

# CLASIFICADOR DE RED NEURONAL CONVOLUCIONAL PARA IDENTIFICAR ENFERMEDADES DEL FRUTO DE AGUACATE (*Persea americana* Mill.) A PARTIR DE IMÁGENES DIGITALES

## CONVOLUTIONAL NEURAL NETWORK CLASSIFIER FOR IDENTIFYING DISEASES OF AVOCADO FRUIT (*Persea americana* Mill.) FROM DIGITAL IMAGES

Ulises Enrique Campos-Ferreira<sup>1</sup>, Juan Manuel González-Camacho<sup>2\*</sup>

<sup>1</sup>Universidad Autónoma Chapingo. Departamento de Parasitología Agrícola. Chapingo, Texcoco 56239, Estado de México, México. <sup>2</sup>Colegio de Postgraduados Campus Montecillo. Programa de Socioeconomía, Estadística e Informática-Cómputo Aplicado. Montecillo, Texcoco 56230, Estado de México, México.

\*Autor para correspondencia: Juan Manuel González Camacho (jmgc@colpos.mx).

ORCID: <https://orcid.org/0000-0001-5479-7316>

### RESUMEN

México es el productor principal de aguacate (*Persea americana* Mill.) en el mundo; sin embargo, este fruto en postcosecha es susceptible a hongos fitopatógenos que reducen su calidad. Por ello, la identificación oportuna de estos organismos con base en herramientas no invasivas, como los análisis por medio de inteligencia artificial resulta de interés para reducir pérdidas económicas. El objetivo de este estudio fue implementar un modelo de aprendizaje automático con una base de datos de imágenes digitales de frutos colectados en campo, a través de crear un clasificador de red neuronal convolucional (CNN), entrenarlo y validarlo para identificar frutos sanos de aguacate cv. Fuerte y frutos infectados con roña (*Sphaceloma perseae* Jenkins) o antracnosis (*Colletotrichum* spp.) de la zona productora en el Estado de Morelos, México. Frutos de aguacate sanos se recolectaron en campo, y también frutos con roña y antracnosis de huertos diferentes para generar un conjunto de 569 imágenes digitales. Las transformaciones de éstas aumentaron el conjunto de datos a 3983 imágenes. El modelo CNN se entrenó con una partición aleatoria de 80% de las imágenes y se validó para predicción con el 20% restante. La CNN alcanzó una precisión global de clasificación correcta de 80% con el conjunto de validación. Además, el clasificador se evaluó con un conjunto de prueba de 100 imágenes no incluidas en el conjunto original y obtuvo una precisión global de clasificación correcta de 87%. El clasificador de aprendizaje profundo por CNN implementado en este estudio representa el uso factible de la inteligencia artificial para su aplicación en la identificación de enfermedades en aguacate a partir de imágenes digitales en la etapa de postcosecha.

### ABSTRACT

Mexico is the main avocado (*Persea americana* Mill.) producer in the world; however, this fruit is susceptible to phytopathogenic fungi that reduce quality during postharvest. Therefore, the timely identification of these organisms using non-invasive tools, such as artificial intelligence analysis, is of interest to reduce economic losses. The objective of this study was to implement a machine learning model with a database of digital images of fruits collected in the field, by creating a convolutional neural network (CNN) classifier, training and validating it to identify healthy avocado cv. Fuerte fruits and fruits infected with scab (*Sphaceloma perseae* Jenkins) or anthracnose (*Colletotrichum* spp.) from the production area in the State of Morelos, Mexico. Healthy avocado fruits were collected in the field, as well as fruits with the scab and anthracnose from different orchards to generate a set of 569 digital images. Transformations of these increased the dataset to 3983 images. The CNN model was trained with a random partition of 80% of the images and validated for prediction with the remaining 20%. CNN achieved an overall correct classification accuracy of 80% with the validation set. In addition, the classifier was evaluated with a test set of 100 images not included in the original set and achieved an overall correct classification accuracy of 87%. The CNN deep learning classifier implemented in this study represents the feasible use of artificial intelligence for application in avocado disease identification from digital images at the postharvest stage.

**Key words:** artificial intelligence, artificial vision, pattern recognition, deep learning, *Persea americana*, *Sphaceloma perseae*.

Recibido: febrero, 2021. Aprobado: noviembre, 2021.

Publicado en *Agrociencia* 55: 695-709. 2021.

**Palabras clave:** inteligencia artificial, visión artificial, reconocimiento de patrones, aprendizaje profundo, *Persea americana*, *Sphaceloma perseae*.

## INTRODUCCIÓN

México es el principal productor de aguacate (*Persea americana* Mill.) en el mundo, con una producción anual de 2 393 000 Mg (SIAP, 2020) que satisface totalmente la demanda interna nacional (SAGARPA, 2017). El cultivo de aguacate es atacado por dos enfermedades muy frecuentes que afectan la calidad del fruto, la antracnosis ocasionada por hongos del género *Colletotrichum*, y la roña o sarna del fruto causada por el hongo *Sphaceloma perseae* (Paull y Duarte, 2011). El manejo inadecuado del cultivo puede originar pérdidas por enfermedades fungosas hasta de 71% antes de la cosecha, que pueden alcanzar el 100% durante el almacenamiento del fruto (Dann *et al.*, 2013).

El aguacate variedad Fuerte es resultado de la hibridación de las razas Mexicana y Guatemalteca y se considera como la primera variedad comercial de excelencia en el mundo. A pesar de haber perdido popularidad ante la variedad Hass, se cultiva en ciertas regiones de México como en los estados de Morelos, Puebla y Oaxaca. Esta variedad es una fuente excelente para polinizar Hass y tolera temperaturas bajas (Crane *et al.*, 2013). En términos de producción, se tiene un registro de 14 municipios como productores de esta variedad de aguacate los cuales producen 5740 Mg anuales (SIAP, 2021).

De manera tradicional, técnicos expertos identifican las enfermedades a través de la inspección visual del tejido de la planta. Estos procedimientos son costosos y poco prácticos para su aplicación generalizada (Dhakal y Shakya, 2018). La disponibilidad y desarrollo de sistemas automatizados para identificar y diagnosticar enfermedades de las plantas ofrece un apoyo valioso a los agrónomos para efectuar diagnósticos, a través de la observación con dispositivos ópticos del tejido infectado de la planta (Ferentinos, 2018).

Estudios diversos reportan el uso de técnicas, como el reconocimiento de patrones, procesamiento de imágenes, aprendizaje automático y aprendizaje profundo para identificar enfermedades en plantas (Türkoğlu y Hanbay, 2019; Harpale *et al.*, 2020; Loey *et al.*, 2020). Estos enfoques contemporáneos

## INTRODUCTION

Mexico is the leading producer of avocado (*Persea americana* Mill.) in the world, with an annual production of 2 393 000 Mg (SIAP, 2020) that fully satisfies domestic demand (SAGARPA, 2017). Avocado in cultivation is attacked by two very frequent diseases that affect fruit quality, anthracnose, caused by fungi of the *Colletotrichum* genus, and the avocado scab caused by the fungus *Sphaceloma perseae* (Paull and Duarte, 2011). Improper crop management can cause losses due to fungal diseases of up to 71% preharvest, which can reach 100% during fruit storage (Dann *et al.*, 2013).

The Fuerte avocado variety is the result of the hybridization of the Mexican and Guatemalan races and is considered the first commercial variety of excellence in the world. Although it has lost popularity to the Hass variety, it is cultivated in certain regions of Mexico such as the states of Morelos, Puebla and Oaxaca. This variety is an excellent source for pollinating Hass and tolerates low temperatures (Crane *et al.*, 2013). In terms of production, 14 municipalities are registered as producers of this avocado variety, which produce 5740 Mg annually (SIAP, 2021).

Traditionally, expert technicians identify diseases through visual inspection of plant tissue. These procedures are costly and impractical for widespread application (Dhakal and Shakya, 2018). The availability and development of automated systems to identify and diagnose plant diseases offers valuable support to agronomists in making diagnoses, through observation with optical devices of infected plant tissue (Ferentinos, 2018).

Diverse studies report the use of techniques, such as pattern recognition, image processing, machine learning and deep learning to identify plant diseases (Türkoğlu and Hanbay, 2019; Harpale *et al.*, 2020; Loey *et al.*, 2020). These contemporary approaches facilitate disease identification from captured crop images. For this purpose, relevant features of the images are used to extract information from the plants of interest (Brahimi *et al.*, 2018).

