Agrociencia

EVALUATION OF MACHINE LEARNING MODELS TO IDENTIFY PEACH VARIETIES BASED ON LEAF COLOR

Daniel **Ayala-Niño**[1], Juan Manuel **González-Camacho**[1*]

[1] Colegio de Postgraduados Campus Montecillo, Posgrado en Socieconomía, Estadística e Informática-Cómputo Aplicado, Carretera México-Texcoco km 36.5, Montecillo, Texcoco, Estado de México, México. C. P. 56230.
* Corresponding author: jmgc@colpos.mx

Citation: Ayala-Niño D, González-Camacho JM. 2022. Evaluation of machine learning models to identify peach varieties based on leaf color. **Agrociencia.** https://doi.org/ 10.47163/agrociencia.v56i4.2810

Editor in Chief:
Dr. Fernando C. Gómez Merino

Received: October 25, 2021.
Approved: March 30, 2022.
Published in Agrociencia: June 20, 2022.

ABSTRACT

Machine learning and deep learning approaches are applied in different areas of the agricultural sector, particularly in the digital image-based identification of characteristics of interest in crops. In this research, the performance of three machine learning classifiers was evaluated: support vector machine (SVM), random forest (RF), and multilayer perceptron (MLP). The aim was to identify four varieties of peach (*Prunus persica* L. Batsch) (CP-03-06, Oro Azteca, Oro San Juan, and Cardenal), based on the color of digital images of the upper and lower side of leaves, represented by two color spaces: RGB (red, green, blue) and HSV (hue, saturation, value). The classifiers were trained and evaluated based on six data input scenarios, defined by the combinations of the upper, lower, and both sides of the leaf with the RGB and HSV color spaces. The three machine learning classifiers (SVM, RF, and MLP) achieved their best prediction performance when they examined the color characteristics of the upper side of leaves transformed to the HSV color space. The SVM classifier outperformed RF and MLP. SVM achieved a global average correct classification accuracy of 84.1 %, *F*1 *macro* of 83.7 %, and area under the ROC curve (*AUC macro*) of 0.93. The Oro Azteca variety reached the highest classification rate with a *F*1 score of 87.9 % and the Oro San Juan variety obtained the lowest rate with a *F*1 of 71.3 %.

Keywords: random forest, support vector machine, supervised classification, *Prunus persica* L. Batsch, neural networks, artificial intelligence.

INTRODUCTION

In Mexico, the peach demand is important: 172 950 Mg were produced in 2020 (SIAP, 2020). In 2018, Mexico ranked seventh in world peach production (SADER, 2018). High productivity and quality peach varieties have high cold requirements; therefore, new peach varieties have been developed with greater productive potential, higher quality, and low cold requirements. A poorly selected variety may not flower (or flower early, during the frost period) causing a reduction or loss of the harvest; therefore, identifying peach varieties is of the utmost importance. The research and evaluation of new genotypes facilitate the selection of varieties with greater productive potential and fruit quality.

Leaf-based plant identification is very common because, under appropriate conditions, leaves last for months or even years (Wang *et al.*, 2016). However, leaf color is not used to identify peach varieties, because it is affected by various biotic, abiotic or phenological factors (UPOV, 2022). Leaf color is rarely used as a descriptor, as a result of the similarity between and within varieties, subject to climatic and health conditions (Wäldchen and Mäder, 2018).

In literature, different leaf-based approaches to identify plants are reported, including the extraction of the characteristics (shape, texture, color, and/or venation) to train a classifier (Cervantes *et al.*, 2018). Xiao *et al.* (2013) designed a system for the detection of pepper leaves based on templates that describe the leaf shape; these authors used active shape models (ASM) and a multilayer perceptron-ASM. The multilayer perceptron-ASM achieved an overall classification accuracy of 85.7 % (simple leaves) and 76.7 % (overlapping leaves).

Liu and Kan (2016) used leaf shape and texture to identify 220 types of leaves; they used a deep belief network (DBN), obtaining an overall classification accuracy of 93.9 %. Caglayan *et al.* (2013) used leaf shape and color to classify 32 types of plants from the Flavia data set (Wu *et al.*, 2007); their random forest classifier obtained an overall classification accuracy of 96 %. Dubey and Jalal (2012) proposed a method to classify apple fruit diseases, based on local binary patterns and a support vector machine, obtaining an overall classification accuracy of 92.6 % (HSV color space) and 88.7 % (RGB).

