Scientific article https://doi.org/10.5154/r.rchsh.2023.04.002

Automatic identification of avocado fruit diseases based on machine learning and chromatic descriptors

Identificación automática de enfermedades en frutos de aguacate con base en máquinas de aprendizaje y descriptores cromáticos

Ulises Enrique Campos-Ferreira; Juan Manuel González-Camacho*; Alfredo Carrillo-Salazar

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

*Corresponding author: jmgc@colpos.mx

Abstract

Timely identification of phytosanitary problems in agricultural crops is essential to reduce production losses. Artificial intelligence algorithms facilitate their rapid and reliable identification. In this research, three learning classifiers, namely random forest (RF), support vector machine (SVM) and multilayer perceptron (MLP), were evaluated to identify three target classes (healthy fruit, anthracnose [*Colletotrichum* spp.] and scab [*Sphaceloma perseae*]) from digital fruit images. Two color descriptor extraction techniques (region selection and image subsampling) were compared with the RF classifier, and an overall classification accuracy (ACC) of 98±0.03 % with region selection and 84±0.08 % with subsampling was obtained. Subsequently, the classifiers were evaluated with color descriptors extracted with region selection. RF and MLP were superior to SVM, with an ACC of 98±0.03 %. Scab and anthracnose were identified with an F1 score of 98 %. The high performance of the classifiers shows the potential for applying artificial intelligence paradigms to identify phytosanitary problems in agricultural crops.

Keywords: *Persea americana*, *Sphaceloma perseae*, *Colletotrichum* spp., machine learning, artificial intelligence.

Resumen

La identificación oportuna de problemas fitosanitarios en cultivos agrícolas es esencial para reducir pérdidas de producción. Los algoritmos de inteligencia artificial facilitan su identificación rápida y confiable. En esta investigación, se evaluaron tres clasificadores de aprendizaje: bosque aleatorio (RF), máquina de soporte vectorial (SVM) y perceptrón multicapa (MLP), para identificar tres clases objetivo (frutos sanos, antracnosis [*Colletotrichum* spp.] y roña [*Sphaceloma perseae*]) a partir de imágenes digitales de frutos. Se compararon dos técnicas de extracción de descriptores de color (selección por región y submuestreo de imágenes) con el clasificador RF, y se obtuvo una precisión global de clasificación (ACC) de 98±0.03 % con selección por región, y de 84±0.08 % con submuestreo. Posteriormente, los clasificadores se evaluaron con descriptores de color extraídos con selección por región. RF y MLP fueron superiores a SVM, con una ACC de 98±0.03 %. La roña y la antracnosis se identificaron con un puntaje F1 de 98 %. El alto desempeño de los clasificadores muestra el potencial de aplicación de los paradigmas de inteligencia artificial para identificar problemas fitosanitarios en cultivos agrícolas.

Palabras clave: *Persea americana*, *Sphaceloma perseae*, *Colletotrichum* spp., aprendizaje automático, inteligencia artificial.

Please cite this article as follows (APA 6): Campos-Ferreira, U. E., González-Camacho, J. M., & Carrillo-Salazar, A. (2023). Automatic identification of avocado fruit diseases based on machine learning and chromatic descriptors *Revista Chapingo Serie Horticultura, 29*(3), 115-130. doi: 10.5154/r.rchsh.2023.04.002

Revista Chapingo Serie Horticultura

Received: March 24, 2023 / Accepted: June 24, 2023

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

Introduction

Mexico is the world's leading producer and exporter of avocado, with production exceeding two million tons in 2021 (Servicio de Información Agroalimentaria y Pesquera [SIAP], 2022); however, postharvest loss is a factor that affects marketing and food security. Two fungal diseases of commercial interest are reported in avocado fruit: scab (*Sphaceloma perseae*) and anthracnose (*Colletotrichum* spp.). Improper management of avocado orchards can cause losses of more than 70 % before harvest, and total postharvest losses if environmental and biological conditions exist for the development of these diseases (Téliz & Mora, 2019). Early detection of phytosanitary problems is an essential step to ensure food safety. In the literature, automatic systems with the ability to detect diseases in plant species based on digital images have been proposed (Saleem, Potgieter, & Arif, 2019).

The machine learning (ML) paradigm is widely used to detect diseases. A basic task in ML is the supervised classification or prediction of a discrete response variable. Input data are associated with each target class to form a training set, and the model learns to predict the target classes on an unseen data set (Ketkar & Moolayil, 2021).

In agriculture, different studies have been conducted using ML algorithms for automatic disease detection in plants. Among the paradigms applied are the support vector machine (SVM) (Sandhya, Balasundaram, & Arunkumar, 2022), random forest (RF) (Srinivasa, Venkata, Anusha, Sai, & Bhanu, 2022) and multilayer perceptron (MLP) (Chen, Dewi, Huang, & Caraka, 2020). These algorithms have been applied to different plant organs, such as leaves, roots, or fruits (Doh et al., 2019).

A deep learning classifier allows automatic feature extraction from an input dataset and reduces user error for feature selection. In general, high overall classification accuracy levels are achieved with this approach when large datasets are available. In agriculture, this has been used in disease identification, crop classification, and yield prediction (Alzubaidi et al., 2021).

The aim of this research was to identify anthracnose-infected, scab-infected and healthy fruits (three target classes) by means of three machine learning models (RF, SVM and MLP) and extracted chromatic descriptors (region selection [BD1] and image subsampling [BD2]) from digital avocado fruit images.

Materials and methods

Database

The set of avocado fruit images used in this study (30 per target class) were selected with different disease

Introducción

México es el principal productor y exportador de aguacate del mundo, con una producción superior a dos millones de toneladas en 2021 (Servicio de Información Agroalimentaria y Pesquera [SIAP], 2022); sin embargo, la pérdida en poscosecha es un factor que afecta la comercialización y la seguridad alimentaria. En los frutos de aguacate se reporta la presencia de dos enfermedades fungosas de interés comercial: la roña (*Sphaceloma perseae*) y la antracnosis (*Colletotrichum* spp.). El manejo inadecuado de los huertos de aguacate puede ocasionar pérdidas superiores a 70 % antes de la cosecha, y pérdidas totales en poscosecha si existen condiciones ambientales y biológicas para el desarrollo de estas enfermeda-des (Téliz & Mora, 2019). La detección temprana de problemas fitosanitarios es una etapa esencial para garantizar la seguridad alimentaria. En la literatura, se han propuesto sistemas automáticos con capacidad para detectar enfermedades en especies vegetales con base en imágenes digitales (Saleem, Potgieter, & Arif, 2019).

El paradigma de aprendizaje automático (ML, *machine learning*) es ampliamente utilizado para detectar enfermedades. Una tarea básica en ML es la clasificación o predicción supervisada de una variable respuesta discreta. Los datos de entrada se asocian con cada clase objetivo para formar un conjunto de entrenamiento, y el modelo aprende a predecir las clases objetivo sobre un conjunto de datos no vistos (Ketkar & Moolayil, 2021).

En la agricultura, se han reportado diferentes trabajos que utilizan algoritmos de ML para la detección automática de enfermedades en plantas. Entre los paradigmas aplicados se tienen los algoritmos máquina de soporte vectorial (SVM, *support vector machine*) (Sandhya, Balasundaram, & Arunkumar, 2022), bosque aleatorio (RF, *random forest*) (Srinivasa, Venkata, Anusha, Sai, & Bhanu, 2022) y perceptrón multicapa (MLP, *multilayer perceptron*) (Chen, Dewi, Huang, & Caraka, 2020). Estos algoritmos se han aplicado a diferentes órganos de la planta, como hojas, raíces o frutos (Doh et al., 2019).

Un clasificador de aprendizaje profundo (*deep learning*) permite la extracción automática de características a partir de un conjunto de datos de entrada, y reduce el error del usuario para la selección de éstas. En general, con este enfoque se alcanzan altos niveles de precisión global de clasificación cuando se dispone de grandes conjuntos de datos. En la agricultura, esto se ha utilizado en la identificación de enfermedades, clasificación de cultivos y predicción de rendimientos (Alzubaidi et al., 2021).

