga, los agricultores necesitan mantenerse actualizados sobre esta condición (Fones et al., 2020). Por lo tanto, se ha vuelto esencial identificar y detectar enfermedades y plagas en cultivos y/o plantas, ya que en los campos agrícolas son

naturalmente susceptibles a diferentes enfermedades fúngicas o bacterianas. No predecir el problema a tiempo conlleva un desastre significativo, afectando tanto la cantidad como la calidad de la producción. Además, las técnicas de identificación de enfermedades son largas y requieren una cuidadosa selección de insecticidas (Tudi et al., 2021). A pesar de la aplicación de pesticidas, varias enfermedades y plagas son resistentes y pueden causar una pérdida parcial o total de la producción (D. Singh et al., 2020), además que el uso de pesticidas a menudo reduce la calidad de los alimentos.

Por estas razones, hay una búsqueda continua de soluciones que puedan mejorar y mitigar los problemas agrícolas, y el aprendizaje automático está emergiendo como una opción viable para detectar plagas y enfermedades en los cultivos. No obstante, cabe mencionar que el procesamiento de imágenes, que es un paso crucial para el funcionamiento adecuado de tales modelos inteligentes. La presencia de sombras, ruido o cambios en las condiciones de iluminación en las imágenes capturadas de cultivos puede causar errores en la predicción de enfermedades (Udeogu et al., 2023), entre otros problemas. Los métodos convencionales pueden resultar imprácticos debido a las diversas enfermedades que pueden presentarse en los cultivos. Por lo tanto, la detección de enfermedades a partir de imágenes se ha convertido en una de las principales actividades de investigación en la Informática y la Agricultura. En los últimos años, la identificación y categorización de enfermedades de las plantas utilizando imágenes digitales (Barman et al., 2023) ha despertado un interés significativo en la comunidad de investigación. El desarrollo de sistemas de diagnóstico asistidos por computadora para aplicaciones agrícolas que utilizan imágenes RGB ha sido un campo de estudio crucial y en rápida expansión. La influencia de los conjuntos de características para clasificar plantas utilizando métodos y reglas de aprendizaje

automático ha sido extensamente estudiada con fines agrícolas. La estimación de la precisión de las técnicas de clasificación basadas en aprendizaje automático ha demostrado un rendimiento efectivo en la identificación de enfermedades de las plantas utilizando distintas técnicas y modelos tales como Deep learning (Atila et al., 2021; Shah et al., 2023), Inception ResNet (Ahmed y Ahmed, 2023), Rat Swarm Optimization-Based Convolutional Neural Network (RSODL-PDDC) (Raja y Karthikeyan, 2023a), Residual deformable attention based transformer network (RDTNet) (A. Kumar et al., 2023), residual network and squeeze-and-excitation (SE-VRNet) (Xiao et al., 2023), Deeper lightweight multi-class classification model (DLMC-Net) (V. Sharma et al., 2023), CNN (Lanjewar y Panchbhai, 2023), todos mostrando una precisión superior a 0.995.

Planteamiento del problema

El papel que desempeña la agricultura en México es fundamental, ya que tiene importancia a nivel económico, alimentario y social. Dentro del panorama mundial, la agricultura es un pilar primordial de la seguridad alimentaria ya que es la principal fuente de alimentos para la población mundial y el ganado. Además de contribuir significativamente a la economía global, mediante la generación de empleos, impulso de desarrollo rural y sirviendo de motor de crecimiento económico para muchos países alrededor del mundo (Duan et al., 2024; Fuglie y Echeverria, 2024; Ogunjobi et al., s.f.). Para México, la agricultura es de suma importancia debido a que es uno de los países con mayor diversidad de climas y suelos, lo que permite la producción de una basta cantidad de cultivos. Gracias a su agricultura México, no se abastece únicamente a si mismo sino también al mercado global, mediante la exportación de frutas, hortalizas y granos.

Debido a esto, resulta imperativo el cuidado y mantenimiento de los cultivos

en un estado saludable, libre de enfermedades y plagas, ya que estas pueden causar daños significativos al afectar su crecimiento, calidad y rendimiento. Esto no solo salvaguarda la inversión económica puesta en el cultivo, sino que también contribuye a la seguridad alimentaria y salud pública. Sin embargo, las técnicas utilizadas actualmente por la comunidad agrícola resultan rústicas y consumen muchos recursos, ya sea por el tiempo que toma observar manualmente cada planta además de las problemáticas que pueden resultar de esto como ineficiencia, inconsistencias, y el error humano que puede resultar en retrasos en la intervención y costos adicionales.

Actualmente, el uso de la Inteligencia Artificial a través del aprendizaje automático está surgiendo como una opción viable para detectar plagas y enfermedades en los cultivos. Sin embargo, existen diferentes desafíos para realizar la detección y/o clasificación automática de plagas y enfermedades. La presencia de sombras, ruido o cambios en las condiciones de iluminación en las imágenes capturadas de los cultivos puede causar una clasificación incorrecta en la predicción de enfermedades.

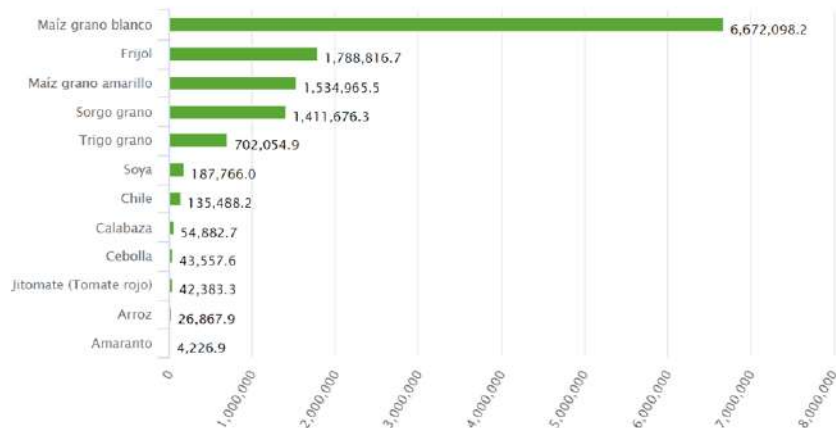
En la literatura se puede encontrar una amplia diversidad de trabajos para la identificación de enfermedades en las hojas de las plantas. No obstante, un gran número de ellos son entrenados con imágenes procedentes de un laboratorio y no se ajustan a las condiciones reales en campo cuando las imágenes son tomadas directamente en los cultivos. En este sentido, el estudio desarrollado en este trabajo de investigación se analizan imágenes complejas tomadas en campo abierto, cuyas características son diversas a diferencia de aquellas que fueron capturadas en un entorno controlado.

Justificación

La agricultura en México representa un porcentaje considerable de la economía del país, De acuerdo con la Secretaría de Agricultura, Ganadería, Desarrollo Rural, Pesca y Alimentación (SAGARPA, 2018), el sector agropecuario contribuye con el 10% al PIB. Dentro de este sector, la agricultura es la actividad económica más valiosa, representando el 70% del valor total del sector primario.. México es conocido por la gran biodiversidad que ofrece. Su agricultura sobresale especialmente por la producción de hortalizas, maíz, trigo, tomate y frijol. Este último, por ejemplo, según la Encuesta Nacional Agropecuaria 2019 realizada por el INEGI es el segundo cultivo con más superficie ocupada siendo esta 1,788,816.7 (un millón setecientos ochenta y ocho mil ochocientos dieciséis puntos siete) hectáreas (ver Figura 1).

Figura 1

Hectáreas destinadas a cultivo por producto.



Nota. Fuente: Encuesta Nacional Agropecuaria 2019.

Esta misma fuente de consulta también menciona que En los últimos 10 años, la superficie cultivada de frijol ha crecido un 12%, pasando de 1.51 a 1.69 millones

de hectáreas. Actualmente, México se posiciona como el noveno mayor exportador de frijol a nivel mundial.(SADER, 2023).

Por lo que el cuidado del cultivo de frijol es de suma importancia para los agricultores mexicanos, que al igual que otros cultivos se ve amenazado por enfermedades o plagas que pueden perjudicar su crecimiento y producción, estas afecciones se presentan en distintas partes de la planta, como lo son flores, frutos raíces y hojas (Buruchara et al., 2010), siendo el análisis de las hojas en el que se centra el presente proyecto de investigación, la detección automática de enfermedades en el frijol puede ayudar a proteger la salud de las plantas y garantizar la calidad de los cultivos, además de permitir a los agricultores ahorrar recursos al evitar la necesidad de utilizar equipos costosos o instrumentos especializados para este mismo fin.

Plagas de frijol

El frijol es una planta altamente susceptible a una variedad de plagas, las cuales pueden afectar gravemente su producción y calidad. Entre estas plagas, los coleópteros juegan un papel destacado debido a la capacidad que tienen para causar daños significativos tanto en el cultivo como durante el almacenamiento del grano. En los siguientes párrafos, se detallan las principales especies de coleópteros que afectan al frijol, sus características biológicas y los patrones de daño que provocan.

Coleopteras

La familia de las coleopteras incluye dos plagas graves, siendo estas el escarabajo de la calabaza (*Epilachna borealis* Fabricius) y el escarabajo mexicano del frijol (*Epilachna varivestis* Mulsant). El escarabajo del frijol a diferencia de la

mayoría de las Coccinellidae, que son carnívoros, se alimenta de plantas, lo que las hace especialmente riesgosos para los agricultores. Selfa y Aneto (1997) en su libro “Los Artrópodos y el Hombre” los menciona como los artrópodos de mayor importancia fitosanitaria. Hernández y Mariselys (2021), refieren que debido a la variedad de cultivos de los cuales se alimenta, la diabrotica balteata podría ser catalogada como especie polífaga y debido a esto es la de más importancia en México de su grupo ya que puede atacar a más de 30 especies de cultivo.

Las coleópteras se encuentran presentes en el continente americano; en México en los estados de Chiapas, Guanajuato, Jalisco, Michoacán, Nayarit, Querétaro, San Luis Potosí, Sinaloa, Tamaulipas y Veracruz. Estas plagas representan una amenaza significativa en los cultivos por lo que identificar las manifestaciones de la plaga de manera oportuna es muy importante. Por otra parte, dado que algunos indicios de la presencia de estas plagas son superficiales y observables en las hojas de las plantas existe una valiosa oportunidad para el análisis de imágenes y el *machine learning*.

Conchuela y diabrotica

Epilachna varivestis Mulsant, Coleóptera: Coccinellidae conocida comúnmente como “conchuela del frijol”, “escarabajo mexicano del frijol”, “maya del frijol”, “petaquita perforadora” (EPPO, 2023). Es considerada, según al NIMF No. 5 “Glosario de términos fitosanitarios”, plaga no cuarentenaria (NIMF, 2018). Se alimentan de la superficie de los tejidos de la hoja, dejando solo las nervaduras y parte de la epidermis (ver Figura 2); el tejido que queda rápidamente muere y se torna café como se muestra en la Figura 3. Provocando defoliación durante la floración y el formado y llenado de las vainas del frijol lo que provoca perdidas en el rendimiento del cultivo.

Figura 2

Daños en el cultivo de frijol común.



Nota. Fuente: Ficha técnica de *Epilachna varivestis* Mulsant CNRF, 2023,

Figura 3

Esqueletización de hoja del frijol a causa del daño ocasionado por la *Epilachna varivestis* Mulsant.



Fragmento de imagen obtenida de Ficha técnica de *Epilachna varivestis* Mulsant (CNRF, 2023).

Diabrotica balteata LeConte, del orden coleóptera, conocida comúnmente

como: “diabrotica”, “catarinita doradilla”, “gusano alfilerillo”, “mayatito con bandas verdes”, “tortuguilla” y “tortuguilla de franjas verdes del pepino” Es considerada, según al (NIMF, 2018) “Glosario de términos fitosanitarios”, plaga no cuarentenaria (CNRF, 2023).

Los adultos se alimentan de las hojas realizando perforaciones en forma ovalada o irregular en toda la lámina de la hoja (Figura 4). Cuando las plantas pequeñas son afectadas sus hojas primarias reducen el crecimiento lo que provoca su marchitamiento y posterior muerte. También se han encontrado daño en las flores y vainas cuando las poblaciones son altas.

Figura 4

Daños causados en el frijol por diabrotica balteata.



Fragmento de imagen obtenida de Ficha técnica de Diabrotica balteata (CNRF, 2023).

Problemáticas de los agricultores por enfermedades o plagas del frijol en México

El manejo de plagas y enfermedades en los cultivos ha sido una preocupación desde los inicios de la agricultura. Tradicionalmente, los agricultores han dependido de métodos manuales y de la experiencia visual para identificar y tratar estos problemas. Sin embargo, con el avance de la tecnología y la necesidad

de aumentar la eficiencia y precisión, se han desarrollado nuevas herramientas para abordar estos desafíos.

Según funcionarios del ámbito agrícola en México, se estima que cada año el país sufre pérdidas de entre el 20 % y el 40 % en la producción de alimentos debido a la presencia de insectos, plagas y enfermedades en los cultivos. Específicamente, los hongos pueden llegar a causar pérdidas que representan hasta el 30 % de la producción total en México.

La Organización de las Naciones Unidas para la Alimentación y la Agricultura (*Food and Agriculture Organization of the United Nations, 2024*) señala que en México hay aproximadamente 23 millones de hectáreas de tierras cultivables, lo que subraya la importancia de tomar medidas para reducir las pérdidas en la agricultura.

Capítulo 2

Antecedentes

En esta sección se proporciona una visión general del contexto histórico y teórico del uso de técnicas de *machine learning* en la detección y predicción de plagas y enfermedades en los cultivos. Se revisan estudios previos, se analizan las tendencias actuales y se identifican las principales brechas en la literatura existente.

R. Kumar et al. (2022) hace un análisis sobre diferentes artículos que utilizan *machine learning* para la clasificación de enfermedades en hojas de plantas; el 70 % se basó en imágenes en campo real y el 30 % restante en imágenes obtenidas en condiciones de laboratorio, con una precisión del 92.2%. En caso del uso de *deep learning*; se utilizó el 25 % de imágenes de hojas en campo real, el 55 % con imágenes de hojas en condiciones de laboratorio y el 20 % restante fueron imágenes de libre acceso, obteniendo una precisión del 98.8%. A continuación, en la Tabla 1 se muestra la información descrita.

Tabla 1

Comparación entre técnicas de aprendizaje automático con base en el análisis sistemático.

Tipo de análisis	de Imágenes de hojas en campo real	Imágenes en de hojas en condiciones de laboratorio	Base de datos de libre acceso	Precisión
Machine learning	70 %	30 %	NA %	92.2 %
Deep learning	25 %	55 %	20 %	98.8 %

En cuanto a los clasificadores más utilizados se encuentra la máquina

de soporte vectorial en la cual se obtuvo una precisión del 95.71 %, las redes neuronales artificiales por su parte con un 95 %, el uso de algoritmos genéticos presentó un 86 % de precisión y el clasificador *K-means* (agrupamiento de k-medias) obtuvo una precisión del 92.15 %. De estos datos se pudo identificar que el uso de *SVM* (máquina de vector de soporte) es quien presenta un mejor desempeño al utilizarse como clasificador para la detección de enfermedades en hojas seguido de las redes neuronales artificiales con una precisión similar (ver Tabla 2).

Tabla 2

Precisión de clasificadores utilizados en análisis sistemático de R. Kumar et al., 2022.

Clasificador	Precisión observada
SVM	95.71 %
ANN	95 %
Algoritmos genéticos	86 %
K-means	92.15 %

En el caso de los estudios que hacen uso de *machine learning* se observa que aproximadamente el 48 % se basan en una red neuronal o una variación de esta, el 32 % de ellos utilizaron *deep learning* y un 16 % uso *SVM*.

En los estudios recopilados por los autores Bhimte y Thool (2018), se evaluaron diferentes clasificadores y técnicas de aprendizaje profundo para la detección de enfermedades en diversos cultivos. En este trabajo se logró una precisión del 98.46 % al utilizar el clasificador *SVM* en el análisis de enfermedades en plantas de algodón. En el mismo año, Hlaing y Zaw reportaron una precisión del 85.1 % en la detección de enfermedades en plantas de tomate mediante el mismo clasificador *SVM*. Kaur et al. (2018) clasificaron enfermedades en plantas de soya, alcanzando una precisión del 90 % también con *SVM*. Atole y Park (2018)

utilizaron el clasificador *DCNN* y obtuvieron una precisión del 91.23 % en el análisis de enfermedades en plantas de arroz. Además, Rangarajan et al. (2018) aplicaron técnicas de aprendizaje profundo, como *ALexNet* y *VGG16*, para la identificación de enfermedades en plantas de tomate, recuperando precisiones del 97.29 % y 97.49 %, respectivamente.