Leaf venation is another characteristic that has been used in digital image-based plant identification. Larese *et al.* (2014) classified different white bean, red bean, and soybean varieties with a support vector machine trained with segmented leaf venation. This model achieved an overall classification accuracy of 95.1 %. Grinblat *et al.* (2016) classified different legume varieties, based on the segmentation of different levels of leaf venation and a convolutional neural network (CNN), obtaining an overall classification accuracy of 96.9 %. Applying convolutional neural networks is relevant for plant identification. Lee *et al.* (2017) used a deconvolutional neural network (DNN) to determine the patterns that a CNN learns from leaves. They proposed a hybrid, two-input CNN model: the first input receives the leaf image, and the second, a cutting of the central part of the leaf. This approach enables an overall classification accuracy of 96.3 %. The model was trained on the Malayakew dataset, which consists of 44 plant types.

The objective of this research was to implement and evaluate the prediction performance of three machine learning classifiers: support vector machine, random forest, and multilayer perceptron. The aim was to classify four varieties of peach (*Prunus persica* L. Batsch): V1, CP-03-06; V2, Oro Azteca; V3, Oro San Juan; and V4, Cardenal, based on digital images of peach leaves (upper and lower side) transformed into two color spaces (RGB and HSV). Each classifier was trained and evaluated in prediction with six data input scenarios: upper and lower side in RGB format; upper and lower side in HSV format; upper side in RGB format; upper side in HSV format; lower side in RGB format; and lower side in HSV format.

MATERIALS AND METHODS

This section describes the steps to evaluate the prediction performance of the three machine learning classifiers: support vector machine (SVM), random forest (RF), and multilayer perceptron (MLP). The classifiers were implemented in the Python programming language (version 3.9.5). The training and evaluation of the models was carried out using the Scikit-learn program library (version 0.24.2) and they were executed in the Google Collaboratory cloud modeling platform.

Database Description

The training database was created based on 741 pairs of digital images of leaves (upper and lower side) of four peach varieties obtained from the peach dataset. The images belong to varieties V1 (CP-03-06), V2 (Oro Azteca), V3 (Oro San Juan), and V4 (Cardenal) (Figure 1). The total number of image pairs for each variety is V1 (101), V2 (207), V3 (204), and V4 (229). The database is available at: https://bit.ly/3cvA9MF. Each peach leaf obtained consists of a pair of images (upper and lower side). Mature peach leaves were collected from healthy trees in the experimental field of the Colegio de Postgraduados, Campus Montecillo. All images have a 72 ppi resolution of, with 720 × 480 pixels. They were saved in the JPG format. They were captured with a Canon EOS REBEL T5i camera, in a controlled environment, with artificial light and white background (Ayala *et al.*, 2020).