El objetivo de esta investigación fue identificar frutos infectados con antracnosis, con roña y frutos sanos (tres clases objetivo) por medio de tres modelos de

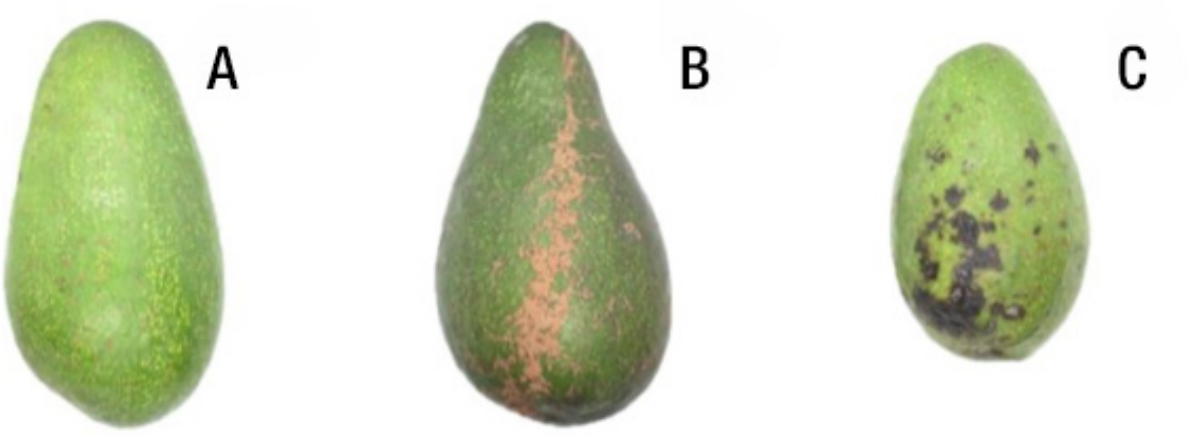

Figure 1. Target classes used in this study: A) healthy fruits, B) fruits with scab, and C) fruits with anthracnose.
Figura 1. Clases objetivo utilizadas en el presente trabajo: A) frutos sanos, B) frutos con roña y C) frutos con antracnosis.

levels from a database with 569 digital images of 256 x 256 pixels in size, belonging to avocado fruits of the Fuerte variety reported in a previous study (Campos-Ferreira & González-Camacho, 2021) (https://www.kaggle.com/datasets/camposfe1/clasifiacin-de-enfermedades-del-aguacatero).

Images of the fruits were captured in the laboratory under homogeneous lighting conditions. The target classes were healthy fruits (H), scab (S, *Sphaceloma perseae*) and anthracnose (A, *Colletotrichum* spp.). Fruits were obtained in the municipalities of Ocuituco (18° 51' 47.6" NL and 98° 46' 49.1" WL) and Tetela del Volcán (18° 53' 36.3" NL and 98° 43' 47.7" WL) in the state of Morelos. Fuerte variety fruits, when ripe, are green, which allows contrasting the lesions caused by the two diseases mentioned (Figure 1).

Software and hardware

Python programming language ver. 3.9 was used as the programming platform, and the scikit-learn ver. 1.0.2 library was used to implement the machine learning classifiers. Training and validation of the classifiers were performed in Spyder development environment ver. 4.2.1 and the MacOs Monterey (12.6) operating system on a Mac mini (Apple) computer, which uses an ARM (M1) chip with 3.2 GHz processing speed and 8 GB of RAM.

Image processing

The set of images represented with the RGB (red, blue and green) color model was transformed to the HSV (hue, saturation and value) color model. This is because the HSV representation is more appropriate than RGB for extracting color features, and allows for improving the performance of learning classifiers (Abdel-Hamid, 2019).

aprendizaje automático (RF, SVM y MLP) y descriptores cromáticos extraídos (selección por región [BD1] y submuestreo de imágenes [BD2]) de imágenes digitales de frutos de aguacate.

Materiales y métodos

Base de datos

El conjunto de imágenes de frutos de aguacate que se utilizaron en este estudio (30 por clase objetivo) se seleccionaron con diferentes niveles de las enfermedades de una base de datos con 569 imágenes digitales de tamaño 256 x 256 píxeles, pertenecientes a frutos de aguacate variedad Fuerte reportadas en un estudio previo (Campos-Ferreira & González-Camacho, 2021) (https://www.kaggle.com/datasets/camposfe1/clasifiacin-de-enfermedades-del-aguacatero).

Las imágenes de los frutos se capturaron en laboratorio y bajo condiciones homogéneas de iluminación. Las clases objetivo fueron: frutos sanos (S), roña (R, *Sphaceloma perseae*) y antracnosis (A, *Colletotrichum* spp.). Los frutos se obtuvieron en los municipios de Ocuituco (18° 51' 47.6" LN y 98° 46' 49.1" LO) y Tetela del Volcán (18° 53' 36.3" LN y 98° 43' 47.7" LO) en el estado de Morelos. Los frutos de la variedad Fuerte, en su periodo de madurez, son de color verde, lo cual permite contrastar las lesiones provocadas por las dos enfermedades mencionadas (Figura 1).

Software and *hardware*

El lenguaje de programación *Python* ver. 3.9 se utilizó como plataforma de programación, y la librería *scikit-learn* ver. 1.0.2 se utilizó para implementar los clasificadores de aprendizaje automático. El entrenamiento y validación de los clasificadores se realizaron en el ambiente de desarrollo *Spyder* ver. 4.2.1 y el sistema

Color feature extraction

The extraction of color features (descriptors) was carried out using two techniques: BD1 and BD2. This was done with the help of the IDENTO program (Ambrosio-Ambrosio, González-Camacho, Rojano-Aguilar, & del Valle-Paniagua, 2023) to create the input datasets from the images for each target class. The BD1 dataset was created using the region extraction technique. This consisted of selecting, in each image, a region of interest, and from a seed, or starting, pixel, the algorithm extracts a set of pixels similar to the seed, where each pixel is represented by the H, S and V color channels. IDENTO allowed generating, for each target class, a set of 45,536 value triplets (H, S and V) (Table 1).

The BD2 dataset was created using the technique of extraction by subsampling a box-shaped section of the image with the area of interest. For this, with the aid of the IDENTO software, three rectangular areas (30 × 30 pixels) with the representative color of each target class were selected. In total, 90 image samples per target class (270 in total) were obtained; subsequently, the total set of 39,675 value triplets (H, S and V), associated with each target class, was created (Table 1).

In both datasets, repeated triplets within and between target classes were removed. The sets of value triplets (H, S, and V), and their associated target class, were exported and saved to a .csv format file for use in training and testing with the RF classifier.

operativo MacOs Monterey (12.6) en una computadora Mac mini (Apple), la cual utiliza un chip ARM (M1) con velocidad de procesamiento de 3.2 GHz y 8 GB de RAM.

Procesamiento de imágenes

El conjunto de imágenes representadas con el modelo de color RGB (*red*, *blue* y *green*) se transformaron al modelo de color HSV (*hue*, *saturation* y *value*). Lo anterior debido a que la representación HSV es más apropiada que RGB para extraer características de color, y permite mejorar el desempeño de los clasificadores de aprendizaje (Abdel-Hamid, 2019).

Extracción de características de color

La extracción de características (descriptores) de color se realizó por medio de dos técnicas: BD1 y BD2. Esto se realizó con la ayuda del programa IDENTO (Ambrosio-Ambrosio, González-Camacho, Rojano-Aguilar, & del Valle-Paniagua, 2023) para crear los conjuntos de datos de entrada a partir de las imágenes por cada clase objetivo. El conjunto de datos BD1 se creó con la técnica de extracción por región. Esta consistió en seleccionar, en cada imagen, una región de interés, y a partir de un píxel semilla, o de inicio, el algoritmo extrae un conjunto de píxeles similares a la semilla, donde cada píxel se representa por los canales de color H, S y V. IDENTO permitió generar, para cada clase objetivo, un conjunto de 45,536 tripletas de valores (H, S y V) (Cuadro 1).

Table 1. Description of the input datasets (pixels or H, S and V color descriptors) obtained with the region extraction (BD1 and image subsampling (BD2) techniques.

Cuadro 1. Descripción de los conjuntos de datos de entrada (pixeles o descriptores de color H, S y V) obtenidos con las técnicas de extracción por región (BD1) y submuestreo de imágenes (BD2).

Class / Clase	BD1	BD2
H / S	15,213	17,746
S / R	14,836	13,416
A	15,487	8,513
Total	45,536	39,675

H = healthy fruits; S = fruits with scab; A = fruits with anthracnose.

S = frutos sanos; R = frutos con roña; A = frutos con antracnosis.

Random forest (RF)

The RF classifier is a supervised ensemble learning model. In the case of a classification problem, this algorithm uses decision trees based on a condition, and each tree obtains a value to classify the data. Each tree casts a unit vote, and the choice of the best decision tree is made according to the one with the highest number of votes from the entire forest (Parmar, Katariya, & Patel, 2019).

