e-ISSN: 2007-4034 / ISSN print: 1027-152X

Scientific article https://doi.org/10.5154/r.rchsh.2025.07.011

Disease detection in potato cultivation using deep learning models

Detección de enfermedades en el cultivo de papa mediante modelos de aprendizaje profundo

Abraham González-Jesús[1]; Juan Manuel González-Camacho[1]*; Juan Pablo Ambrosio-Ambrosio[1]; Luis Arturo Soriano-Avendaño[2]; José Alfredo Carrillo-Salazar[1]

[1]Colegio de Postgraduados-Campus Montecillo. Carretera México-Texcoco km 36.5, Montecillo, Texcoco, Estado de México, C. P. 56264, México.

[2]Universidad Autónoma Chapingo. Carretera México-Texcoco km 38.5, Chapingo, Texcoco, Estado de México, C. P. 56230, México.

*Corresponding author: jmgc@colpos.mx, tel. 595 95 20 200, ext. 1431, 1462.

Abstract

The potato is the fourth most important crop worldwide. In 2024, Mexico reached a production of 2.12 million tons, with an average yield of 32.71 t·ha^{-1}. The incidence of diseases is a limiting factor in ensuring production; therefore, the application of machine / computer vision techniques is of great value for their timely detection. The aim of this research was to implement and evaluate the performance of three deep learning models to identify six diseases in potato cultivation from digital images: late blight (*Phytophthora infestans*), early blight (*Alternaria solani*), bacterial infections (*Candidatus Phytoplasma*), *Potato virus Y*, potato mosaic virus X (*Potato virus X*), and *Potato leafroll virus*, as well as damage from pests and healthy leaves. The performance of a standard convolutional neural network (CNN) and two CNN models with transfer learning (VGG19 and YOLOv11n) was evaluated. The YOLOv11n model outperformed VGG19 and the standard CNN, achieving an accuracy of 98.1 %, compared to 92.8 and 90.1 %, respectively. At the class level, YOLOv11n obtained *F1-score* values of 95 to 100 %, while VGG19 recorded values of 85 to 100 % and CNN of 72 to 98 %. These results demonstrate that YOLOv11n exhibits a high classification capacity for detecting the six diseases analyzed, pest damage, and healthy leaves. The observed performance confirms the potential of deep learning models as a foundation for developing smart technologies aimed at detecting phytosanitary problems.

Keywords: convolutional neural networks, transfer learning, YOLOv11n, pattern recognition, early warning.

Resumen

La papa es el cuarto cultivo más importante a nivel mundial. En 2024, México alcanzó una producción de 2.12 millones de toneladas, con un rendimiento promedio de 32.71 t·ha^{-1}. La incidencia de enfermedades es un factor limitante para garantizar la producción; por ello, la aplicación de técnicas de visión artificial es de gran valor para su detección oportuna. El objetivo de esta investigación fue implementar y evaluar el desempeño de tres modelos de aprendizaje profundo para identificar seis enfermedades en el cultivo de papa a partir de imágenes digitales: tizón tardío (*Phytophthora infestans*), tizón temprano (*Alternaria solani*), infecciones bacterianas (*Candidatus Phytoplasma*), virus Y de la papa (*Potato virus Y*), virus del mosaico de la papa (*Potato virus X*) y virus del enrollado de la papa (*Potato leafroll virus*), así como daños por plagas y hojas sanas. Se evaluó el desempeño de una red neuronal convolucional (CNN) estándar y dos modelos CNN con transferencia de aprendizaje (VGG19 y YOLOv11n). El modelo YOLOv11n superó a VGG19 y a la CNN estándar, al alcanzar una precisión de 98.1 %, en comparación con 92.8 y 90.1 %, respectivamente. A nivel de clases, YOLOv11n obtuvo valores de *F1-score* de 95 a 100 %, mientras que VGG19 registró valores de 85 a 100 % y CNN de 72 a 98 %. Estos resultados evidencian que YOLOv11n presenta una capacidad de clasificación alta para detectar las seis enfermedades analizadas, daños por plagas y hojas sanas. El desempeño observado confirma el potencial de los modelos de aprendizaje profundo como base para el desarrollo de tecnologías inteligentes orientadas a detectar problemas fitosanitarios.

Palabras clave: redes neuronales convolucionales, transferencia de aprendizaje, YOLOv11n, reconocimiento de patrones, alerta temprana.

Please cite this article as follows (APA 7): González-Jesús, A., González-Camacho, J. M., Ambrosio-Ambrosio, J. P., Soriano-Avendaño, L. A., & Carrillo-Salazar, J. A. (2026). Disease detection in potato cultivation using deep learning models. *Revista Chapingo Serie Horticultura*, *32*, e25011. https://doi.org/10.5154/r.rchsh.2025.07.011

Revista Chapingo Serie Horticultura

Received: July 5, 2025 | **Accepted:** December 10, 2025 | **Published online:** May 14, 2026

https://revistas.chapingo.mx/horticultura/

Introduction

The potato (*Solanum tuberosum* L.) is a basic crop for global food security. According to the Food and Agriculture Organization of the United Nations (FAO, 2025), this tuber ranks fourth among the world's most important food crops and is produced in more than 150 countries. In 2024, potato production reached a volume of 2.12 million tons in Mexico, with a harvested area of 64 917 ha and an average yield of 32.71 t·ha^{-1}, which generated an estimated production value of 20 258 million pesos (MXN) (Servicio de Información Agroalimentaria y Pesquera [SIAP], 2024).

Plant health is one of the main challenges for sustainable potato production, due to the crop's susceptibility to various diseases that affect the leaves, tubers, and roots. Among the most frequent are early blight (*Alternaria solani*), late blight (*Phytophthora infestans*), *Potato virus Y*, potato mosaic virus (*Potato virus X*), *Potato leafroll virus*, bacterial infections, and other pathogens of phytosanitary importance (Charkowski et al., 2019). These diseases reduce the quality and yield of tubers, in addition to increasing production costs due to the intensive use of agrochemicals and the implementation of corrective management practices. Furthermore, pests and infectious agents decrease crop resistance and contribute to persistent production losses in subsequent growing seasons (Chakrabarti et al., 2022).

Timely detection of diseases in potato crops is a key strategy to minimize economic losses, optimize phytosanitary management, and reduce the indiscriminate use of fungicides. Traditionally, diagnosis is performed through visual inspection; however, in recent years, computer vision techniques have gained relevance due to their ability to process complex data quickly and accurately (Saleem & Chishti, 2021). Rahman et al. (2020) evaluated convolutional neural networks (CNNs) with transfer learning, such as VGG16, InceptionV3, and MobileNet, to classify eight rice crop diseases and pests, and reported a global classification accuracy (ACC) of 97.1 % with VGG16. Similarly, Ferentinos (2018) analyzed the performance of AlexNet, GoogLeNet, and VGGNet on images of 58 classes corresponding to 25 plant species (healthy or diseased) and obtained an ACC of 99.5 % with VGGNet.

In potato cultivation, Islam et al. (2019) implemented CNN models with transfer learning for the early detection of early blight and late blight and reported an ACC of 99.4 % with VGG16. Ghosh et al. (2023) compared VGG19, DenseNet121, and ResNet50 for the identification of potato foliar diseases and observed better performance with DenseNet121 (ACC of 97.92 %), followed by VGG19 and ResNet50. For their

Introducción

La papa (*Solanum tuberosum* L.) es un cultivo básico para la seguridad alimentaria global. De acuerdo con la *Food and Agriculture Organization of the United Nations* (FAO, 2025), este tubérculo ocupa el cuarto lugar entre los cultivos alimentarios más importantes del mundo y se produce en más de 150 países. En 2024, la producción de papa alcanzó un volumen de 2.12 millones de toneladas en México, con una superficie cosechada de 64 917 ha y un rendimiento promedio de 32.71 t·ha^{-1}, lo cual generó un valor de producción estimado en 20 258 millones de pesos (MXN) (Servicio de Información Agroalimentaria y Pesquera [SIAP], 2024).

La sanidad vegetal constituye uno de los principales desafíos para la producción sostenible de papa, debido a la susceptibilidad del cultivo a diversas enfermedades que afectan las hojas, los tubérculos y las raíces. Entre las más frecuentes se encuentran el tizón temprano (*Alternaria solani*), el tizón tardío (*Phytophthora infestans*), el virus Y de la papa (*Potato virus Y*), el virus del mosaico de la papa (*Potato virus X*), el virus del enrollado de la papa (*Potato leafroll virus*), las infecciones bacterianas y otros patógenos de importancia fitosanitaria (Charkowski et al., 2019). Estas enfermedades reducen la calidad y el rendimiento de los tubérculos, además de incrementar los costos de producción debido al uso intensivo de agroquímicos y a la implementación de prácticas de manejo correctivo. Asimismo, las plagas y agentes infecciosos disminuyen la resistencia del cultivo y favorecen la persistencia de pérdidas productivas en ciclos agrícolas subsecuentes (Chakrabarti et al., 2022).