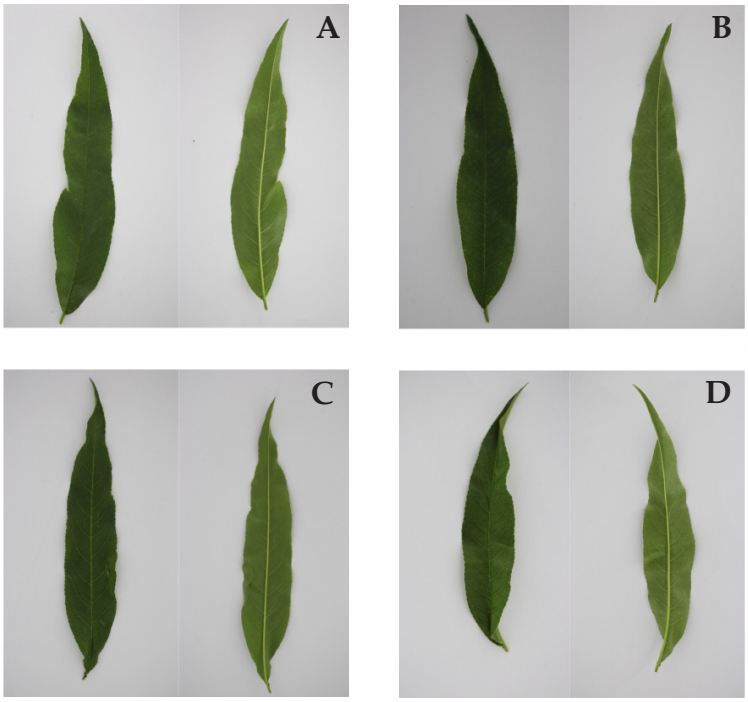

Figure 1. Upper and lower sides of the four peach varieties: A, CP-03-06 variety (V1); B, Oro Azteca (V2); C, Oro San Juan (V3); and D, Cardenal (V4).

The implementation of machine learning models was based on the following stages: dataset obtaining, image preprocessing, characteristic extraction, hyperparameter selection, and classifier prediction testing (Figure 2).

Image Preprocessing

The images were segmented with the Otsu algorithm, in order to select the object of interest (leaf) from the background of the image. This algorithm establishes an optimal threshold to separate the image from the background and delivers a binary image, with a numerical value of zero (background) and one (leaf area). Subsequently, a 5 × 5 Gaussian filter was applied to remove imperfections from the image of the segmented leaf. Based on the importance of leaf venation as the main identifier of plants, three samples (30 × 30 pixels) were extracted from each preprocessed image, in two specific leaf areas: the central nerve and the tertiary nerve (Grinblat *et al.*, 2016; Lee *et al.*, 2017). Image samples were transformed to RGB (red, green, blue) and HSV (hue, saturation, value) color spaces. Image representation in the HSV color space improves the performance of learning models (Łuszczkiewicz-Piątek, 2014; García-Mateos *et al.*, 2015). RGB is an additive color space represented by three color channels (red, green, and blue); each channel has a 0-255 variation in the discrete range. HSV is a nonlinear transformation of the RGB color space, defined by three characteristics: hue, with 0-360 values; saturation, with 0-1 values; and brightness, with 0-1 values.

Creating the Training Datasets

Six data input scenarios were developed with the triplets obtained from the two RGB and HSV color spaces and the three leaf positions (upper, lower, and both sides): E1, upper and lower side in RGB format; E2, upper and lower side in HSV format; E3,

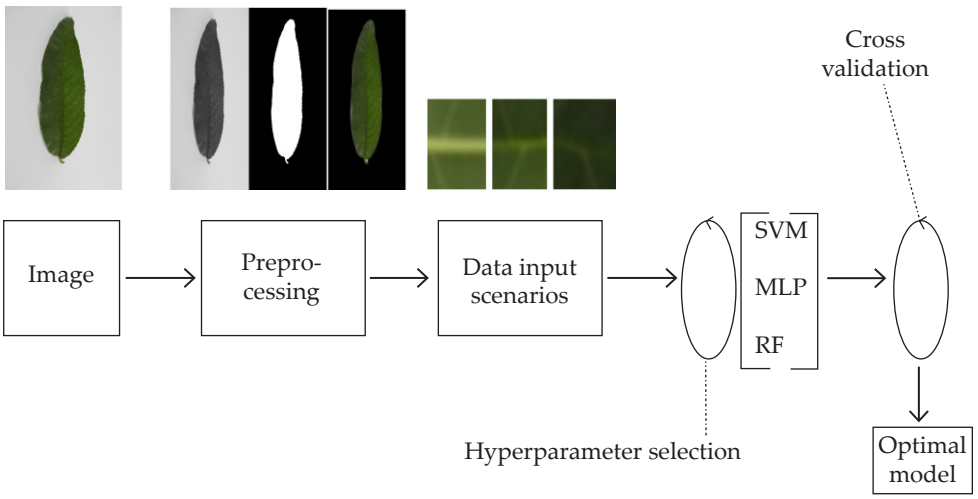

Figure 2. Stages for the implementation of machine learning support vector machine (SVM), random forest (RF), and multilayer perceptron (MLP) models.

upper side in RGB format; E4, upper side in HSV format; E5, lower side in RGB format; and E6, lower side in HSV format. The three scenarios with the RGB color space are:

E1 = [upper or lower side, R, G, B]

E3 = [upper side, R, G, B]

E5 = [lower side, R, G, B]

where E1 has four characteristics: leaf position (upper or lower side), RGB triplet (*R*, *G*, *B*), E3 RGB triplet (upper side), and E5 RGB triplet (lower). For the HSV space, scenarios E2, E4, and E6 are defined with the HSV triplet (H, S, V), in a similar way to E1, E3, and E5.