Mohanty *et al.* (2016) implemented the use of deep learning algorithms to identify diseases of various crops from a public database (PlantVillage dataset) and in some crops achieved an overall

facilitan la identificación de enfermedades a partir de imágenes capturadas de los cultivos. Para ello, se utilizan características relevantes de las imágenes para extraer información de las plantas de interés (Brahmi *et al.*, 2018).

Mohanty *et al.* (2016) implementaron el uso de algoritmos de aprendizaje profundo para identificar enfermedades de varios cultivos a partir de una base de datos pública (PlantVillage dataset) en algunos cultivos alcanzaron una precisión global en predicción de 99%. Toda y Okura (2018) mostraron la importancia de las redes neuronales convolucionales para la detección automática de enfermedades con conjuntos de datos de acceso público con base en características de color y textura de plantas. Thangaraj *et al.* (2020) reportaron la aplicación de un clasificador de aprendizaje profundo para detectar enfermedades en aguacate con una precisión global de clasificación correcta de 96%.

El objetivo de este estudio fue implementar un modelo de aprendizaje profundo con una base de datos de imágenes digitales de frutos colectados en campo, a través de crear un clasificador de red neuronal convolucional (CNN), entrenarlo y validarlo para identificar frutos sanos de aguacate cv. Fuerte y frutos infectados con roña (*Sphaceloma perseae* Jenkins) o antracnosis (*Colletotrichum* spp.) provenientes de la zona productora de aguacate en el Estado de Morelos, México.

## MATERIALES Y MÉTODOS

### Sitio experimental y colecta del material vegetativo

Para crear la base de datos de imágenes digitales, se realizó una colecta de frutos de aguacate sanos y con las dos enfermedades de interés, en la zona oriente del estado de Morelos, en los municipios de Ocuituco (18° 51' 47.6" N 98° 46' 49.1" O) y Tetela del Volcán (18° 53' 36.3" N 98° 43' 47.7" O). Esta zona es la región productora de aguacate variedad Fuerte más importante de México y comparte condiciones geográficas y climatológicas similares con su centro de origen en Atlixco, Puebla.

En este estudio se seleccionó la variedad de aguacate Fuerte debido a que el fruto presenta una coloración verde durante toda su etapa de desarrollo, a diferencia de la variedad Hass que cambia del color verde a negro cuando madura. Esta característica de la variedad Fuerte contrasta con las enfermedades que se presentan en el fruto y facilita su reconocimiento con base en imágenes transformadas al espacio de color RGB, y un clasificador de red neuronal convolucional.

prediction accuracy of 99%. Toda and Okura (2018) showed the importance of convolutional neural networks for automatic disease detection with publicly available datasets based on color and texture characteristics of plants. Thangaraj *et al.* (2020) reported the application of a deep learning classifier to detect diseases in avocado with an overall correct classification accuracy of 96%.

The objective of this study was to implement a deep learning model with a database of digital images of fruits collected in the field, by creating a convolutional neural network (CNN) classifier, training and validating it to identify healthy fruits of avocado cv. Fuerte and fruits infected with scab (*Sphaceloma perseae* Jenkins) or anthracnose (*Colletotrichum* spp.) from the production area of avocado in the State of Morelos, Mexico.

## MATERIALS AND METHODS

### Experimental site and vegetative material collection

To create the digital image database, healthy avocado fruit with the two diseases of interest were collected in the eastern zone of the state of Morelos, in the municipalities of Ocuituco (18° 51' 47.6" N 98° 46' 49.1" W) and Tetela del Volcán (18° 53' 36.3" N 98° 43' 47.7" W). This area is the most important avocado producing region in Mexico and shares similar geographic and climatological conditions with its center of origin in Atlixco, Puebla.

In this study, the Fuerte avocado variety was selected because the fruit has a green color throughout its development stage, unlike the Hass variety, which changes from green to black when ripe. This characteristic of the Fuerte variety contrasts with the diseases present in the fruit and facilitates its recognition based on images transformed to RGB colour space and a convolutional neural network classifier.

Fruit collection at the Ocuituco site was made in commercial orchards, backyard orchards and orchards with low agricultural activity. Healthy fruit, and with anthracnose or scab, at physiological and commercial maturity were selected. Fruit collection at the Tetela del Volcán site was purchased at an arbitrary point of sale in the main square; 320 avocado fruits were collected, approximately 80 kg weight.

### Capturing images of avocado fruits

Digital images of avocado fruits were obtained in a semi-professional PULUZ® brand studio box with integrated LED lighting in a Nikon® model D3400 digital camera. The images

La colecta de frutos en el sitio de Ocuituco se realizó en huertos comerciales, de traspatio y con actividad agrícola baja; se seleccionaron frutos sanos, con antracnosis y roña, en madurez fisiológica y comercial. La colecta de frutos en el sitio de Tetela del Volcán se llevó a cabo en un punto de venta arbitrario de la plaza principal; se colectaron 320 frutos de aguacate con un peso aproximado de 80 kg.

#### Captura de imágenes de los frutos de aguacate

Las imágenes digitales de los frutos de aguacate se obtuvieron en una caja de estudio semi profesional marca PULUZ® con iluminación LED integrada con una cámara digital Nikon® modelo D3400. Las imágenes se tomaron sobre el fruto en el plano horizontal, la cámara se colocó arriba en el orificio superior de la caja de estudio y se fijó por medio de un trípode. Condiciones similares de espacio y de la cámara se definieron para homogeneizar la calidad de las imágenes y su representación en el espacio de color RGB (Cuadro 1).

A cada fruto se le capturó al menos una imagen, y a ésta se le asoció una de las tres clases objetivo, S para frutos sano, R para roña o A para antracnosis. Cuando el fruto tenía más de una característica de interés (por ejemplo, dos enfermedades) se tomaron imágenes complementarias. La utilidad DigiCamControl se utilizó para controlar la cámara en forma remota desde la computadora para almacenamiento directo de las fotografías digitales; se obtuvieron 569 imágenes digitales primarias (Figura 1).

#### Preprocesamiento e incremento de imágenes

Las imágenes originales de 4496 x 3000 píxeles se redimensionaron a 300 x 300 píxeles de resolución para reducir requerimientos de memoria y de cálculo. El programa IrfanView® se utilizó para recortar las imágenes y transferirlas a un lote completo.

Después, las imágenes se homogeneizaron con un fondo blanco, con el editor de fotografías Photoshop®. IrfanView® se utilizó para aumentar el número de imágenes; se generaron imágenes transformadas en forma horizontal, vertical y rotadas 90 y 180°. Con este procedimiento, la base de datos se incrementó a 3983 imágenes, las cuales se utilizaron para entrenar y validar el desempeño del clasificador CNN. Las imágenes se etiquetaron con un nombre secuencial: Avocado 0001.jpg, Avocado 0002, ... Avocado 3983.jpg, como la última fotografía (Figura 2).

Las tres clases objetivo S, A y R se recodificaron con sus vectores unitarios respectivos (1,0,0), (0,1,0) y (0,0,1) para su tratamiento numérico.

#### Entrenamiento y validación del clasificador CNN

El clasificador CNN se implementó en la plataforma de modelación Keras y Tensorflow en lenguaje Python v. 3.7 y entorno

#### Cuadro 1. Condiciones de captura de imágenes digitales de los frutos de aguacate.

Table 1. Conditions for digital images capture of avocado fruits.