La detección oportuna de enfermedades en el cultivo de papa constituye una estrategia clave para minimizar pérdidas económicas, optimizar el manejo fitosanitario y reducir el uso indiscriminado de fungicidas. Tradicionalmente, el diagnóstico se realiza mediante inspección visual; sin embargo, en años recientes, las técnicas de visión por computadora han adquirido relevancia debido a su capacidad para procesar datos complejos de manera rápida y precisa (Saleem & Chishti, 2021). Rahman et al. (2020) evaluaron redes neuronales convolucionales (CNN, por sus siglas en inglés) con transferencia de aprendizaje, como VGG16, InceptionV3 y MobileNet, para clasificar ocho enfermedades y plagas del cultivo de arroz, y reportaron una precisión global de clasificación (ACC, del inglés *accuracy*) de 97.1 % con VGG16. De manera similar, Ferentinos (2018) analizó el desempeño de AlexNet, GoogLeNet y VGGNet en imágenes de 58 clases correspondientes a 25 especies vegetales (sanas o enfermas) y obtuvo una ACC de 99.5 % con VGGNet.

En el cultivo de papa, Islam et al. (2019) implementaron modelos CNN con transferencia de aprendizaje para

part, Wang et al. (2022) proposed an improved version of YOLOv5 (You Only Look Once) to classify diseases in peanut leaves, with a reduction in the number of parameters and improvements in both speed (11.8 %) and accuracy (3.98 %), with which they obtained an ACC of 92.57 %.

Aldakheel et al. (2024) used YOLOv4 with the *PlantVillage* dataset and achieved an ACC of 99.99 %, higher than that observed with DenseNet and AlexNet. Similarly, Srivastava et al. (2024) used YOLOv7 for the classification and detection of diseases in potato leaves and obtained an ACC of 98.1 %. In this context, the YOLOv11n architecture improves the classification speed and accuracy of its predecessors; furthermore, it facilitates the early detection of crop diseases, as well as its practical application in agricultural settings (Ultralytics Inc., 2024).

Based on the above, the objective of this work was to implement and evaluate the performance of three deep learning models from digital images: a standard CNN and two CNN models with transfer learning (VGG19 and YOLOv11n), for the identification of eight target classes in potato cultivation. The classes included late blight, early blight, bacterial infections (*Candidatus Phytoplasma*), potato virus Y, potato mosaic virus, potato leafroll virus, pest damage, and healthy leaves.

Materials and methods

Dataset

The dataset consisted of 6 153 images, distributed into eight classes corresponding to diseases, pest damage, and healthy leaves of the potato crop (Table 1). The images were stored in JPG format.

In order to standardize data entry and facilitate the model optimization process, all images were resized to 224 × 224 pixels, with a resolution of 96 pixels per inch (ppi) (Figure 1).

Hardware and software

The deep learning models were implemented on a workstation equipped with a 12th generation Intel Core i9-12900F processor, with a frequency of 2.40 GHz and multi-core architecture, which optimized parallel processing during model training. The system included 128 GB of RAM and an NVIDIA 4090Ti graphics card.

The algorithms were developed in Python version 3.9.18. TensorFlow 2.10 was used as the *backend* for the implementation of the standard CNN and the VGG19 model, and Keras 2.10 as the high-level interface. The YOLOv11n model was implemented

la detección temprana de tizón temprano y tizón tardío, y reportaron una ACC de 99.4 % con VGG16. Ghosh et al. (2023) compararon VGG19, DenseNet121 y ResNet50 para la identificación de enfermedades foliares de papa y observaron un mejor desempeño con DenseNet121 (ACC de 97.92 %), seguido por VGG19 y ResNet50. Por su parte, Wang et al. (2022) propusieron una versión mejorada de YOLOv5 (*You Only Look Once*) para clasificar enfermedades en hojas de maní, con una reducción en la cantidad de parámetros y mejoras tanto en velocidad (11.8 %) como en precisión (3.98 %), con lo cual obtuvieron una ACC de 92.57 %.

Aldakheel et al. (2024) emplearon YOLOv4 con el conjunto de datos *PlantVillage* y alcanzaron una ACC de 99.99 %, superior al observado con DenseNet y AlexNet. Asimismo, Srivastava et al. (2024) utilizaron YOLOv7 para la clasificación y detección de enfermedades en hojas de papa y obtuvieron una ACC de 98.1 %. En este contexto, la arquitectura YOLOv11n mejora la velocidad y precisión de clasificación de sus predecesores; además, facilita la detección temprana de enfermedades en cultivos, así como su aplicación práctica en entornos agrícolas (Ultralytics Inc., 2024).

Con base en lo anterior, el objetivo de este trabajo fue implementar y evaluar el desempeño de tres modelos de aprendizaje profundo a partir de imágenes digitales: una CNN estándar y dos modelos CNN con transferencia de aprendizaje (VGG19 y YOLOv11n), para la identificación de ocho clases objetivo en el cultivo de papa. Las clases incluyeron tizón tardío, tizón temprano, infecciones bacterianas (*Candidatus Phytoplasma*), virus Y de la papa, virus del mosaico de la papa, virus del enrollado de la papa, daños por plagas y hojas sanas.

Materiales y métodos

Conjunto de datos

El conjunto de datos estuvo conformado por 6 153 imágenes, distribuidas en ocho clases correspondientes a enfermedades, daños por plagas y hojas sanas del cultivo de papa (Cuadro 1). Las imágenes se almacenaron en formato JPG.

Con el propósito de estandarizar la entrada de datos y facilitar el proceso de optimización de los modelos, todas las imágenes se redimensionaron a 224 × 224 píxeles, con una resolución de 96 píxeles por pulgada (ppp) (Figura 1).

Hardware y software

Los modelos de aprendizaje profundo se implementaron en una estación de trabajo equipada con un procesador Intel Core i9-12900F de 12ª generación, con una

Table 1. Information on the images collected from the literature.
Cuadro 1. Información de las imágenes recopiladas de la literatura.

Scientific name / Nombre científico	Target classes / Clases objetivo	Images / Imágenes	Source / Fuente
Candidatus Phytoplasma	BI / BA	1 274	Shabrina et al. (2024)
-	PD / PO	602	
Potato virus Y (cracking)	PY	442	Raigonda et al. (2022)
Potato virus X	MV / VM	444	
Potato leafroll virus	PV	273	
-	HL / HS	828	Hughes & Salathe (2015), Laizer et al. (2023), Raigonda et al. (2022), Shabrina et al. (2024), Tilahun (2022)
Phytophthora infestans	LB / TO	1 240	
Alternaria solani	EB / TP	1 050	

BI: bacterial infection; PD: pest damage; PY: potato virus Y; MV: mosaic virus; PV: potato leafroll virus; HL: healthy leaf; LB: late blight; EB: early blight.

BA: infección bacteriana; PO: daños por plagas; PY: virus Y de la papa; VM: virus del mosaico; PV: virus del enrollamiento de la hoja de papa; HS: hoja sana; TO: tizón tardío; TP: tizón temprano.

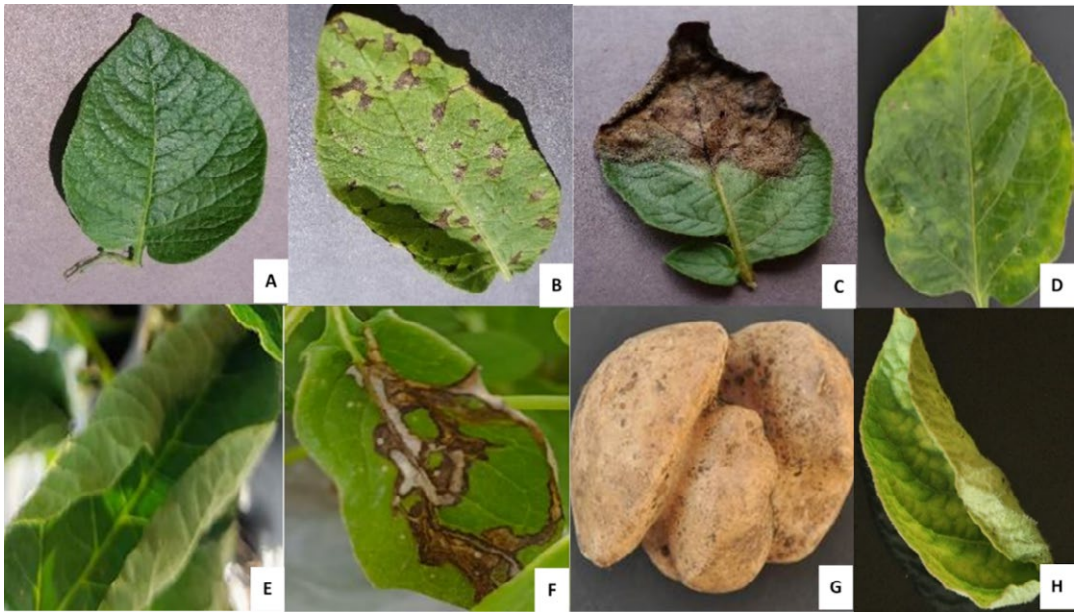


Figure 1. Illustration of the eight target classes: A) healthy leaf, B) early blight (*Alternaria solani*), C) late blight (*Phytophthora infestans*), D) bacterial infection (*Candidatus Phytoplasma*), E) *Potato virus Y*, F) pest damage, G) mosaic virus (*Potato virus X*) and H) *Potato leafroll virus*.