In RF, each tree is constructed from a randomly drawn sample, with replacement, from the training dataset (bootstrap); subsequently, multiple training sets are produced with values other than the initial set. From each sample, a model is built (bagging) and the data from the original sample is inputted, its predicted class is determined and the difference with the actual value is analyzed, thus obtaining the classification error. Ensemble models allow for reducing the variance of the RF estimator and avoiding overfitting of the model (Knauer et al., 2019).

In an RF model, the aim is to maximize the information gain, which is defined by:

$$IG(D_p,f) = I(D_p) - \sum^{m}_{j=1} \frac{N_j}{N_p} I(D_j)$$

where f is the condition dividing the parent node, D_p and D_j belong to the data of the parent node and the j-th child, I is the impurity metric, N_p is the number of samples in the parent node and N_j is the number of samples from the j-th child. The information gain is the difference between the impurity of the parent node and the sum of the impurities of the child nodes; the lower the impurities of the child nodes, the greater the information gain (Raschka, Liu, & Mirjalili, 2022).

An objective function used in RF is the Gini criterion (I_G) and is calculated as:

$$I_G = 1 - \sum_{k=1}^{K} (p_{mk})^2$$

where K corresponds to the number of target classes an $\hat{p}_{mk}$ represents the proportion of training observations in the m-th region that belong to the k-th class of interest (Géron, 2022). Another criterion is entropy (I_E), which is defined by:

$$I_E = -\sum_{k=1}^{K} \hat{p}_{mk} log_2(\hat{p}_{mk})$$

In this work, the *random forest classifier* function was used, which takes as its main arguments the number of trees (*n_estimators*, NE), the criterion (*criterion*, Cr), tree depth

El conjunto de datos BD2 se creó con la técnica de extracción por submuestreo de un recuadro de la imagen con el área de interés. Para ello, con la ayuda del programa IDENTO, se seleccionaron tres áreas rectangulares (30 x 30 píxeles) con el color representativo de cada clase objetivo. En total, se obtuvieron 90 muestras de imágenes por clase objetivo (270 en total); posteriormente, se creó el conjunto total de 39,675 tripletas de valores (H, S y V), asociadas a cada clase objetivo (Cuadro 1).

En ambos conjuntos de datos, se eliminaron las tripletas repetidas dentro de clases y entre clases objetivo. Los conjuntos de tripletas de valores (H, S y V), y su clase objetivo asociada, se exportaron y guardaron en un archivo con formato .csv para su uso en el entrenamiento y prueba con el clasificador RF.

Bosque aleatorio (RF)

El clasificador RF es un modelo de aprendizaje supervisado de ensamble. En el caso de un problema de clasificación, este algoritmo utiliza árboles de decisión a partir de una condición, y cada árbol obtendrá un valor para clasificar los datos. Cada árbol emite un voto unitario, y la elección del mejor árbol de decisión se hace de acuerdo con el que tenga el mayor número de votos de todo el bosque (Parmar, Katariya, & Patel, 2019).

En RF, cada árbol se construye a partir de una muestra extraída de manera aleatoria, con reemplazo, del conjunto de datos de entrenamiento (*bootstrap*); posteriormente, se producen múltiples conjuntos de entrenamiento con valores distintos al conjunto inicial. A partir de cada muestra, se construye un modelo (*bagging*) y se introducen los datos de la muestra original, se determina su clase predicha y se analiza la diferencia con el valor real, con lo cual se obtiene el error de clasificación. Los modelos de ensamble permiten reducir la varianza del estimador de RF y evitar el sobre-ajuste del modelo (Knauer et al., 2019).

En un modelo RF se busca maximizar la ganancia de información, la cual está definida por:

$$GI(D_p,f) = I(D_p) - \sum^{m}_{j=1} \frac{N_j}{N_p} I(D_j)$$

donde f es la condición que divide el nodo padre, D_p y D_j pertenecen a los datos del nodo padre y del j-ésimo hijo, I es la métrica de impureza, N_p es el número de muestras en el nodo padre y N_j es el número de muestras del j-ésimo hijo. La ganancia de información es la diferencia entre la impureza del nodo padre y la suma de las impurezas de los nodos hijo, mientras más bajas sean las impurezas de los nodos hijo, más grande es la ganancia de información (Raschka, Liu, & Mirjalili, 2022).

(*max_depth*, MD) and maximum features (*max_features*, MF) (Raschka et al., 2022). These hyperparameters were used to optimize the RF model based on the following intervals and values (selected by trial and error): NE = 100, 200, 300 and 1000; Cr = '*gini*' and '*entropy*'; MD = 4, 6 and 8; MF = '*sqrt*' and 'auto'.

Support vector machine (SVM)

SVM is a model that associates the data from the original set, from an input space with a high-dimensional feature space, making it simpler in the feature space. This is done to optimally separate the classes in a hyperplane, minimize the generalization error and maximize the margin.

In case the classes are separable from each other from a linear classifier, the SVM model determines the hyperplane that minimizes the generalization error (by means of a test dataset). Otherwise, when at least one class is not separable from the others, SVM tries to find the hyperplane that maximizes the margin and, at the same time, minimizes it by an amount proportional to the number of misclassifications. Based on the selected hyperplane, the maximum margin between classes will be obtained, that is, the sum of the distances between the separation hyperplane and the closest points for each class (Das, Singh, Mohanty, & Chakravarty, 2020).

The equation of a hyperplane is defined as $w \times x + b = 0$, where w is a normal vector to the hyperplane and b is an offset. In the case of a multiclass classification, the following optimization problem is used:

$$\min_{w^j,\xi_i^j} \frac{1}{2} \|w^j\|^2 + c \sum_{i=1}^{n} \xi_i^j$$

where ξ_i ($i = 1, \ldots, n$) are slack variables, which may allow some data to remove the constraints that define the minimum margin required for the training data for a separable case. C is a user-defined penalty parameter to control the margin of error of the training set; the larger the value of C, the larger the penalty. On the other hand, the j-th SVM is trained with the data in the j-th class with labels belonging to the class, and the other classes with labels different from the class (Pisner & Schnyer, 2020).

There is a kernel function, which is used to transform the training data set so that a nonlinear decision surface is transformed into a linear equation in a larger number of dimensional spaces. In general, this function returns the inner product between two points in a feature dimension. The most commonly used kernels are linear:

$$K(x_i, x_j) = x_i \times x_j$$

Una función objetivo que se utiliza en RF es el criterio de Gini (I_G) y se calcula como:

$$I_G = 1 - \sum_{k=1}^{K} (p_{mk})^2$$

donde K corresponde al número de clases objetivo y $\hat{p}_{mk}$ representa la proporción de observaciones de entrenamiento en la m-ésima región que pertenecen a la k-ésima clase de interés (Géron, 2022). Otro criterio es la entropía (I_E), la cual está definida por:

$$I_E = -\sum_{k=1}^{K} \hat{p}_{mk} log_2(\hat{p}_{mk})$$

En este trabajo, se utilizó la función *randomforestclassifier*, que toma como argumentos principales el número de árboles (*n_estimators*, NE), el criterio (*criterion*, Cr), la profundidad del árbol (*max_depth*, MD) y las características máximas (*max_features*, MF) (Raschka et al., 2022). Estos hiperparámetros se utilizaron para optimizar el modelo RF con base en los siguientes intervalos y valores (seleccionados por prueba y error): NE = 100, 200, 300 y 1000; Cr = '*gini*' y '*entropy*'; MD = 4, 6 y 8; MF = '*sqrt*' y 'auto'.

Máquina de soporte vectorial (SVM)

SVM es un modelo que asocia los datos del conjunto original, a partir de un espacio de entrada con un espacio de características de alta dimensión, volviéndolo más simple en el espacio de características. Esto se realiza para separar, de manera óptima, las clases en un hiperplano y minimizar el error de generalización, además de maximizar el margen.

En caso de que las clases sean separables entre sí a partir de un clasificador lineal, el modelo SVM determina el hiperplano que minimiza el error de generalización (mediante un conjunto de datos de prueba). En caso contrario, cuando al menos una clase no es separable de las otras, SVM intenta buscar el hiperplano que maximice el margen y, al mismo tiempo, minimice a una cantidad proporcional al número de clasificaciones incorrectas. Con base en el hiperplano seleccionado, se tendrá el máximo margen entre clases; es decir, la suma de las distancias entre el hiperplano de separación y los puntos más cercanos para cada clase (Das, Singh, Mohanty, & Chakravarty, 2020).