Características del espacio	Datos
Dimensiones de la caja de estudio	22 x 24 x 23 cm
Intensidad de luz	450 lm
Fondo	Blanco
Colocación del fruto	Al centro
Características de la cámara	Datos
Extensión de archivo	jpg
Dimensiones	4496 x 3000 px
Relación de aspecto	4:3
Apertura del diafragma	f / 3.8
Velocidad de obturación	1 / 100 s
Valor ISO	200
Distancia focal	24 mm
Compensación de exposición	0 paso
Uso de flash incorporado	No

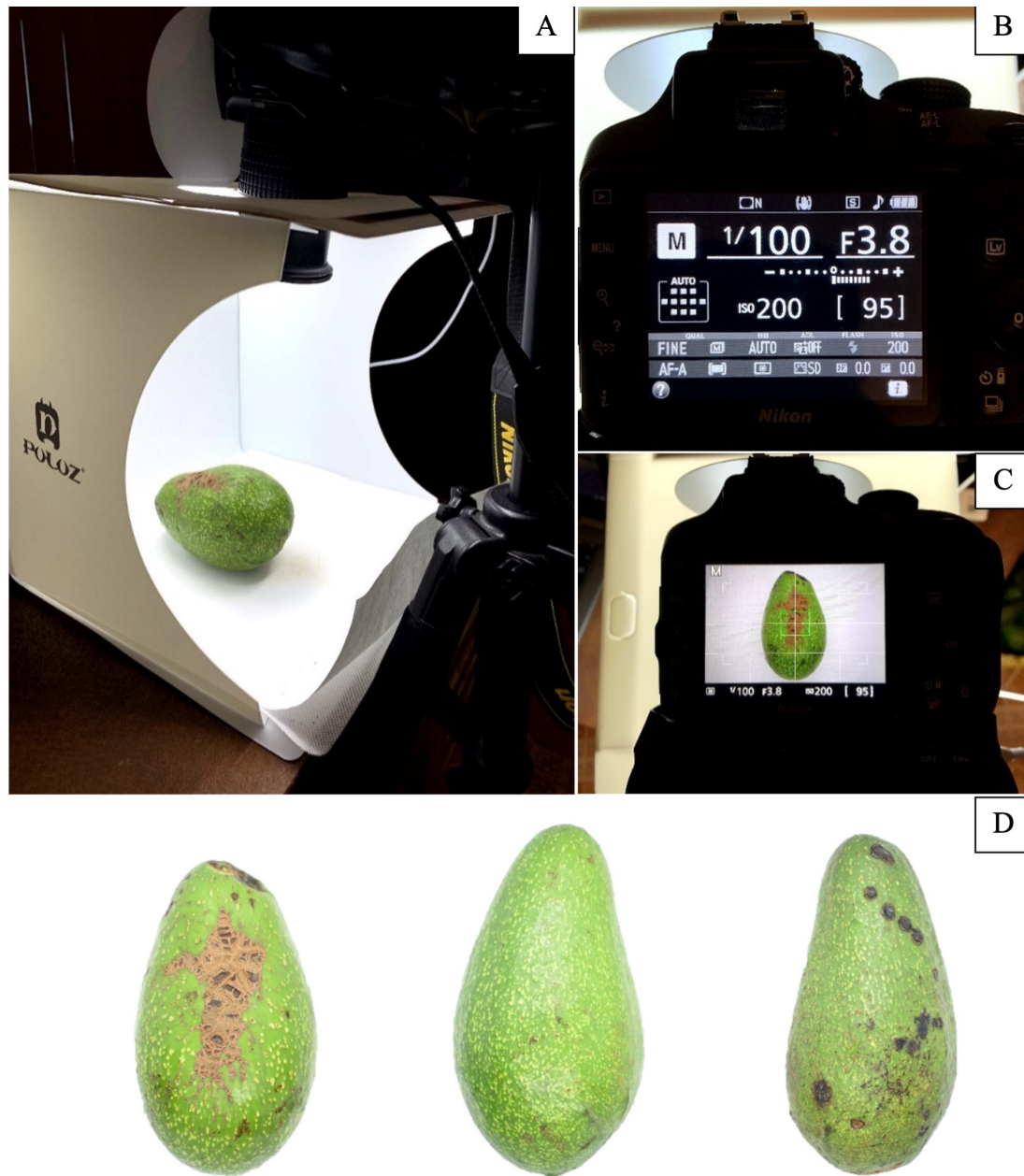
were taken on the fruit in the horizontal plane, the camera was placed on top in the upper hole of the studio box and fixed by means of a tripod. Similar space and camera conditions were defined to homogenize the quality of the images and their representation in RGB colour space (Table 1).

At least one image was captured for each fruit, and it was associated to any of the three target classes, S for healthy fruit, R for scab, or A for anthracnose, as it was the case. When the fruit had more than one characteristic of interest (*e.g.*, two diseases), complementary images were taken. The DigiCamControl utility was used to control the camera remotely from the computer for direct storage of the digital photographs; 569 primary digital images were obtained (Figure 1).

#### Pre processing and additions to the image database

The original 4496 x 3000 pixels images were resized to 300 x 300 pixels resolution to reduce memory and computational requirements. The IrfanView® program was used to crop the images into a complete batch.

The images were then homogenized with a white background using Photoshop® photo editor. IrfanView® was used to increase the number of images; transformed images were generated horizontally, vertically and rotated 90 and 180°. With this procedure, the database was increased to 3983 images, which



**Figura 1.** Condiciones de digitalización de los frutos de aguacate. A: fruto dentro de la caja de estudio, arriba de ésta se instaló la cámara fotográfica; B: metadatos de la cámara fotográfica; C: previsualización de la imagen; D: imágenes digitales de frutos.

**Figure 1.** Conditions of digitization of avocado fruits. A: fruit inside the study box, on top of which the camera was installed; B: camera metadata; C: image preview; D: digital images of fruits.

de desarrollo Jupyter Notebook® v. 6.0.1 (Chollet, 2018). El entrenamiento y validación del clasificador CNN se realizó con una computadora portátil con Windows 10, Intel® Core i3, velocidad de procesamiento 2.20 GHz y 20 GB en RAM. El conjunto de datos de entrada consistió en 3983 imágenes de 300 x 300 píxeles; como se indicó, a cada imagen se le asoció su clase objetivo (S, R, o A).

were used to train and validate the performance of the CNN classifier. The images were labelled with a sequential name: Avocado 0001.jpg, Avocado 0002, ... Avocado 3983.jpg, as the last photograph (Figure 2).

The three target classes S, A and R were recoded with their respective unit vectors (1,0,0), (0,1,0) and (0,0,1) for numerical processing.

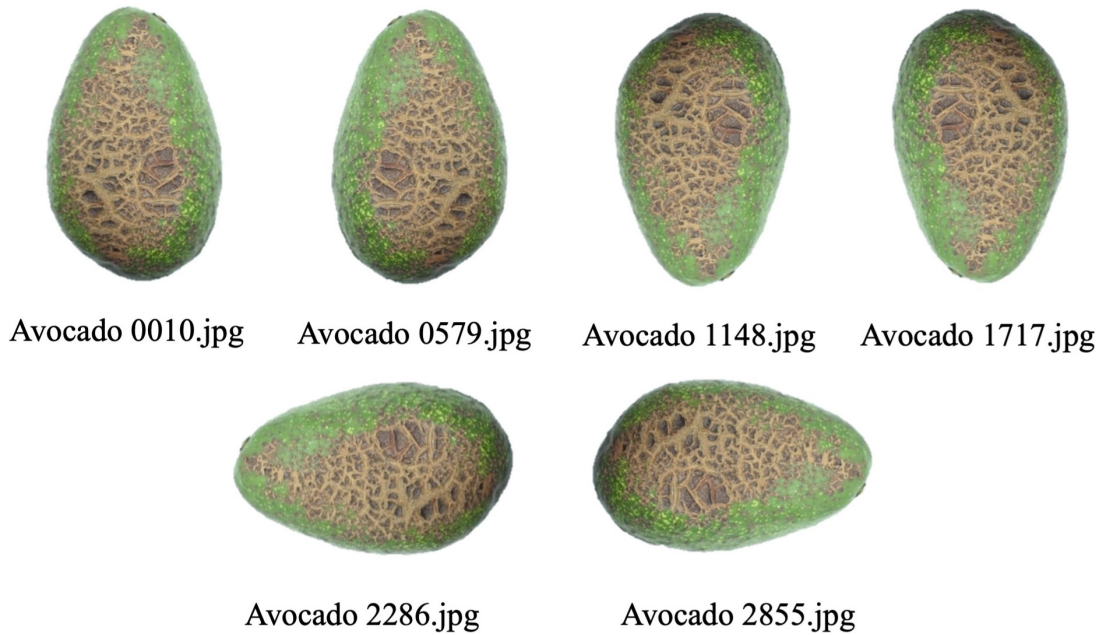


Figura 2. Imágenes transformadas en posiciones distintas del fruto de aguacate digitalizado y almacenadas con nombres diferentes.