Figura 1. Ilustración de las ocho clases objetivo: A) hoja sana, B) tizón temprano (*Alternaria solani*), C) tizón tardío (*Phytophthora infestans*), D) infección bacteriana (*Candidatus Phytoplasma*), E) virus Y de la papa (*Potato virus Y*), F) daño por plagas, G) virus del mosaico (*Potato virus X*) y H) virus de enrollamiento de la hoja de papa (*Potato leafroll virus*).

in PyTorch. NumPy 1.26.4 was used for numerical operations, and Matplotlib 3.8.4 for chart visualization.

Convolutional neural network

A CNN consists of three main elements: convolution layers, pooling layers, and fully connected layers. The process begins with the input of an image represented as a three-dimensional array with width, height, and color channel values. Filters (e.g., 3 × 3) are applied to this matrix, representing hyperparameters that are optimized during model training. These filters

frecuencia de 2.40 GHz y arquitectura multinúcleo, lo cual permitió optimizar el procesamiento en paralelo durante el entrenamiento de los modelos. El sistema contó con 128 GB de memoria RAM y una tarjeta gráfica NVIDIA 4090Ti.

Los algoritmos se desarrollaron en Python versión 3.9.18. Para la implementación de la CNN estándar y del modelo VGG19 se utilizaron TensorFlow 2.10 como *backend* y Keras 2.10 como interfaz de alto nivel. El modelo YOLOv11n se implementó en PyTorch. Asimismo, se emplearon las librerías Numpy

allow for the extraction of relevant features through the dot product between the input pixel vectors and the corresponding filter (Dong et al., 2020).

The filtering process generates a feature map (tensor) that highlights texture and edge details (Figure 2). Subsequently, a pooling operation is applied to reduce the tensor's dimensionality and retain the most relevant information. For example, *max pooling* divides the input into regions and selects the maximum value from each region, while *average pooling* calculates the average value for each region. The pooling method is a hyperparameter that influences the model's performance.

Finally, the extracted features are transformed into a one-dimensional vector that is used as input to a fully connected layer of neurons. In the last layer, the network learns classification patterns from the extracted features and assigns the image to one of the target classes (Dong et al., 2020).

In this study, the CNN received 224 × 224 pixel RGB images as input. The architecture included three convolutional blocks with 32, 64, and 128 filters (3 × 3), each followed by a 2 × 2 *max pooling* layer. Next, the feature maps of the last convolutional layer were flattened to form a one-dimensional vector, which

1.26.4 para operaciones numéricas y Matplotlib 3.8.4 para visualizar gráficos.

Red neuronal convolucional

Una CNN se compone de tres elementos principales: capas de convolución, capas de agrupación y capas completamente conectadas. El proceso inicia con la entrada de una imagen representada como una matriz tridimensional con valores de ancho, alto y canales de color. A esta matriz se le aplican filtros (por ejemplo, 3 × 3), los cuales representan hiperparámetros que se optimizan durante el entrenamiento del modelo. Estos filtros permiten extraer características relevantes mediante el producto escalar entre los vectores de píxeles de entrada y el filtro correspondiente (Dong et al., 2020).

El proceso de filtrado genera un mapa de características (tensor) que resalta detalles de texturas y bordes (Figura 2). Posteriormente, se aplica una operación de agrupación para reducir la dimensión del tensor y conservar la información más relevante. Por ejemplo, *max pooling* divide la entrada en regiones y selecciona el valor máximo de cada región, mientras que *average pooling* calcula el valor medio de cada región. El método de agrupación constituye un hiperparámetro que influye en el desempeño del modelo.

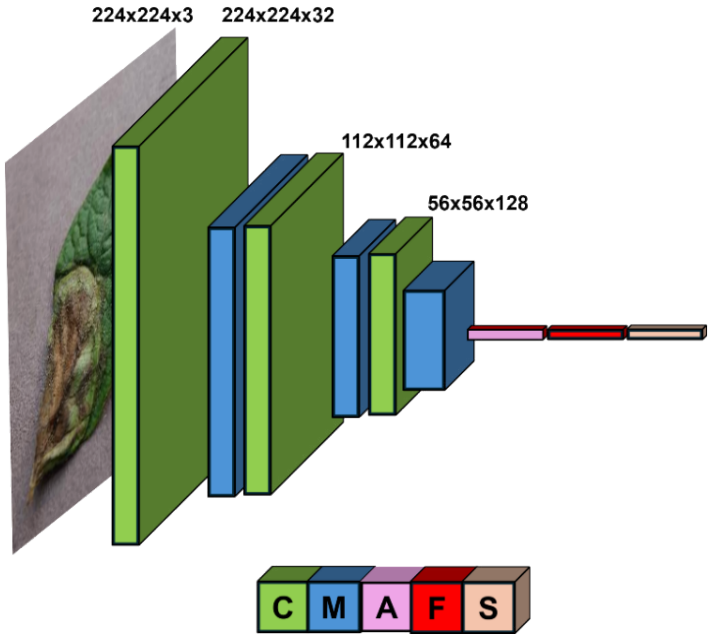


Figure 2. Architecture of the convolutional neural network (CNN). C: convolutional layers; M: *max pooling* layers; A: flattening layer; F: fully connected layer with ReLU activation function; S: output layer with *softmax* activation function for classifying the eight target classes in potato cultivation. Source: Own elaboration.

Figura 2. Arquitectura de la red neuronal convolucional (CNN). C: capas convolucionales; M: capas de agrupamiento *max pooling*; A: capa de aplanado; F: capa completamente conectada con función de activación ReLU; S: capa de salida con función de activación *softmax* para la clasificación de las ocho clases objetivo en el cultivo de papa. Fuente: Elaboración propia.

fed into a hidden dense layer of 256 neurons with ReLU (*Rectified Linear Unit*) activation function. Finally, the output layer included eight neurons with *softmax* activation function for multiclass classification.

VGG19 model

VGG19 is a CNN with transfer learning, developed by the Visual Geometry Group at the University of Oxford. This model is characterized by its depth and simplicity, with convolution layers that consistently use 3 × 3 size filters throughout the network (Figure 3). VGG19 contains 19 parameterized layers, of which 16 are convolutional layers, and three are fully connected layers. Each pair of convolutional layers is accompanied by a *max pooling* (2 × 2) layer, whose function is to reduce spatial dimensionality and facilitate the extraction of scale-invariant features (Simonyan & Zisserman, 2014)

At the end of the convolutional layers, the extracted features are flattened and connected to fully connected layers with 4096 neurons and a ReLU activation function. The output layer is adjusted to the target number of classes, and the *softmax* function is applied to estimate the membership probabilities for each class (Simonyan & Zisserman, 2014).

Finalmente, las características extraídas se transforman en un vector unidimensional que se utiliza como entrada en una capa de neuronas completamente conectada. En la última capa, la red aprende patrones de clasificación a partir de las características extraídas y asigna la imagen a una de las clases objetivo (Dong et al., 2020).

En este estudio, la CNN recibió como entrada imágenes RGB de 224 × 224 píxeles. La arquitectura incluyó tres bloques convolucionales con 32, 64 y 128 filtros de 3 × 3, cada uno seguido de una capa de *max pooling* de 2 × 2. Después, los mapas de características de la última capa convolucional se aplanaron para formar un vector unidimensional, el cual alimentó una capa densa oculta de 256 neuronas con función de activación ReLU (*Rectified linear unit*). Finalmente, la capa de salida incluyó ocho neuronas con función de activación *softmax* para la clasificación multiclase.