The datasets for each scenario were refined through the elimination of repeated records within each class (based on the leaf side). Subsequently, the same procedure was applied to eliminate repeated records between classes, in order to obtain unique values for each class.

Machine Learning Models

Multiclass Support Vector Machine

The multiclass support vector machine (SVM) classifier is a machine learning algorithm proposed by Boser *et al.* (1992). The SVM optimization finds an optimal hyperplane as a solution to the learning problem. This hyperplane defines the decision limits of the input characteristic vectors. A new sample is classified according to its distance from the hyperplane. An optimal hyperplane has the maximum distance to the data point closest to each target class; this maximum distance is known as the margin (Evgeniou and Pontil, 2001).

An optimization problem with restrictions has a dual form. That is, the Karush-Kuhn-Tucker (KKT) conditions solve two problems: the primal problem and the dual problem.

The primal problem is defined as:

$$J(w, b, \xi) = \min_{w,b,\xi} \frac{1}{2} w^T w + c \sum_{i=1}^{n} \xi_i$$

where the target function of SVM is to maximize the margin, which is equivalent to minimizing $\|w\|^2 = w^T w$, subject to the following restriction:

$$y_i (w_0 \phi (x_i) + b) \geq 1 - \xi_i,$$

where w is the weight vector of the hyperplane; b is the bias; $\xi_i \geq 0$, $i = 1, \ldots, n$ is a slack variable aimed at avoiding convergence problems in non-linearity cases; $x_i \in \mathbb{R}^p$ is the vector of p input characteristics; y_i is the target class of the i-th sample; and n is the number of samples.

The C hyperparameter controls the penalty of an erroneous classification. This hyperparameter is used to control the width of the margin.
The dual problem is defined as:

$$\min_{\alpha} \frac{1}{2}\alpha^T Q\alpha - e^T\alpha$$

Subject to: $y^T\alpha = 0$ and $0 \leq \alpha_i \leq C$, $i = 1, ..., n$.

where e is a vector of ones; and Q is a positive semidefinite $n \times n$ matrix, set forth as follows:

$$Q_{ij} = y_i y_j K(x_i, x_j)$$

where $K(x_i, x_j) = \phi(x_i)^T \phi(x_i)$ is a kernel function. The response of the model optimized for a x_i sample, is defined by:

$$\hat{Y} = \sum_{i \in SV} y_i \alpha_i K(x_i, x_j) + b$$

where $K(x_i, x_j)$ is used to solve nonlinear problems. The training data are transformed to another dimension by means of a $\phi()$ function; then, SVM is trained with the new characteristics that map the new space. The Gaussian kernel (*rbf*, radial basis function) is the most common and is expressed as:

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

where $\gamma = \frac{1}{2\sigma^2}$ is another hyperparameter to be optimized, σ is the width of the Gaussian kernel (Savas and Dovis, 2019). Linear and sigmoid kernels are also used.

Random Forest

RF is a learning algorithm based on a technique known as ensemble learning and consists of a set of decision trees (Breiman, 2001). This method is known as a hard voting classifier. This technique uses a set of classifiers to predict, based on a x_i sample; the predicted class gets most of the votes.
RF training uses the bagging resampling method, which consists of a new sampling in which the training set for each algorithm in the set is replaced (Breiman, 1996). A pasting resampling method without replacement can also be used (Breiman, 1999).
The target function of a decision tree seeks to increase the information gain (IG) at each node. Starting at the root of the tree, data are divided based on IG; that is to say, the degree of nodes with samples that belong to a single class depends on the impurity. The IG function is set forth as:

$$IG(D_P, f) = I(D_p) - \sum_{j=1}^{n} \frac{N_j}{N_p} I(D_j)$$

where f is the function that divides the data; D_p and D_j are the data sets of the parent node and the j-th son; I is the impurity metric; N_p is the number of samples in the parent node; N_j is the number of samples in the j-th son.