En 2019, se llevaron a cabo estudios sobre diferentes cultivos utilizando técnicas de aprendizaje profundo. Los autores de Ramcharan et al. obtuvieron una precisión del 80.6 % en la detección de enfermedades en plantas de yuca mediante el uso de técnicas de aprendizaje profundo. Adedoja et al. (2019) lograron una precisión del 93.8 % al aplicar técnicas de aprendizaje profundo en el análisis de enfermedades en varios cultivos. Türkoğlu y Hanbay (2019) utilizaron *ALexNet* y *VGG16* para analizar enfermedades en múltiples cultivos, alcanzando precisiones mayores al 95 % respectivamente. En el caso de la remolacha azucarera, (Ozguven y Adem, 2019) obtuvieron una precisión del 95.48 % mediante el uso de técnicas de aprendizaje profundo. Hu et al. (2019) aplicaron el clasificador *DCNN* en el análisis de enfermedades en plantas de té, con una precisión del 92.5 %. U. P. Singh et al. (2019) utilizaron *MCNN* en el análisis de enfermedades en plantas de mango, alcanzando una precisión del 97.13 %. Elhassouny y Smarandache (2019) aplicaron *CNN* en el análisis de enfermedades en plantas de tomate, obteniendo una precisión del 90.3 % En 2020, Arora, Agrawal et al. evaluaron enfermedades en plantas de maíz utilizando el clasificador deep forest, logrando una precisión del 96.25 %. En el mismo año, S. H. Lee et al. utilizaron diferentes técnicas de aprendizaje profundo en el análisis de enfermedades en múltiples cultivos. Utilizando *GoogLeNetBN*, obtuvieron una precisión del 99.09 %. Con *VGG16*, alcanzaron una precisión del 99 %. Aplicando Inception V3, lograron una precisión del 99.31 %. Asimismo, con *GoogLeNet*; la cual de acuerdo con Thenmozhi y Reddy (2019) es una red neuronal

convolucional con 22 capas de profundidad, 2 capas de convolución, 4 capas de agrupación máxima, 9 módulos iniciales linealmente apilados y una agrupación promedio que se aplica al final del último módulo inicial. Puede cargar una versión preentrenada de la red entrenada con los conjuntos de datos de *ImageNet* o *Places365*. La red entrenada con ImageNet puede clasificar imágenes en 1000 categorías de objetos, obtuvieron una precisión del 99.35 %. Zeng y Li (Zeng y Li, 2020) utilizaron *SACNN (AESCD9214)* y *MK-D2* en el análisis de enfermedades en múltiples cultivos, obteniendo precisiones del 95.33 % y 98 %, respectivamente. J. Chen et al. (2020) aplicaron técnicas de aprendizaje profundo y lograron una precisión del 92 % para la clasificación de enfermedades en diversos cultivos. Li y Yang (2020) evaluaron enfermedades en plantas de algodón utilizando CNN, obteniendo una precisión del 95.4 %. Sethy et al. (2020) hicieron uso de una *SVM* en el análisis de enfermedades en plantas de arroz, logrando una precisión del 98.38 %. Li et al. (2020) aplicaron una combinación de *shallow CNN* y técnicas de aprendizaje automático para el estudio de enfermedades en varios cultivos, obteniendo una precisión del 94 %. Mientras que I. Ahmad et al. (2020) evaluaron enfermedades en plantas de tomate utilizando *CNN*, logrando una precisión del 93.40 % en condiciones de laboratorio y 85 % en condiciones reales.

Oyewola et al. (2021) obtuvieron una precisión del 96.75 % al utilizar *CNN* en el análisis de enfermedades en plantas de yuca. Atila et al. (2021) aplicaron técnicas de aprendizaje profundo en el análisis de enfermedades en múltiples cultivos, logrando una precisión del 99.91 %. Tuncer (2021) utilizaron *CNN* para la clasificación de enfermedades en diferentes cultivos, alcanzando una precisión del 99 %. Bi et al. (2022) evaluaron enfermedades en plantas de manzana utilizando *MobileNet*, obteniendo una precisión del 73.5 %.

Estos estudios demuestran el amplio espectro de clasificadores y técnicas

de aprendizaje profundo utilizado en la detección de enfermedades en diversos cultivos, así como las altas precisiones obtenidas, lo que destaca la efectividad de estas herramientas en el monitoreo fitosanitario y el manejo de enfermedades en la agricultura. Se realizó la Tabla 3 para mostrar de manera sintetizada la información anteriormente descrita.

Tabla 3

Revisión del estado de arte de técnicas de machine learning para la detección y/o clasificación de enfermedades en plantas del 2018 al 2020.

Autor	Planta analizada	Clasificador	Precisión (%)
Bhimte y Thool, 2018	algodón	SVM	98.46
Hlaing y Zaw, 2018	tomate	SVM	85.10
Kaur et al., 2018	soya	SVM	90.00
Atole y Park, 2018	arroz	DCNN	91.23
Rangarajan et al., 2018	tomate	ALexNet, VGG16	97.29, 97.49
Ramcharan et al., 2023	yuca	deep learning	80.60
Adedoja et al., 2019	múltiples	deep learning	93.80
Türkoğlu y Hanbay, 2019	múltiples	ALexNet, VGG16	95.00
Ozguven y Adem, 2019	remolacha azucarera	deep learning	95.48
Hu et al., 2019	te	DCNN	92.50

U. P. Singh et al., 2019	mango	MCNN	97.13
Elhassouny y Smarandache, 2019	tomate	CNN	90.30
Arora, Agrawal et al., 2020	maíz	deep forest	96.25
S. H. Lee et al., 2020	múltiples	GoogLeNetBN, VGG16, Inception V3, GoogLeNet	99.09, 99.00, 99.31, 99.35
Zeng y Li, 2020	múltiples	SACNN, AES-CD9214, MK-D2	95.33, 98.00
J. Chen et al., 2020	múltiples	deep learning	92.00
Li y Yang, 2020	algodón	CNN	95.40
Sethy et al., 2020	arroz	SVM	98.38
Li et al., 2020	múltiples	shallow CNN & ML	94.00
I. Ahmad et al., 2020	tomate	CNN	85.00
Oyewola et al., 2021	yuca	CNN	96.75
Atila et al., 2021	múltiples	deep learning	99.91
Tuncer, 2021	múltiples	CNN	99.00

DCNN; Deep convolutional neural network, SVM; Support vector machine, SACNN, Self-Attention convolutional neural network, ML; Machine learning, KNN; K-Nearest Neighbor.

Para los siguientes años 2022-2023, se realizó la Tabla 4.

Tabla 4

Revisión del estado de arte de técnicas de *machine learning* para la detección y/o clasificación de enfermedades en plantas de 2022 a la actualidad.

Autor	Planta analizada	Clasificador	Precisión (%)
Bi et al., 2022	manzana	MobileNet	73.50
C. Zhang et al., 2023	Garbanzo	FAIMS	94.00
Soares et al., 2022	Café	Imágenes multiespectrales	80.00
Chug et al., 2023	Tomate	EfficientNet y técnicas de ML	87.55
Peng et al., 2022	Arroz	WeedDet basado en RetinaNet	94.00
Marco-Detchart et al., 2023	Varios (Tomate, Uva, Fresa, Manzana)	Edge-AI	-
Pan et al., 2023	Arroz	RiceNet basado en YOLOX	95.85
Kalyani et al., 2023	EfficientNet transferencia aprendizaje	con Yuca de	89.00
Bayram et al., 2022	Tomate	CNN (AlexNet, GoogLeNet, ShuffleNet, EfficientNetB0, ResNet50, InceptionV3)	99.50

Gehlot y Gandhi, 2023	Varios	EfNet-TS basado en EfficientNetV2	99.00
Hasan et al., 2022	Manzana	Machine learning y visión por computadora	98.63
Kumar K y E, 2022	Arroz	VGG-16 y Faster R-CNN	98.80
Khan et al., 2022	Manzana	Deep learning	88.00
I. Ahmad et al., 2022	Plagas de insectos	YOLOv5x	98.30
Vishnoi et al., 2023	Manzana	CNN	98.00
Lanjewar y Panchbhai, 2023	Té	CNN en PaaS cloud	100.00
Dhande y Malik, 2023	Cultivos generales	Red convolucional profunda	70.00
Rajpoot et al., 2023	Arroz	VGG-16 y Faster R-CNN	97.30
Yong et al., 2023	Palma de aceite	VGG16 y Mask RCNN	91.93
Ma et al., 2023	Cultivos generales	Redes neuronales	90.15
Altalak et al., 2022	Tomate	CNN y SVM	97.20
Shoaib et al., 2022	Pimiento, Tomate, Papa	CNN 3D basado en CANet	92.00
Orchi et al., 2023	Varios (Manzana, Pimiento, Cereza, Maíz, Uva, Durazno, Papa, Fresa)	InceptionV3	98.01

Rahaman et al., 2023	Mango	DenseNet169	97.81
Shahoveisi et al., 2023	Varios	EfficientNetB4	94.29
Anim-Ayeko et al., 2023	Papa y tomate	ResNet-9	99.25
Aabidi, EL Makrani et al., 2023	Varios (Manzana, Pimiento, Cereza, Naranja, Papa, Calabaza, Fresa, Tomate)	CNN	99.00
Mitra et al., 2023	Tomate	Redes neuronales profundas	89.00
Ayaz et al., 2023	Cordia dichotoma	YOLOv4	95.00
Abuhayi y Mossa, 2023	Café	GoogLeNet y RESNET	99.08
P. D. Kumar et al., 2023	Varios	Redes neuronales	98.00
D. Zhang et al., 2023	Tomate	ResNet-50	95.03
Vishnoi et al., 2023	Manzana	CNN	98.00
Kalpana et al., 2023	Black gram	CNN	92.50
V. Sharma et al., 2023	Varios (Cítricos, Pepino, Uva, Tomate)	DLMC-Net	93.56, 92.34, 99.50, 96.56

Rajeena P. P et al., 2023	Maíz	EfficientNet	98.85
Shedthi B et al., 2023	Varios (Arroz, Pimiento, Uva)	Algoritmo híbrido de clustering	97.90
Kannagi et al., 2023	General	CNN	95.00
M.-G. Lee et al., 2023	Pino	DeepLab V3+ (ResNet50)	-
Milke et al., 2023	Café	CNN	98.10
Wu et al., 2023	Fresa	Imagen hiperespectral	93.33
Xiao et al., 2023	Varios (Tomate, Uva, Maíz, Durazno, Pimiento, Fresa, Manzana, Arándano, Calabaza)	SE-VRNet	99.73
Bensaadi y Louchene, 2023	Tomate	CNN	97.04
P. Kumar et al., 2023	Manzana	Deep learning basado en DCDM	98.78
A. Kumar et al., 2023	Arroz	RDTNet	99.53
Shrivastava et al., 2023	Papa	Deep learning optimizado	99.22
Ahmed y Ahmed, 2023	Palma	ResNet e Inception ResNet	99.62
Raja y Karthikeyan, 2023a	Uva y manzana	RSODL-PDDC	99.96

Khalid et al., 2023	Planta del dinero	Redes neuronales convolucionales profundas	93.00
Jana et al., 2023	General	DnCNN	97.73
Padamata y Atluri, 2023	Tomate	Algoritmo de segmentación semántica	97.29
Kalbande y Patil, 2023	Tomate	CNN	96.46
A. Alharbi et al., 2023	Trigo	Few-shot learning (EfficientNet)	93.19
Xue et al., 2023	Té	YOLO-Tea (YOLOv5)	-
Aabidi, Makrani et al., 2023	Manzana	Bag of Words model	70.08
Shah et al., 2023	Arroz	Deep learning y transfer learning	99.75
W. Ahmad et al., 2023	Tomate	LTriTP	94.50
Towfek y Khodadadi, 2023	Varios (Manzana, Arándano, Cereza, Uva, Durazno, Papa, Soya, Tomate)	Deep CNN	96.46
Esomonu et al., 2023	Tomate	CNN y SVM	88.00
Odukoya et al., 2023	Algodón	SVM	99.05
Bansal et al., 2023	Varios (14 especies)	CNN y LSTM	95.11

Rao et al., 2023	Pino	Balance Mixup y PWDNet	90.00
Mulla et al., 2023	Plantas verdes pequeñas	CNST	96.90
Sahu y Pandey, 2023	Varios (Manzana, Arándano, Cereza, Maíz, Uva, Naranja, Durazno, Pimiento, Fresa, Papa, Frambuesa, Calabaza)	HXTL-COKELM	98.90
Mishra et al., 2023	Varios (Manzana, Cereza, Maíz, Uva, Papa, Pimiento, Tomate)	I-LDD	93.22
Raja y Karthikeyan, 2023b	Uva y manzana	RDODL-APDC	-
M. Alharbi et al., 2023	Plantas medicinales	ConvLSTM U-Net	85.39
Baljon, 2023	Pimiento	GoogleNet	99.50
Önler, 2023	Frijol	Redes neuronales artificiales	98.33
Thai et al., 2023	Yuca	LeIAP	-
Kethineni y Pradeepini, 2023	General	GA-SVM	91.30
Jain y Dharavath, 2023	Maíz, arroz, uva	MSSOA	93.60, 79.10, 95.00
B. Zhang et al., 2023	Goji	SVM, KNN	90.70, 84.00

Khoramshahi et al., 2023	Cebada	Machine learning	60.50
Xie et al., 2023	Varios (Manzana, Maíz, Uva, Café)	SSAFS	-
Hipsch et al., 2023	Papa	Imagen redox	-
Lin et al., 2023	Fresa	StrawberryTalk	96.88
Wei et al., 2023	Olmo	Machine Learning e imagen hiperespectral	-
Vélez et al., 2023	Vid	Machine learning y UAV	99.00
Casas et al., 2023	Palma	UAV y machine learning	87.00
Meenakshi, 2023	Varias (Betel, Albahaca, Cítricos, Hibisco, Mango, Pimiento)	Logistic regression	83.75
Sawyer et al., 2023	Vid	Machine learning e imagen hiperespectral	87.00

DCNN; Deep convolutional neural network, SVM; Support vector machine, SACNN, Self-Attention convolutional neural network, ML; Machine learning, KNN; K-Nearest Neighbor.

El diagnóstico de enfermedades en plantas puede presentar diversos factores y problemas que pueden influir en su identificación como lo son:

- Problemas con las características de la enfermedad y el tipo de clasificador

utilizado.

- La mayoría de los estudios usan imágenes en condiciones de laboratorio controladas como base para el análisis.
- La eliminación del fondo y la segmentación precisa del área de la imagen donde se localiza la enfermedad de la hoja.
- Las imágenes en campo real suelen presentar fondos complejos afectando el modelo de detección.
- Se dificulta discernir si la planta presenta enfermedad o deficiencias de minerales o nutrientes.
- Coexistencia de múltiples enfermedades o plagas en una misma hoja, como lo pueden ser las coleópteras o diabrotica, siendo esta última mencionada por Edwards et al. (2013) como “una de las plagas más dañinas del maíz en el mundo.”

Además de estas tablas, se generaron los siguientes gráficos para resumir aspectos relevantes encontrados en la literatura.

El tomate la planta con más incidencia en los artículos analizados, en segundo lugar se encuentra la uva seguida del arroz, la manzana y la papa. Otros cultivos que también se ubicaron fueron maíz, fresa, durazno, café, cereza, pimientos entre otros. En la Figura (5) se encuentra esta información de manera desglosada. Se generó la Figura 6, con el fin de resumir las técnicas de aumento de datos más frecuentemente mencionadas en este estudio.

Figura 5

Plantas mencionadas en más de un artículo y su frecuencia.

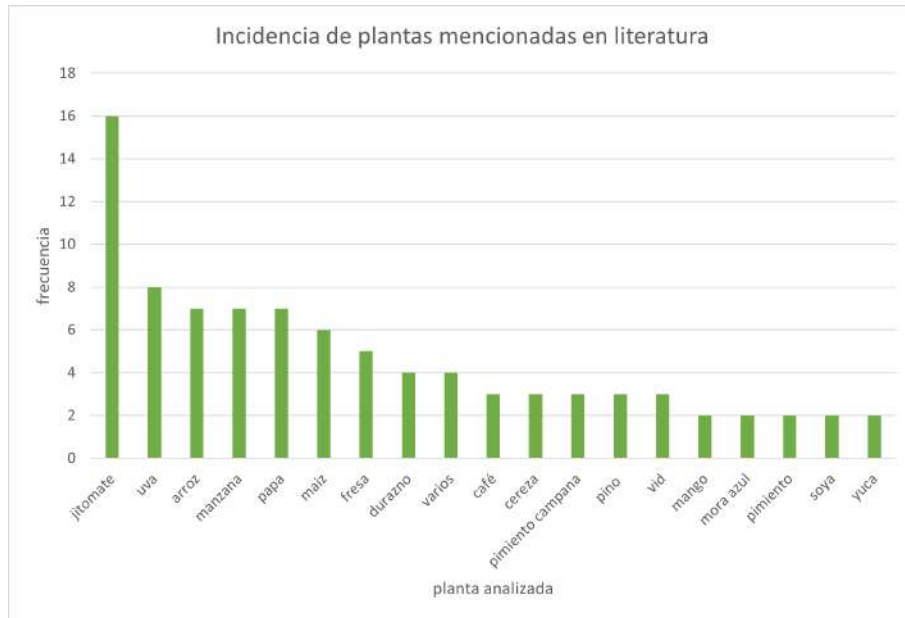
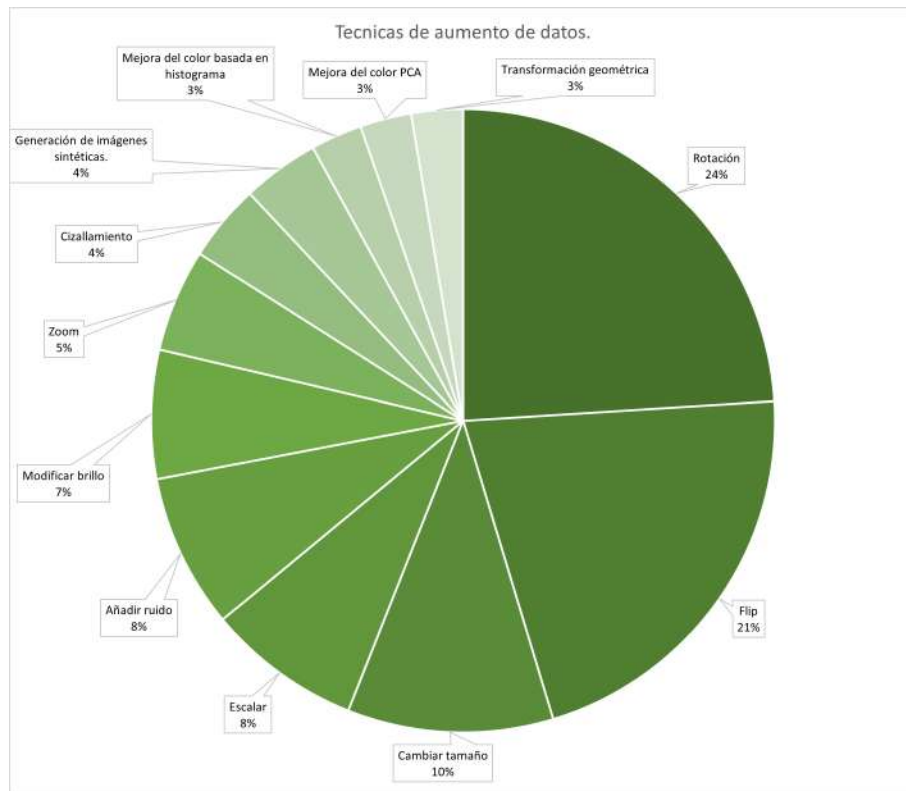


Figura 6

Técnicas de aumento de datos.