Modelo VGG19

VGG19 es una CNN con transferencia de aprendizaje, desarrollada por el *Visual Geometry Group* de la Universidad de Oxford. Este modelo se caracteriza por su profundidad y simplicidad, con capas de convolución que utilizan filtros de tamaño 3 × 3 de forma consistente a lo largo de la red (Figura 3). VGG19 contiene 19 capas

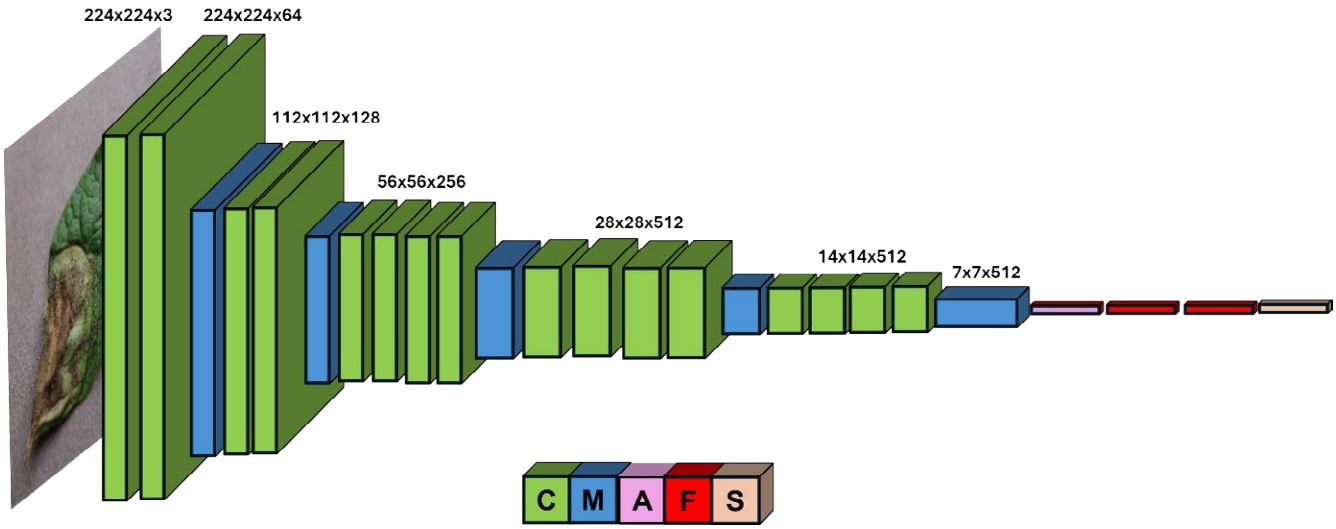


Figure 3. Architecture of the VGG19 convolutional neural network used in this study. C: convolutional layers; M: *max pooling* layers; A: flattening layer; F: fully connected layer with ReLU activation function; S: output layer with *softmax* activation function for classifying the eight target classes in potato cultivation. Source: Own elaboration.

Figura 3. Arquitectura de la red neuronal convolucional VGG19 utilizada en este estudio. C: capas convolucionales; M: capas de agrupamiento *max pooling*; A: capa de aplanado; F: capa completamente conectada con función de activación ReLU; S: capa de salida con función de activación *softmax* para la clasificación de las ocho clases objetivo en el cultivo de papa. Fuente: Elaboración propia.

In this work, VGG19 was used with frozen convolutional and pooling layers (*conv and max pooling*) to leverage its learning transfer characteristics. Subsequently, a flattening layer and two dense layers with ReLU activation function were added. The classifier was finalized with an output layer of eight neurons and *softmax* activation function, which allowed for obtaining the probabilities associated with each target class.

YOLO model

YOLO is a family of models designed for real-time object detection and classification. It processes entire images in a single pass using a CNN capable of predicting the location, size of objects, and the probability of belonging to each class. This approach overcomes the limitations of traditional two-stage methods and offers a precise and efficient solution (Redmon et al., 2016).

The YOLO architecture is structured in three main modules: i) the *backbone* transforms the input image into feature maps of different scales using convolutional networks, ii) the *neck* connects the *backbone* to the prediction stage to refine and merge features at various scales, and iii) the *head* generates predictions of location and classification of objects from the extracted feature maps.

YOLOv11 incorporates significant improvements in computer vision tasks, with increased accuracy in identifying and locating objects in complex scenes (Viso.ai, 2024). Its architecture reduces the number of parameters by 22 % compared to YOLOv8 (Ultralytics Inc., 2024). The nano version (YOLOv11n) was designed for applications on devices with limited computing resources by substantially reducing the number of parameters and operations, which allows for faster inference and lower memory consumption without sacrificing accuracy (Khanam et al., 2024).

In this study, YOLOv11n was configured with the default parameters of the Ultralytics library (Ultralytics Inc., 2024), with two modifications: the input was set to 224 × 224-pixel images, and the *early stopping* time was set to 10 training cycles. Among the preserved hyperparameters were the batch size of 16, the number of training cycles of 100, and the initial learning rate (λ) of 0.01.

Performance metrics

The performance of the CNN, VGG19, and YOLOv11n models was evaluated using the global classification accuracy (*ACC*), precision (*P*), sensitivity (*S*), *F1-score*, and *F1-macro* metrics. These metrics were calculated from a confusion matrix constructed with the test dataset, which describes the relationship between the observed and predicted classes.

con parámetros, de las cuales 16 corresponden a capas convolucionales y tres a capas completamente conectadas. Cada par de capas convolucionales se acompaña de una capa de agrupamiento *max pooling* (2 × 2), cuya función es reducir la dimensionalidad espacial y favorecer la extracción de características invariantes a la escala (Simonyan & Zisserman, 2014).

Al final de las capas convolucionales, las características extraídas se aplanan y se conectan a capas completamente conectadas con 4096 neuronas y función de activación ReLU. La capa de salida se ajusta al número de clases objetivo y se aplica la función *softmax* para estimar las probabilidades de pertenencia de cada clase (Simonyan & Zisserman, 2014).

En este trabajo, VGG19 se utilizó con las capas convolucionales y de agrupamiento (*conv* y *max pooling*) congeladas para aprovechar las características de transferencia de aprendizaje. Posteriormente, se agregó una capa de aplanado y dos capas densas con función de activación ReLU. El clasificador finalizó con una capa de salida de ocho neuronas y función de activación *softmax*, que permitió obtener las probabilidades asociadas a cada clase objetivo.

Modelo YOLO

YOLO corresponde a una familia de modelos diseñada para la detección y clasificación de objetos en tiempo real. Procesa imágenes completas en una sola pasada mediante una CNN capaz de predecir la ubicación, el tamaño de los objetos y las probabilidades de pertenencia de cada clase. Este enfoque supera las limitaciones de los métodos tradicionales de dos etapas y ofrece una solución precisa y eficiente (Redmon et al., 2016).

La arquitectura YOLO se estructura en tres módulos principales: i) la red troncal (*backbone*) transforma la imagen de entrada en mapas de características de distintas escalas mediante redes convolucionales, ii) el cuello (*neck*) conecta la red troncal con la etapa de predicción para refinar y fusionar las características en varias escalas, y iii) el frente (*head*) genera las predicciones de localización y clasificación de objetos a partir de los mapas de características extraídas.

YOLOv11 incorpora mejoras significativas en tareas de visión por computadora, con incrementos en precisión para identificar y localizar objetos en escenas complejas (Viso.ai, 2024). Su arquitectura reduce el número de parámetros en 22 % con respecto a YOLOv8 (Ultralytics Inc., 2024). La versión nano (YOLOv11n) se diseñó para aplicaciones en dispositivos con recursos computacionales limitados, al disminuir de forma sustancial el número de parámetros y operaciones, lo cual permite una inferencia más rápida y un menor consumo de memoria, sin sacrificar precisión (Khanam et al., 2024).

In the binary case, the confusion matrix consists of true positives (*TP*: samples that are in the positive class and are classified as positive), true negatives (*TN*: samples that are in the negative class and are classified as negative), false positives (*FP*: samples that belong to the negative class and are classified as positive), and false negatives (*FN*: samples that belong to the positive class but are classified as negative) (Raschka & Mirjalili, 2019). From these values, the metrics used to evaluate the classifiers' performance were calculated:

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

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

$$F1 - score = \frac{2 \times P \times E}{P+E} \quad (3)$$

$$F1 - macro = \frac{1}{C}\sum_{i=1}^{C} F1_i \quad (4)$$

where *C* is the total number of target classes.

$$ACC = \frac{TP+TN}{TP+FP+TN+FN} \quad (5)$$

Training and testing of classifiers

The images were organized into a main folder with subfolders corresponding to each of the eight target classes. Of the total dataset, 10 % was allocated for testing and 90 % for cross-validation (k = 5). Additionally, the files were renamed to facilitate their identification and management during model training and evaluation.