IG is higher when the impurity of the son is lower. Impurity is a criterion for the division of the data along the tree. The Gini and cross-entropy criteria are the cost or loss functions used to optimize RF. The cross-entropy criterion is defined by:

$$I_G(t) = -\sum_{i=1}^{c} p(i \mid t) \log_2 p(i \mid t)$$

where $p(i \mid t)$ is the rate of samples belonging to the i target class in the t node; and c is the total number of target classes. The entropy is zero if all the samples of the node belong to the same class and it is greater if the classes are uniformly distributed. The entropy criterion seeks to maximize the mutual information in the tree, while the Gini criterion seeks to minimize the probability of an erroneous classification, just like the entropy criterion, Gini is higher if the classes are uniformly distributed. The Gini index is defined by:

$$I_G(t) = 1 - \sum_{i=1}^{c} p(i \mid t)^2$$

RF is an ensemble method for random decision trees, which has two basic hyperparameters: the number of random trees (*nar*) for the algorithm and their maximum depth (*pmax*); each random tree is assigned a different sample of the data (bagging).

Multilayer Perceptron

The multilayer perceptron (MLP) classifier is a machine learning algorithm that learns from a training dataset and a $f(\cdot)$: $\mathbb{R}^p \rightarrow \mathbb{R}^c$ function where p is the number of input characteristics and e is the total number of target classes. Given a $x \in \mathbb{R}^p$ data set and $y \in \mathbb{R}^c$ target classes, the algorithm approximates a nonlinear function in order to classify an input sample with its target classes. A standard MLP model has an input layer that represents the input variables, a hidden layer, and an output layer. The hidden layer has *nn* neurons (hyperparameter) and each neuron transforms the input data and weights (*i.e.*, the $w_1 x_1 + w_2 x_2 + ... + w_m x_m$ linear combination), by means of an activation function. The output layer receives the values from the hidden layer and transforms them into output target classes, via another activation function. The process that takes

data from the input layer to the last output layer is known as forward propagation. The training of the MLP classifier uses the backpropagation error algorithm (Rumelhart *et al.*, 1986).

Backpropagation error consists of using the responses obtained with forward propagation in each cycle to calculate the gradients of the cost function with regard to the parameters of the last layer. This procedure is applied in the hidden layer(s). The weights are updated in each layer until the training epochs are finished (1000 in this research) or a stop condition is defined (tolerance error). The function to be optimized by means of the descending gradient is the multiclass cross-entropy cost (J) function (Ho and Wookey, 2020) which is defined as follows:

$$J = -\frac{1}{n}\sum_{k}^{c}\sum_{j}^{n} y_i^k \log\left(f_w(x_j)\right)$$

where: n is the number of samples used in training, c is the number of classes, y_i^k is the class to which the i sample of the k class belongs, and f_n is the output of the MLP function that receives as input the vector of the x_j characteristics.

The following optimization procedures were used in this research: stochastic gradient descent (SGD), adam (Kingma and Ba, 2014), and L-BFGS (Limited-memory Broyden-Fletcher-Goldfarb-Shanno).

The activation functions that were used to optimize the MLP were linear rectifier —defined as $ReLU = max\ (0,\ x)$— and hyperbolic tangent —defined as $tanh\ (x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ —, where x is an input data.

Performance Metrics

The metrics used to evaluate the performance of the classifiers were: overall classification accuracy (*Acc*), *F*1 *macro*, *AUC macro*, accuracy (*P*), sensitivity (*S*), *F*1, and area under the ROC curve (*AUC*). *Acc* is the total rate of correctly predicted samples, which in case of binary classification is obtained as follows:

$$Acc = (TP + TN)\ N^{-1}$$

where *TP* are the true positives (number of samples correctly predicted as positive); *TN* are the true negatives (number of samples correctly predicted as negative); and *N* is the total number of predicted samples.