La ecuación de un hiperplano se define como $w \times x + b = 0$, donde w es un vector normal al hiperplano y b es un sesgo. En el caso de una clasificación multiclase, se utiliza el siguiente problema de optimización:

$$\min_{w^j,\xi_i^j} \frac{1}{2} \|w^j\|^2 + c \sum_{i=1}^{n} \xi_i^j$$

And the Radial basis function (RBF) kernel, also known as the Gaussian kernel:

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

where $\sigma > 0$ is the parameter controlling the kernel width. This expression can be simplified as follows:

$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right)$$

where $\gamma = \frac{1}{2\sigma^2}$. This value can be understood as the cutoff, and is defined by the user; in it, the influence or scope of the training samples is increased, which leads to a soft or narrow decision boundary (Raschka et al., 2022).

Classification methods are included in the scikit-learn program library, within the *svm* module. In the present work, the support vector classification (SVC) function was used, which takes as its main hyperparameters the penalty or *C* value (C), the *gamma* value (γ) and the kernel mask size (K). By trial and error, the following intervals and values were defined: C = 0.01, 0.1, 1, 10 and 100; γ = 0.001, 0.1, 1, 10 and 100; kernels = '*rbf* ' and '*linear*' (Raschka et al., 2022).

Multilayer perceptron (MLP)

In MLP, the basic element is known as an artificial neuron. This artificial neuron, of the feed forward type, consists of input and output elements that are processed in the central unit. The input layers depend on the data used for training, while the hidden layers and the output layers pertain to the number of classes of interest.

The MLP architecture consists of an input layer, one or more hidden layers, and an output layer. The input layer depends on the number of input data, the hidden layers represent the level of complexity that exists between the input and output layer, and the output layer represents the number of target classes and gives the predicted class (Edmond & Girsang, 2020).

The function that transforms the input data is known as the activation function. The most common is the ReLU (Rectified Linear Unit) function, which is expressed as:

$$f(z) = \max(0, z)$$

where the response is z if the input is positive and *0* if it is negative. ReLU is used to filter the data in the intermediate layers. In the output layer, the softmax activation function is used, which is expressed as:

$$p(z) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}}$$

donde ξ_i ($i = 1, \ldots, n$) son variables de holgura, las cuales pueden permitir algunos datos para eliminar las limitaciones que definen el margen mínimo requerido para los datos de entrenamiento para un caso separable. *C* es un parámetro de penalización definido por el usuario para controlar el margen de error del conjunto de entrenamiento; mientras más grande sea el valor de *C*, más grande es la penalización. Por otra parte, el j-ésimo SVM es entrenado con los datos en la j-ésima clase con etiquetas pertenecientes a la clase, y las demás clases con etiquetas diferentes a la clase (Pisner & Schnyer, 2020).

Existe una función *kernel*, la cual es utilizada para transformar el conjunto de datos de entrenamiento, para que una superficie de decisión no lineal se transforme en una ecuación lineal en una mayor cantidad de espacios dimensionales. De manera general, esta función devuelve el producto interno entre dos puntos en una dimensión de características. Los *kernel* más utilizados son el lineal:

$$K(x_i, x_j) = x_i \times x_j$$

Y el *kernel Radial basis function* (RBF), también conocido como *kernel* Gaussiano:

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

donde $\sigma > 0$ es el parámetro que controla el ancho del *kernel*. Esta expresión se puede simplificar de la siguiente forma:

$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right)$$

donde $\gamma = \frac{1}{2\sigma^2}$. Este valor se puede entender como el corte, y es definido por el usuario; en él, se aumenta la influencia o alcance de las muestras de entrenamiento, las cuales llevan a un límite de decisión suave o estrecho (Raschka et al., 2022).

En la biblioteca de programas *scikit-learn*, dentro del módulo *svm*, se incluyen los métodos para clasificación. En el presente trabajo, se utilizó la función *support vector classification (SVC)*, la cual toma como principales hiperparámetros la penalización o el valor *C* (C), el valor *gamma* (γ) y el tamaño de la máscara *kernel* (K). Mediante prueba y error se definieron los siguientes intervalos y valores: C = 0.01, 0.1, 1, 10 y 100; γ = 0.001, 0.1, 1, 10 y 100; kernels = '*rbf* ' y '*linear*' (Raschka et al., 2022).

Perceptrón multicapa (MLP)

En MLP, al elemento básico se le conoce como neurona artificial. Esta neurona artificial, de tipo hacia adelante (*feed forward*), consta de elementos de entrada y salida

where *p(z)* is the probability that an input *z* belongs to the i-th class (Chollet, 2018).

For MLP training, the adam (Adaptive moment estimation) optimizer was used, which is a variant of the gradient descent method. This method uses the momentum and variance of the loss function's gradient to update the weights, which allows smoothing the learning curve and improving the learning of the classifier (Géron, 2022).

The scikit-learn neural network module contains the MLPClassifier function, and takes the following hyperparameters as arguments: hidden layer size (HL), number of iterations (It), activation function (AF), optimizer (Op), learning range (LR) and batch size of samples entering the model at each iteration step (BS) (Raschka et al., 2022). The intervals and search values of the hyperparameters were defined by trial and error, and the following were considered: HL = 50, 100 and 500; It = 50, 100 and 500; AF = 'softmax' and 'ReLU'; Op = Broyden-Fletcher-Goldfarb-Shanno (lbfgs) and adam algorithm; LR = 'constant' and 'adaptative'; BS = 8, 16 and 32.

Performance metrics

The metrics for determining overall performance and each target class of the classifiers are obtained from the confusion matrix, based on the test data. In the case of a binary classification, this matrix has four possible outcomes: true positive (TP), true negative (TN), false positive (FP) and false negative (FN) (Kulkarni, Chong, & Batarseh, 2020). Based on these values, the following metrics are defined:

Precision (*P*), which measures how accurate the model is in predicting positive values.

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

Recall or sensitivity (R), which measures the strength to predict positive outcomes.

$$R = \frac{TP}{TP + FN}$$

The *F1* score, which is the harmonic mean between *P* and *R*.

$$F1 = \frac{2 \times P \times R}{P + R}$$

These three metrics are used to evaluate the performance of the classifier to predict each class. The evaluation of the overall performance of the classifiers was made based on the overall classification accuracy (*ACC*)

que se procesan en la unidad central. Las capas de entrada dependen de los datos utilizados para el entrenamiento, mientras que las capas ocultas y las capas de salida pertenecen al número de clases de interés.

La arquitectura MLP se compone de una capa de entrada, una o más capas ocultas y una capa de salida. La capa de entrada depende del número de datos de entrada, las capas ocultas representan el nivel de complejidad que existe entre la capa de entrada y de salida, y la capa de salida representa el número de clases objetivo y da la clase predicha (Edmond & Girsang, 2020).

La función que transforma los datos de entrada se conoce como función de activación. La más común es la función ReLU (*Rectified Linear Unit*), la cual se expresa como:

$$f(z) = \max(0, z)$$

donde la respuesta es *z* si la entrada es positiva y *0* si es negativa. ReLU se utiliza para filtrar los datos en las capas intermedias. En la capa de salida se utiliza

$$p(z) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}}$$

la función de activación *softmax*, que se expresa como:

donde *p(z)* es la probabilidad de pertenencia de una entrada *z* a la clase i-ésima (Chollet, 2018).

Para el entrenamiento de MLP, se utilizó el optimizador adam (*Adaptative moment estimation*), que es una variante del método gradiente descendente. Este método utiliza el *momentum* y la varianza del gradiente de la función de pérdida para actualizar los pesos, lo cual permite suavizar la curva de aprendizaje y mejorar el aprendizaje del clasificador (Géron, 2022).

El modulo *neural_network* de *scikit-learn* contiene la función *MLPClassifier*, y toma como argumentos los siguientes hiperparámetros: tamaño de la capa oculta (CO), número de iteraciones (It), función de activación (FA), optimizador (Op), rango de aprendizaje (RA) y tamaño del lote de muestras que entran al modelo en cada paso de iteración (TL) (Raschka et al., 2022). Los intervalos y valores de búsqueda de los hiperparámetros se definieron por prueba y error, y se consideraron los siguientes: CO = 50, 100 y 500; It = 50, 100 y 500; FA = 'softmax' y 'ReLU'; Op = aproximación del algoritmo Broyden–Fletcher–Goldfarb–Shanno (lbfgs) y adam; RA = 'constant' y 'adaptative'; TL = 8, 16 y 32.