Prior to training and testing the models, different partitioning ratios of the dataset were evaluated (80:20 and 70:30). The 90:10 configuration showed greater stability in performance metrics and better generalizability, so it was adopted as the final scheme in this study.

Hyperparameter selection

The optimization of the CNN model hyperparameters was performed using the *grid search* technique, which consists of exploring the possible combinations of a set of values and selecting the one that maximizes the target metric (in this case, ACC).

For the VGG19 model, a random hyperparameter search was applied using the *Keras Tuner* procedure, which allows exploring a subset of possible combinations to identify the combination of values that maximizes ACC. The adjusted hyperparameters included the number of neurons in the dense layer (*nn*), the *dropout* rate (*d*), the learning rate (λ), and the early stopping criterion, to avoid overfitting the model.

En este estudio, YOLOv11n se configuró con los parámetros predeterminados de la librería Ultralytics (Ultralytics Inc., 2024), con dos modificaciones: la entrada se estableció en imágenes de 224 × 224 píxeles y la detención temprana del clasificador (*early stopping*) se fijó en 10 ciclos de entrenamiento. Entre los hiperparámetros conservados destacan el tamaño de lote de 16, el número de ciclos de entrenamiento de 100 y la tasa de aprendizaje inicial (λ) de 0.01.

Métricas de desempeño

El desempeño de los modelos CNN, VGG19 y YOLOv11n se evaluó mediante las métricas de precisión global de clasificación (*ACC*), precisión (*P*), sensibilidad (*S*), *F1-score* y *F1-macro*. Estas métricas se calcularon a partir de una matriz de confusión construida con el conjunto de datos de prueba, la cual describe la relación entre las clases observadas y las predichas por el clasificador.

En el caso binario, la matriz de confusión se compone de verdaderos positivos (*TP*: muestras que son de la clase positiva y que se clasifican como positivas), verdaderos negativos (*TN*: muestras que son de la clase negativa y que se clasifican como negativas), falsos positivos (*FP*: muestras que pertenecen a la clase negativa y que se clasifican como positivas) y falsos negativos (*FN*: muestras que pertenecen a la clase positiva, pero que se clasifican como negativas) (Raschka & Mirjalili, 2019). A partir de estos valores se calcularon las métricas empleadas para evaluar el desempeño de los clasificadores:

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

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

$$F1 - score = \frac{2 \times P \times E}{P+E} \quad (3)$$

$$F1 - macro = \frac{1}{C}\sum_{i=1}^{C} F1_i \quad (4)$$

donde *C* es el número total de clases objetivo.

$$ACC = \frac{TP+TN}{TP+FP+TN+FN} \quad (5)$$

Entrenamiento y prueba de los clasificadores

Las imágenes se organizaron en una carpeta principal con subcarpetas correspondientes a cada una de las ocho clases objetivo. Del conjunto total de datos, se asignó 10 % para prueba y 90 % para validación cruzada (k = 5). Adicionalmente, los archivos se renombraron para facilitar su identificación y gestión durante el entrenamiento y la evaluación de los modelos.

Previo al entrenamiento y prueba de los modelos, se evaluaron diferentes proporciones de partición del

The YOLOv11n model was trained with the default hyperparameter values and early stopping criterion. This choice was based on the model's robust and previously optimized design, which incorporates configurations suitable for classification tasks without requiring extensive hyperparameter searching.

The application of different adjustment strategies responded to the particularities of each architecture and its optimization requirements.

Results and discussion

Optimal hyperparameters

Hyperparameter optimization using *k*-fold cross-validation ($k = 5$) and grid search allowed us to identify the best-performing CNN configuration. The model achieved optimal performance with 32, 64, and 128 filters in the three convolutional layers, a fully connected dense layer with 256 neurons, $\lambda = 1\times10^{-3}$, and a ReLU activation function. This configuration favored the extraction of discriminant features and allowed for stable convergence during training.

In the case of the VGG19 architecture, the optimal hyperparameters included 896 neurons in the first dense layer, $d = 0.5$, and $\lambda = 1\times10^{-3}$. This combination provided a suitable balance between generalizability and overfitting prevention, as well as ensuring stability during the optimization process.

To ensure reproducibility of the results, the YOLOv11n model was trained using default values, with two specific adjustments: the input image size was set to 224 pixels, and the *early stopping* criterion was set to 10 epochs. Among the parameters retained were the number of epochs (*epochs* = 100), batch size (*batch* = 16), the initial optimizer settings (*learning rate* = 0.01, *momentum* = 0.937, and *weight decay* = 0.0005), as well as the activation of validation mode (*val* = *True*), the use of a fixed random seed (*seed* = 0), and execution in deterministic mode (*deterministic* = *True*).

Learning curves

The CNN learning curves showed that the ACC in training and validation stabilized around 95 %. The validation curve remained slightly below the training curve, indicating a good fit of the model (Figure 4A). For its part, the VGG19 achieved better ACC in training and validation in epoch 13 (Figure 4B). From that point on, accuracy in validation remained stable, while in training, it showed a slight increase.

Confusion matrices

At the target class level, CNN showed its lowest performance in the classification of pest damage (PD),

conjunto de datos (80:20 y 70:30). La configuración 90:10 mostró mayor estabilidad en las métricas de desempeño y una mejor capacidad de generalización, por lo que se adoptó como esquema definitivo en este estudio.

Selección de hiperparámetros

La optimización de los hiperparámetros del modelo CNN se realizó mediante la técnica de búsqueda por rejilla (*grid search*), que consiste en explorar las combinaciones posibles de un conjunto de valores y seleccionar la que maximice la métrica objetivo (en este caso la ACC).

Para el modelo VGG19 se aplicó una búsqueda aleatoria de hiperparámetros con el procedimiento *Keras Tuner*, el cual permite explorar un subconjunto de combinaciones posibles para identificar la combinación de valores que maximice la ACC. Los hiperparámetros ajustados incluyeron el número de neuronas en la capa densa (*nn*), la tasa de *dropout* (*d*), la tasa de aprendizaje (λ) y el criterio de detención temprana, con el objetivo de evitar el sobreajuste del modelo.

El modelo YOLOv11n se entrenó con los valores predeterminados de los hiperparámetros y el criterio de detención temprana. Esta elección se basó en el diseño robusto y previamente optimizado del modelo, que incorpora configuraciones adecuadas para tareas de clasificación, sin requerir una búsqueda extensiva de hiperparámetros.

La aplicación de diferentes estrategias de ajuste respondió a las particularidades de cada arquitectura y a sus requerimientos de optimización.

Resultados y discusión

Hiperparámetros óptimos

La optimización de hiperparámetros mediante validación cruzada *k*-fold ($k = 5$) y búsqueda por rejilla permitió identificar la configuración con mejor desempeño para la CNN. El modelo alcanzó su rendimiento óptimo con 32, 64 y 128 filtros en las tres capas convolucionales, una capa densa completamente conectada con 256 neuronas, $\lambda = 1\times10^{-3}$ y función de activación ReLU. Esta configuración favoreció la extracción de características discriminantes y permitió una convergencia estable durante el entrenamiento.

En el caso de la arquitectura VGG19, los hiperparámetros óptimos incluyeron 896 neuronas en la primera capa densa, $d = 0.5$ y $\lambda = 1\times10^{-3}$. Esta combinación proporcionó un equilibrio adecuado entre la capacidad de generalización y la prevención del sobreajuste, además de asegurar estabilidad durante el proceso de optimización.

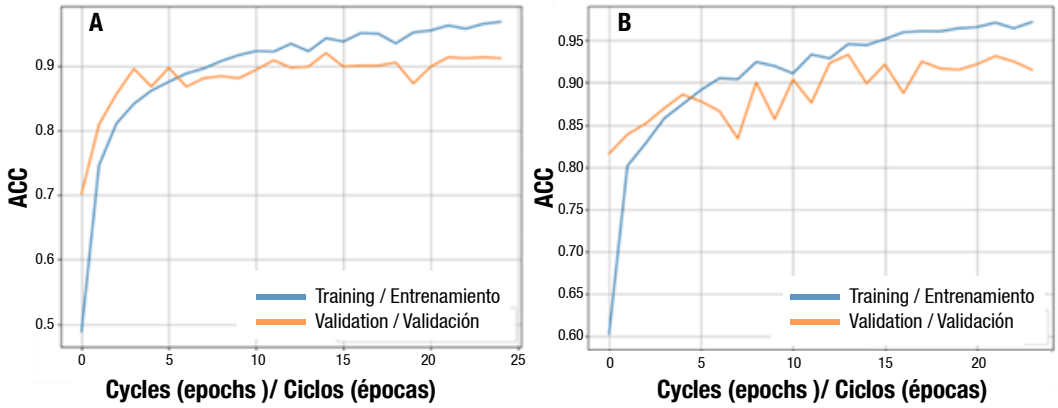


Figure 4. Learning curves (ACC vs epochs) during training and validation of convolutional neural networks (CNNs): A) Standard CNN and B) VGG19. ACC: overall accuracy.