P is the rate of true positives with respect to the total of predicted positives; it is obtained as follows:

$$P = TP\ (TP + FP)^{-1}$$

where *FP* are the false positives (number of negative samples predicted as positive). *S* is the rate of true positives with respect to the total number of positives observed, as defined by:

$$S = TP\ (TP + FN)^{-1}$$

where *FN* are the false negatives, number of positive samples predicted as negative. Finally, *F*1 it is the harmonic mean of *P* and *S*, which is obtained by:

$$F1 = 2(P \cdot S)\ (P + S)^{-1}$$

Where *F*1 *macro* is the average of the *F*1 values obtained by target class; it is obtained as follows:

$$F1\ macro = \frac{1}{c}\sum_{i=1}^{c} F1_i$$

where *c* is the total number of target classes; and $F1_i$ is the value of the *F*1 metric of the *i* class.

Values close to one of the described metrics indicate good model performance. *AUC* is the area under the ROC curve, which is the graph of the *IP* rate (*IVP*) *versus* the *FP* rate (*TFP*) for different decision thresholds; a value close to one represents an optimal model. This metric provides information on the ability of the model to discriminate between classes; the closer the area is to a given model, the model's ability to discriminate between classes improves (Fawcett, 2006). *AUC macro* is the mean of the area under the ROC curve evaluated for each target class.

Classifier Training

The training of the SVM, RF, and MLP classifiers consists of two stages: the selection of optimal hyperparameters and the evaluation of the prediction performance ability to obtain an optimal model.

Data Standardization

The input characteristics of the six data scenarios were standardized as follows:

$$f_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}$$

where f_{ij} represents an element of the standardized characteristic matrix; $i = 1, 2, ..., n$; n is the total number of observations or samples; $j = 1, 2, ..., m$; m is the number of characteristics of the i-th sample; x_{ij} is the j-th characteristic of the i-th sample; σ_j is the

standard deviation of the *j*-th characteristic; and μ_j is the mean of the *j*-th characteristic. The data is standardized with $\mu = 0$ and $\sigma = 1$.

Selection of Optimal Hyperparameters

The optimal hyperparameters for each classifier (SVM, RF, and MLP) were selected through the application of the grid search method and the cross-validation (CV) procedure, with $k = 5$ random partitions of the training data subset (80 %). The CV procedure consists of subdividing the training set into *k* disjoint random partitions and generating *k* training and validation subsets. Each training subset is formed with *k* - 1 partitions, and the validation subset is formed with the remaining partition.

Based on a finite set of hyperparameter values (Table 1), a combination is selected, and the model is trained *k* times by means of a CV, obtaining the average performance (*Acc*); subsequently, the combination of the values with the highest average performance of the classifier is selected.

Evaluation of the Prediction Performance of the Classifiers

The evaluation of the prediction performance of the SVM, RF, and MLP classifiers was carried out using the optimal hyperparameters, selected by means of a CV, with k = 10 partitions of the complete dataset. Ten training data subsets (90 %) and ten test subsets (10 %) were developed. For each test subset, the *Acc*, *F*1 *macro*, and *AUC macro* global performance metrics were calculated, as well as the *P*, *S*, *F*1 and *AUC* metrics per target class. The average prediction performance of each metric for each classifier was then obtained.

RESULTS AND DISCUSSION

Preprocessed Images

Leaf images were segmented from the background and filtered for analysis (Figure 3). From the total set of collected images, 80 pairs of images (upper and lower side)

Table 1. Hyperparameter sets and value intervals used in the grid search for the support vector machine (SVM), random forest (RF), and multilayer perceptron (MLP) models.

Models	Hyperparameters			
SVM	†C [0.0001, 0.001, 0.01, 0.1, 1.0, 10.0, 100.0]	γ ¶ [0.0001, 0.001, 0.01, 0.1, 1.0,10.0, 100.0, 1000.0]	*k*§ sigmoid and *rbf*	
RF	*nar*Þ [50, 100, 150, 200]	*pmax*¤ [10, 15, 20]	*crit*†† Gini and entropy	
MLP	*fa*¶¶ ReLU and *tanh*	*opt*§§ *sgd, adam,* L-BFGS	*nn*ÞÞ [50, 100, 150, 200]	α¤¤ [0.001, 0.5, 1]

†C: penalty hyperparameter; ¶γ: gamma hyperparameter; §*k*: kernel; Þ*nar:* number of trees; ¤pmax: maximum depth; ††*crit*: impurity criterion; ¶¶*fa*: activation function; §§*opt*: optimizer; ÞÞ*nn*: number of neurons; ¤¤α: alpha hyperparameter.