Métricas de desempeño

Las métricas para determinar el desempeño global y para cada clase objetivo de los clasificadores se obtienen a partir de la matriz de confusión, con base

and the area under the curve (AUC) for the ROC (receiver operating characteristic) curve.

ACC is the ratio of correct classifications to the total number of samples, and is expressed as:

$$ACC = \frac{TP + TN}{TP + FP + TN + FN}$$

The ROC is a graph that is constructed with values for different probability thresholds of the true positive rate (*TPR*) versus the false positive rate (*FPR*) (Jiang, Li, & Safara, 2021). AUC varies between 0 and 1, and measures the performance of the model for each target class. *TPR* is obtained from the following equation:

$$TPR = \frac{TP}{TP + FN}$$

and *FPR* is calculated as:

$$FPR = \frac{FP}{FP + TN}$$

Machine learning model selection

In the literature, it is reported that the RF, SVM and MLP classifiers, in general, achieve good overall classification accuracy (Yuvali, Yaman, & Tosun, 2022). In preliminary tests, these classifiers performed well in terms of classifying avocado fruit images.

Training and testing of classifiers

The training and testing of the classifiers consisted of two stages. In the first, the RF classifier was trained based on the BD1 and BD2 datasets to compare the chromatic descriptor extraction and image subsampling techniques, and to select the one that generates the best RF performance in terms of . In the second stage, the three classifiers (RF, SVM and MLP) were trained with the most appropriate dataset from stage 1. The optimal hyperparameters for each classifier were obtained by grid search and cross-validation (Raschka et al., 2022).

Selection of optimal hyperparameters

To optimize each classifier, we first performed a stratified random partitioning by each target class of the dataset at an 80:20 ratio (80 % for training and 20 % for the prediction test). This ratio establishes a balance between the training and test data to measure the performance of the classifiers. The training data were standardized to homogenize the input descriptors using the following expression:

$$x_{std} = \frac{x - \mu_x}{\sigma_x}$$

en los datos de prueba. En el caso de una clasificación binaria, esta matriz tiene cuatro posibles resultados: verdadero positivo (VP), verdadero negativo (VN), falso positivo (FP) y falso negativo (FN) (Kulkarni, Chong, & Batarseh, 2020). Con base en estos valores se definen las siguientes métricas:

La precisión (*P*), la cual mide que tan acertado es el modelo para predecir los valores positivos.

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

La exhaustividad o sensibilidad (*E*), que mide la fuerza para predecir muestras positivas.

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

El puntaje *F1*, que es la media armónica entre *P* y *E*.

$$F1 = \frac{2 \times P \times E}{P + E}$$

Estas tres métricas se utilizan para evaluar el desempeño que tiene el clasificador al predecir cada clase. La evaluación del desempeño global de los clasificadores se realizó con base en la precisión global de clasificación correcta (*ACC*) y el área bajo la curva (AUC) ROC (curva característica operativa del receptor).

La *ACC* es la proporción de clasificaciones correctas con respecto al total de muestras, y se expresa como:

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

La ROC es una gráfica que se construye con valores para diferentes umbrales de probabilidad de la tasa de verdaderos positivos (*TVP*) *versus* la tasa de falsos positivos (*TFP*) (Jiang, Li, & Safara, 2021). AUC varía entre 0 y 1, y mide el desempeño del modelo para cada clase objetivo. *TVP* se obtiene a partir de la siguiente ecuación:

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

y *TFP* se calcula como:

$$TFP = \frac{FP}{FP + VN}$$

Selección de modelos de aprendizaje automático

En la literatura se reporta que los clasificadores RF, SVM y MLP, en general, alcanzan buena precisión global de clasificación (Yuvali, Yaman, & Tosun, 2022). En pruebas preliminares, estos clasificadores obtuvie-

where *x* corresponds to each descriptor of the input set, μ_x is the mean of the set of *x* values and σ_x is the sampling variance of the set of *x* values.

Optimal hyperparameter selection was performed using a grid search and cross-validation with k = 10 disjoint groups. For each classifier, value intervals of each hyperparameter were defined to select the combination of values that maximize the average *ACC* of the classifier (Liashchynskyi & Liashchynskyi, 2019).

Cross-validation consists of dividing the training set randomly into k disjoint groups, where *k-1* groups are used as training sets, and the remaining group as a validation set. This process is repeated k times for each combination of hyperparameter values, and an average *ACC* of k model runs is obtained (Raschka et al., 2022). The optimal combination of hyperparameter values is the one that generates the maximum average *ACC* value.

Classifier prediction test

With the set of optimal hyperparameters obtained in the training stage, a cross-validation procedure was performed with k = 10 groups with the total input dataset (100 %) to recalculate the weights or parameters of each classifier. In each k run, the *ACC* prediction performance was determined for each k-th test set. After k runs, the average *ACC* performance of each classifier was obtained.

The codes implemented for the present work can be found at the following link: https://github.com/Camposfe1/Avocado-disease-classification.git.

Results and discussion

Selection of optimal hyperparameters

The optimal hyperparameter values for each classifier were NE = 10, MF = 'auto', MD = 8 and Cr = 'entropy' for RF, C = 10, γ = 10 and K = 'rbf' for SVM, and BS = 8, HL = 150, AF = 'ReLU', It = 100 and Op = 'adam' for MLP.

Comparison of descriptor extraction techniques

Comparison of color feature or descriptor extraction techniques, by region and image subsampling, as well as their effect on RF classifier performance, showed that selection by region (BD1) allows generating color pixel sets with more information to differentiate the three target classes (H, S and A). RF obtained greater accuracy to classify the three target classes. The values of FP and FN, for each class, were lower than those corresponding to BD2. Likewise, extraction by region generated better balanced class sizes than by subsampling (Figure 2).

ron buen desempeño para clasificar imágenes de frutos de aguacate.

Entrenamiento y prueba de los clasificadores

El entrenamiento y prueba de los clasificadores constó de dos etapas. En la primera se entrenó el clasificador RF con base en los conjuntos de datos BD1 y BD2 para comparar las técnicas de extracción de descriptores cromáticos y por submuestreo de imágenes, y para seleccionar la que genere el mejor desempeño de RF en términos de ACC. En la segunda etapa, los tres clasificadores (RF, SVM y MLP) se entrenaron con el conjunto de datos más apropiado de la etapa 1. Los hiperparámetros óptimos de cada clasificador se obtuvieron por medio de una búsqueda por retícula y validación cruzada (Raschka et al., 2022).

Selección de hiperparámetros óptimos

Para optimizar cada clasificador, primero se realizó una partición aleatoria estratificada por cada clase objetivo del conjunto de datos en proporción 80:20 (80 % para entrenamiento y 20 % para la prueba en predicción). Esta proporción establece un balance entre los datos de entrenamiento y de prueba para medir el desempeño de los clasificadores. Los datos de entrenamiento se estandarizaron para homogeneizar los descriptores de entrada mediante la siguiente expresión:

$$x_{std} = \frac{x - \mu_x}{\sigma_x}$$

donde *x* corresponde a cada descriptor del conjunto de entrada, μ_x es la media del conjunto de valores de *x* y σ_x es la varianza muestral del conjunto de valores de *x*.

La selección óptima de hiperparámetros se realizó por medio de una búsqueda por retícula y validación cruzada con k = 10 grupos disjuntos. Para cada clasificador, se definieron intervalos de valores de cada hiperparámetro para seleccionar la combinación de valores que maximicen la *ACC* promedio del clasificador (Liashchynskyi & Liashchynskyi, 2019).

La validación cruzada consiste en dividir el conjunto de entrenamiento de forma aleatoria en k grupos disjuntos, donde *k-1* grupos se utilizan como conjuntos de entrenamiento, y el grupo restante como conjunto de validación. Este proceso se repite k veces para cada combinación de valores de los hiperparámetros, y se obtiene una *ACC* promedio de k corridas del modelo (Raschka et al., 2022). La combinación de valores óptimos de los hiperparámetros es la que genera el máximo valor de *ACC* promedio.