Figure 2. Images transformed from the digitized avocado fruit, in distinct positions and stored under different names.

### Arquitectura del clasificador CNN

La arquitectura básica del modelo CNN para resolver un problema de clasificación multiclase se configuró con los hiperparámetros (Figura 3): *sequential* para conectar las capas de neuronas ocultas; *conv2D* para realizar convoluciones y extraer características; *MaxPooling2D* para extraer los vectores de mayor ponderación y crear una submuestra de la imagen; *flatten* para convertir los arreglos bidimensionales a unidimensionales; y *dense* para conectar por completo las neuronas con las siguientes; *dropout* para desactivar un porcentaje específico de neuronas y reducir el sobreajuste del clasificador); y una capa de salida con tres neuronas para las clases objetivo S, R, y A.

En cada capa convolucional se utilizó la función de activación *ReLU* (Rectified linear unit), definida por:

$$\text{ReLU}(z) = \max(0, z) = \begin{cases} 0 & \text{para } x < 0 \\ x & \text{para } x \geq 0 \end{cases}$$

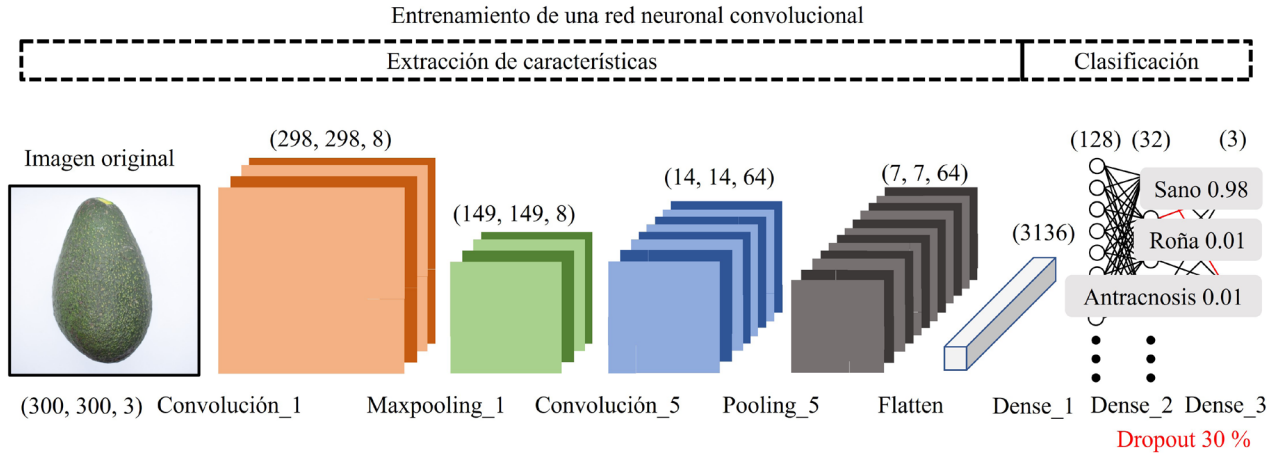
*ReLU* transforma los valores de entrada, anula los valores negativos y deja los positivos, tal y como entran. En el procesamiento de imágenes, los valores negativos no son relevantes; por lo tanto, el número de neuronas conectadas con la siguiente capa oculta se redujo.

### CNN classifier training and validation

The CNN classifier was implemented on the Keras and Tensorflow modelling platform in Python v. 3.7 language and the Jupyter Notebook® v. 6.0.1 development environment (Chollet, 2018). The training and validation of the CNN classifier was performed with a notebook computer Windows 10, Intel® Core i3, 2.20 GHz processing speed and 20 GB RAM. The input dataset consisted of 3983 images of 300 x 300 pixels; as it was described, each image was associated with its target class (S, R, or A).

### CNN classifier architecture

The basic architecture of the CNN model to solve a multiclass classification problem was configured with the hyperparameters (Figure 3): *sequential* to connect the hidden neuron layers; *conv2D* to perform convolutions and extract features; *MaxPooling2D* to extract the highest weighted vectors and create a subsample of the image; *flatten* to convert the two-dimensional arrays to one-dimensional; and *dense* to fully connect the neurons to the following neurons; *dropout* to disable a specific percentage of neurons reducing classifier overfitting; and an output layer with three neurons for the target classes S, R, and A.



**Figura 3.** Arquitectura del clasificador de red neuronal convolucional (CNN). La imagen original representada como un arreglo pasa por varios filtros de convolución y *pooling* para capturar información relevante y reducir el tamaño del arreglo. La última capa de *pooling* se transforma en un vector que conecta a tres capas densas de neuronas para generar las respuestas. Los números entre paréntesis representan el ancho, largo y profundidad de los arreglos.

**Figure 3.** Convolutional neural network (CNN) classifier architecture. The original image represented as an array passes through several convolution and pooling filters to capture relevant information and reduce the array size. The last pooling layer is transformed into a vector connecting three dense layers of neurons to generate the responses. The numbers in parentheses represent the width, length and depth of the arrays.

Para la capa de salida se utilizó la función de activación *Softmax* que asigna una probabilidad de pertenencia a cada clase y se define como:

$$\text{Softmax}(x)_i = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}}$$

donde  $x$  es un vector de entrada con valores reales (positivo, cero, o negativo); la función exponencial estándar  $e^{x_i}$  se aplica a cada elemento del vector de entrada y se divide por  $\sum_{j=1}^K e^{x_j}$  para obtener valores en el intervalo  $[0, 1]$ . Este término de normalización asegura que los valores que entran a la función *Softmax* sumen 1 y que cada valor de probabilidad se encuentre entre 0 y 1; y  $K$  es el número de clases objetivo del clasificador.

Las funciones de pérdida *binary\_crossentropy* y *categorical\_crossentropy* y los optimizadores *adam* (adaptive moment estimation) y *adamax* se utilizaron para estimar los parámetros del clasificador CNN estos hiperparámetros se utilizan con frecuencia en problemas de clasificación multiclase.

### Entrenamiento del clasificador CNN

Particiones aleatorias de la base de datos se realizaron; 80% (3186 imágenes) para el entrenamiento y 20% (797 imágenes) para la validación del clasificador CNN. El conjunto de validación consistió en 336 imágenes (clase S), 252 (clase R) y 209

The activation function (Rectified linear unit) was used in each convolutional layer, defined by:

$$\text{ReLU}(z) = \max(0, z) = \begin{cases} 0 & \text{para } x < 0 \\ x & \text{para } x \geq 0 \end{cases}$$

*ReLU* transforms the input values, cancels the negative values and leaves the positive values as they enter. In image processing, negative values are not relevant; therefore, the number of neurons connected to the next hidden layer was reduced.

For the output layer, the *Softmax* activation function was used, which assigns a probability of belonging to each class and is defined as follows:

$$\text{Softmax}(x)_i = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}}$$

where  $x$  is an input vector with real values  $x_i$  (positive, zero, or negative); the standard exponential function  $e^{x_i}$  is applied to each element of the input vector and is divided by  $\sum_{j=1}^K e^{x_j}$  to obtain values in the interval  $[0, 1]$  This standardization term ensures that the values entering the *Softmax* function add up to 1; and that each probability value lies between 0 and 1; and  $K$  is the number of target classes of the classifier.

The loss functions *binary\_crossentropy* and *categorical\_crossentropy* and the optimizers *adam* (adaptive moment

(clase A). En el entrenamiento del modelo se utilizó un procesamiento por lotes de 128 muestras de imágenes y 25 iteraciones (epochs). También se consideraron 100 imágenes suplementarias, no incluidas en el entrenamiento y validación del modelo CNN, para evaluar su capacidad predictiva con entradas nuevas de datos. Los pesos del clasificador se inicializaron de manera aleatoria.