A pesar de los avances, existen aún áreas que no han sido suficientemente exploradas. Por ejemplo, la integración de datos de diferentes fuentes (como imágenes de cultivos en campo, datos del suelo, imágenes sin tratamientos de color etc) para mejorar la precisión de los modelos predictivos en campo real es un área en desarrollo; por lo que, la implementación práctica de estos modelos en entornos agrícolas reales aún enfrenta desafíos significativos.

En resumen, en el estado del arte se destaca la evolución y el estado actual del uso de *machine learning* en la detección y predicción de plagas y enfermedades en cultivos. Este análisis revela importantes oportunidades de investigación, especialmente en la implementación en entornos reales. La presente investigación pretende contribuir significativamente a este campo, abordando

las brechas identificadas y proponiendo soluciones prácticas para la agricultura moderna.

Capítulo 3

Fundamentos teóricos

Modelos preentrenados

Un modelo preentrenado es un modelo de aprendizaje automático que ha sido entrenado con un gran conjunto de datos y puede ajustarse para una tarea específica. Los modelos preentrenados a menudo se utilizan como punto de partida para desarrollar modelos de *ML*, ya que proporcionan un conjunto inicial de pesos y sesgos que se pueden ajustar para una tarea particular.

Entre sus ventajas se encuentran: aprovechamiento de conocimiento y experiencias anteriores, ahorro de tiempos y recursos, mejora de rendimiento del modelo, entre otros. Lo que proporciona una base sólida para ajuste y mejora significativa del rendimiento de modelos futuros.

Las *CNN* son frecuentemente utilizadas como base para la creación de modelos de clasificación de imágenes, ya que están entrenados para categorizar imágenes predeterminadamente (Marcelino, 2018).

YOLO

You Only Look Once (solo miras una vez, por sus siglas en inglés) es un sistema de detección de objetos en tiempo real de última generación que utiliza redes neuronales para identificar objetos en imágenes.

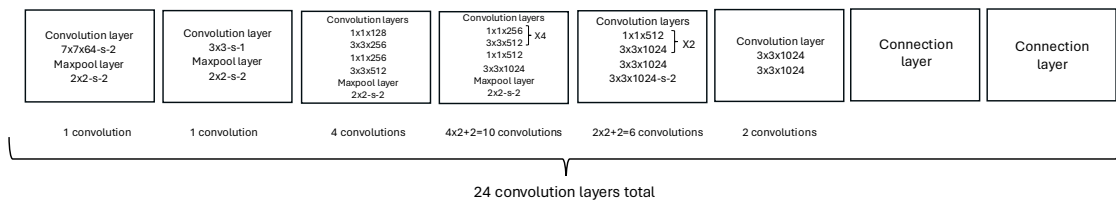
Los sistemas de detección anteriores reutilizaban clasificadores o localizadores para realizar la detección, aplicando el modelo a una imagen en múltiples ubicaciones y escalas. Las regiones de la imagen con altas puntuaciones se consideraban detecciones.

YOLO adopta un enfoque diferente, aplicando una única red neuronal a

la imagen completa. La red divide la imagen en regiones y predice cuadros delimitadores y probabilidades para cada región. Estos cuadros delimitadores se ponderan según las probabilidades predichas (Rahman et al., 2019).

Su arquitectura esta compuesta de 24 capas convolucionales las cuales combinan convoluciones 3x3 con 1x1 para la reducción de canales. A la salida se encuentra una capa completamente conectada que genera una cuadrícula de 7x7 con 30 valores para cada celda de la cuadrícula para acomodar diez coordenadas del cuadro delimitador (2 cuadros) con 20 categorías (Terven et al., 2023), para mejor entendimiento ver Figura 7.

Figura 7
Arquitectura de YOLO.



Fuente: autoría propia.

Segmentador SAM

El modelo segmenta cualquier cosa (*SAM*, por sus siglas en inglés) es un modelo de segmentación de imágenes (He et al., 2023).

Características principales de SAM

- Tarea de segmentación con indicación: *SAM* está diseñado para una tarea de segmentación con indicaciones, permitiendo generar máscaras de segmentación válidas a partir de cualquier tipo de indicación, como pistas

espaciales o textuales que identifiquen un objeto.

- **Arquitectura avanzada:** El modelo *SAM* utiliza un potente codificador de imágenes, un codificador de indicaciones y un decodificador de máscaras ligero.
- **Conjunto de datos SA-1B:** Introducido por *SAM*, incluye más de 1.000 millones de máscaras en 11 millones de imágenes. Al ser el conjunto de datos de segmentación más grande hasta la fecha, proporciona a *SAM* una fuente de datos de entrenamiento amplia y diversa.
- **Rendimiento de disparo cero:** *SAM* demuestra un rendimiento de disparo cero sobresaliente en diversas tareas de segmentación.

Capítulo 4

Hipótesis

Preguntas de investigación

- ¿Es posible crear una herramienta inteligente que determine si la hoja del frijol presenta algún tipo de enfermedad o plaga mediante el análisis de características presentadas en las hojas?
- ¿Cuáles son las técnicas más efectivas de segmentación de imágenes para identificar y aislar las regiones de interés en las hojas de las plantas de frijol afectadas por plagas o enfermedades?
- ¿Cómo se puede diseñar y desarrollar un descriptor con las características suficientes para la clasificación de las enfermedades presentes en las hojas del frijol?

Planteamiento de la Hipótesis

Se plantea que al desarrollar un algoritmo basado en redes neuronales con técnicas de segmentación de imágenes y un descriptor de características identifique de manera automática el daño causado en la hoja por la presencia de Coleóptera: Coccinellidae y/o Chrysomelidae.

Capítulo 5

Objetivos

Objetivo General

Desarrollar un algoritmo basado en *machine learning* que permita identificar los daños en la hoja del frijol causado por la presencia de Coleóptera: Coccinellidae y/o Chrysomelidae mediante técnicas de procesamiento de imágenes y redes neuronales artificiales.

Objetivos Específicos

1. Crear un conjunto de datos a partir de muestras con el propósito de entrenar el algoritmo de *machine learning*.
2. Procesar la información a través de un descriptor que permita caracterizar las hojas de las clases sanas/enfermas.
3. Diseñar la red neuronal artificial para identificar si las hojas presentan ocasionado por las plagas.
4. Evaluar el rendimiento con diferentes técnicas de identificación para validar el algoritmo.

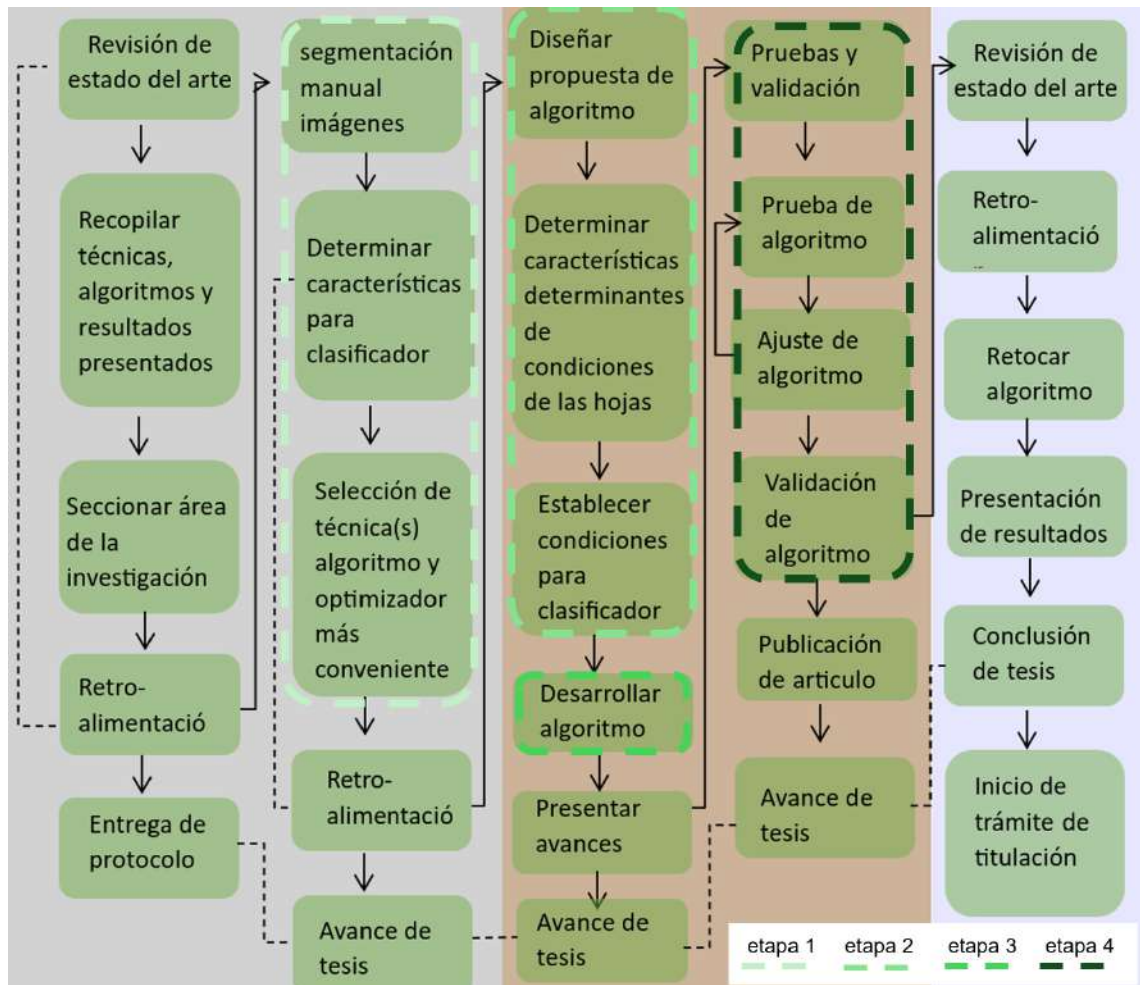
Capítulo 6

Metodología

En el presente estudio, la metodología se fundamenta en el procesamiento de imágenes y el uso de técnicas de aprendizaje automático. La Figura 8, muestra los pasos esenciales de dicha metodología, desde la adquisición de las imágenes hasta su análisis detallado, lo que proporciona una visión general del enfoque de la investigación.

Figura 8

Metodología propuesta.



Nota. De la metodología general del proyecto de investigación se pueden observar resaltadas las cuatro etapas principales las cuales serán descritas a profundidad en párrafos posteriores. Fuente: Autoría propia.

La metodología que se llevo a cabo para el desarrollo del proyecto consistió en las etapas que se describen a continuación:

Etapas 1. Creación de un conjunto de datos:

- Recolección de muestras de hojas de frijol sanas y afectadas por las plagas de coleóptera mencionadas. Para la creación del conjunto de datos se tomó una muestra de imágenes que fueron directamente tomadas de surcos de frijol Pérez et al., 2022. La muestra consta de 531 imágenes RGB de 1920 x 1080 px , con un tamaño de 519 KB (532,381 bytes) en formato JPEG. Cabe mencionar que las imágenes fueron capturadas en campos de cultivo de frijol situados en regiones del estado de Zacatecas, México.
- La cámara que se utilizó para la captura de las imágenes es de la marca Logitech C920. Las imágenes se capturaron en condiciones de luz natural, durante el medio día para asegurar una iluminación uniforme.
- Las fotografías se tomaron a una distancia de aproximadamente 170 cm del suelo, con un ángulo de 90°.
- De las imágenes obtenidas en los cultivos de frijol, capturadas por el robot agrícola Pérez et al., 2022, se eliminaron aquellas que tuvieran desenfoques y malezas.

En la Figura 9 se muestran ejemplos de las imágenes que quedaron fuera del conjunto final al no cumplir alguna de las características planteadas. Por el contrario en las Figuras 10 y 11 se muestran ejemplos de las imágenes que fueron seleccionadas para formar parte del conjunto final.

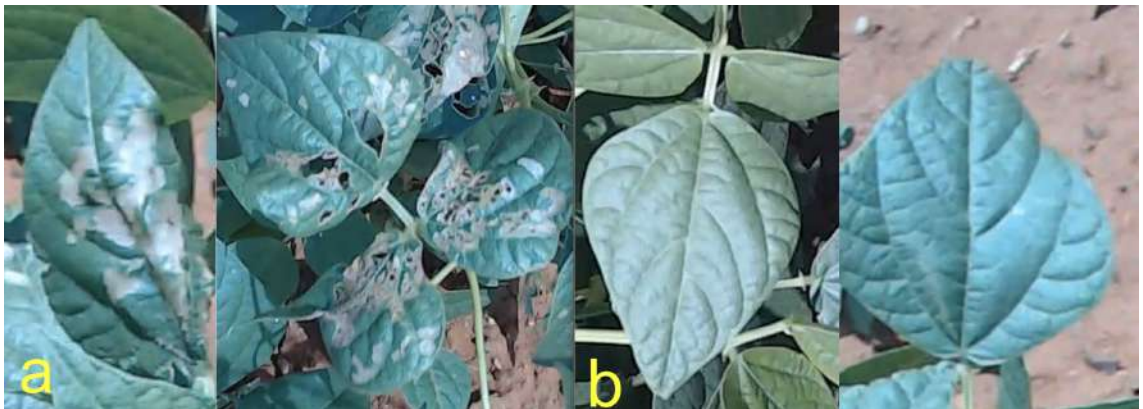
Figura 9

Ejemplo de imágenes descartadas.



Figura 10

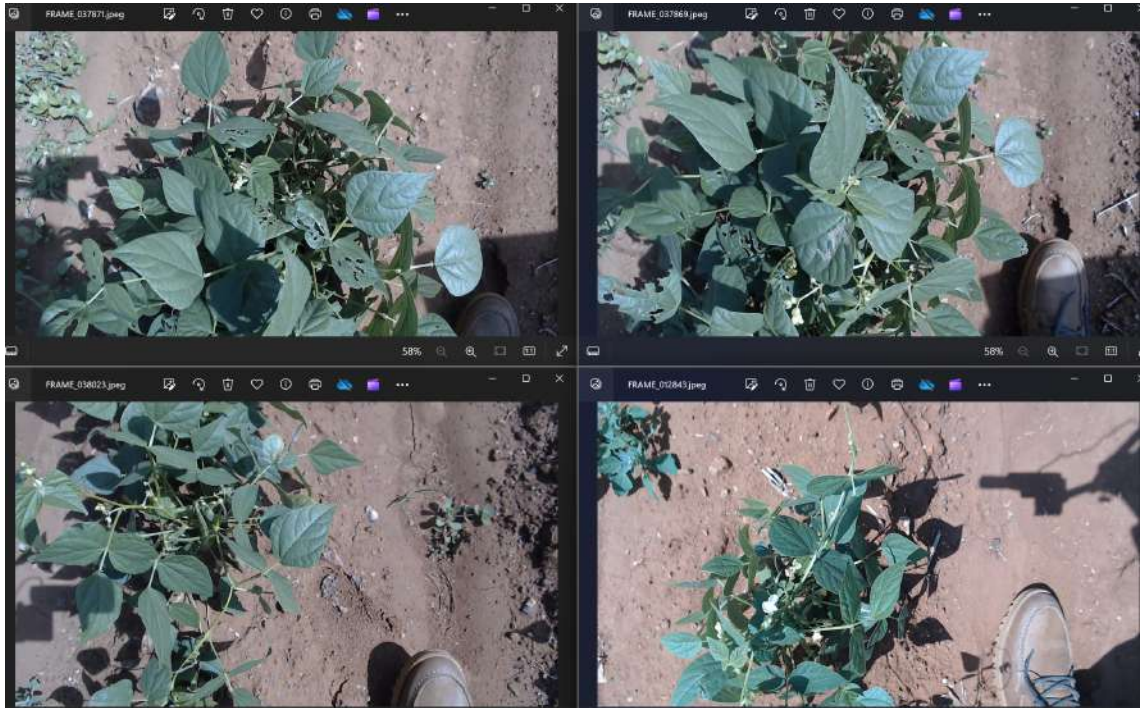
Ejemplo de imágenes aceptadas.



Nota. Las imágenes fueron aceptadas ya que aparece una planta de frijol, la imagen es nítida, y se encuentran múltiples hojas. a) ejemplos de hoja infectada aceptada, b) ejemplos de hoja sana aceptada.

Figura 11

Ejemplo de imágenes aceptadas.



Después de realizar este proceso, el conjunto final quedó de 508 imágenes.

Organizar y almacenar los datos de manera estructurada

Con el fin de entrenar y evaluar los modelos de manera efectiva, las 508 imágenes se dividieron en tres conjuntos distintos: entrenamiento (*train*), prueba (*test*), y validación (*validation*).