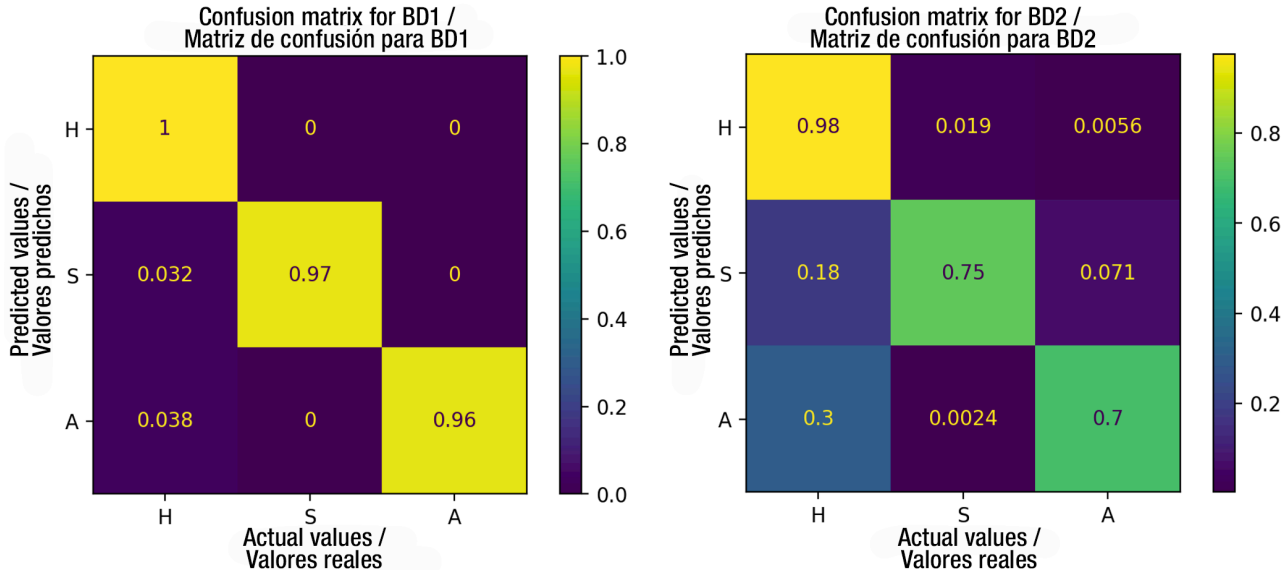

Figure 2. Confusion matrices of the random forest (RF) classifier based on the datasets generated with region extraction (BD1) and image subsampling (BD2). Target classes: H = healthy fruit; S = fruit with scab; A = fruit with anthracnose. Predicted versus actual color pixels by target class.

Figura 2. Matrices de confusión del clasificador bosque aleatorio (RF) con base en los conjuntos de datos generados con extracción por región (BD1) y submuestreo de imágenes (BD2). Clases objetivo: S = frutos sanos; R = frutos con roña; A = frutos con antracnosis. Píxeles de color predichos *versus* reales por clase objetivo.

Similarly, RF performance, overall and at the class level, was superior with BD1 than with BD2. With BD1, RF achieved an average *ACC* of 98 %, while with BD2 it was 84 %. At the class level, F1 scores were higher than 97 % with BD1 and higher than 76 % with BD2 (Table 2).

Evaluation of the prediction of RF, SVM and MLP classifiers

The prediction performance of the three classifiers was high. The confusion matrices for each classifier show that RF and MLP achieved the highest *ACC* for all three classes. All three classifiers have the highest FP value for predicting class H; that is, the classifiers predict samples from classes S or A as H. For this analysis, there are pixel samples with scab that are predicted to be healthy, and healthy samples that are predicted to have anthracnose (Figure 3).

In terms of overall classifier performance, RF and MLP were superior to SVM with an *ACC* of 98 %. Likewise, both classifiers obtained an F1 score above 97 % for each target class (Table 3).

The BD1 method of extracting features or color descriptors had a significant effect on classifier performance. This method allowed obtaining pixels or color samples representative of each target class and balanced class sizes, which led to better classifier performance. The BD2 feature extraction method

Prueba en predicción de los clasificadores

Con el conjunto de hiperparámetros óptimos obtenidos en la etapa de entrenamiento, se realizó un procedimiento de validación cruzada con $k = 10$ grupos con el conjunto total de datos de entrada (100 %) para recalcular los pesos o parámetros de cada clasificador. En cada corrida k, se determinó el desempeño de predicción *ACC* para cada conjunto de prueba k-ésimo. Después de k corridas, se obtuvo el desempeño *ACC* promedio de cada clasificador.

Los códigos implementados para el presente trabajo se encuentran en el siguiente enlace: https://github.com/Camposfe1/Avocado-disease-classification.git.

Resultados y discusión

Selección de hiperparámetros óptimos

Los valores óptimos de los hiperparámetros de cada clasificador fueron NE = 10, MF = 'auto', MD = 8 y Cr = 'entropy' para RF, C = 10, γ = 10 y K = 'rbf' para SVM, y TL = 8, CO = 150, FA = 'ReLU', It = 100 y Op = 'adam' para MLP.

Comparación de técnicas de extracción de descriptores

La comparación de las técnicas de extracción de características o descriptores de color, por región y submuestreo de imágenes, así como su efecto en el

Table 2. Prediction performance metrics of the random forest (RF) classifier based on region-based color descriptor extraction (BD1) and image subsampling (BD2) methods.

Cuadro 2. Métricas de desempeño en predicción del clasificador bosque aleatorio (RF) con base en los métodos de extracción de descriptores de color por región (BD1) y submuestreo de imágenes (BD2).

Metric / Métrica	BD1			BD2		
	H/S	S/R	A	H/S	S/R	A
P	0.94	1.00	1.00	0.77	0.97	0.76
R/E	1.00	0.97	0.96	0.98	0.75	0.84
F1	0.97	0.98	0.98	0.86	0.84	0.76
AUC	1.00	1.00	1.00	0.97	0.94	0.95
ACC	0.98 ± 0.03			0.84 ± 0.08		

H = healthy fruits; S = fruits with scab; A = fruits with anthracnose; P = precision; R = recall; F1 = F1 score; *ACC* = overall classification accuracy; AUC = area under the ROC curve.

S = frutos sanos; R = frutos con roña; A = frutos con antracnosis; P = precisión; E = exhaustividad; F1 = puntaje F1; ACC = precisión global de clasificación; AUC = área bajo la curva ROC.

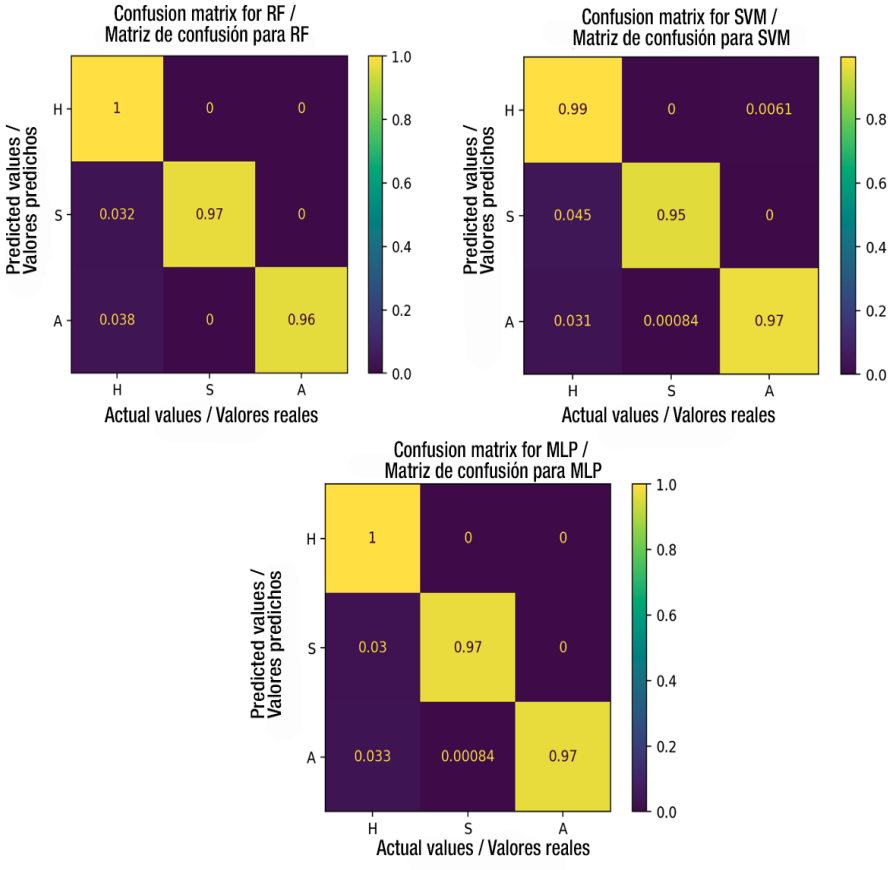

Figura 3. Matrices de confusión de los clasificadores bosque aleatorio (RF), máquina de soporte vectorial (SVM y perceptrón multicapa (MLP) con base en la extracción de descriptores por región (BD1). Clases objetivo: S = frutos sanos; R = frutos con roña; A = frutos con antracnosis.