### Métricas de desempeño del clasificador CNN

Para evaluar la capacidad predictiva del clasificador CNN se utilizaron métricas que se definen en términos de una matriz de confusión, la cual se describe para el caso de una respuesta binaria con cuatro valores: verdaderos positivos ( $VP$ ), verdaderos negativos ( $VN$ ), falsos positivos ( $FP$ ), y falsos negativos ( $FN$ ). Las métricas definidas en términos de la matriz de confusión fueron las siguientes:

La precisión ( $P$ ) se define por:

$$P = \frac{VP}{VP + FP}$$

La exhaustividad ( $E$ ) se calcula por:

$$E = \frac{VP}{VP + FN}$$

La métrica  $F1_s$  es una combinación de  $P$  y  $E$  y se calcula como:

$$F1_s = 2 * \frac{P * E}{P + E}$$

Las métricas  $P$ ,  $E$  y  $F1_s$ , permiten evaluar el desempeño del clasificador para cada clase objetivo. La precisión global de clasificación ( $PG$ ) se calcula por:

$$PG = \frac{VP + VN}{VP + VN + FP + FN}$$

## RESULTADOS Y DISCUSIÓN

### Entrenamiento y validación del modelo

El entrenamiento del modelo CNN tuvo un tiempo de ejecución promedio de tres horas. Durante el entrenamiento se analizaron las curvas de desempeño

(estimation) and *adamax* were used to estimate the parameters of the CNN classifier; these hyperparameters are frequently used in multiclass classification problems.

### CNN classifier training

Random partitions of the database were performed; 80% (3186 images) for training and 20% (797 images) for validation of the CNN classifier. The validation set consisted of 336 (class S), 252 (class R) and 209 (class A) images. Batch processing of 128 image samples and 25 iterations (epochs) was used in the model training. Also, 100 supplementary images, not included in the training and validation of the CNN model, were considered to evaluate its predictive capability with new data inputs. The classifier weights were randomly initialized.

### CNN classifier performance metrics

To evaluate the predictive ability of the CNN classifier, metrics were used that are defined in terms of a confusion matrix, which is described for the case of a binary response with four values: true positives ( $VP$ ), true negatives ( $VN$ ), false positives ( $FP$ ) and false negatives ( $FN$ ). The metrics defined in terms of the confusion matrix were as follows:

Precision ( $P$ ) is defined by:

$$P = \frac{VP}{VP + FP}$$

Recall ( $E$ ) is calculated by:

$$E = \frac{VP}{VP + FN}$$

The  $F1_s$  metrics is a combination of  $P$  and  $E$ , it is calculated as:

$$F1_s = 2 * \frac{P * E}{P + E}$$

Metrics  $P$ ,  $E$  and  $F1_s$ , allow to evaluate the performance of the classifier for each target class. The overall classification accuracy ( $PG$ ) is calculated by:

$$PG = \frac{VP + VN}{VP + VN + FP + FN}$$

del modelo CNN en entrenamiento y validación (Figura 4). Estas curvas permiten dar seguimiento sistematizado a las tendencias de la evolución de precisión global de clasificación, y a la pérdida o error del modelo en función del número de iteraciones. El modelo CNN obtuvo una *PG* de 89% en entrenamiento y *PG* de 87% en validación después de 25 iteraciones con el optimizador *adam* y función de pérdida *binary\_crossentropy* (Figura 4-A).

En cambio, el clasificador CNN alcanzó una *PG* de 0.70 en entrenamiento y validación con el optimizador *adamax* y función de pérdida *categorical\_crossentropy* (Figura 4-C). De manera similar, CNN obtuvo una pérdida de 0.25 en entrenamiento y de 0.30 en validación con el optimizador *adam* y *binary\_crossentropy* (Figura 4-B), y de 0.70 con el optimizador *adamax* y *categorical\_crossentropy* (Figura 4-D).

La matriz de confusión muestra los resultados de validación (predicción) del clasificador CNN. La

## RESULTS AND DISCUSSION

### Model training and validation

The training of the CNN model had an average run time of three hours. During training, the performance curves of the CNN model during training and validation were analysed (Figure 4). These curves allow monitoring of the trends in the evolution of the overall classification accuracy, and the loss or error of the model as a function of the number of iterations. The CNN model obtained a *PG* of 89% in training and *PG* of 87% in validation after 25 iterations with the *adam* optimizer and *binary\_crossentropy* loss function (Figure 4-A).

In contrast, the CNN classifier achieved a *PG* of 0.70 in training and validation with the *adamax* optimizer and *categorical\_crossentropy* loss function (Figure 4-C). Similarly, CNN obtained a loss of 0.25

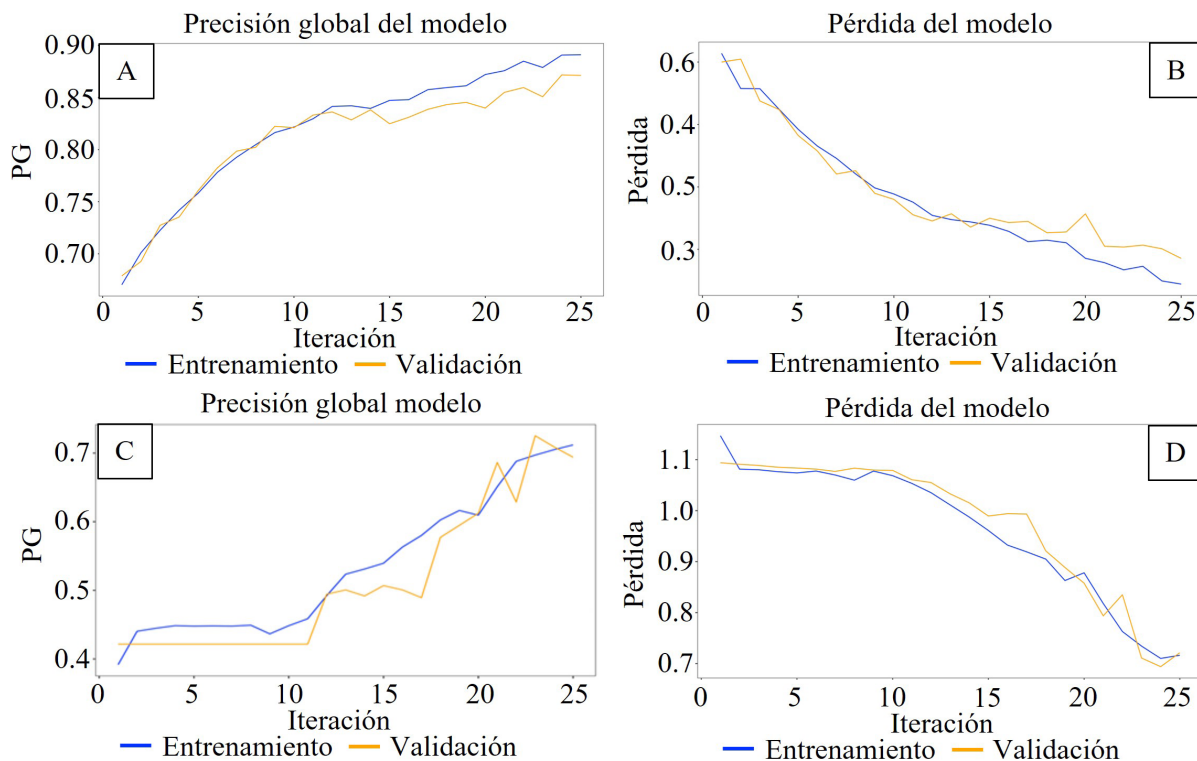


Figura 4. Comparación del desempeño del clasificador CNN en el entrenamiento y validación de la precisión global de clasificación correcta (*PG*) y la pérdida en función del número de iteraciones, con diferentes hiperparámetros. En A y B se utilizaron el optimizador *adam* y *binary\_crossentropy*; en C y D, *adamax* y *categorical\_crossentropy*.

Figure 4. Comparison of CNN classifier performance in training and validation of the overall classification accuracy (*PG*) and loss as a function of the number of iterations, with different hyperparameters. In A and B, the *adam* optimizer y *binary\_crossentropy* was used; in C and D, *adamax* and *categorical\_crossentropy*.

clase A obtuvo los errores de predicción mayores debido a que frutos con antracnosis se clasificaron como sanos. La mayor cantidad de *FP* y *FN* se presentaron en las clases S y A. Esto es, imágenes de frutos con A se clasificaron como S y *viceversa* (Figura 5).