Figura 4. Curvas de aprendizaje durante el entrenamiento y la validación de las redes neuronales convolucionales (CNN): A) CNN estándar y B) VGG19. ACC: precisión global de clasificación.

while achieving its highest accuracy in the identification of early blight (EB). The main classification errors corresponded to model confusion between PD and bacterial infection (BI) or mosaic virus (MV), as well as less frequent confusion between EB and late blight (LB) (Figure 5A). The VGG19 model showed confusion between PD, BI, and healthy leaf (HL), as well as reverse errors between HL and PD. Classification errors between MV and PD were also identified. Despite these occasional confusions, the model showed a high level of accuracy in the remaining classes (Figure 5B).

Performance of the YOLOv11n model

The confusion matrix of the YOLOv11n model showed outstanding performance, with accuracy values close to 100 % in most target classes. Unlike CNN and VGG19, YOLOv11n identified the PD with greater reliability. However, this class exhibited some classification errors, attributable to the high visual variability of the samples and their limited representation. The difference between the class with the lowest and highest accuracy was only 7 %, indicating balanced performance among the target classes (Figure 6).

During the cross-validation procedure ($k = 5$), YOLOv11n consistently outperformed the CNN and VGG19 in terms of ACC (Figure 7).

Performance in prediction with unseen test data

The optimized CNN, VGG19, and YOLOv11n models were evaluated using the test dataset to estimate their performance in unseen scenarios. The global ACC and F1-macro metrics for each model were similar, indicating low sensitivity to class size imbalances. However, the YOLOv11n model outperformed CNN and VGG19 in the evaluated global metrics (Table 2).

Para garantizar la reproducibilidad de los resultados, el entrenamiento del modelo YOLOv11n se realizó con los valores predeterminados, con dos ajustes específicos: el tamaño de imagen de entrada se fijó en 224 píxeles y el criterio de detención temprana se estableció en 10 ciclos. Entre los parámetros conservados destacaron el número de épocas (*epochs* = 100), el tamaño de lote (*batch* = 16), la configuración inicial del optimizador (*learning rate* = 0.01, *momentum* = 0.937 y *weight decay* = 0.0005), así como la activación del modo de validación (*val* = *True*), el uso de una semilla aleatoria fija (*seed* = 0) y la ejecución en modo determinístico (*deterministic* = *True*).

Curvas de aprendizaje

Las curvas de aprendizaje de la CNN mostraron que la ACC en el entrenamiento y la validación se estabilizó alrededor del 95 %. La curva de validación se mantuvo ligeramente por debajo de la curva de entrenamiento, lo cual indica un ajuste adecuado del modelo (Figura 4A). Por su parte, el VGG19 alcanzó mejor ACC en entrenamiento y validación en el ciclo 13 (Figura 4B). A partir de ese punto, la precisión en validación permaneció estable, mientras que en entrenamiento presentó un leve incremento.

Matrices de confusión

A nivel de clases objetivo, la CNN presentó su menor desempeño en la clasificación de daños por plagas (PO), mientras que alcanzó su mayor precisión en la identificación del tizón temprano (TP). Los principales errores de clasificación correspondieron a confusiones del modelo entre PO e infección bacteriana (BA) o virus del mosaico (VM), así como confusiones de menor frecuencia entre TP y tizón tardío (TO) (Figura 5A). El modelo VGG19 presentó confusión de PO con BA y hoja sana (HS), así como errores inversos de HS con PO. Asimismo, se identificaron errores de clasificación

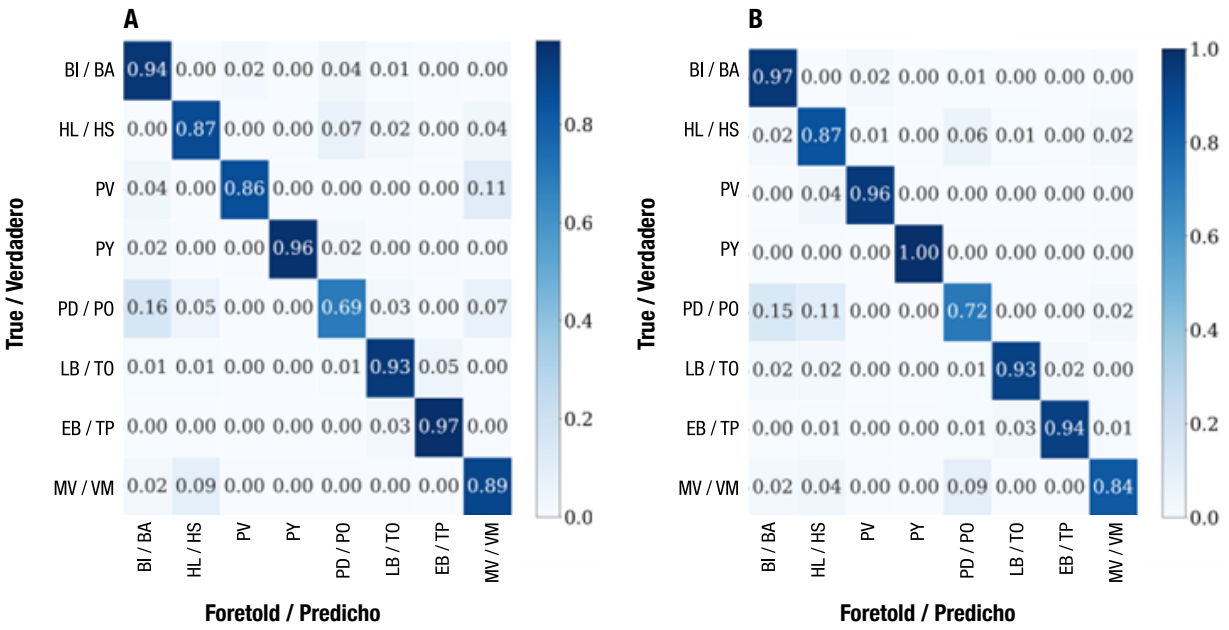


Figure 5. Confusion matrices of the accuracy of convolutional neural networks (CNNs) for the classification of eight target classes in potato cultivation: A) Standard CNN and B) VGG19. BI: bacterial infection (*Candidatus Phytoplasma*); HL: healthy leaf; PV: potato leafroll virus; PY: potato virus Y; PD: pest damage; LB: late blight (*Phytophthora infestans*); EB: early blight (*Alternaria solani*); MV: mosaic virus (*Potato virus X*).

Figura 5. Matrices de confusión de la precisión de las redes neuronales convolucionales (CNN) para la clasificación de ocho clases objetivo en el cultivo de papa: A) CNN estándar y B) VGG19. BA: infección bacteriana (*Candidatus Phytoplasma*); HS: hoja sana; PV: virus del enrollamiento de la hoja de papa (*Potato leafroll virus*); PY: virus Y de la papa (*Potato virus Y*); PO: daños por plagas; TO: tizón tardío (*Phytophthora infestans*); TP: tizón temprano (*Alternaria solani*); VM: virus del mosaico (*Potato virus X*).

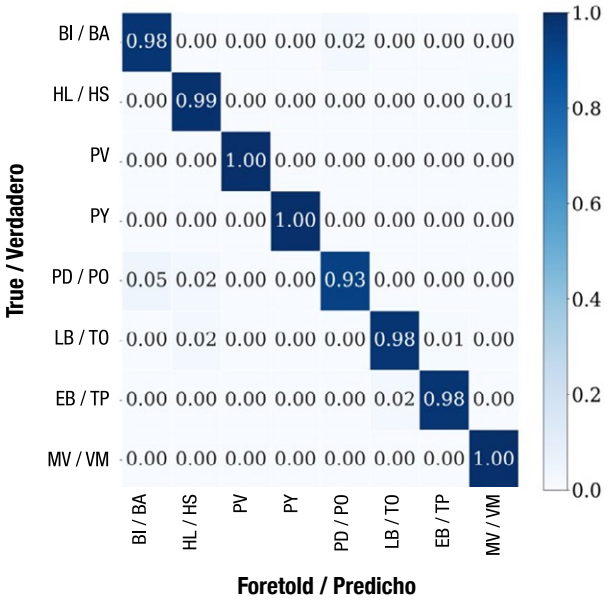


Figure 6. Confusion matrix of the accuracy of the YOLOv11n model for the classification of eight target classes in potato cultivation. BI: bacterial infection (*Candidatus Phytoplasma*); HL: healthy leaf; PV: potato leafroll virus; PY: potato virus Y; PD: pest damage; LB: late blight (*Phytophthora infestans*); EB: early blight (*Alternaria solani*); MV: potato mosaic virus.