- Conjunto de Entrenamiento (*Train*): 308 imágenes.
- Conjunto de Prueba (*Test*): 100 imágenes.
- Conjunto de Validación (*Validation*): 100 imágenes.

La separación de las imágenes en estos tres conjuntos se realizó con el

objetivo de procesar los modelos de aprendizaje automático, asegurando así los siguientes aspectos:

- Proporción apropiada para el entrenamiento: Se asignaron 308 imágenes (aproximadamente el 60% del total) al conjunto de entrenamiento. Esta proporción asegura que el modelo tenga suficiente cantidad de datos para aprender las características de las hojas de frijol.
- Evaluación y ajuste del Modelo: Se reservaron 100 imágenes (cerca del 20% del total) para el conjunto de validación. Este conjunto se utilizó para ajustar los hiperparámetros del modelo y prevenir el sobre ajuste, asegurando que el modelo generalice bien a datos no vistos.
- Medición de la precisión: Finalmente, se destinaron 100 imágenes (aproximadamente el 20% del total) al conjunto de prueba. Este conjunto se utilizó exclusivamente para evaluar el rendimiento final del modelo. La elección de 100 imágenes facilitó el cálculo de métricas de precisión basadas en un porcentaje del total de imágenes, permitiendo una interpretación directa de los resultados en una escala de 0 a 100. Con el fin de garantizar una aleatoriedad se creó el código de MATLAB nombrado código A (ver anexo A para mayor detalle).

Etapas 2. Procesamiento de la información:

- Análisis y extracción de características relevantes de las muestras de las hojas mediante técnicas de procesamiento de imágenes:
De acuerdo a lo observado en la literatura acerca de las manifestaciones de las coleópteras en las hojas del frijol se optó por la generación de dos clases para el etiquetado, *healthy* (sana), e *infected* (infectada) (ver Figura 12)

Figura 12

Definición de las clases en la sección categorías del programa de etiquetas.

```
"categories": [  
  {  
    "id": 0,  
    "name": "healthy",  
    "supercategory": "leaves"  
  },  
  {  
    "id": 1,  
    "name": "infected",  
    "supercategory": "leaves"  
  }  
]
```

- Desarrollo de un sistema de etiquetado y clasificación para identificar automáticamente las muestras y asignarlas a las clases correspondientes de acuerdo con sus características:

Se utilizó una modificación de *SAM* para crear las etiquetas de las imágenes de acuerdo a las categorías definidas anteriormente. El uso de *SAM* facilitó la segmentación detallada de las imágenes, permitiendo una identificación clara de las regiones de interés dentro de cada imagen. Esta segmentación precisa es crucial para la correcta caracterización de los objetos, en este caso, las plantas sanas e infectadas. Además, el proceso manual de etiquetado asegura que las etiquetas asignadas a cada imagen son exactas, lo cual es fundamental para el rendimiento del modelo de aprendizaje profundo.

El descriptor desarrollado se compone de las características extraídas de

las imágenes segmentadas, como la textura, color, forma y contorno de las plantas. Estas características son esenciales para que posteriormente el modelo YOLO pueda aprender y distinguir entre plantas sanas e infectadas.

En la Figura 13 se muestra un ejemplo de la segmentación, en la Figura 14 se muestra la selección de las hojas y algunas etiquetas de las hojas en la imagen completa.

Figura 13

Segmentador.

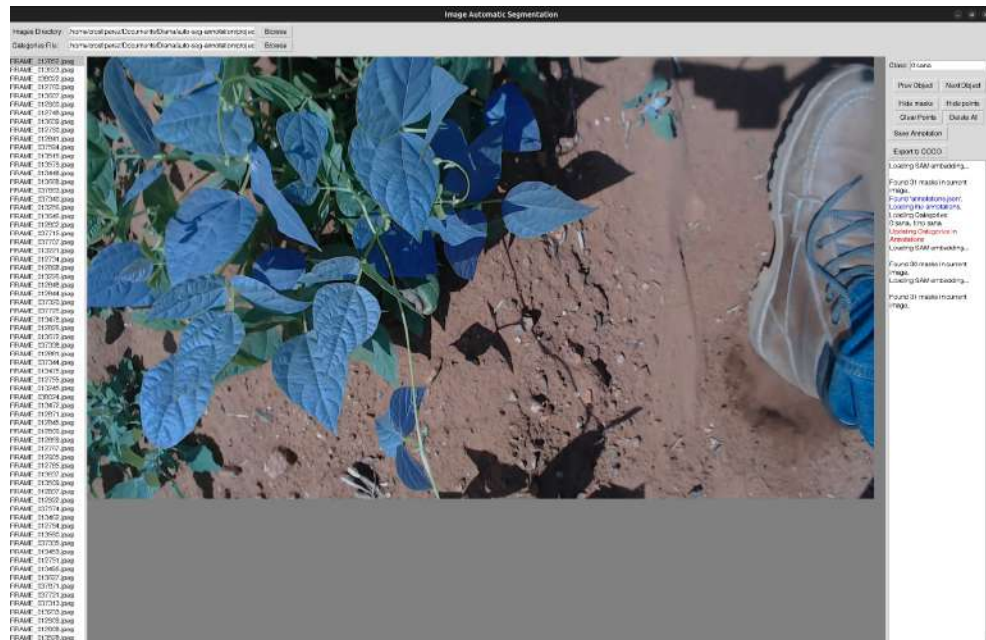
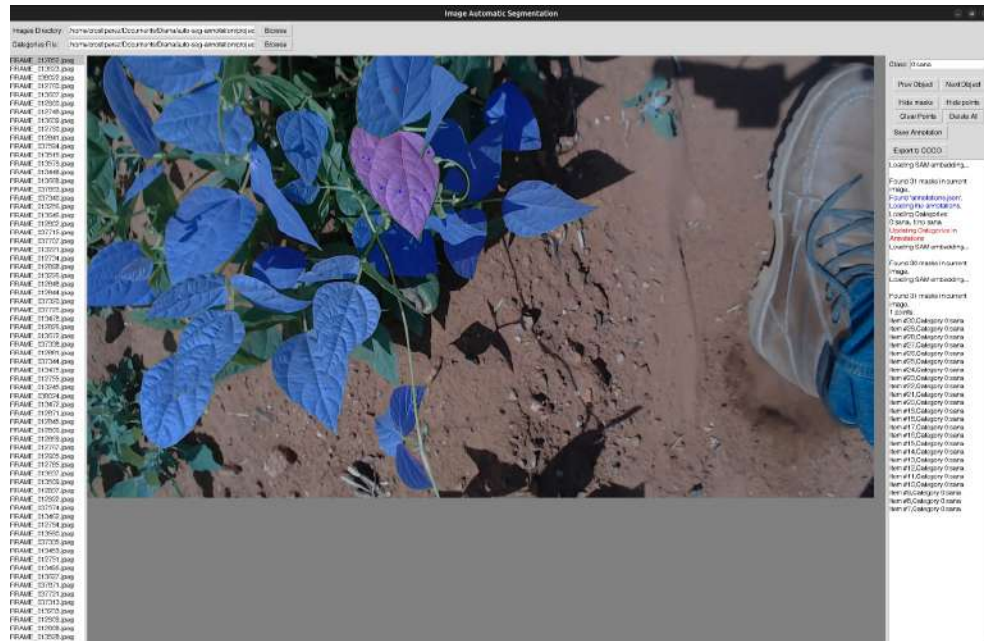


Figura 14

Etiqueta de segmentado.



Nota. En la parte derecha se observan el valor y nombre de la clase asignada.

Se realizó este proceso hasta asegurar que todas las hojas de frijol presentes en cada imágenes se encontraran correctamente etiquetadas como se puede apreciar en la Figura 15.

Figura 15

Ejemplos de imagen etiquetada.



Nota. A cada hoja de frijol le corresponde un cuadro delimitador (*bounding box*).

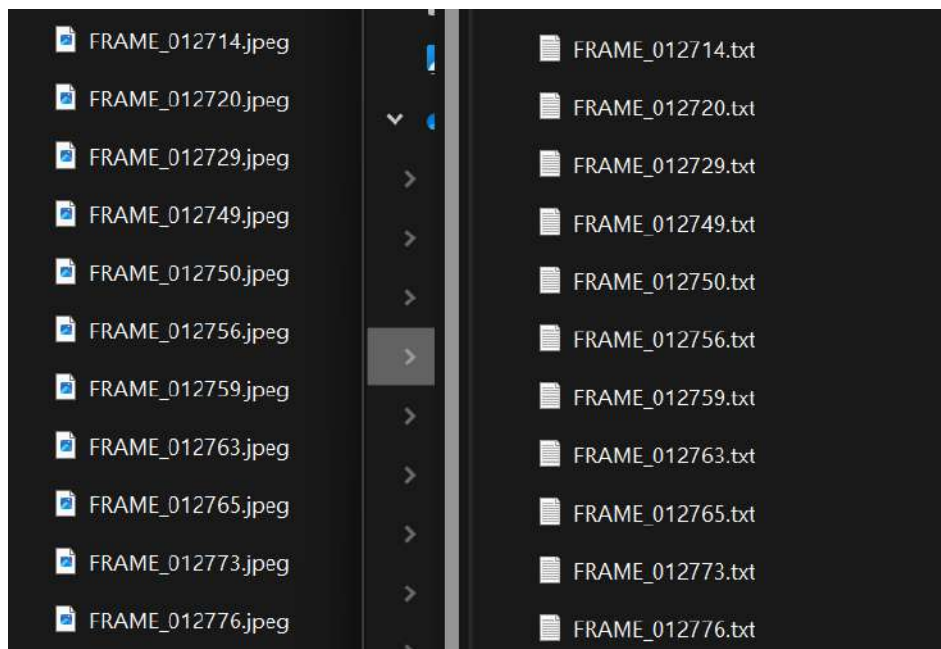
Etapa 3. Arquitectura de la red neuronal artificial:

- Seleccionar y configurar una arquitectura de red neuronal artificial adecuada para el problema de clasificación de las muestras de hojas de frijol afectadas. Para abordar el problema de la clasificación de hojas de frijol afectadas por plagas, se seleccionó la arquitectura de red YOLO. La elección de YOLO se basó en su capacidad para realizar detección de objetos en tiempo real con alta precisión y eficiencia. YOLO ha demostrado ser altamente eficaz en tareas de visión por computadora, particularmente en la identificación y clasificación de objetos en imágenes complejas, sumado a esto se comparó el rendimiento de YOLO con otros modelos de clasificación utilizados en trabajos previos para asegurar que la elección de YOLO era la más adecuada. Este enfoque sistemático y detallado asegura que el modelo desarrollado no solo es preciso y eficiente, sino también robusto y generalizable a nuevas muestras. En el modelo YOLO, a cada una de las imágenes le corresponde un archivo de texto (.txt) que contiene información de los objetos segmentados, cada imagen puede tener múltiples objetos de interés y múltiples etiquetas y así mismo

el archivo de texto asociado. Ya que la segmentación que se realizó genera archivo de anotación COCO (*Common Objects in Context* u objetos comunes en contexto al español) fue necesario transformarlos para lo cual se desarrolló el programa que se muestra en el anexo B, el cuál garantizaba un archivo .txt asociado mediante el nombre a cada imagen como se puede observar en la Figura 16.

Figura 16

Imágenes y su respectivo archivo de etiquetas.



Nota. A la izquierda se observan parte de las imágenes de un subconjunto de imágenes, a la derecha se observan los archivos de texto asociados a estas imágenes.

Se probaron distintos modelos de YOLO, específicamente YOLO5n, YOLO5s, YOLO5m, YOLO5l, YOLO5x, YOLO8n, YOLO8s, YOLO8m, YOLO8l, YOLO8x, YOLO9c, YOLO9e para determinar cuál proporcionaba los mejores resultados en términos de precisión, velocidad y capacidad de generalización. La

selección final se hizo basada en una comparación exhaustiva de estas versiones en función de los resultados obtenidos en el conjunto de datos específico de hojas de frijol.

- Definir los parámetros de la red neuronal. Una vez seleccionada la versión de YOLO más adecuada, se procedió a la configuración de los parámetros de la red. Esto incluyó:

Tamaño del modelo: Se evaluaron diferentes tamaños de modelo (pequeño, mediano, grande) para balancear la precisión y la velocidad.

Hiperparámetros: Se ajustaron hiperparámetros como la tasa de aprendizaje, el número de épocas y el tamaño del lote para optimizar el rendimiento del modelo.

Anclas: Se definieron las anclas (*anchors*) específicas para el conjunto de datos de hojas de frijol, ajustando las dimensiones para mejorar la detección de los daños característicos de las plagas.

Tabla 5

hiperparámetros estandarizados para la red.

hiperparámetro	Valor
Épocas	200
Tamaño del lote	16
Paciencia	20
Tamaño de la imagen	640

En el proceso de entrenamiento del modelo YOLO, primero se cargan las configuraciones y rutas de los datos desde los archivos `config.yaml` y `data.yaml`, respectivamente. El archivo `config.yaml` proporciona parámetros esenciales como el tamaño del lote, el número de épocas, la tasa de

aprendizaje, el tamaño de la imagen y la configuración del modelo. El archivo `data.yaml` define las rutas a los conjuntos de datos de entrenamiento y validación, así como los nombres de las clases (sano e infectado). Con esta información, se inicializa el modelo YOLO y se configuran los parámetros de entrenamiento. Luego, se cargan los conjuntos de datos correspondientes y se compila el modelo especificando el optimizador, la función de pérdida y las métricas de evaluación. El modelo se entrena utilizando los datos de entrenamiento y validación durante el número de épocas especificado y con el tamaño de lote definido. Finalmente, el modelo entrenado se guarda en un archivo y se evalúa utilizando el conjunto de datos de validación, mostrando los resultados de la evaluación (ver Figura 17).

Figura 17

Pseudo código del algoritmo.

```
# Importar las bibliotecas necesarias

import yaml

import biblioteca_modelo_yolo # Biblioteca hipotética para YOLO

# Cargar configuración y rutas de datos

with open('config.yaml', 'r') as file:

    config = yaml.safe_load(file)

with open('data.yaml', 'r') as file:

    data = yaml.safe_load(file)

# Inicializar el modelo con la configuración

modelo = biblioteca_modelo_yolo.ModeloYOLO(config['config_modelo'])

# Configurar parámetros de entrenamiento

tamano_lote = config['tamano_lote']

epocas = config['epocas']

tasa_aprendizaje = config['tasa_aprendizaje']

tamano_imagen = config['tamano_imagen']

# Cargar conjuntos de datos

conjunto_entrenamiento = cargar_conjunto_datos(data['entrenamiento'],
tamano_imagen)

conjunto_validacion = cargar_conjunto_datos(data['validacion'], tamano_imagen)

# Compilar modelo

modelo.compile(optimizador='adam', perdida='categorical_crossentropy',
metricas=['accuracy'])

# Entrenar modelo

modelo.entrenar(conjunto_entrenamiento,
datos_validacion=conjunto_validacion, epocas=epocas,
tamano_lote=tamano_lote)

# Guardar modelo entrenado

modelo.guardar('modelo_yolo_entrenado.h5')

# Evaluar modelo

evaluacion = modelo.evaluar(conjunto_validacion)

print('Evaluación:', evaluacion)
```

- Entrenar la red neuronal utilizando la muestra de datos creada previamente.

El entrenamiento de la red se realizó utilizando el conjunto de datos creado previamente, que contiene imágenes etiquetadas de hojas de frijol con y sin daños causados por Coleóptera: Coccinellidae y/o Chrysomelidae. El proceso de entrenamiento incluyó los siguientes pasos:

Se realizó un entrenamiento utilizando todo el conjunto de datos, aplicando técnicas de aumento de datos (*data augmentation*) para mejorar la robustez del modelo las cuales se encuentran detalladas en la Tabla 6.