Figure 3. Confusion matrices of the random forest (RF), support vector machine (SVM) and multilayer perceptron (MLP classifiers based on descriptor extraction by region (BD1). Target classes: H = healthy fruits; S = fruits with scab; A = fruits with anthracnose.

Table 3. Prediction performance metrics of random forest (RF), support vector machine (SVM) and multilayer perceptron (MLP) machine learning classifiers based on descriptor extraction by region (BD1).

Cuadro 3. Métricas de desempeño en la predicción de los clasificadores de aprendizaje automático bosque aleatorio (RF), máquina de soporte vectorial (SVM) y perceptrón multicapa (MLP) con base en la extracción de descriptores por región (BD1).

Metric/ Métrica	RF			SVM			MLP		
	H/S	S/R	A	H/S	S/R	A	H/S	S/R	A
P	0.94	1.00	1.00	0.94	1.00	0.99	0.95	1.00	1.00
R/E	1.00	0.97	0.96	0.99	0.95	0.97	1.00	0.97	0.97
F1	0.97	0.98	0.98	0.96	0.98	0.98	1.00	0.98	0.98
AUC	1.00	1.00	1.00	0.99	0.99	0.99	0.99	0.99	1.00
ACC	0.98 ± 0.03			0.97 ± 0.06			0.98 ± 0.03		

H = healthy fruit; S = fruit with scab; A = fruit with anthracnose; P = precision; R = recall; F1 = F1 value; AUC = area under the ROC curve; ACC = overall classification accuracy.

S = frutos sanos; R = frutos con roña; A = frutos con antracnosis; P = precisión; E = exhaustividad; F1 = valor F1; AUC = área bajo la curva ROC; ACC = precisión global de clasificación.

generated a dataset with unbalanced classes and the H class of healthy fruits was favored. The imbalance between class sizes in BD2 had a negative effect on the performance of the RF classifier, which reached an ACC of 84 %. Some publications state that there is no difference between feature extraction methods (Jones, Faiz, Qiu, & Zheng, 2022; Suresh & Mohan, 2020); however, in this study the BD1 selection technique allowed obtaining a better ACC of the RF model than BD2, which depends on the ability to select the areas of interest (Chen et al., 2020). In the case of unbalanced classes, the F1 metric (harmonic mean of P and R) is a more suitable measure to evaluate classifier performance (Fourure, Javaid, Posocco, & Tihon, 2021).

When there are unbalanced classes, the classifiers bias the prediction to the majority class; hence, there is a misclassification for the minority class (Kaur, Singh, & Kaur, 2019). This is because it is easier to get healthy fruits than ones with disease symptoms, since certain conditions are needed for symptoms to be expressed in the fruit (Wardhani, Rochayani, Iriany, Sulistyono, & Lestantyo, 2019).

desempeño del clasificador RF, mostró que la selección por región (BD1) permite generar conjuntos de píxeles de color con más información para diferenciar las tres clases objetivo (S, R y A). RF obtuvo mayor precisión para clasificar las tres clases objetivo. Los valores de FP y FN, para cada clase, fueron menores que los correspondientes a BD2. Asimismo, la extracción por región generó tamaños de clase mejor balanceados que por submuestreo (Figura 2).

De manera similar, el desempeño de RF, a nivel global y a nivel de clase, fue superior con BD1 que con BD2. Con BD1, RF alcanzó una *ACC* promedio de 98 %, mientras que con BD2 ésta fue de 84 %. A nivel de clase, los puntajes F1 fueron superiores a 97 % con BD1 y superiores a 76 % con BD2 (Cuadro 2).

Evaluación de la predicción de los clasificadores RF, SVM y MLP

El desempeño, en cuanto a la predicción de los tres clasificadores, fue alto. En las matrices de confusión de cada clasificador se observa que RF y MLP alcanzaron

The three classifiers, RF, SVM, and MLP, categorized scab and anthracnose fruit with an F1 score of 98 %, and were the best-classified classes. Fruits with scab are visually differentiated from healthy or anthracnose fruits; therefore, pixel values between classes are more contrasting.

The machine learning classifiers used in this study show that high performance levels can be achieved with small datasets and shorter computational times compared to deep learning classifiers, particularly convolutional neural networks, which require longer computational times. However, in more complex identification problems, with a large number of classes (greater than 10) and large datasets (thousands of images), deep learning algorithms are more appropriate (Alzubaidi et al., 2021).

Conclusions

The extraction of color descriptors with the region selection method induced a better overall classification accuracy (ACC = 98 %) of the random forest classifier, in contrast to the image subsampling extraction method (ACC = 84 %). All three classifiers (random forest, vector support machine and multilayer perceptron) obtained a ACC greater than 97 % for classifying the surface of avocado fruits as being healthy, with scab or with anthracnose. Likewise, the fruit surface classes with scab and with anthracnose obtained an F1 score of 98 % with all three classifiers.

Acknowledgments

We would like to thank the National Science and Technology Council (CONACyT) for awarding a scholarship to the first author for graduate studies.

End of English version

References

Abdel-Hamid, L. (2019). Glaucoma detection using statistical features: comparative study in RGB, HSV and CIEL*a*b* color models. *Tenth International Conference on Graphics and Image Processing, 11069*. doi: 10.1117/12.2524215

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamarìa, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data, 8*(53), 1-74. doi: 10-1186/s40537-021-00444-8

Ambrosio-Ambrosio, J. P., González-Camacho, J. M., Rojano-Aguilar, A., & del Valle-Paniagua, D. (2023). Identification of disease in tomato leaves using machine learning classifiers and digital images.

la mayor *ACC* para las tres clases. Los tres clasificadores tienen el mayor valor de FP para predecir la clase S; es decir, los clasificadores predicen muestras de las clases R o A como S. Para este análisis, hay muestras de píxeles con roña que se predicen como sanas, y muestras sanas que se predicen con antracnosis (Figura 3).

En términos del desempeño global de los clasificadores, RF y MLP fueron superiores a SVM con una *ACC* de 98 %. Asimismo, ambos clasificadores obtuvieron un puntaje F1 superior a 97 % para cada clase objetivo (Cuadro 3).

El método de extracción de características o descriptores de color BD1 tuvo un efecto importante en el desempeño de los clasificadores. Este método permitió obtener píxeles o muestras de color representativas de cada clase objetivo y tamaños de clase balanceados, lo cual condujo a un mejor desempeño de los clasificadores. El método de extracción de características BD2 generó un conjunto de datos con clases desbalanceadas y se privilegió a la clase S de frutos sanos. El desbalance entre tamaños de clase en BD2 tuvo un efecto negativo en el desempeño del clasificador RF, el cual alcanzó una *ACC* de 84 %. En algunas publicaciones se menciona que no hay diferencias entre los métodos de extracción de características (Jones, Faiz, Qiu, & Zheng, 2022; Suresh & Mohan, 2020); sin embargo, en este estudio la técnica de selección BD1 permitió obtener una mejor *ACC* del modelo RF que BD2, el cual depende de la habilidad para seleccionar las áreas de interés (Chen et al., 2020). En el caso de clases desbalanceadas, la métrica F1 (media armónica de P y E) es una medida más adecuada para evaluar el desempeño del clasificador (Fourure, Javaid, Posocco, & Tihon, 2021).

Cuando se tienen clases desbalanceadas, los clasificadores sesgan la predicción a la clase mayoritaria; por lo cual, hay una mala clasificación para la clase minoritaria (Kaur, Singh, & Kaur, 2019). Lo anterior se debe a que es más fácil conseguir frutos sanos que frutos con los síntomas de las enfermedades, ya que se necesitan ciertas condiciones para que se puedan expresar los síntomas en la fruta (Wardhani, Rochayani, Iriany, Sulistyono, & Lestantyo, 2019).

Los tres clasificadores RF, SVM y MLP catalogaron a los frutos con roña y anctracnosis con puntaje F1 de 98 %, y fueron las clases que se clasificaron mejor. Los frutos con roña se diferencian visualmente de los frutos sanos o con antracnosis; por ello, los valores de los píxeles entre clases son más contrastantes.