Figura 6. Matriz de confusión de la precisión del modelo YOLOv11n para la clasificación de ocho clases objetivo en el cultivo de papa. BA: infección bacteriana (*Candidatus Phytoplasma*); HS: hoja sana; PV: virus del enrollamiento de la hoja de papa (*Potato leafroll virus*); PY: virus Y de la papa (*Potato virus Y*); PO: daños por plagas; TO: tizón tardío (*Phytophthora infestans*); TP: tizón temprano (*Alternaria solani*); VM: virus del mosaico (*Potato virus X*).

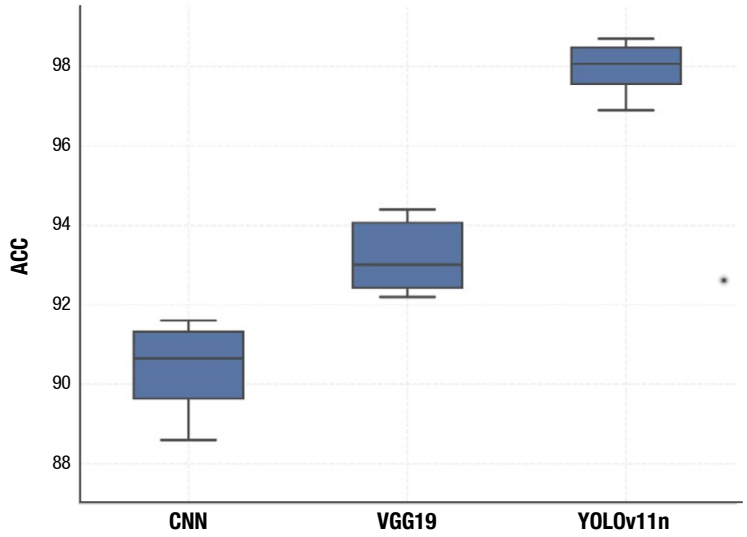


Figure 7. Comparison of the global classification accuracy (ACC) of the standard convolutional neural network (CNN) models, VGG19 and YOLOv11n, using cross-validation (*k* = 5).

Figura 7. Comparación de la precisión global de clasificación (ACC) de los modelos de redes neuronales convolucionales estándar (CNN), VGG19 y YOLOv11n mediante validación cruzada (*k* = 5).

Table 2. Prediction performance metrics of the standard convolutional neural network (CNN) models, VGG19 and YOLOv11n with the test dataset.

Cuadro 2. Métricas de desempeño de predicción de los modelos de redes neuronales convolucionales estándar (CNN), VGG19 y YOLOv11n con el conjunto de datos de prueba.

Model / Modelo	Class / Clase	Performance (%) / Desempeño (%)					
		P	S	F1	ACC	F1-macro	Soporte
CNN	BI / BA	90	94	92			128
	HL / HS	90	87	88			83
	PV	92	86	89			28
	PY	100	96	98			44
	PD / PO	76	69	72			128
	LB / TO	93	93	93			124
	EB / TP	94	97	96			105
	MV / VM	80	89	84			45
					90.1	89	
VGG19	BI / BA	92	97	94			128
	HL / HS	97	83	90			83
	PV	96	89	93			28
	PY	100	100	100			44
	PD / PO	85	85	85			128
	LB / TO	89	97	93			124
	EB / TP	97	97	97			105
	MV / VM	93	84	88			45
					92.8	92.5	
YOLOv11n	BI / BA	98	95	97			128
	HL / HS	99	96	98			83
	PV	100	100	100			28
	PY	100	100	100			44
	PD / PO	93	98	95			128
	LB / TO	97	98	98			124
	EB / TP	98	99	98			105
	MV / VM	100	99	100			45
					98	98.1	

P: precision; S: sensitivity; F1: F1 value; ACC: global classification accuracy; BI: bacterial infection (*Candidatus Phytoplasma*); HL: healthy leaf; PV: *Potato leafroll virus*; PY: *Potato virus Y*; PD: pest damage; LB: late blight (*Phytophthora infestans*); EB: early blight (*Alternaria solani*); MV: potato mosaic virus (*Potato virus X*).

P: precisión; S: sensibilidad; F1: valor F1; ACC: precisión global de clasificación; BA: infección bacteriana (*Candidatus Phytoplasma*); HS: hoja sana; PV: virus del enrollamiento de la hoja de papa (*Potato leafroll virus*); PY: virus Y de la papa (*Potato virus Y*); PO: daños por plagas; TO: tizón tardío (*Phytophthora infestans*); TP: tizón temprano (*Alternaria solani*); VM: virus del mosaico (*Potato virus X*).

Table 3. Performance of convolutional models reported in the literature.
Cuadro 3. Desempeño de modelos convolucionales reportados en la literatura.

Models / Modelos	Data set / Conjunto de datos	Global classification accuracy (ACC) / Precisión global de clasificación (ACC)	Classes / Clases	References / Referencia
VGG16 y VGG19	*Plant Village*	91.0 %	3	Sholihati et al. (2020)
VGG16, VGG19, resNet50 y MobileNet	*Plant Village*	97.89 %	3	Chakraborty et al. (2022)
EfficientNetV2B3, MobileNetV3-Large, VGG-16, ResNet50 y DenseNet121	Own / Propios	73.63 %	7	Shabrina et al. (2024)
CNN, VGG19 y YOLOv11	*Plant Village*, *Mendeley Data* y *Zenodo*	98.1 %	8	Present study

For reference, the performance obtained in this study was compared with results reported for CNN models and architectures with transfer learning, such as VGG16, VGG19 and MobileNet, trained with different datasets and numbers of target classes (Table 3). This comparison showed that the same model can exhibit variations in performance depending on the quality, diversity, and balance of the dataset. Even though some studies used similar datasets or included common classes, the reported accuracy levels were lower than those obtained in this work.

The three models evaluated (CNN, VGG19, and YOLOv11n) showed high predictive capacity and adequate adaptation to the variability of data from different sources (*Plant Village, Mendeley Data*, and *Zenodo*). However, YOLOv11n stood out for achieving an ACC of 98.1 % in the classification of the eight target classes.

Conclusions

The CNN deep learning models VGG19 and YOLOv11n showed high capacity for classifying eight target classes of potato crops from digital images. Among the models, YOLOv11n achieved the best performance, with an ACC of 98.1 %, followed by VGG19 (92.8 %) and CNN (90.1 %).

These results confirm the potential of applying deep learning models to develop phytosanitary diagnostic tools in potato cultivation using digital images.

Acknowledgments

The lead author thanks the Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT), now the Secretaría de Ciencia, Humanidades, Tecnología e Innovación (SECIHTI), for the support provided through a postgraduate scholarship, which made this research possible.

entre VM y PO. A pesar de estas confusiones puntuales, el modelo mostró un alto nivel de precisión en las clases restantes (Figura 5B).

Desempeño del modelo YOLOv11n

La matriz de confusión del modelo YOLOv11n evidenció un desempeño sobresaliente, con valores de precisión cercanos al 100 % en la mayoría de las clases objetivo. A diferencia de la CNN y VGG19, YOLOv11n identificó el PO con mayor confiabilidad. No obstante, esta clase presentó algunos errores de clasificación, atribuibles a la alta variabilidad visual de las muestras y a su escasa representación. La diferencia entre la clase con menor y mayor precisión fue de solo 7 %, lo cual indica un desempeño equilibrado entre clases objetivo (Figura 6).

Durante el procedimiento de validación cruzada ($k = 5$), YOLOv11n superó consistentemente a la CNN y a VGG19 en términos de ACC (Figura 7).

Desempeño en predicción con datos de prueba no vistos

Los modelos optimizados de CNN, VGG19 y YOLOv11n se evaluaron con el conjunto de datos de prueba para estimar su desempeño en escenarios no vistos. Las métricas globales ACC y F1-macro de cada modelo fueron similares, lo que indica una baja sensibilidad de los modelos al desbalance en el tamaño de las clases. Sin embargo, el modelo YOLOv11n presentó un desempeño superior al de CNN y VGG19 en las métricas globales evaluadas (Cuadro 2).