Tabla 6

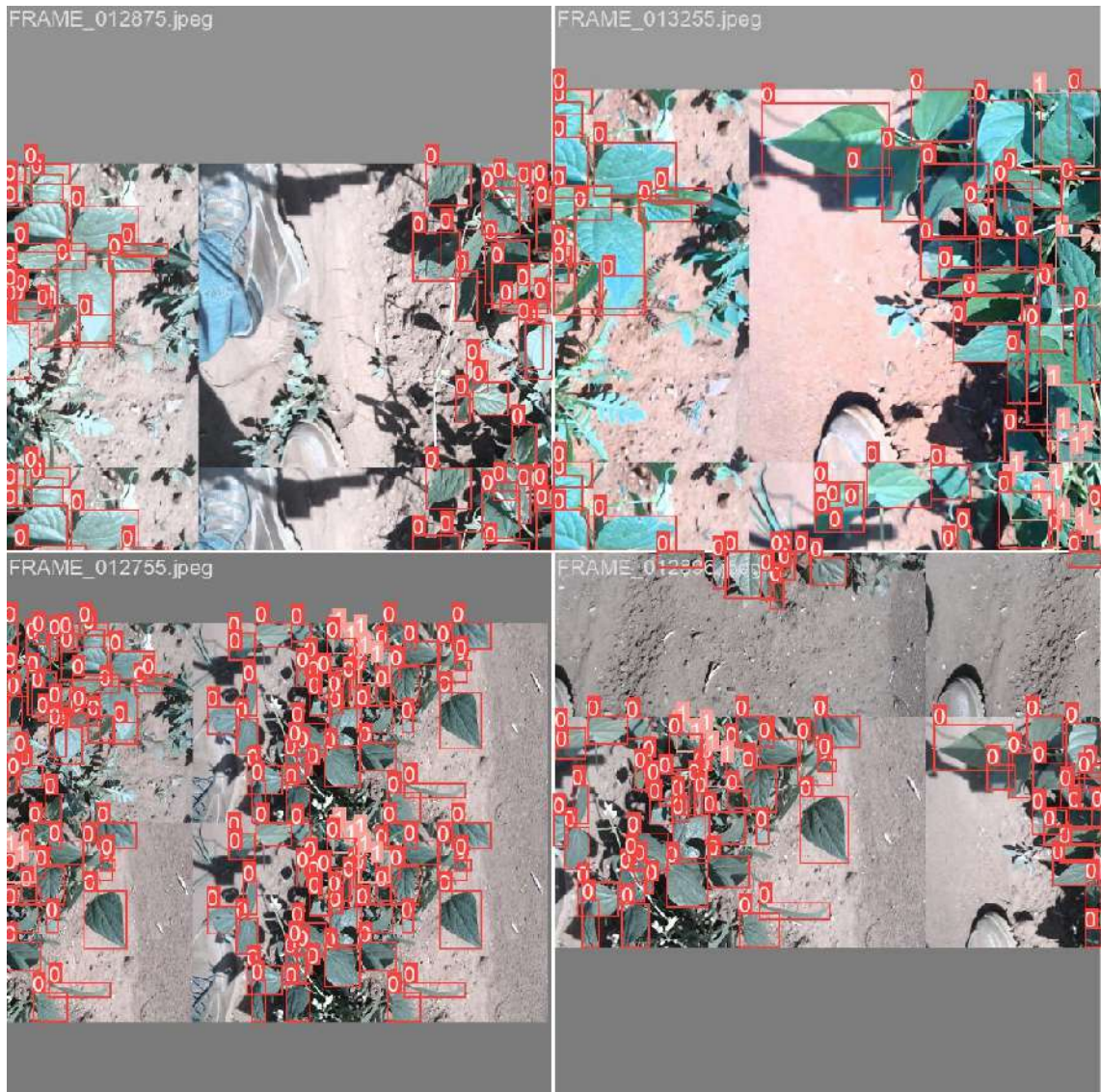
Hiperparámetros de Aumento de datos.

Nombre	Descripción	Probabilidad
hsv_hue_augmentation	Aumento de tonalidad en el espacio de color	0.015
hsv_saturation_augmentation	Aumento de saturación en el espacio de color HSV	0.7
hsv_value_augmentation	Aumento de valor brillo en el espacio de color HSV	0.4
image_translation	Traslación de la imagen como fracción del tamaño de la imagen	0.1
image_scale	Escalado de la imagen como fracción del tamaño original	0.5
flip_left_right	Probabilidad de voltear la imagen horizontalmente	0.5
mosaic_augmentation	Probabilidad de aplicar el aumento de mosaico	1.0
random_erasing	Probabilidad de borrar partes de la imagen durante el entrenamiento	0.4
image_crop_fraction	Fracción de recorte de la imagen para clasificación	1.0

Mediante este proceso se generaron imágenes como las que se observan en la Figura 18 para el entrenamiento del modelo YOLO9e el cual presenta resultados notablemente mejores al resto de los modelos.

Figura 18

Ejemplo de imágenes creadas por el modelo.



Nota. Las imágenes fueron creados por el modelo para su entrenamiento basados en los parámetros de aumento de datos (*data augmentation*) establecidos.

Etapa 4. Evaluación de rendimiento:

- Pruebas de validación cruzada y evaluación del rendimiento de la red neuronal

Con el fin de realizar una correcta comparación y evaluación de desempeño de los distintos modelos generados se tomaron en cuenta distintas métricas como lo son las siguientes curvas, *Precision-Recall* (Figura 19) las cuales nos ofrece información crucial para entender la capacidad de los modelos para manejar diferentes distribuciones de clases.

La curva de confianza contra F1 (Figura 20) que proporciona una medida integral del rendimiento del modelo

Las curvas *Recall-Confidence* (Figura 21) y *Precision-Confidence* (Figura 22) las cuales ayudan a identificar el nivel de confianza en el que cada modelo se desempeña mejor en términos de sensibilidad y precisión.

En adición a esto, se evaluó el rendimiento de los modelos para la detección de objetos empleando distintas métricas (X. Chen et al., 2024) las cuales son las siguientes:

Precisión, conocida también como $Box(P)$, mide la proporción de observaciones positivas correctamente predichas en relación con el total de observaciones positivas predichas. Se calcula como 1:

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

donde TP son los verdaderos positivos y FP los falsos positivos.

Esta métrica nos indica cuantas predicciones fueron correctamente identificadas

Recall (R) mide la proporción de observaciones positivas correctas en relación con

las observaciones en la clase real. se define como 2:

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

donde FN son los falsos negativos. Recall destaca la capacidad del modelo para identificar todas las instancias relevantes dentro del conjunto de datos.

El **Mean Average Precision at IoU=0.50 (mAP50)** es una métrica ampliamente utilizada que promedia la precisión a través de todas las clases en un umbral de Intersección sobre Unión (IoU) de 0.50. Esta métrica proporciona información sobre el rendimiento del modelo en la detección de objetos con un criterio de solapamiento moderado.

Para una evaluación más detallada, se utiliza también **mean Average Precision at IoU thresholds from 0.50 to 0.95 (mAP50-95)**. Esta métrica promedia la precisión a través de múltiples umbrales de IoU , que van desde 0.50 hasta 0.95 con un tamaño de paso de 0.05, ofreciendo una visión más completa de la capacidad del modelo para predecir con precisión las cajas delimitadoras en diferentes niveles de solapamiento.

Capítulo 7

Resultados y Discusión

Resultados obtenidos

En las siguientes figuras, se puede observar el desempeño de los modelos bajo distintas métricas como lo son la curva de *precision-recall* (figura 19) en la que se espera un comportamiento cercano a 1 en sus dos ejes, la curva f1-confianza (figura20) de la que se espera una relación armónica entre estas dos métricas, recal-confianza (figura 21) de la que esperamos valores altos de recal para distintos valores de confianza, y por ultimo la curva precision-confianza (figura22) en la que se espera un comportamiento directamente proporcional entre los dos ejes.

Figura 19

Curva de precisión contra *recall* para los 12 modelos de YOLO comparados bajo diferentes umbrales.

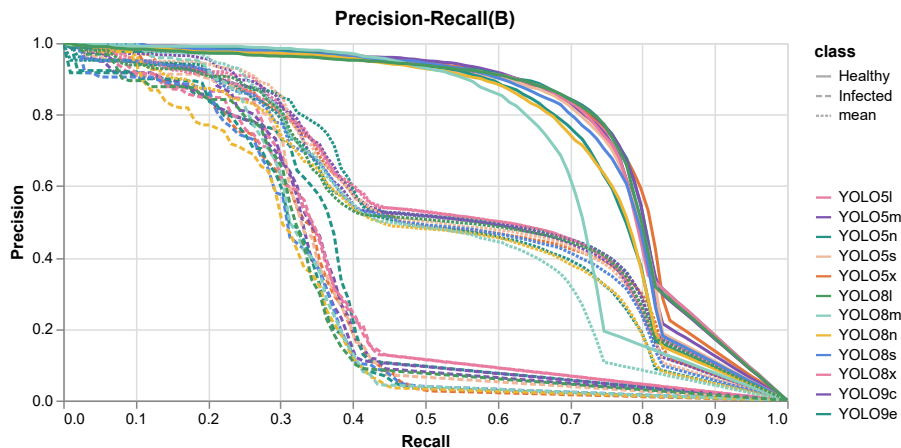
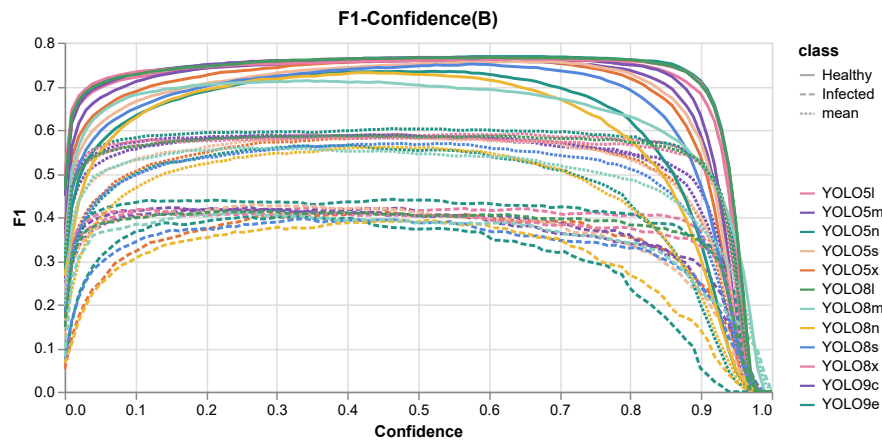


Figura 20

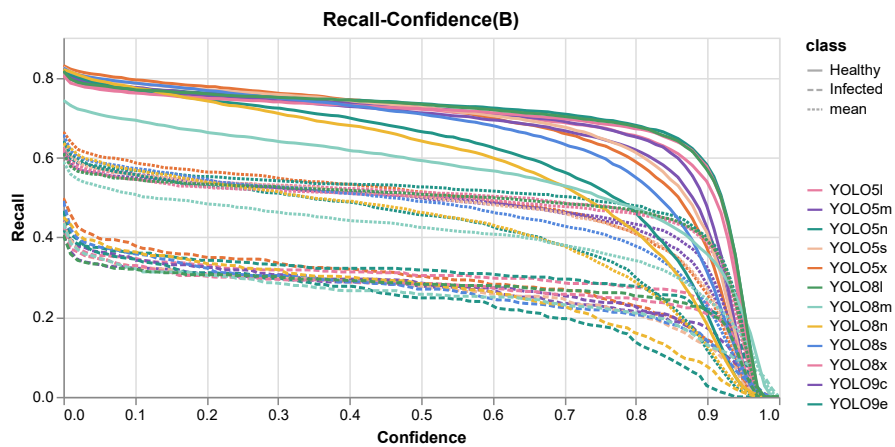
Curva de confianza contra F1 para 12 versiones de modelos YOLO.



Nota. Se muestra la variación en la puntuación F1 a través de diferentes umbrales de confianza para identificar el equilibrio óptimo entre precisión y exhaustividad.

Figura 21

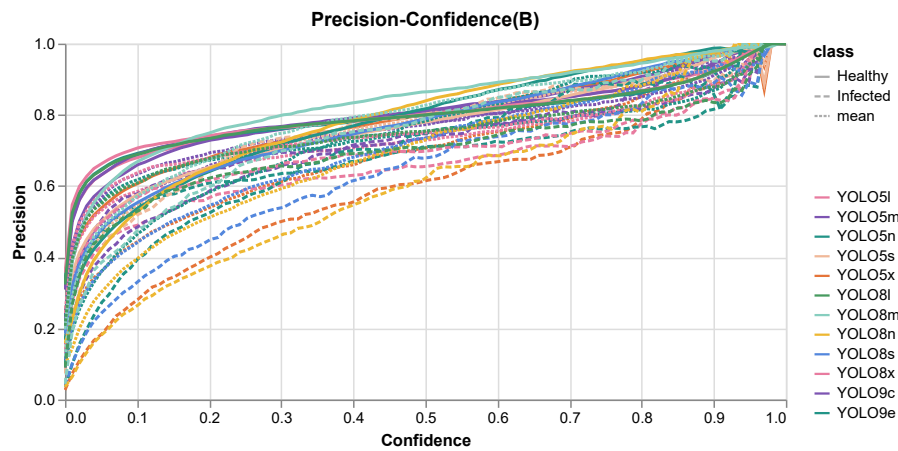
Curva de confianza contra recall para los 12 modelos de YOLO.



Nota. Se muestran los cambios en la exhaustividad con diferentes umbrales de confianza, destacando la sensibilidad de cada modelo.

Figura 22

Curva de confianza contra precisión para 12 versiones de modelos YOLO.



Nota. Se ilustra cómo cambia la precisión con diferentes umbrales de confianza, lo que indica la exactitud de cada modelo en varios niveles.

En la Tabla 7 se muestran las métricas de desempeño en el conjunto de prueba de cada modelo en precisión, recall y (mAP). Esta tabla sirve como base para identificar el modelo de YOLO más efectivo para detectar manifestaciones de plagas en hojas de frijol.

Tabla 7

Métricas de rendimiento de YOLO.

Model	Class	Images	Instances	Box(P)	Box(R)	mAP50	mAP50-95
yolov5n	all	100	5131	0.685	0.512	0.532	0.421
yolov5n	Healthy	100	4647	0.789	0.698	0.752	0.599
yolov5n	Infected	53	484	0.581	0.326	0.312	0.244
yolov5s	all	100	5131	0.732	0.519	0.558	0.471
yolov5s	Healthy	100	4647	0.786	0.73	0.767	0.651
yolov5s	Infected	53	484	0.648	0.308	0.349	0.291
yolov5m	all	100	5131	0.733	0.509	0.56	0.478
yolov5m	Healthy	100	4647	0.819	0.716	0.774	0.673
yolov5m	Infected	53	484	0.648	0.302	0.345	0.283
yolov5l	all	100	5131	0.684	0.557	0.583	0.513
yolov5l	Healthy	100	4647	0.762	0.745	0.783	0.694
yolov5l	Infected	53	484	0.606	0.368	0.383	0.333
yolov5x	all	100	5131	0.738	0.529	0.567	0.49
yolov5x	Healthy	100	4647	0.821	0.72	0.783	0.68
yolov5x	Infected	53	484	0.655	0.339	0.352	0.3
yolov8n	all	100	5131	0.697	0.505	0.541	0.428
yolov8n	Healthy	100	4647	0.819	0.668	0.75	0.591
yolov8n	Infected	53	484	0.575	0.343	0.332	0.265
yolov8s	all	100	5131	0.76	0.494	0.55	0.455
yolov8s	Healthy	100	4647	0.847	0.677	0.764	0.637
yolov8s	Infected	53	484	0.674	0.31	0.335	0.273
yolov8m	all	100	5131	0.738	0.53	0.566	0.473
yolov8m	Healthy	100	4647	0.793	0.713	0.767	0.647
yolov8m	Infected	53	484	0.682	0.347	0.365	0.3
yolov8l	all	100	5131	0.718	0.528	0.565	0.499
yolov8l	Healthy	100	4647	0.799	0.73	0.781	0.696
yolov8l	Infected	53	484	0.636	0.326	0.348	0.302
yolov8x	all	100	5131	0.732	0.529	0.571	0.503
yolov8x	Healthy	100	4647	0.808	0.722	0.778	0.691
yolov8x	Infected	53	484	0.656	0.336	0.364	0.316
yolov9c	all	100	5131	0.762	0.522	0.575	0.505
yolov9c	Healthy	100	4647	0.821	0.717	0.782	0.689
yolov9c	Infected	53	484	0.703	0.326	0.369	0.322
yolov9e	all	100	5131	0.725	0.553	0.578	0.514
yolov9e	Healthy	100	4647	0.792	0.744	0.785	0.706
yolov9e	Infected	53	484	0.658	0.362	0.371	0.322

También se compararon los resultados de las predicciones obtenidas con las diferentes versiones del modelo YOLO: YOLOv5n (Figura 23), YOLOv5s (figura 24), YOLOv5m (Figura 25), YOLOv5l (Figura 26), YOLOv5x (Figura 27), YOLOv8n (Figura 28), YOLOv8s (Figura 29), YOLOv8m (Figura 30), YOLOv8l (Figura 31), YOLOv8x (Figura 32), YOLOv9c (Figura 33) y YOLOv9e (Figura 34), para la misma imagen. Como se puede observar, el modelo YOLO9e (Figura 34) es el que realiza predicciones más cercanas a las etiquetas reales, demostrando una mayor precisión en comparación con los otros modelos.

Figura 23

Predicción con YOLO5n.



Figura 24

Predicción con YOLO5s.



Figura 25

Predicción con YOLO5m.



Figura 26

Predicción con YOLO5l.



Figura 27

Predicción con YOLO5x.

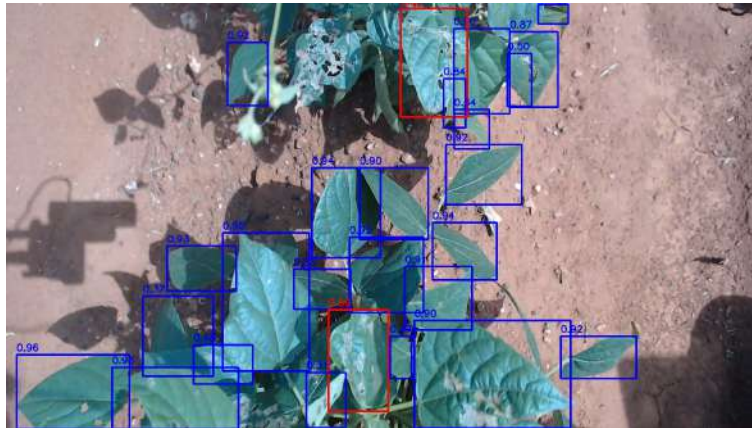


Figura 28

Predicción con YOLO8n.

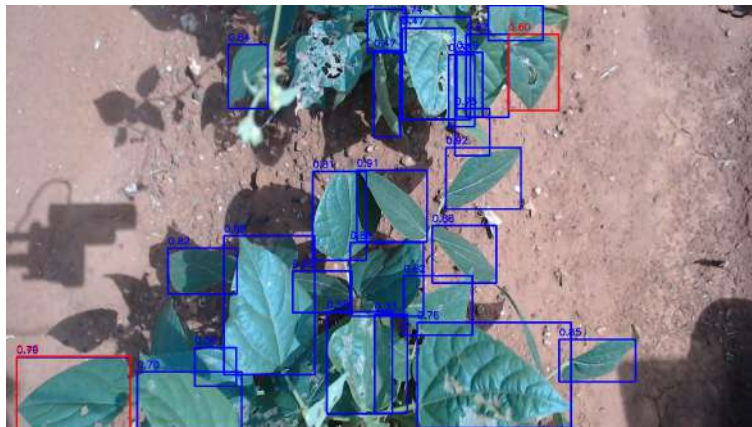


Figura 29

Predicción con YOLO8s.