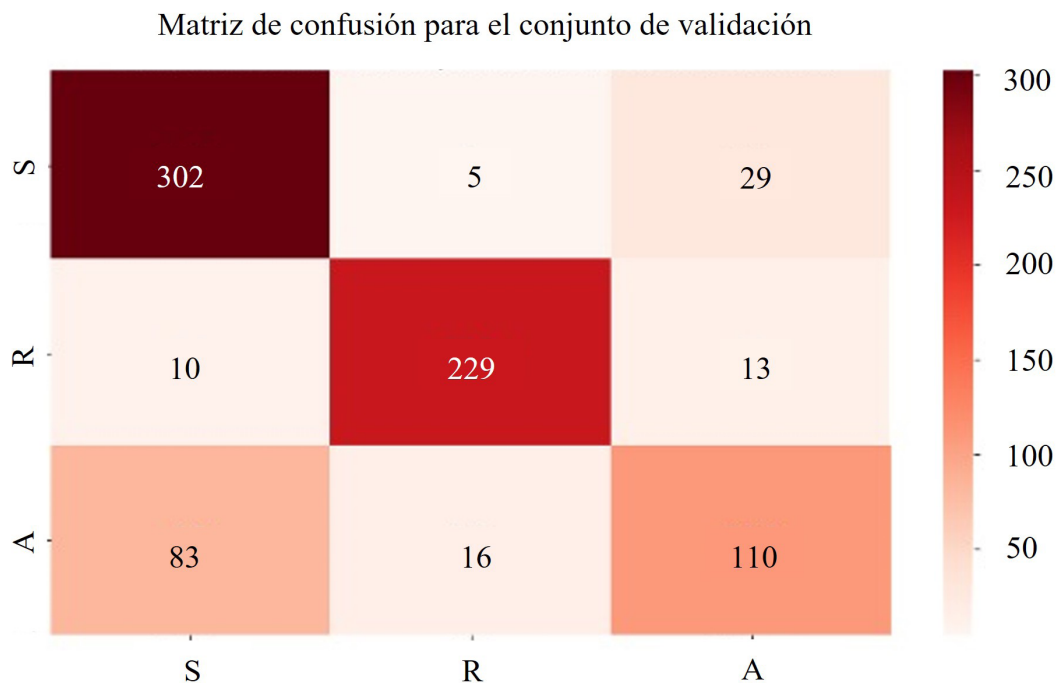
La antracnosis presenta dos tipos de síntomas; el primero son lesiones protuberantes, pequeñas, de color negro, y el segundo son áreas grandes, dañadas en forma dispersa, resultado de la infección en el fruto (Trinidad *et al.*, 2017). En este estudio, la mayoría de las imágenes de frutos con antracnosis son del primer tipo, en etapas tempranas de la enfermedad; por ello, presentaron similitud con los frutos sanos. El total de muestras por clase (S: 336, R: 252, y A: 209) mostró un desbalance entre clases (Figura 5).

Las métricas de desempeño en predicción del modelo CNN *P*, *E*, *F1\_s*, para cada clase objetivo (S, R y A) y *PG* se registraron. El clasificador CNN obtuvo *P* = 92% para identificar la clase R y *P* = 72% para la clase A. La clase R obtuvo un *F1\_score* de 91%, lo que indica un buen desempeño de CNN para identificar la

in training and 0.30 in validation with the *adam* optimizer and *binary\_crossentropy* (Figure 4-B), and 0.70 with the *adamax* optimizer and *categorical\_crossentropy* (Figure 4-D).

The confusion matrix shows the validation (prediction) results of the CNN classifier. Class A had the highest prediction errors because fruit with anthracnose were classified as healthy. The highest number of *FP* and *FN* occurred in classes S and A. That is, images of fruits with A were classified as S and *vice versa* (Figure 5).

Anthracnose presents two types of symptoms; the first are protruding, small, black lesions, and the second are large, scattered damaged areas resulting from infection in the fruit (Trinidad *et al.*, 2017). In this study, most of the images of fruits with anthracnose are of the first type, in early stages of the disease; therefore, they presented similarity with healthy fruits. The total number of samples per class (S: 336, R: 252, and A: 209) showed an imbalance among classes (Figure 5).



**Figura 5.** Matriz de confusión del desempeño en predicción del modelo CNN con el conjunto de validación. Clases objetivo A (antracnosis), R (roña) y S (sanos); filas representan clases observadas y columnas clases predichas. La escala de color aumenta en intensidad de color respecto al número de muestras clasificadas por clase objetivo.

**Figure 5.** Confusion matrix of the prediction performance of the CNN model with the validation set. Target classes A (anthracnose), R (scab) and S (healthy); rows represent observed classes and columns predicted classes. The colour scale increases in colour intensity in regard to the number of samples classified by target class.

roña.  $PG = 80\%$  indica un desempeño aceptable para identificar las tres clases objetivo (Cuadro 2).

Cuando existe un desbalance entre clases en los datos de entrenamiento, los modelos sobre clasifican la clase mayoritaria (frutos sanos). Como resultado, los datos que pertenecen a la clase minoritaria A se clasifican erróneamente con mayor frecuencia que los pertenecientes a la clase S. El desbalance entre clases induce entonces un desempeño menor del clasificador. Además, la métrica de evaluación  $P$  resulta alta y sesga el desempeño del clasificador. Sin embargo, el uso de la métrica  $F1_s$  es un estimador más conservador del desempeño del clasificador (Johnson y Khoshgoftaar, 2019); para el caso de la clase A,  $F1_s$  fue 0.61.

El clasificador se evaluó en predicción con un conjunto prueba de 100 imágenes no utilizadas en el entrenamiento ni en validación del clasificador CNN. Los resultados se describen en una matriz de confusión (Figura 6) y con las métricas para evaluar desempeño (Cuadro 3).

En este caso el clasificador alcanzó una  $PG = 87\%$  que muestra un buen desempeño en predicción; sin embargo, el modelo obtuvo  $P = 57\%$  para la clase A, la cual refleja el efecto del desbalance entre clases para la clase A con menor número de muestras. El modelo mostró eficacia mayor para identificar la clase R, luego la clase S y por último la clase A. El valor de  $F1_s = 0.66$  pondera la diferencia en el desempeño del clasificador CNN entre  $P$  y  $E$ .

Los resultados de la predicción del clasificador CNN se ilustran con ocho imágenes del conjunto de

The prediction performance metrics of the CNN model  $P$ ,  $E$ ,  $F1_s$ , for each target class (S, R and A) and  $PG$  were recorded. The CNN classifier obtained  $P = 92\%$  for identifying the class R and  $P = 72\%$  for the class A. The R class obtained a  $F1_{score} = 91\%$ , which indicates a good performance of CNN for identifying scab.  $PG = 80\%$  indicates an acceptable performance for identifying the three target classes (Table 2).

When there is an imbalance between classes in the training data, the models overclassify the majority class (healthy fruit). As a result, data belonging to the minority class A are misclassified more frequently than those belonging to the class S. The imbalance among classes then induces lower classifier performance. In addition, the evaluation metric  $P$  shift to be high and biases the classifier performance. However, using the metric  $F1_s$  is a more conservative estimator of classifier performance (Johnson and Khoshgoftaar, 2019); for the case of class A,  $F1_s$  was 0.61.

The classifier was evaluated in prediction with a test set of 100 images not used in training nor in validation of the CNN classifier. Results are described in a confusion matrix (Figure 6) and with the metrics to evaluate performance (Table 3).

In this case the classifier achieved a  $PG = 87\%$  which shows a good prediction performance. However, the model obtained  $P = 57\%$  for class A, which reflects the effect of the imbalance between classes for class A with fewer samples. The model showed higher efficiency in identifying the R class, then the S class and finally the A class. The

**Cuadro 2. Métricas de desempeño en predicción del modelo de red neuronal convolucional (CNN) con el conjunto de validación.**  
**Table 2. Prediction performance metrics of the convolutional neural network (CNN) model on the validation set.**

Clases	Métricas			
	$P$	$E$	$F1_s$	$n$
S	0.76	0.90	0.83	336
R	0.92	0.91	0.91	252
A	0.72	0.53	0.61	209
$PG$	0.80			

$P$  es la precisión;  $E$  es la exhaustividad;  $F1_s$  es la media armónica de  $P$  y  $E$ ;  $n$  es el número de imágenes;  $PG$  es la precisión global de clasificación correcta; clases S (sano), R (roña) y A (antracnosis). ♦  $P$ : precision;  $E$ : recall;  $F1_s$ , the harmonic mean of  $P$  and  $E$ ;  $n$ , number of images;  $PG$ , the overall classification accuracy; classes S (healthy), R (scab) and A (anthracnose).