Los clasificadores de aprendizaje automático utilizados en este estudio muestran que es posible alcanzar altos niveles de desempeño con conjuntos de datos pequeños y tiempos de cómputo menores, en comparación con los clasificadores de aprendizaje profundo; en particular, con las redes neuronales convolucionales,

Agrociencia, 57(3), 476-507. doi: 10.47163/agrociencia.v57i3.2462

Campos-Ferreira, U. E., & González-Camacho, J. M. (2021). Clasificador de red neuronal convolucional para identificar enfermedades del fruto de aguacate (*Persea americana* Mill.) a partir de imágenes digitales. *Agrociencia*, *5*(8), 695-709. doi: 10.47163/agrociencia.v55i8.2662

Chen, R. C., Dewi, C., Huang, S. W., & Caraka, R. E. (2020). Selectig critical features for data classification based on machine learning methods. *Journal of Big Data, 7*(52), 1-26. doi: 10.1186/s40537-020-00327-4

Chollet, F. (2018). Getting started with neural networks. *In*: Chollet, F. (Ed.), *Deep Learning with Python* (pp. 56-92). New York: Manning Publications Co.

Das, D., Singh, M., Mohanty, S. S., & Chakravarty, S. (2020). Leaf disease detection using support vector machine. *International Conference on Communication and Signal Processing, 2020*. 1036-1040. doi: 10.1109/ICCSP48568.2020.9182128

Doh, B., Zhang, D., Shen, Y., Hussain, F., Doh, R. F., & Ayepah, K. (2019). Automatic citrus fruit disease detection by phenotyping using machine learning. *25th International Conference on Automatic and Computing, 2019*, 1-5. doi: 10.23919/IConAC.2019.8895102

Edmond, C., & Girsang, A. S. (2020). Classification performance for credit scoring using neural network. *International Journal of Emerging Trends in Engineering Research*, *8*(5), 1592-1599. doi: 10.30534/ijeter/2020/19852020

Fourure, D., Javaid, M. U., Posocco, N., & Tihon, S. (2021). Anomaly detection: how to artificially increase your f1-score with a biased evaluation protocol. In: Dong, Y., Kourtellis, N., Hammer, B., & Lozano, J. A. (Eds.), *Machine learning and knowledge discovery in databases. Applied data science track* (pp. 3-18). New York: Springer. doi: 10.48550/arXiv.2106.16020

Géron, A. (2022). Ensemble learning and random forests. In: Géron, A. (Ed.), *Hands-on machine learning with scikit-learn, keras & tensorflow. concepts, tools, and techniques to build intelligent systems* (pp. 337-373). Sebastopol: O'Reilly Media.

Jiang, H., Li, X., & Safara, F. (2021). Iot-based agriculture: deep learning in detecting apple fruit diseases. *Microprocessors and Microsystems, 14*, 1-23. doi: 10.1016/j.micpro.2021.104321

Jones, M. A., Faiz, R., Qiu, Y., & Zheng, B. (2022). Improving mammography lesion classification by optimal fusion of handcrafted and deep transfer learning features. *Physics in Medicine & Biology*, *67*(5). doi: 10.1088/1361-6560/ac5297

Kaur, H., Singh, P. H., & Kaur, M. A. (2019). A systematic review on imbalanced data challenges in machine learning: applications and solutions. *ACM Comput, 52*(4), 1-36. doi: 10.1145/3343440

Ketkar, N., & Moolayil, J. (2021). Feed-forward neural networks. In: Ketkar, N., & Moolayil, J. (Eds.), *Deep learning with python* (pp. 93-132). Pune: APress.

las cuales requieren mayores tiempos de cómputo. No obstante, en problemas de identificación más complejos, con un gran número de clases (mayor de 10) y grandes conjuntos de datos (miles de imágenes), los algoritmos de aprendizaje profundo resultan más apropiados (Alzubaidi et al., 2021).

Conclusiones

La extracción de descriptores de color con el método de selección por región indujo una mejor precisión global de clasificación del clasificador bosque aleatorio (*ACC* = 98 %), en contraste con el método de extracción con submuestreo de imágenes (*ACC* = 84 %). Los tres clasificadores (bosque aleatorio, máquina de soprte vectorial y perceptrón multicapa) obtuvieron *ACC* mayores a 97 % para clasificar la superficie de frutos de aguacate sano, con roya o antracnosis. Asimismo, las clases superficie del fruto con roña y con antracnosis obtuvieron un puntaje F1 de 98 % con los tres clasificadores.

Agradecimientos

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por la asignación de la beca al primer autor para los estudios de posgrado.

Fin de la versión en español

Knauer, U., Rekowski, C. S. v., Stecklina, M., Krokotsch, T., Pham, M. T., Hauffe, V., Kilias, D., Ehrhardt, I., Sagischewski, H., Chmara, S., & Seiffert, U. (2019). Tree species classification based on hybrid ensembles of a convolutional neural network (cnn) and random forest classifiers. *Remote Sensing, 11*(23), 1-15. doi: 10.3390/rs11232788

Kulkarni, A., Chong, D., & Batarseh, F. A. (2020). Foundations of data imbalance and solutions for a data democracy. *In*: Batarseh, F. A., & Yang, R. (Eds.), *Data Democracy* (pp. 83-106). Cambridge: Academic Press.

Liashchynskyi, P., & Liashchynskyi, P. (2019). Grid search, random search, genetic algorithm: A Big Comparison for NAS. *arXiv, 12*, 1-11. doi: 10.48550/arXiv.1912.06059

Parmar, A., Katariya, R., & Patel, V. (2019). A review on random forest: an ensemble classifier. In: Hemanth, J., Fernando, X., Lafata, P., & Baig, Z. (Eds), *International Conference on Intelligent Data Communication Technologies and Internet of Things, 2018*, 758-763. Cham: Springer Nature Switzerland.

Pisner, D. A., & Schnyer, D. M. (2020). Support vector machine In: Mechelli, A., & Vieira, S. (Eds.), *Machine learning – methods and applications to brain disorders*. Cambridge: Academic Press.

Raschka, S., Liu, Y., & Mirjalili, V. (2022). A tour of machine learning classifiers using scikit-learn. In: Raschka, S., Liu, Y., & Mirjalili, V. (Eds.), *Machine Learning with PyTorch and Scikit-Learn*. Birmingham: Packt Publishing.

Saleem, M. H., Potgieter, J., & Arif, K. M. (2019). Plant disease detection and classification by deep learning. *Plants*, *8(11)*, 1-22. doi: 10.3390/plants8110468

Sandhya, S., Balasundaram, A., & Arunkumar, S. (2022). Deep learning and computer vision based model for detection of diseased mango leaves. *International Journal on Recent and Innovation Trends in Computing and Communication*, *10*(6), 70-79. doi: 10.17762/ijritcc.v10i6.5555

Servicio de Información Agroalimentaria y Pesquera (SIAP) (2022). *Aguacate. Panorama Agroalimentario*. México: Servicio de Información Agroalimentaria y Pesquera.

Suresh, S., & Mohan, S. (2020). ROI-based feature learning for efficient true positive prediction using convolutional neural network for lung cancer diagnosis. *Neural Computing and Applications*, *32*(20), 15989-16009. doi: 10.1007/s00521-020-04787-w

Srinivasa, G. N., Venkata, R. P., Anusha, T. M., Sai, H. V., & Bhanu, P. B. (2022). Detection of plant leaf diseases using random forest classifier. *International Journal of Innovative Research in Technology*, *9(1)*, 1300-1302. Retrieved from https://ijirt.org/master/publishedpaper/IJIRT155613_PAPER.pdf

Téliz, D., & Mora, A. (2019). Enfermedades. In: Téliz, D., & Mora, A. (Eds.), *El aguacate y su manejo integrado* (pp. 171-173). Texcoco: Biblioteca Básica de Agricultura.

Wardhani, N. W., Rochayani, M. Y, Iriany, A., Sulistyono, A. D., & Lestantyo, P. (2019). Cross-validation metrics for evaluating classification performance on imbalanced data. *International Conference on Computer, Control, Informatics and its Applications, 2019*, 14-18. doi: 10.1109/IC3INA48034.2019.8949568

Yuvali, M., Yaman, B., & Tosun, O. (2022). Classification comparison of machine learning algorithms using two independent CAD datasets. *Mathematics*, *10*(3), 1-15. doi: 10.3390/math10030311