Figura 30

Predicción con YOLO8m.

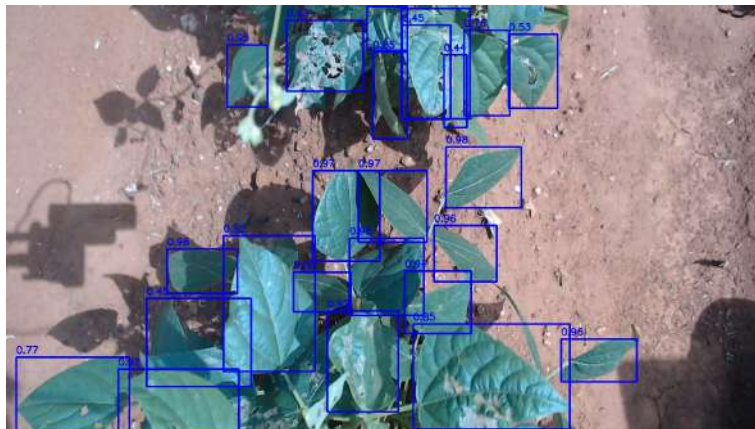


Figura 31

Predicción con YOLO8l.

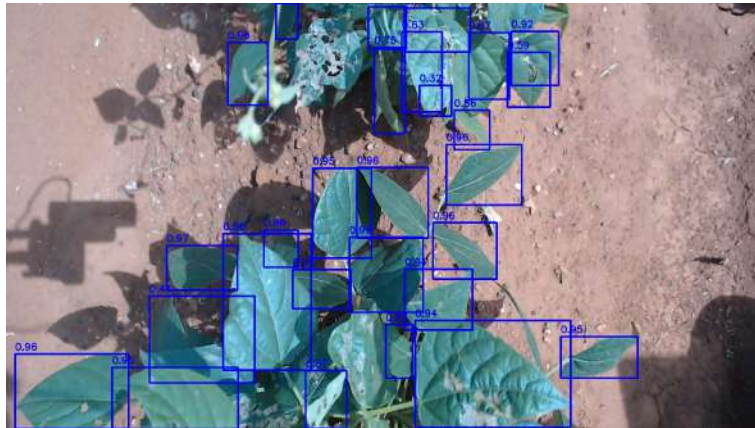


Figura 32

Predicción con YOLO8x.

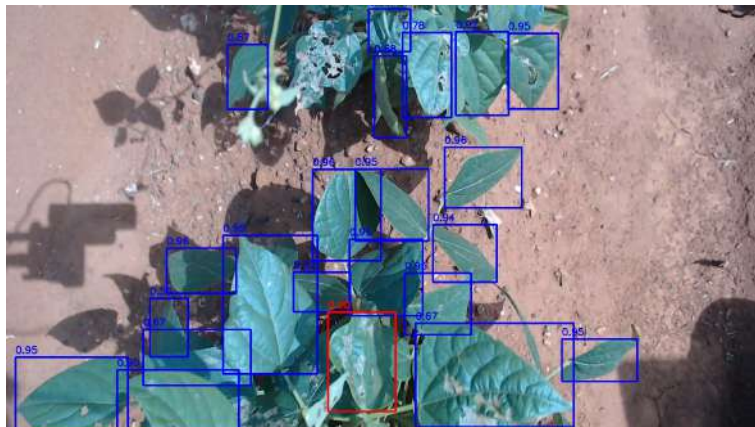


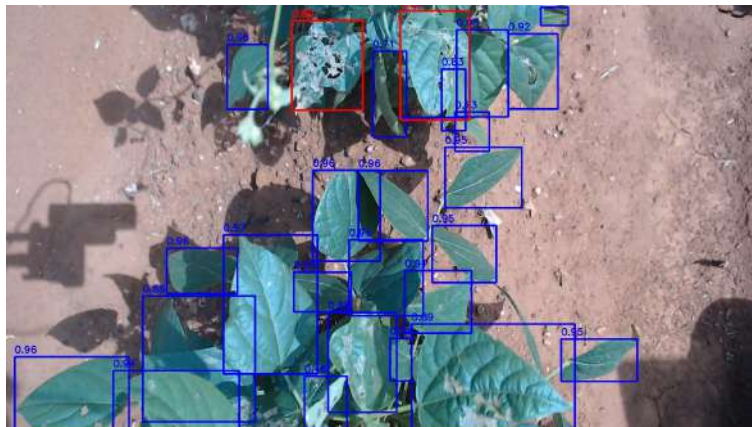
Figura 33

Predicción con YOLO9c.



Figura 34

Predicción con YOLO9e.



Discusión

En la Tabla 7 se presentan los resultados obtenidos de la evaluación de los diversos modelos de YOLO sobre un conjunto de datos que incluye imágenes de instancias “sana” e “infectada”. Se evaluaron las métricas de precisión, recall, mAP50 y mAP50-95 para cada modelo y clase.

De acuerdo con las métricas obtenidas, el modelo YOLOv9e mostró el mejor

desempeño global. Este modelo alcanzó un mAP50 de 0.578 y un mAP50-95 de 0.514, ambos los más altos entre los modelos evaluados. Para la clase “sana”, YOLOv9e también obtuvo los mejores resultados en mAP50 (0.785) y mAP50-95 (0.706). En la clase ‘infectada’, aunque el mAP50 no fue el más alto, el desempeño en mAP50-95 (0.322) fue comparable al del modelo con el mejor mAP50 en esta clase.

En las curvas de desempeño, el modelo YOLOv9e también demostró ser superior en varias métricas. La curva de Precision-Confidence (Figura 22) muestra que YOLOv9e mantiene una alta precisión a lo largo de diferentes niveles de confianza, superando a otros modelos especialmente en niveles altos de confianza. De manera similar, en la curva de Recall-Confidence (Figura 21), YOLOv9e mantiene un recall consistente y alto, lo que indica una capacidad robusta para detectar instancias positivas a diferentes niveles de confianza.

Además, la curva de Precision-Recall (figura 19) muestra que YOLOv9e mantiene una alta precisión y recall simultáneamente, lo cual refuerza su desempeño en el balance entre ambas métricas comparado con otros modelos, esto puede atribuirse a la arquitectura y los ajustes específicos de este modelo que permitieron una mejor generalización y capacidad de detección en diferentes niveles de confianza.

Comparacion de trabajos similares y sus aportaciones a la literatura.

Con el fin de resumir y esclarecer la relevancia y aportes del presente trabajo, se generó la tabla 8 en la cual se muestran estas características clave en contraste con trabajos similares observados en la literatura.

Tabla 8

Comparación de características en diferentes trabajos.

Trabajo	Uso de fondos complejos (reales)	de Incidencia de clases en misma imagen	Distintas condicionales de brillo	Aplicado al frijol o soya	Uso de aumento de datos
Propuesto	x	x	x	x	x
Raja Karthikeyan, 2023b	y -	-	-	-	-
Chug et al., 2023	-	-	-	-	x
Towfek Khodadadi, 2023	y -	-	-	x	x
Xue et al., 2023	x	x	x	-	-
Khoramshahi et al., 2023	x	-	-	-	-
Aabidi, EL Makrani et al., 2023	-	-	-	x	-
Sahu y Pandey, 2023	-	-	-	x	x

Acciones Potenciales.

Como sugerencia para futuros trabajos se recomienda considerar los siguientes puntos:

- Ajuste de Hiperparámetros: Ajustar la tasa de aprendizaje y otros hiperparámetros específicos que puedan ayudar a manejar el desequilibrio de clases y mejorar el rendimiento general del modelo.
- Aumento de Datos: Implementar distintas técnicas de *data augmentation* para equilibrar el conjunto de datos y aumentar la representación de las clases.
- Rebalanceo del conjunto de datos: Asegurar que tanto el conjunto de datos de entrenamiento como el de validación estén equilibrados en términos de clases.
- Regularización: Aumentar la regularización para prevenir el sobreajuste a los datos de entrenamiento.

Capítulo 8

Conclusiones

En este trabajo se desarrolló un algoritmo basado en *machine learning* y técnicas de procesamiento de imágenes para la detección automática de daño en hojas de frijol causado por Coleoptera: Coccinellidae y/o Chrysomelidae. El planteamiento de la hipótesis se verificó mediante la implementación de redes neuronales YOLO en varias versiones, evaluando su rendimiento en términos de precisión, recall y mAP sobre un conjunto de datos específico.

A lo largo del desarrollo, se lograron los siguientes objetivos específicos:

- Creación de un conjunto de datos robusto que permitió el entrenamiento y validación del algoritmo.
- Procesamiento de la información utilizando descriptores de características que diferenciaron efectivamente entre hojas sanas e infectadas.
- Diseño e implementación de una red neuronal capaz de identificar los daños con alta precisión.
- Evaluación del rendimiento del algoritmo con diversas técnicas de clasificación, destacando la superioridad del modelo YOLOv9e.

Los resultados obtenidos mostraron que el modelo YOLOv9e es el más adecuado para la tarea planteada, ya que alcanzó los mejores resultados en términos de mAP50, mAP50-95 y consistencia en las curvas de Precision-Confidence y Recall-Confidence. Este modelo demostró un equilibrio óptimo entre precisión y recall, lo que es crucial para minimizar tanto falsos positivos como falsos negativos en la detección de daños por dichas plagas.

Por lo tanto, se concluye que la hipótesis planteada es válida y el objetivo general se alcanzó satisfactoriamente. El desarrollo de este algoritmo proporciona una herramienta eficaz y precisa para la identificación automática de daños en hojas de frijol, lo cual tiene un impacto significativo en la agricultura, permitiendo una detección temprana y precisa de plagas facilitando así la implementación de medidas de control más efectivas. Este trabajo sienta las bases para futuras investigaciones y mejoras en algoritmos de detección de plagas utilizando técnicas de aprendizaje automático.

Bibliografía

- Aabidi, M. H., EL Makrani, A., Jabir, B., & Zaimi, I. (2023). A Model Proposal for Enhancing Leaf Disease Detection Using Convolutional Neural Networks (CNN): Case Study [Cited by: 0; All Open Access, Gold Open Access]. *International journal of online and biomedical engineering*, 19(12), 127-143. <https://doi.org/10.3991/ijoe.v19i12.40329>
- Aabidi, M. H., Makrani, A. E., Jabir, B., & Zaimi, I. (2023). A NEW APPROACH OF LEAF DISEASE DETECTION USING BAG OF VISUAL WORDS [Cited by: 0]. *Journal of Theoretical and Applied Information Technology*, 101(14), 5645-5664. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85173522773&partnerID=40&md5=65364f2c8b88da0bde8c73d899e0db59>
- Abuhayi, B. M., & Mossa, A. A. (2023). Coffee disease classification using Convolutional Neural Network based on feature concatenation [Cited by: 2; All Open Access, Gold Open Access]. *Informatics in Medicine Unlocked*, 39. <https://doi.org/10.1016/j.imu.2023.101245>
- Adedoja, A., Owolawi, P. A., & Mapayi, T. (2019). Deep learning based on nasnet for plant disease recognition using leave images. *2019 international conference on advances in big data, computing and data communication systems (icABCD)*, 1-5.
- Ahmad, I., Hamid, M., Yousaf, S., Shah, S. T., & Ahmad, M. O. (2020). Optimizing pretrained convolutional neural networks for tomato leaf disease detection. *Complexity*, 2020, 1-6.
- Ahmad, I., Yang, Y., Yue, Y., Ye, C., Hassan, M., Cheng, X., Wu, Y., & Zhang, Y. (2022). Deep learning based detector YOLOv5 for identifying insect pests. *Applied Sciences*, 12(19), 10167.

- Ahmad, W., Adnan, S. M., & Irtaza, A. (2023). Local triangular-ternary pattern: a novel feature descriptor for plant leaf disease detection [Cited by: 0]. *Multimedia Tools and Applications*. <https://doi.org/10.1007/s11042-023-16420-8>
- Ahmed, M., & Ahmed, A. (2023). Palm tree disease detection and classification using residual network and transfer learning of inception ResNet [Cited by: 1; All Open Access, Gold Open Access, Green Open Access]. *PLoS ONE*, *18*(3 March). <https://doi.org/10.1371/journal.pone.0282250>
- Alharbi, A., Khan, M. U. G., & Tayyaba, B. (2023). Wheat Disease Classification Using Continual Learning [Cited by: 0; All Open Access, Gold Open Access]. *IEEE Access*, *11*, 90016-90026. <https://doi.org/10.1109/ACCESS.2023.3304358>
- Alharbi, M., Rajagopal, S. K., Rajendran, S., & Alshahrani, M. (2023). Plant Disease Classification Based on ConvLSTM U-Net with Fully Connected Convolutional Layers [Cited by: 4; All Open Access, Bronze Open Access]. *Traitement du Signal*, *40*(1), 157-166. <https://doi.org/10.18280/ts.400114>
- Altalak, M., Uddin, M. A., Alajmi, A., & Rizg, A. (2022). A hybrid approach for the detection and classification of tomato leaf diseases. *Applied Sciences*, *12*(16), 8182.
- Amiri, N., Khebiza, M. Y., & Messouli, M. (2023). The Impact of Climate Change on Insect Pests Damaging Crops: How Insect Pests Damage Crops. En *Climate Change and the Economic Importance and Damages of Insects* (pp. 73-101). IGI Global.
- Anim-Ayeko, A. O., Schillaci, C., & Lipani, A. (2023). Automatic blight disease detection in potato (*Solanum tuberosum* L.) and tomato (*Solanum lycopersicum*, L. 1753) plants using deep learning [Cited by: 6; All Open

- Access, Gold Open Access, Green Open Access]. *Smart Agricultural Technology*, 4. <https://doi.org/10.1016/j.atech.2023.100178>
- Arora, J., Agrawal, U., et al. (2020). Classification of Maize leaf diseases from healthy leaves using Deep Forest. *Journal of Artificial Intelligence and Systems*, 2(1), 14-26.
- Atila, Ü., Uçar, M., Akyol, K., & Uçar, E. (2021). Plant leaf disease classification using EfficientNet deep learning model. *Ecological Informatics*, 61, 101182.
- Atole, R. R., & Park, D. (2018). A multiclass deep convolutional neural network classifier for detection of common rice plant anomalies. *International Journal of Advanced Computer Science and Applications*, 9(1).
- Ayaz, M., Shah, S. K., Ullah, K., Iqbal, T., Khan, A., Ghadi, Y. Y., & Eldin, S. M. (2023). Automatic Early Diagnosis of Dome Galls in *Cordia Dichotoma* G. Forst. Using Deep Transfer Learning [Cited by: 0; All Open Access, Gold Open Access]. *IEEE Access*, 11, 59511-59523. <https://doi.org/10.1109/ACCESS.2023.3283568>
- Baljon, M. (2023). A Framework for Agriculture Plant Disease Prediction using Deep Learning Classifier [Cited by: 0; All Open Access, Gold Open Access]. *International Journal of Advanced Computer Science and Applications*, 14(8), 1098-1111. <https://doi.org/10.14569/IJACSA.2023.01408119>
- Bansal, P., Ranvijay & Yadav, M. (2023). Automatic detection of plant leaf diseases using deep learning [Cited by: 0; All Open Access, Gold Open Access]. *International Journal of Computing and Digital Systems*, 13(1), 901-910. <https://doi.org/10.12785/ijcds/130171>
- Barman, U., Pathak, C., & Mazumder, N. K. (2023). Comparative assessment of Pest damage identification of coconut plant using damage texture and color analysis. *Multimedia Tools and Applications*, 82(16), 25083-25105.

- Bayram, H. Y., Bingol, H., & Alatas, B. (2022). Hybrid deep model for automated detection of tomato leaf diseases. *Traitement du Signal*, 39(5), 1781.
- Bensaadi, S., & Louchene, A. (2023). Low-cost convolutional neural network for tomato plant diseases classification [Cited by: 1; All Open Access, Gold Open Access, Green Open Access]. *IAES International Journal of Artificial Intelligence*, 12(1), 162-170. <https://doi.org/10.11591/ijai.v12.i1.pp162-170>
- Bhimte, N. R., & Thool, V. (2018). Diseases detection of cotton leaf spot using image processing and SVM classifier. *2018 Second international conference on intelligent computing and control systems (ICICCS)*, 340-344.
- Bi, C., Wang, J., Duan, Y., Fu, B., Kang, J.-R., & Shi, Y. (2022). MobileNet based apple leaf diseases identification. *Mobile Networks and Applications*, 1-9.
- Boliko, M. C. (2019). FAO and the situation of food security and nutrition in the world. *Journal of nutritional science and vitaminology*, 65(Supplement), S4-S8.
- Buruchara, R. A., Mukaruziga, C., & Ampofo, K. (2010). Bean disease and pest identification and management.
- Casas, E., Arbelo, M., Moreno-Ruiz, J. A., Hernández-Leal, P. A., & Reyes-Carlos, J. A. (2023). UAV-Based Disease Detection in Palm Groves of Phoenix canariensis Using Machine Learning and Multispectral Imagery [Cited by: 0; All Open Access, Gold Open Access]. *Remote Sensing*, 15(14). <https://doi.org/10.3390/rs15143584>
- Chen, J., Chen, J., Zhang, D., Sun, Y., & Nanekaran, Y. A. (2020). Using deep transfer learning for image-based plant disease identification. *Computers and electronics in agriculture*, 173, 105393.
- Chen, X., Liu, T., Han, K., Jin, X., Wang, J., Kong, X., & Yu, J. (2024). TSP-yolo-based deep learning method for monitoring cabbage seedling emergence. *European Journal of Agronomy*, 157, 127191.