Como referencia, se comparó el desempeño obtenido en este estudio con resultados reportados para modelos CNN y arquitecturas con transferencia de aprendizaje, como VGG16, VGG19 y MobileNet, entrenados con diferentes conjuntos de datos y números de clases objetivo (Cuadro 3). Esta comparación evidenció que

Declaration of no conflict of interest

The authors declare that we have no known economic conflicts of interest or personal relationships that could have influenced the work presented in this article.

Declaration of the use of artificial intelligence (AI)

The authors declare that they have not used generative AI or AI-assisted technologies for the development of this contribution.

End of English version

References / Referencias

Aldakheel, E. A., Zakariah, M., & Alabdalall, A. H. (2024). Detection and identification of plant leaf diseases using YOLOv4. *Frontiers in Plant Science*, *15*, 1355941. https://doi.org/10.3389/fpls.2024.1355941

Chakrabarti, S. K., Sharma, S., & Shah, M. A. (2022). Potato pests and diseases: A global perspective. In S. K. Chakrabarti, S. Sharma, & M. A. Shah (Eds.), *Sustainable management of potato pests and diseases* (pp. 1-23). Springer. https://doi.org/10.1007/978-981-16-7695-6_1

Chakraborty, K. K., Mukherjee, R., Chakroborty, C., & Bora, K. (2022). Automated recognition of optical image based potato leaf blight diseases using deep learning. *Physiological and Molecular Plant Pathology*, *117*, 101781. https://doi.org/10.1016/j.pmpp.2021.101781

Charkowski, A., Sharma, K., Parker, M. L., Secor, G. A., & Elphinstone, J. (2019). Bacterial diseases of potato. In H. Campos & O. Ortiz (Eds.), *The potato crop: Its agricultural, nutritional and social contribution to humankind* (pp. 351-388). Springer. https://doi.org/10.1007/978-3-030-28683-5_10

Dong, K., Zhou, C., Ruan, Y., & Li, Y. (2020). MobileNetV2 model for image classification. *Institute of Electrical and Electronics Engineers*, *2*, 476-480. https://doi.org/10.1109/ITCA52113.2020.00106

Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, *145*, 311-318. https://doi.org/10.1016/j.compag.2018.01.009

Food and Agriculture Organization of the United Nations (FAO). (2025). *Celebrating the potato: A global treasure for food security and biodiversity*. FAO. https://www.fao.org/cgrfa/news/news-detail/celebrating-the-potato--a-global-treasure-for-food-security-and-biodiversity/en

Ghosh, H., Rahat, I. S., Shaik, K., Khasim, S., & Yesubabu, M. (2023). Potato leaf disease recognition and prediction using convolutional neural networks. *EAI Endorsed Transactions on Scalable Information Systems*, *10*(6), 1-8. https://doi.org/10.4108/eetsis.3937

un mismo modelo puede presentar variaciones en su desempeño dependiendo de la calidad, diversidad y balance del conjunto de datos. Aun cuando algunos estudios emplearon conjuntos de datos similares o incluyeron clases comunes, los niveles de precisión reportados fueron inferiores a los obtenidos en este trabajo.

Los tres modelos evaluados (CNN, VGG19 y YOLOv11n) mostraron una alta capacidad de predicción y una adecuada adaptación a la variabilidad de los datos provenientes de diferentes fuentes (*Plant Village*, *Mendeley Data* y *Zenodo*). No obstante, YOLOv11n destacó por alcanzar una ACC de 98.1 % en la clasificación de las ocho clases objetivo.

Conclusiones

Los modelos de aprendizaje profundo CNN, VGG19 y YOLOv11n mostraron alta capacidad para clasificar ocho clases objetivo del cultivo de papa a partir de imágenes digitales. En los modelos, YOLOv11n alcanzó el mejor desempeño, con una ACC de 98.1 %, seguida por VGG19 (92.8 %) y CNN (90.1 %).

Estos resultados confirman el potencial de aplicación de los modelos de aprendizaje profundo para desarrollar herramientas de diagnóstico fitosanitario en el cultivo de papa a partir de imágenes digitales.

Agradecimientos

El autor principal agradece al Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT), ahora Secretaría de Ciencia, Humanidades, Tecnología e Innovación (SECIHTI), por el apoyo brindado a través de una beca de posgrado, la cual hizo posible la realización de esta investigación.

Declaración de no conflicto de interés

Los autores declaramos que no tenemos conflictos de intereses económicos ni relaciones personales conocidas que pudieran haber influido en el trabajo presentado en este artículo.

Declaración del uso de inteligencia artificial (IA)

Los autores declaramos no haber utilizado IA generativa o tecnologías asistidas por IA para el desarrollo de la presente contribución.

Fin de la versión en español

Hughes, D. P., & Salathe, M. (2015). *An open access repository of images on plant health to enable the development of mobile disease diagnostics*. arXiv. https://arxiv.org/abs/1511.08060v2

Islam, F., Hoq, M. N., & Rahman, C. M. (2019). Application of transfer learning to detect potato disease from leaf image. *Institute of Electrical and Electronics Engineers*, 127-130. https://doi.org/10.1109/RAAICON48939.2019.53

Khanam, R., Hussain, M., Hill, R., & Allen, P. (2024). A comprehensive review of convolutional neural networks for defect detection in industrial applications. *Institute of Electrical and Electronics Engineers*, *12*, 94250-94295. https://doi.org/10.1109/ACCESS.2024.3425166

Laizer, H., Mduma, N., Machuve, D., Lyimo, T., Babirye, C., Swai, J., & Siwingwa, A. (2023). *Irish potato imagery dataset for early detection of crop diseases*. Zenodo. https://doi.org/10.5281/zenodo.8286529

Rahman, C. R., Arko, P. S., Ali, M. E., Khan, M. A. I., Apon, S. H., Nowrin, F., & Wasif, A. (2020). Identification and recognition of rice diseases and pests using convolutional neural networks. *Biosystems Engineering*, *194*, 112-120. https://doi.org/10.1016/j.biosystemseng.2020.03.020

Raigonda, M. R., Terdal, S. P., & Bainoor, R. (2022). *Potato viral disease dataset on both foliar and tuber*. Mendeley Data. https://doi.org/10.17632/rgfhzd5mzw.1

Raschka, S., & Mirjalili, V. (2019). *Machine learning and deep learning with Python, Scikit-Learn, and TensorFlow 2* (3ra ed.). Packt Publishing.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: unified, real-time object detection. *Institute of Electrical and Electronics Engineers*, 779-788. https://doi: 10.1109/CVPR.2016.91

Saleem, T. J., & Chishti, M. A. (2021). Deep learning for the internet of things: Potential benefits and use-cases. *Digital Communications and Networks*, *7*(4), 526-542. https://doi.org/10.1016/j.dcan.2020.12.002

Servicio de Información Agroalimentaria y Pesquera (SIAP). (2024). *Anuario estadístico de la producción agrícola*. SIAP. https://nube.agricultura.gob.mx/cierre_agricola/

Shabrina, N. H., Indarti, S., Maharani, R., Kristiyanti, D. A., Irmawati, Prastomo, N., & Adilah, T. (2024). A novel dataset of potato leaf disease in uncontrolled environment. *Data in Brief*, *52*, 109955. https://doi.org/10.1016/j.dib.2023.109955

Sholihati, R. A., Sulistijono, I. A., Risnumawan, A., & Kusumawati, E. (2020). Potato leaf disease classification using deep learning approach. *Institute of Electrical and Electronics Engineers*, 392-397. https://doi.org/10.1109/IES50839.2020.9231784

Simonyan, K., & Zisserman, A. (2014). *Very deep convolutional networks for large-scale image recognition*. arXiv. https://arxiv.org/abs/1409.1556v6

Srivastava, A., Rawat, B. S., Bajpai, P., & Dhondiyal, S. A. (2024). Potato leaf disease detection method based on the YOLO model. *International Conference on Data Engineering and Communication Systems*, *4*. https://doi.org/10.1109/ICDECS59733.2023.10502511

Tilahun, N. (2022). *Potato leaf (healthy and late blight)*. Mendeley Data. https://doi.org/10.17632/v4w72bsts5.2

Ultralytics Inc. (2024). *YOLOv11 - Ultralytics YOLO Docs* [software]. Ultralytics Inc. https://docs.ultralytics.com/models/yolo11/

Viso.ai. (2024). *YOLOv11: A new iteration of "You Only Look Once"* [software]. Viso.ai. https://viso.ai/computer-vision/yolov11/

Wang, H., Shang, S., Wang, D., He, X., Feng, K., & Zhu, H. (2022). Plant disease detection and classification method based on the optimized lightweight YOLOv5 model. *Agriculture*, *12*(7), 931. https://doi.org/10.3390/agriculture12070931