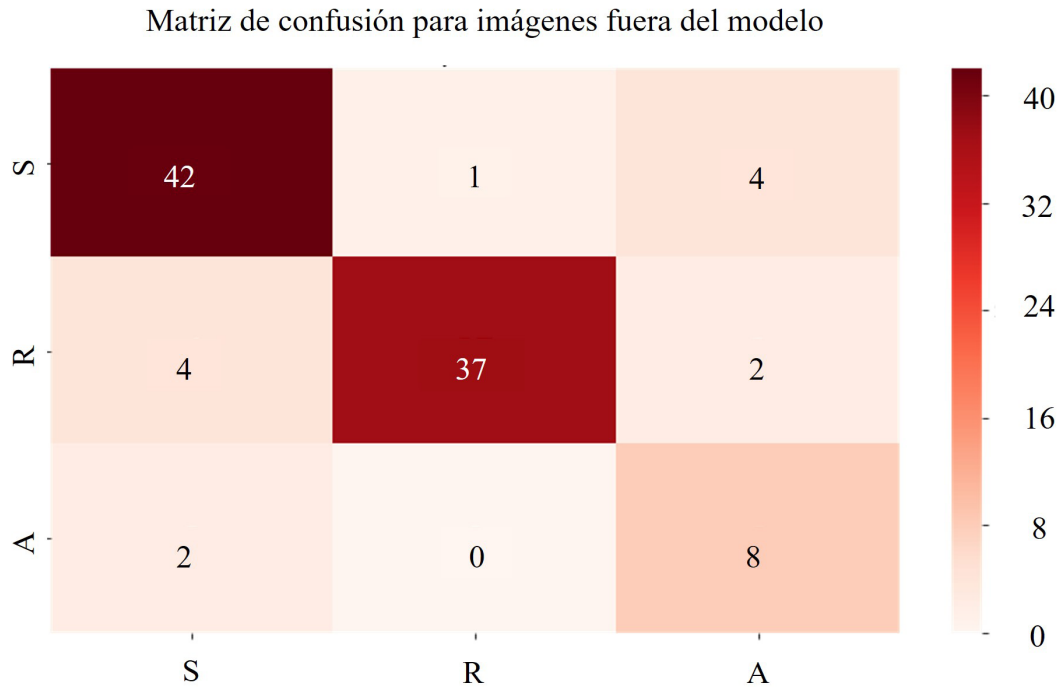


Figura 6. Matriz de confusión de predicción con 100 imágenes no vistas por el modelo; filas representan clases observadas y columnas clases predichas. Clases S, sano; R, roña; y A, antracnosis. La escala indica a mayor intensidad de color, mayor frecuencia de clasificación.

Figure 6. Prediction confusion matrix with 100 images not seen by the model; rows represent observed classes and columns represent predicted classes. Classes: S, healthy; R, scab; and A, anthracnose. The scale indicates the higher the colour intensity, the higher the frequency of classification.

Cuadro 3. Métricas de desempeño en predicción del modelo de red neuronal convolucional (CNN) obtenidas con el conjunto de prueba de 100 imágenes no incluidas en el modelo.

Table 3. Prediction performance metrics of the convolutional neural network (CNN) model, obtained with the test set of 100 images not included in the model.

Clases	Métricas			
	<i>P</i>	<i>E</i>	<i>F1<sub>s</sub></i>	<i>n</i>
S	0.84	0.89	0.86	47
R	0.97	0.86	0.91	43
A	0.57	0.8	0.66	10
<i>PG</i>	0.87			

*P* es la precisión; *E* es la exhaustividad; *F1<sub>s</sub>* es la media armónica de *P* y *E*; *n* es el número de imágenes; *PG* es la precisión global. Las clases S (sano), R (roña) y A (antracnosis). ♦ *P*, precision; *E*, recall; *F1<sub>s</sub>*, the harmonic mean of *P* and *E*; *n*, number of images; *PG*, the overall classification accuracy. The classes S (healthy), R (scab) and A (anthracnose).

validación y ocho imágenes del conjunto de prueba. En texto negro se describen imágenes con clasificación correcta y la probabilidad de pertenencia a cada clase objetivo, y en texto rojo imágenes mal clasificadas (Figura 7).

$F1_s = 0.66$  weights the difference in CNN classifier performance between  $P$  and  $E$ .

The prediction results of the CNN classifier are illustrated with eight images from the validation set and eight images from the test set. Images with

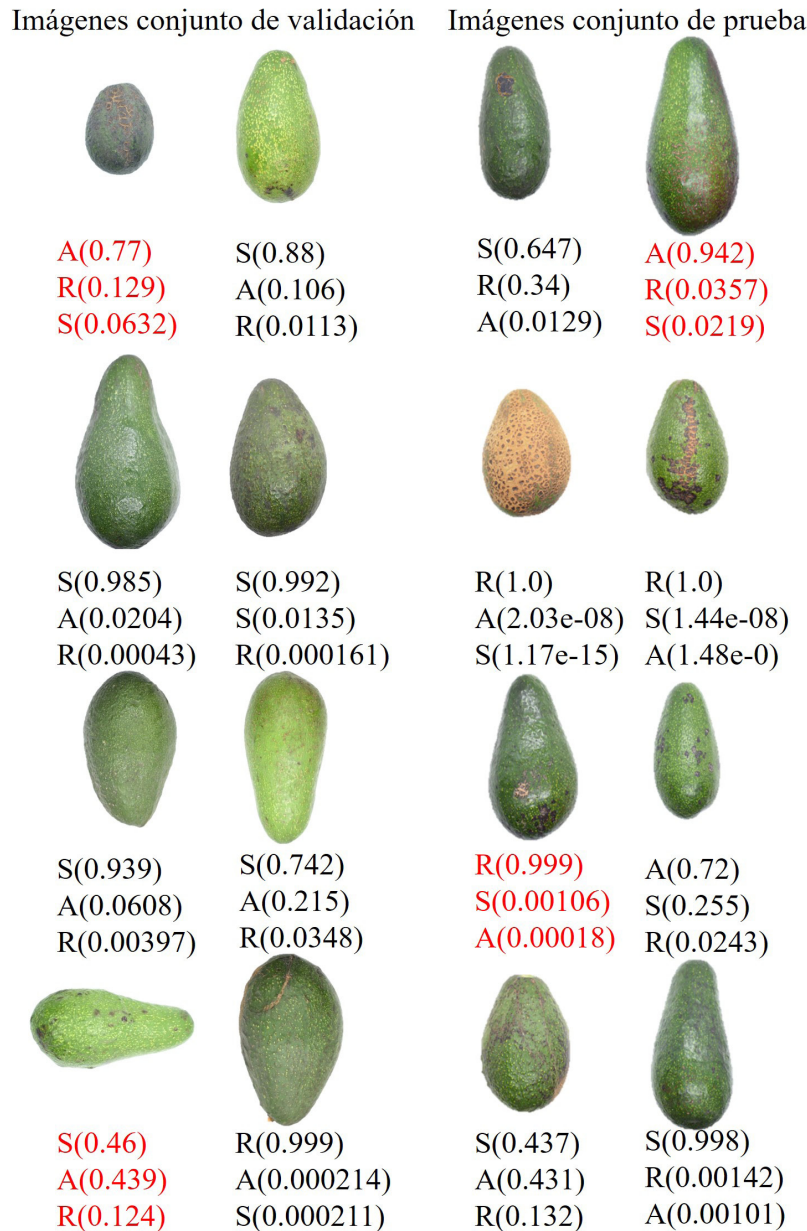


Figura 7. Ejemplos de resultados de la predicción de frutos de aguacate con roña (R), antracnosis (A) o frutos sanos (S) con el modelo de red neuronal convolucional (CNN) con imágenes del conjunto de validación y del conjunto de prueba. Texto en (rojo) indica predicción incorrecta y (negro) correcta. Valores entre paréntesis indican la probabilidad de pertenencia predicha para cada clase objetivo.

Figure 7. Examples of prediction results of avocado fruit with scab (R), anthracnose (A) or healthy fruit (S) with the convolutional neural network (CNN) model with images of the validation set and the test set. Text in (red) indicates incorrect and (black) correct prediction. Values in parentheses indicate the predicted belonging probability for each target class.