- Chug, A., Bhatia, A., Singh, A. P., & Singh, D. (2023). A novel framework for image-based plant disease detection using hybrid deep learning approach [Cited by: 7]. *Soft Computing*, 27(18), 13613-13638. <https://doi.org/10.1007/s00500-022-07177-7>
- CNRF. (2023). Centro nacional de referencia fitosanitaria.
- Dhande, A., & Malik, R. (2023). Design of a highly efficient crop damage detection ensemble learning model using deep convolutional networks [Cited by: 1]. *Journal of Ambient Intelligence and Humanized Computing*, 14(8), 10811-10821. <https://doi.org/10.1007/s12652-022-04352-4>
- Duan, W., Jiang, M., & Qi, J. (2024). Agricultural certification, market access and rural economic growth: Evidence from poverty-stricken counties in China. *Economic Analysis and Policy*, 81, 99-114.
- Edwards, C. R., Gort, J. A., & Pons, X. (2013). La diabrotica, una plaga invasiva del maíz que puede amenazar España. *Phytoma España: La revista profesional de sanidad vegetal*, (251), 21.
- Elhassouny, A., & Smarandache, F. (2019). Smart mobile application to recognize tomato leaf diseases using Convolutional Neural Networks. *2019 International Conference of Computer Science and Renewable Energies (ICCSRE)*, 1-4.
- EPPO. (2023). European and Mediterranean Plant Protection Organization.
- Esomonu, N., Eze, U., John-Otumu, A., Ayogu, I., Nwokonkwo, O., Oshoiribhor, E., Okolie, S., Nwokorie, E., Mbamala, C., & Dokun, O. (2023). A Hybrid Model using MobileNetv2 and SVM for Enhanced Classification and Prediction of Tomato Leaf Diseases [Cited by: 0; All Open Access, Hybrid Gold Open Access]. *SSRG International Journal of Electronics and Communication Engineering*, 10(8), 37-50. <https://doi.org/10.14445/23488379/IJEEE-V10I8P104>

- Fones, H. N., Bebber, D. P., Chaloner, T. M., Kay, W. T., Steinberg, G., & Gurr, S. J. (2020). Threats to global food security from emerging fungal and oomycete crop pathogens. *Nature Food*, 1(6), 332-342.
- Food and Agriculture Organization of the United Nations. (2024).
- Fuglie, K. O., & Echeverria, R. G. (2024). The economic impact of CGIAR-related crop technologies on agricultural productivity in developing countries, 1961–2020. *World Development*, 176, 106523.
- Gehlot, M., & Gandhi, G. C. (2023). “EffiNet-TS”: A deep interpretable architecture using EfficientNet for plant disease detection and visualization [Cited by: 3]. *Journal of Plant Diseases and Protection*, 130(2), 413-430. <https://doi.org/10.1007/s41348-023-00707-x>
- Hasan, S., Jahan, S., & Islam, M. I. (2022). Disease detection of apple leaf with combination of color segmentation and modified DWT. *Journal of King Saud University-Computer and Information Sciences*, 34(9), 7212-7224.
- He, S., Bao, R., Li, J., Grant, P. E., & Ou, Y. (2023). Accuracy of segment-anything model (sam) in medical image segmentation tasks. *arXiv preprint arXiv:2304.09324*.
- Hernández, F., & Mariselys, P. (2021). *Insectos plagas del frijol Cul 156 en el agroecosistema Zequeira: taxonomía y etología* [Tesis doctoral, Universidad de Matanzas. Facultad de Ciencias Agropecuarias].
- Hipsch, M., Michael, Y., Lampl, N., Sapir, O., Cohen, Y., Helman, D., & Rosenwasser, S. (2023). Early detection of late blight in potato by whole-plant redox imaging [Cited by: 1; All Open Access, Green Open Access, Hybrid Gold Open Access]. *Plant Journal*, 113(4), 649-664. <https://doi.org/10.1111/tpj.16071>

- Hlaing, C. S., & Zaw, S. M. M. (2018). Tomato plant diseases classification using statistical texture feature and color feature. *2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS)*, 439-444.
- Hu, G., Yang, X., Zhang, Y., & Wan, M. (2019). Identification of tea leaf diseases by using an improved deep convolutional neural network. *Sustainable Computing: Informatics and Systems*, 24, 100353.
- Jain, S., & Dharavath, R. (2023). Memetic salp swarm optimization algorithm based feature selection approach for crop disease detection system [Cited by: 12]. *Journal of Ambient Intelligence and Humanized Computing*, 14(3), 1817-1835. <https://doi.org/10.1007/s12652-021-03406-3>
- Jana, S., Thilagavathy, S., Shenbagavalli, S., Srividhya, G., Gowtham Prasad, T., & Hemavathy, R. (2023). Plant Leaf Disease Prediction Using Deep Dense Net Slice Fragmentation and Segmentation Feature Selection Using Convolution Neural Network [Cited by: 0]. *International Journal of Intelligent Systems and Applications in Engineering*, 11(6s), 76-85. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85167978973&partnerID=40&md5=6d562749314899678cdcf139adbd588d>
- Kalbande, K., & Patil, W. (2023). The Convolutional Neural Network for Plant Disease Detection Using Hierarchical Mixed Pooling Technique with Smoothing to Sharpening Approach [Cited by: 0; All Open Access, Gold Open Access]. *International Journal of Computing and Digital Systems*, 14(1), 357-377. <https://doi.org/10.12785/ijcds/140127>
- Kalpana, M., Karthiba, L., Senguttuvan, K., & Parimalarangan, R. (2023). Diagnosis of Major Foliar Diseases in Black gram (*Vigna mungo* L.) using Convolution Neural Network (CNN) [Cited by: 0; All Open Access, Hybrid Gold Open

Access]. *Legume Research*, 46(7), 940-945. <https://doi.org/10.18805/LR-5083>

- Kalyani, G., Sai Sudheer, K., Janakiramaiah, B., & Narendra Kumar Rao, B. (2023). Hyperparameter Optimization for Transfer Learning-based Disease Detection in Cassava Plants [Cited by: 0]. *Journal of Scientific and Industrial Research*, 82(5), 536-545. <https://doi.org/10.56042/jsir.v82i05.1089>
- Kannagi, V., Muthumanickam, S., Dhivya, K., Inbamalar, T., Vivekanand, C. V., & Rithika, M. (2023). Detection of Diseases in Flora Through Leaf Image Classification by Convolution Neural Network [Cited by: 0; All Open Access, Bronze Open Access]. *International Journal on Recent and Innovation Trends in Computing and Communication*, 11, 139-144. <https://doi.org/10.17762/ijritcc.v11i5s.6637>
- Kaur, S., Pandey, S., & Goel, S. (2018). Semi-automatic leaf disease detection and classification system for soybean culture. *IET Image Processing*, 12(6), 1038-1048.
- Kethineni, K., & Pradeepini, G. (2023). Identification of Leaf Disease Using Machine Learning Algorithm for Improving the Agricultural System [Cited by: 2; All Open Access, Gold Open Access]. *Journal of Advances in Information Technology*, 14(1), 122-129. <https://doi.org/10.12720/JAIT.14.1.122-129>
- Khalid, M., Sarfraz, M. S., Iqbal, U., Aftab, M. U., Niedbała, G., & Rauf, H. T. (2023). Real-Time Plant Health Detection Using Deep Convolutional Neural Networks [Cited by: 4; All Open Access, Gold Open Access]. *Agriculture (Switzerland)*, 13(2). <https://doi.org/10.3390/agriculture13020510>
- Khan, A. I., Quadri, S., Bandy, S., & Shah, J. L. (2022). Deep diagnosis: A real-time apple leaf disease detection system based on deep learning. *computers and Electronics in Agriculture*, 198, 107093.

- Khoramshahi, E., Näsi, R., Rua, S., Oliveira, R. A., Päivänsalo, A., Niemeläinen, O., Niskanen, M., & Honkavaara, E. (2023). A Novel Deep Multi-Image Object Detection Approach for Detecting Alien Barleys in Oat Fields Using RGB UAV Images [Cited by: 0; All Open Access, Gold Open Access, Green Open Access]. *Remote Sensing*, 15(14). <https://doi.org/10.3390/rs15143582>
- Kumar, A., Yadav, D. P., Kumar, D., Pant, M., & Pant, G. (2023). Multi-scale feature fusion-based lightweight dual stream transformer for detection of paddy leaf disease [Cited by: 2]. *Environmental Monitoring and Assessment*, 195(9). <https://doi.org/10.1007/s10661-023-11628-5>
- Kumar, P. D., Suhasini, A., & Anand, D. (2023). Crop Disease Detection Using 2D CNN Based Deep Learning Architecture [Cited by: 0]. *International Journal of Intelligent Systems and Applications in Engineering*, 11(2), 461-470. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85160867090&partnerID=40&md5=053412a47b7db1ba95cb8e08da8e0513>
- Kumar, P., Raghavendran, S., Silambarasan, K., Kannan, K. S., & Krishnan, N. (2023). Mobile application using DCDM and cloud-based automatic plant disease detection. *Environmental Monitoring and Assessment*, 195(1), 44.
- Kumar, R., Chug, A., Singh, A. P., Singh, D., et al. (2022). A Systematic analysis of machine learning and deep learning based approaches for plant leaf disease classification: a review. *Journal of Sensors*, 2022.
- Kumar K, K., & E, K. (2022). Detection of rice plant disease using AdaBoostSVM classifier. *Agronomy journal*, 114(4), 2213-2229.
- Lanjewar, M. G., & Panchbhai, K. G. (2023). Convolutional neural network based tea leaf disease prediction system on smart phone using paas cloud [Cited by: 8]. *Neural Computing and Applications*, 35(3), 2755-2771. <https://doi.org/10.1007/s00521-022-07743-y>

- Lee, M.-G., Cho, H.-B., Youm, S.-K., & Kim, S.-W. (2023). Detection of Pine Wilt Disease Using Time Series UAV Imagery and Deep Learning Semantic Segmentation [Cited by: 0; All Open Access, Gold Open Access]. *Forests*, 14(8). <https://doi.org/10.3390/f14081576>
- Lee, S. H., Goëau, H., Bonnet, P., & Joly, A. (2020). New perspectives on plant disease characterization based on deep learning. *Computers and Electronics in Agriculture*, 170, 105220.
- Li, Y., Nie, J., & Chao, X. (2020). Do we really need deep CNN for plant diseases identification? *Computers and Electronics in Agriculture*, 178, 105803.
- Li, Y., & Yang, J. (2020). Few-shot cotton pest recognition and terminal realization. *Computers and Electronics in Agriculture*, 169, 105240.
- Lin, Y.-B., Liu, C.-Y., Chen, W.-L., Chang, C.-H., Ng, F.-L., Yang, K., & Hsung, J. (2023). IoT-based Strawberry Disease Detection with Wall-mounted Monitoring Cameras [Cited by: 1]. *IEEE Internet of Things Journal*, 1-1. <https://doi.org/10.1109/JIOT.2023.3288603>
- Ma, W., Yu, H., Fang, W., Guan, F., Ma, D., Guo, Y., Zhang, Z., & Wang, C. (2023). Crop Disease Detection against Complex Background Based on Improved Atrous Spatial Pyramid Pooling [Cited by: 1; All Open Access, Gold Open Access]. *Electronics (Switzerland)*, 12(1). <https://doi.org/10.3390/electronics12010216>
- Marcelino, P. (2018). Transfer learning from pre-trained models. *Towards data science*, 10(330), 23.
- Marco-Detchart, C., Carrascosa, C., Julian, V., & Rincon, J. (2023). Robust Multi-Sensor Consensus Plant Disease Detection Using the Choquet Integral † [Cited by: 1; All Open Access, Gold Open Access, Green Open Access]. *Sensors*, 23(5). <https://doi.org/10.3390/s23052382>

- Martin, W. (2019). Economic growth, convergence, and agricultural economics. *Agricultural Economics*, 50, 7-27.
- Meenakshi, T. (2023). Automatic Detection of Diseases in Leaves of Medicinal Plants Using Modified Logistic Regression Algorithm [Cited by: 1]. *Wireless Personal Communications*, 131(4), 2573-2597. <https://doi.org/10.1007/s11277-023-10555-5>
- Milke, E. B., Gebiremariam, M. T., & Salau, A. O. (2023). Development of a coffee wilt disease identification model using deep learning [Cited by: 1; All Open Access, Gold Open Access]. *Informatics in Medicine Unlocked*, 42. <https://doi.org/10.1016/j.imu.2023.101344>
- Mishra, R., Kavita, Rajpal, A., Bhatia, V., Rajpal, S., Agarwal, M., & Kumar, N. (2023). I-LDD: an interpretable leaf disease detector [Cited by: 0]. *Soft Computing*. <https://doi.org/10.1007/s00500-023-08512-2>
- Mitra, D., Goyal, A., Gupta, S., Kanyal, H. S., Kaushik, S., & Kumar, K. (2023). AUTOMATED TOMATO LEAF DISEASE DETECTION TECHNIQUE USING DEEP LEARNING [Cited by: 0]. *Journal of Theoretical and Applied Information Technology*, 101(14), 5734-5744. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85172031980&partnerID=40&md5=014d5e6ca859327aa2732097cf065321>
- Mulla, R. A., Pawar, M. E., Banait, S. S., Ajani, S. N., Borawake, M. P., & Hundekari, S. (2023). Design and Implementation of Deep Learning Method for Disease Identification in Plant Leaf [Cited by: 8; All Open Access, Bronze Open Access]. *International Journal on Recent and Innovation Trends in Computing and Communication*, 11, 278-285. <https://doi.org/10.17762/ijritcc.v11i2s.6147>

- NIMF. (2018). *Norma internacional para Medidas Fitosanitarias*. Organización de las Naciones Unidas para la Alimentación y la Agricultura.
- Odukoya, O., Aina, S., & Dégbéssé, F. (2023). A Model for Detecting Fungal Diseases in Cotton Cultivation using Segmentation and Machine Learning Approaches [Cited by: 0; All Open Access, Gold Open Access]. *International Journal of Advanced Computer Science and Applications*, 14(1), 628-636. <https://doi.org/10.14569/IJACSA.2023.0140169>
- Ogunjobi, J., Oladipo, O. A., Eluyela, D. F., EKIRAN, J. O., & Adigbole, E. (s.f.). Impact of Agricultural Financing on Poverty Reduction and Economic Growth in Nigeria. *Available at SSRN 4699722*.
- Önler, E. (2023). Feature fusion based artificial neural network model for disease detection of bean leaves [Cited by: 1; All Open Access, Gold Open Access]. *Electronic Research Archive*, 31(5), 2409-2427. <https://doi.org/10.3934/era.2023122>
- Orchi, H., Sadik, M., Khaldoun, M., & Sabir, E. (2023). Automation of Crop Disease Detection through Conventional Machine Learning and Deep Transfer Learning Approaches [Cited by: 1; All Open Access, Gold Open Access]. *Agriculture (Switzerland)*, 13(2). <https://doi.org/10.3390/agriculture13020352>
- Oyewola, D. O., Dada, E. G., Misra, S., & Damaševičius, R. (2021). Detecting cassava mosaic disease using a deep residual convolutional neural network with distinct block processing. *PeerJ Computer Science*, 7, e352.
- Ozguven, M. M., & Adem, K. (2019). Automatic detection and classification of leaf spot disease in sugar beet using deep learning algorithms. *Physica A: statistical mechanics and its applications*, 535, 122537.
- Padamata, R. B., & Atluri, S. K. (2023). Tomato Crop Disease Classification Using Semantic Segmentation Algorithm in Deep Learning [Cited by: 1; All

- Open Access, Bronze Open Access]. *Revue d'Intelligence Artificielle*, 37(2), 415-423. <https://doi.org/10.18280/ria.370218>
- Pan, J., Wang, T., & Wu, Q. (2023). RiceNet: A two stage machine learning method for rice disease identification [Cited by: 3]. *Biosystems Engineering*, 225, 25-40. <https://doi.org/10.1016/j.biosystemseng.2022.11.007>
- Peng, H., Li, Z., Zhou, Z., & Shao, Y. (2022). Weed detection in paddy field using an improved RetinaNet network. *Computers and Electronics in Agriculture*, 199, 107179.
- Pérez, K. M., et al. (2022). Sistema de reconocimiento del Lenguaje de Señas Mexicano basado en una cámara RGB-D y aprendizaje automático.
- Rahaman, N., Chowdhury, M. R., Rahman, A., Ahmed, H., Hossain, M., Rahman, H., Biswas, S., Kader, A., Noyan, T. A., & Biswas, M. (2023). A Deep Learning Based Smartphone Application for Detecting Mango Diseases and Pesticide Suggestions [Cited by: 2; All Open Access, Gold Open Access]. *International Journal of Computing and Digital Systems*, 13(1), 1273-1286. <https://doi.org/10.12785/ijcnds/1301104>
- Rahman, F., Ritun, I. J., Farhin, N., & Uddin, J. (2019). An assistive model for visually impaired people using YOLO and MTCNN. *Proceedings of the 3rd International Conference on Cryptography, Security and Privacy*, 225-230.
- Raja, D., & Karthikeyan, M. (2023a). Plant Disease Detection and Classification Based on Rat Swarm Optimization using Deep Learning Approach [Cited by: 0]. *International Journal of Engineering Trends and Technology*, 71(7), 42-52. <https://doi.org/10.14445/22315381/IJETT-V71I7P204>
- Raja, D., & Karthikeyan, M. (2023b). Red Deer Optimization with Deep Learning Enabled Agricultural Plant Disease Detection and Classification Model [Cited by: 0; All Open Access, Bronze Open Access]. *International Journal of*

Intelligent Engineering and Systems, 16(5), 21-30. <https://doi.org/10.22266/ijies2023.1031.03>

- Rajeena P. P., F., S. U, A., Moustafa, M. A., & Ali, M. A. S. (2023). Detecting Plant Disease in Corn Leaf Using EfficientNet Architecture—An Analytical Approach [Cited by: 1; All Open Access, Gold Open Access]. *Electronics (Switzerland)*, 12(8). <https://doi.org/10.3390/electronics12081938>
- Rajpoot, V., Tiwari, A., & Jalal, A. S. (2023). Automatic early detection of rice leaf diseases using hybrid deep learning and machine learning methods [Cited by: 2]. *Multimedia Tools and Applications*, 82(23), 36091-36117. <https://doi.org/10.1007/s11042-023-14969-y>
- Ramcharan, A., McCloskey, P., Baranowski, K., Mbilinyi, N., Mrisho, L., & Ndalaha, M. (2023). & Hughes, DP (2019). A mobile-based deep learning model for cassava disease diagnosis. *Frontiers in plant science*, 10, 425916.
- Rangarajan, A. K., Purushothaman, R., & Ramesh, A. (2018). Tomato crop disease classification using pre-trained deep learning algorithm. *Procedia computer science*, 133, 1040-1047.
- Rao, D., Zhang, D., Lu, H., Yang, Y., Qiu, Y., Ding, M., & Yu, X. (2023). Deep learning combined with Balance Mixup for the detection of pine wilt disease using multispectral imagery [Cited by: 0]. *Computers and Electronics in Agriculture*, 208. <https://doi.org/10.1016/j.compag.2023.107778>
- Rehman, A., Saba, T., Kashif, M., Fati, S. M., Bahaj, S. A., & Chaudhry, H. (2022). A revisit of internet of things technologies for monitoring and control strategies in smart agriculture. *Agronomy*, 12(1), 127.
- SADER. (2023). La importancia de la producción de frijol en México. *Secretaría de Agricultura y Desarrollo Rural*.

- Sahu, S. K., & Pandey, M. (2023). Hybrid Xception transfer learning with crossover optimized kernel extreme learning machine for accurate plant leaf disease detection [Cited by: 0]. *Soft Computing*, 27(19), 13797-13811. <https://doi.org/10.1007/s00500-023-09048-1>
- Sawyer, E., Laroche-Pinel, E., Flasco, M., Cooper, M. L., Corrales, B., Fuchs, M., & Brillante, L. (2023). Phenotyping grapevine red blotch virus and grapevine leafroll-associated viruses before and after symptom expression through machine-learning analysis of hyperspectral images [Cited by: 0; All Open Access, Gold Open Access, Green Open Access]. *Frontiers in Plant Science*, 14. <https://doi.org/10.3389/fpls.2023.1117869>
- Selfa, J., & Anento, J. L. (1997). Plagas agrícolas y forestales. *Bol. SEA*, 20, 75-91.
- Sethy, P. K., Barpanda, N. K., Rath, A. K., & Behera, S. K. (2020). Deep feature based rice leaf disease identification using support vector machine. *Computers and Electronics in Agriculture*, 175, 105527.
- Shah, S. R., Qadri, S., Bibi, H., Shah, S. M. W., Sharif, M. I., & Marinello, F. (2023). Comparing Inception V3, VGG 16, VGG 19, CNN, and ResNet 50: A Case Study on Early Detection of a Rice Disease [Cited by: 2; All Open Access, Gold Open Access]. *Agronomy*, 13(6). <https://doi.org/10.3390/agronomy13061633>
- Shahoveisi, F., Taheri Gorji, H., Shahabi, S., Hosseinirad, S., Markell, S., & Vasefi, F. (2023). Application of image processing and transfer learning for the detection of rust disease [Cited by: 3; All Open Access, Gold Open Access, Green Open Access]. *Scientific Reports*, 13(1). <https://doi.org/10.1038/s41598-023-31942-9>
- Sharma, S., Kooner, R., & Arora, R. (2017). Insect pests and crop losses. *Breeding insect resistant crops for sustainable agriculture*, 45-66.

- Sharma, V., Tripathi, A. K., & Mittal, H. (2023). DLMC-Net: Deeper lightweight multi-class classification model for plant leaf disease detection [Cited by: 10]. *Ecological Informatics*, 75. <https://doi.org/10.1016/j.ecoinf.2023.102025>
- Shedthi B, S., Siddappa, M., Shetty, S., Shetty, V., & Suresh, R. (2023). Detection and classification of diseased plant leaf images using hybrid algorithm [Cited by: 0]. *Multimedia Tools and Applications*, 82(21), 32349-32372. <https://doi.org/10.1007/s11042-023-14751-0>
- Shoaib, M., Shah, B., Hussain, T., Ali, A., Ullah, A., Alenezi, F., Gechev, T., Ali, F., & Syed, I. (2022). A deep learning-based model for plant lesion segmentation, subtype identification, and survival probability estimation. *Frontiers in Plant Science*, 13, 1095547.
- Shrivastava, V. K., Shelke, C. J., Shrivastava, A., Mohanty, S. N., & Sharma, N. (2023). Optimized Deep Learning Model for Disease Prediction in Potato Leaves [Cited by: 0; All Open Access, Gold Open Access]. *EAI Endorsed Transactions on Pervasive Health and Technology*, 9(1). <https://doi.org/10.4108/eetpht.9.4001>
- Singh, D., Singh, S. K., Modi, A., Singh, P. K., Zhimo, V. Y., & Kumar, A. (2020). Impacts of agrochemicals on soil microbiology and food quality. En *Agrochemicals detection, treatment and remediation* (pp. 101-116). Elsevier.
- Singh, U. P., Chouhan, S. S., Jain, S., & Jain, S. (2019). Multilayer convolution neural network for the classification of mango leaves infected by anthracnose disease. *IEEE access*, 7, 43721-43729.
- Soares, A. d. S., Vieira, B. S., Bezerra, T. A., Martins, G. D., & Siquieroli, A. C. S. (2022). Early detection of coffee leaf rust caused by *Hemileia vastatrix* using multispectral images. *Agronomy*, 12(12), 2911.

- Terven, J., Córdova-Esparza, D.-M., & Romero-González, J.-A. (2023). A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine Learning and Knowledge Extraction*, 5(4), 1680-1716.
- Thai, H.-T., Le, K.-H., & Nguyen, N. L.-T. (2023). FormerLeaf: An efficient vision transformer for Cassava Leaf Disease detection [Cited by: 11]. *Computers and Electronics in Agriculture*, 204. <https://doi.org/10.1016/j.compag.2022.107518>
- Thenmozhi, K., & Reddy, U. S. (2019). Crop pest classification based on deep convolutional neural network and transfer learning. *Computers and Electronics in Agriculture*, 164, 104906.
- Towfek, S., & Khodadadi, N. (2023). Deep Convolutional Neural Network and Metaheuristic Optimization for Disease Detection in Plant Leaves [Cited by: 0]. *Journal of Intelligent Systems and Internet of Things*, 10(1), 66-75. <https://doi.org/10.54216/JISIoT.100105>
- Tudi, M., Daniel Ruan, H., Wang, L., Lyu, J., Sadler, R., Connell, D., Chu, C., & Phung, D. T. (2021). Agriculture development, pesticide application and its impact on the environment. *International journal of environmental research and public health*, 18(3), 1112.
- Tuncer, A. (2021). Cost-optimized hybrid convolutional neural networks for detection of plant leaf diseases. *Journal of Ambient Intelligence and Humanized Computing*, 12(8), 8625-8636.
- Türkoğlu, M., & Hanbay, D. (2019). Plant disease and pest detection using deep learning-based features. *Turkish Journal of Electrical Engineering and Computer Sciences*, 27(3), 1636-1651.

- Udeogu, C. U., Nwakanma, C. I., Ayoade, I. A., Amadi, C. S., & Eze, U. F. (2023). Agro-vision IoT-enabled Crop Pest Recognition System based on VGG-16. *2023 2nd International Conference on Multidisciplinary Engineering and Applied Science (ICMEAS)*, 1-5.
- Vélez, S., Ariza-Sentís, M., & Valente, J. (2023). Mapping the spatial variability of Botrytis bunch rot risk in vineyards using UAV multispectral imagery. *European Journal of Agronomy*, *142*, 126691.
- Vishnoi, V. K., Kumar, K., Kumar, B., Mohan, S., & Khan, A. A. (2023). Detection of Apple Plant Diseases Using Leaf Images Through Convolutional Neural Network [Cited by: 9; All Open Access, Gold Open Access]. *IEEE Access*, *11*, 6594-6609. <https://doi.org/10.1109/ACCESS.2022.3232917>
- Wei, X., Zhang, J., Conrad, A. O., Flower, C. E., Pinchot, C. C., Hayes-Plazolles, N., Chen, Z., Song, Z., Fei, S., & Jin, J. (2023). Machine learning-based spectral and spatial analysis of hyper- and multi-spectral leaf images for Dutch elm disease detection and resistance screening [Cited by: 0; All Open Access, Gold Open Access]. *Artificial Intelligence in Agriculture*, *10*, 26-34. <https://doi.org/10.1016/j.aiia.2023.09.003>
- Wu, G., Fang, Y., Jiang, Q., Cui, M., Li, N., Ou, Y., Diao, Z., & Zhang, B. (2023). Early identification of strawberry leaves disease utilizing hyperspectral imaging combing with spectral features, multiple vegetation indices and textural features [Cited by: 8]. *Computers and Electronics in Agriculture*, *204*. <https://doi.org/10.1016/j.compag.2022.107553>
- Xiao, Z., Shi, Y., Zhu, G., Xiong, J., & Wu, J. (2023). Leaf Disease Detection Based on Lightweight Deep Residual Network and Attention Mechanism [Cited by: 0; All Open Access, Gold Open Access]. *IEEE Access*, *11*, 48248-48258. <https://doi.org/10.1109/ACCESS.2023.3272985>

- Xie, X., Xia, F., Wu, Y., Liu, S., Yan, K., Xu, H., & Ji, Z. (2023). A Novel Feature Selection Strategy Based on Salp Swarm Algorithm for Plant Disease Detection [Cited by: 1; All Open Access, Gold Open Access, Green Open Access]. *Plant Phenomics*, 5. <https://doi.org/10.34133/plantphenomics.0039>
- Xue, Z., Xu, R., Bai, D., & Lin, H. (2023). YOLO-Tea: A Tea Disease Detection Model Improved by YOLOv5 [Cited by: 13; All Open Access, Gold Open Access]. *Forests*, 14(2). <https://doi.org/10.3390/f14020415>
- Yong, L. Z., Khairunniza-Bejo, S., Jahari, M., & Muharam, F. M. (2023). Automatic Disease Detection of Basal Stem Rot Using Deep Learning and Hyperspectral Imaging [Cited by: 7; All Open Access, Gold Open Access]. *Agriculture (Switzerland)*, 13(1). <https://doi.org/10.3390/agriculture13010069>
- Zeng, W., & Li, M. (2020). Crop leaf disease recognition based on Self-Attention convolutional neural network. *Computers and Electronics in Agriculture*, 172, 105341.
- Zhang, B., Zhou, C., Mao, J., Zhao, H., Rao, Z., & Gong, X. (2023). Application of Chlorophyll Fluorescence Dynamics Imaging in Leaf Blight Disease Diagnosis of Barbary Wolfberry by Implementing Machine Learning. *Russian Journal of Plant Physiology*, 70(4), 88.
- Zhang, C., Chen, T., Chen, W., & Sankaran, S. (2023). Non-invasive evaluation of Ascochyta blight disease severity in chickpea using field asymmetric ion mobility spectrometry and hyperspectral imaging techniques [Cited by: 2]. *Crop Protection*, 165. <https://doi.org/10.1016/j.cropro.2022.106163>
- Zhang, D., Huang, Y., Wu, C., & Ma, M. (2023). Detecting tomato disease types and degrees using multi-branch and destruction learning. *Computers and Electronics in Agriculture*, 213, 108244.

Apéndice A

codigo A

```
Repartir imagenes aleatoriamente.
*Definir la ruta de la carpeta principal* mainFolderPath = 'ruta de tu carpeta de imagenes';
*Leer todas las imágenes en la carpeta* imageFiles = dir(fullfile(mainFolderPath, '*.jpeg')); numImages = length(imageFiles);
*Barajar las imágenes aleatoriamente* rng('shuffle'); *Para asegurar que la distribución sea aleatoria cada vez que se ejecute el script* shuffledIndices =
randperm(numImages);
*Definir los números de imágenes para cada conjunto en este caso* 308,100,100 numTrain = 308; numTest = 100; numValidation = 100;
*Crear las carpetas de destino si no existen* trainFolder = fullfile(mainFolderPath, 'train'); testFolder = fullfile(mainFolderPath, 'test'); validationFolder =
fullfile(mainFolderPath, 'validation');
if exist(trainFolder, 'dir') mkdir(trainFolder); end
if exist(testFolder, 'dir') mkdir(testFolder); end
if exist(validationFolder, 'dir') mkdir(validationFolder); end
*Copiar las imágenes a las carpetas correspondientes* for i = 1:numTrain sourceFile = fullfile(imageFiles(shuffledIndices(i)).folder,
imageFiles(shuffledIndices(i)).name); destinationFile = fullfile(trainFolder, imageFiles(shuffledIndices(i)).name); copyfile(sourceFile, destinationFile); end
for i = numTrain+1:numTrain+numTest sourceFile = fullfile(imageFiles(shuffledIndices(i)).folder, imageFiles(shuffledIndices(i)).name); destinationFile =
fullfile(testFolder, imageFiles(shuffledIndices(i)).name); copyfile(sourceFile, destinationFile); end
for i = numTrain+numTest+1:numTrain+numTest+numValidation sourceFile = fullfile(imageFiles(shuffledIndices(i)).folder,
imageFiles(shuffledIndices(i)).name); destinationFile = fullfile(validationFolder, imageFiles(shuffledIndices(i)).name); copyfile(sourceFile, destinationFile);
end
disp('Las imágenes han sido divididas y copiadas a las carpetas correspondientes.');
```


Apéndice B

codigo B

```
import json
import cv2
import numpy as np
import os

def save_yolo_format(image_id, image_info, bounding_boxes, save_dir):
    txt_filename = os.path.join(save_dir, f"{image_info['file_name'].split('.')[0]}.txt")
    with open(txt_filename, 'w') as f:
        for bbox in bounding_boxes:
            class_id = bbox[0]
            x_center = bbox[1] / image_info['width']
            y_center = bbox[2] / image_info['height']
            width = bbox[3] / image_info['width']
            height = bbox[4] / image_info['height']
            f.write(f"{class_id} {x_center} {y_center} {width} {height}\n")

def load_coco_annotations(annotations_path, images_dir, save_dir):
    with open(annotations_path, 'r') as file:
        coco_data = json.load(file)

    # Group annotations by image_id
    annotations_by_image = {}
    for annotation in coco_data['annotations']:
        image_id = annotation['image_id']
        if image_id not in annotations_by_image:
            annotations_by_image[image_id] = []
        annotations_by_image[image_id].append(annotation)

    # Loop through images
    for image_info in coco_data['images']:
        image_id = image_info['id']

        if image_id not in annotations_by_image:
            print(f"Image Id:{image_id}, {image_info['file_name']}
                  does not have annotations.")
            continue # Skip images without annotations

        print(f"Image Id:{image_id}, {image_info['file_name']}
              has {len(annotations_by_image[image_id])} annotations.")

        image_path = os.path.join(images_dir, image_info['file_name'])
        image = cv2.imread(image_path)
        if image is None:
            print(f"Error loading image: {image_path}")
            continue
```

```

bounding_boxes = []

# Draw all annotations for this image
for annotation in annotations_by_image[image_id]:
    annotation_id = annotation["id"]
    # Assuming an extra level of nesting, iterate through each set of polygons
    for polygons in annotation['segmentation']:
        for polygon in polygons: # Now iterating over the actual polygons
            if not isinstance(polygon, list):
                print(f"Segmentation format error for image ID {image_id}.
                    Is not a list. Skipping...")
                continue

            try:
                # Reshape polygon to a 2D array where each row is a point
                poly = np.array(polygon).reshape((-1, 2))
                cv2.polylines(image, [poly.astype(np.int32)], isClosed=True,
                    color=np.random.randint(0, 256, size=3).tolist(), thickness=2)

                # Extract bounding box from polygon
                x, y, w, h = cv2.boundingRect(poly.astype(np.int32))
                cv2.rectangle(image, (x, y), (x+w, y+h), (0, 255, 0), 2)

                # Convert bounding box to YOLO format
                x_center = x + w / 2
                y_center = y + h / 2
                bounding_boxes.append((annotation['category_id'],
                    x_center, y_center, w, h))

            except Exception as e:
                print(f"Error processing segmentation for image ID {image_id},
                    ann ID {annotation_id}: {e}")
                continue

        # Save bounding boxes in YOLO format
        save_yolo_format(image_id, image_info, bounding_boxes, save_dir)

        # Display the image with all annotations
        cv2.imshow('Image with Masks and Bounding Boxes', image)
        cv2.waitKey(0)
    cv2.destroyAllWindows()

# Example usage
annotations_path = "tu dirección"
images_dir = "tu dirección"
save_dir = "tu dirección"
if not os.path.exists(save_dir):
    os.makedirs(save_dir)
load_coco_annotations(annotations_path, images_dir, save_dir)

```