En artículos publicados en años recientes, relacionados con la detección y clasificación de enfermedades en plantas, se han obtenido intervalos de precisión de 78.8 a 99.5% a través de técnicas como aprendizaje por transferencia, las cuales son arquitecturas CNN pre entrenadas que se utilizan para la clasificación de imágenes (Qiao *et al.*, 2017; Moolayil, 2019; Saleem *et al.*, 2019). Las principales arquitecturas utilizadas son AlexNet, VGG16, VGG19, ResNet50, Inception V3 (Weiss *et al.*, 2016; Rosebrock, 2017; Ferentinos, 2018).

En este estudio, los resultados de la precisión global de clasificación obtenidos con el clasificador CNN implementado a partir de imágenes de frutos del aguacate, mostraron un desempeño aceptable. El desarrollo del código y la base de datos se encuentran en: <https://github.com/Camposfe1208/Avocado-fruit-diseases-classification>.

## CONCLUSIONES

El modelo de red neuronal convolucional (CNN) alcanzó una precisión global de clasificación correcta de 80% en el conjunto de imágenes de validación y de 87% en el conjunto de imágenes de prueba para identificar frutos de aguacate variedad Fuerte sanos, y con la enfermedad de roña (*Sphaceloma perseae*) y antracnosis (*Colletotrichum* spp.).

El clasificador CNN entrenado obtuvo en el conjunto de prueba en predicción una media armónica entre precisión y exhaustividad (F1<sub>s</sub>) de 91% para identificar frutos con roña; de 66% para identificar frutos con antracnosis y de 86% para identificar frutos sanos. Este estudio comprueba el uso potencial de inteligencia artificial a través de algoritmos de redes convolucionales para realizar la identificación de enfermedades en el fruto de aguacate a partir de imágenes digitales en la etapa de postcosecha.

## LITERATURA CITADA

- Brahimi, M., M. Arsenovic, S. Laraba, S. Sladojevic, K. Boukhalifa, and A. Moussaoui. 2018. Deep learning for plant diseases: detection and saliency map visualization. *In*: Zhou, J., and F. Chen (eds). Human and Machine Learning: Visible, Explainable, Trustworthy and Transparent. Charm. Springer International Publishing. Cham, Switzerland. pp: 93-117.
- Chollet, F. 2018. Deep Learning with Python. New York. Manning Publications Co. 360 p.
- Crane, J. H., G. Douhan, B. A. Faber, M. L. Arpaia, G. S. Bender, C. F. Balerdi, and A. F. Barrientos-Priego. 2013. Cultivars

correct classification and the probability of belonging to each target class are described in black text, and misclassified images are described in red text (Figure 7).

In papers published in recent years, related to plant disease detection and classification, accuracy ranges from 78.8 to 99.5% have been obtained through techniques such as transfer learning, which are pre-trained CNN architectures used for image classification (Qiao *et al.*, 2017; Moolayil, 2019; Saleem *et al.*, 2019). The main architectures used are AlexNet, VGG16, VGG19, ResNet50, Inception V3 (Weiss *et al.*, 2016; Rosebrock, 2017; Ferentinos, 2018).

In this study, the overall accuracy results obtained with the CNN classifier implemented from avocado fruit images showed acceptable performance. The code development and database can be found at: <https://github.com/Camposfe1208/Avocado-fruit-diseases-classification>

## CONCLUSIONS

The convolutional neural network (CNN) model achieved an overall classification accuracy of 80% in the validation image set and 87% in the test image set for identifying avocado cv. Fuerte fruits, healthy and with diseases, scab (*Sphaceloma perseae*) and anthracnose (*Colletotrichum* spp.).

The trained CNN classifier obtained in the prediction test set a harmonic mean between accuracy and recall (F1<sub>s</sub>) of 91% for identifying fruits with scab; 66% for identifying fruits with anthracnose and 86% for identifying healthy fruits. This study affirms the potential use of artificial intelligence through convolutional network algorithms to identify avocado fruit diseases from digital images at the postharvest stage.

—End of the English version—

---\*---

- and rootstocks. *In*: Schaffer, B., N. B. Wolstenholme, and A. W. Whiley (eds). *The Avocado: Botany, Production and Uses*. 2nd ed. CAB International. Wallingford, UK. pp: 200-233.
- Dann, E. K., R. C. Ploetz, L. M. Coates, and K. G. Pegg. 2013. Foliar, fruit and soilborne diseases. *In*: Schaffer, B., N. B. Wolstenholme, and A. W. Whiley (eds). *The Avocado: Botany, Production and Uses*. 2nd ed. CAB International. Wallingford, UK. pp: 380-422.
- Dhakal, A., and S. Shakya. 2018. Image-based plant disease detection with deep learning. *International Journal of Computer Trends and Technology*. 61: 26-29.
- Ferentinos, K. P. 2018. Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*. 145: 311-318.
- Harpale, D., S. Jadhav, K. Lakhani, and K. Thyagarajane. 2020. Plant disease identification using image processing. *International Research Journal of Engineering and Technology*. 7: 3571-3573.
- Johnson, J. M., and T. M. Khoshgoftaar. 2019. Survey on deep learning with class imbalance. *Journal of Big Data* 6: 1-54.
- Loey, M., A. ElSawy, and M. Afify. 2020. Deep learning in plant diseases detection for agricultural crops: A Survey. *Int. J. Serv. Sci. Manag. Eng. Tech*. 11: 41-58.
- Mohanty, S. P., D. P. Hughes, and M. Salathé. 2016. Using deep learning for image-based plant disease detection. *Frontiers in Plant Science* 7: 1419. <https://doi.org/10.3389/fpls.2016.01419>
- Moolayil, J. 2019. *Learn Keras for Deep Neural Networks: a fast-track approach to modern deep learning with Python*. A Press. Vancouver, Canada. 182 p.
- Paull, R. E., and O. Duarte. 2011. *Avocado*. *In*: Paull, R. E., and O. Duarte (eds). *Tropical Fruits: Vol. 1*. CAB International. Wallingford, UK. pp: 153-184.
- Qiao, Y., J. Shen, T. Xiao, Q. Yang, M. Wen, and C. Zhang. 2017. FPGA-accelerated deep convolutional neural networks for high throughput and energy efficiency. *Concurrency and Computation: Practice and Experience* 29: 1-20.
- Rosebrock, A. 2017. *Deep Learning for Computer Vision with Python: Starter bundle*. PyImageSearch. 330 p.
- SAGARPA (Secretaría de Agricultura, Ganadería, Pesca y Alimentación). 2017. *Planeación Agrícola Nacional 2017-2030: Aguacate mexicano*. Ciudad de México, México. 16 p.
- Saleem, M. H., J. Potgieter, and K. M. Arif. 2019. Plant disease detection and classification by deep learning. *Plants*. 8: 1-22.
- SIAP (Servicio de Información Agrícola y Pesquera). 2020. *Panorama Agroalimentario*. Ciudad de México, México. pp: 26-27.
- SIAP (Servicio de Información Agrícola y Pesquera). 2021. *Anuario Estadístico de la Producción Agrícola*. <https://nube.siap.gob.mx/cierreagricola/> (Consulta: noviembre 2021).
- Thangaraj, R., D., Dinesh, S. Hariharan, S. Rajendar, D. Gokul, T.R. Hariskarthi. 2020. Automatic recognition of avocado fruit diseases using modified deep convolutional neural network. *International Journal of Grid and Distributed Computing*. 13: 1550-1559.
- Toda, Y., and F. Okura (2019) How convolutional neural networks diagnose plant disease. *Plant Phenomics* 2019: 9237136. <https://doi.org/10.34133/2019/9237136>
- Trinidad Á., E., F. J. Ascencio V., J. A. Ulloa, J. C. Ramírez R., J. A. Ragazzo S., M. Calderón S., y P. U. Bautista R. 2017. Identificación y caracterización de *Colletotrichum* spp. causante de antracnosis en aguacate Nayarit, México. *Revista Mexicana de Ciencias Agrícolas*. 19: 3953-3964.
- Türkoğlu, M., and D. Hanbay. 2019. Plant disease and pest detection using deep learning-based features. *The Turkish Journal of Electrical Engineering & Computer Sciences*. 27: 1636-1651.
- Weiss, K., T. M. Khoshgoftaar, and D. Wang. 2016. A survey of transfer learning. *Journal of Big Data*, sin punto al final. 3: 1-40.