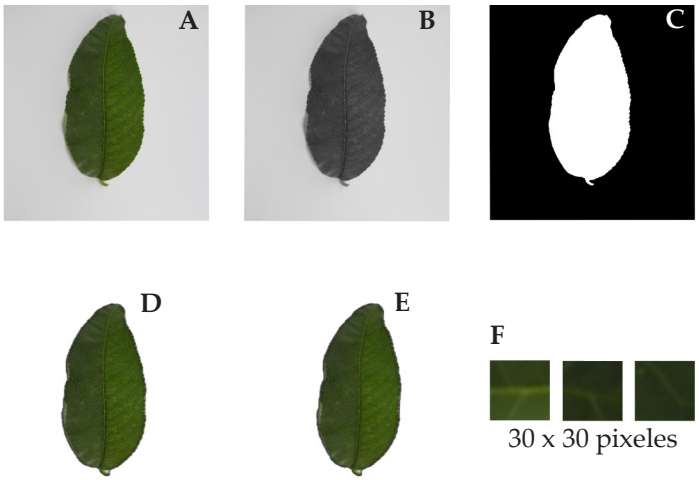

Figure 3. Image preprocessing. A: original image; B: gray scale; C: Otsu segmentation; D: object of interest; E: Gaussian filter application; and F: cuttings of areas of interest, central nerve, and secondary nerve (30 × 30 pixels each).

were randomly selected per each peach variety or target class. Three 30 × 30 pixel samples from two leaf areas (central nerve and tertiary nerve) were extracted from each image. For each target class, 480 image samples were obtained —*i.e.*, 432 000 pixels represented in three RGB or HSV color channels.

Color pixel sets were reduced to approximately 0.5 % of the original set, once repeated pixels were removed from classes and target classes. The high color similarity between the four peach varieties explained those repetitions (Table 2).

Data Input Scenarios

The datasets in each scenario varied according to the position of the leaf (upper and/or lower side) and the target class (Table 2).

Table 2. Description of the six data input scenarios and the number of color pixel samples representative of each target class (peach variety).

Clases	Data input scenarios					
	E1†	E2¶	E3§	E4Þ	E5¤	E6††
V1¶¶	2778		1871		907	
V2§§	1906		738		1168	
V3ÞÞ	2285		1041		1244	
V4¤¤	2538		676		1862	

†E1: upper and lower side in RGB format; ¶E2: upper and lower side in HSV format; §E3: upper side in RGB format; ÞE4: upper side in HSV format; ¤E5: lower side in RGB format; ††E6: lower side in HSV format; ¶¶V1: CP-03-06 variety; §§V2: Oro Azteca variety; ÞÞV3: Oro San Juan variety; ¤¤V4: Cardenal variety.

Optimal Hyperparameters

The values of the optimal hyperparameters of the SVM, RF, and MLP classifiers were selected based on the maximum average *Acc* for each classifier and input scenario; the best optimization of SVM was achieved through a Gaussian kernel (Table 3). The optimal hyperparameters of each classifier were used to evaluate the global and specific prediction performance for each peach variety.

Evaluation of the Prediction Performance

The evaluation of the prediction performance of SVM, RF, and MLP was carried out with the *Acc* and *F*1 *macro* average metrics for the six input scenarios. SVM outperformed RF and MLP in E4, with 84.1 % (*Acc*) and 83.7 % (*F*1 *macro*). The three classifiers obtained the best performance in scenario E4, upper side of the leaves transformed to the HSV color space (Table 4); these results match other color-based classification studies, including the one carried out by Dubey and Jalal (2012), where they confirm that the representation in the HSV color space (instead of RGB) improves the performance of the classifiers.

The lowest performance metrics were obtained in E5 and E6 (color of the lower side of the leaf). Including the color of the lower side did not improve performance in scenarios E1 and E2 (Table 2). The models trained with HSV color data (E2, E4, and E6) achieved better performance than models trained with RGB data (E1, E3, and E5) (Table 4). In terms of the *Acc* metric, SVM and RF performed better than MLP. The performance of the three classifiers showed a similar dispersion (Figure 4).

The discrimination performance of the SVM classifier for each peach variety or target class was obtained with the *P*, *S*, *F*1, and *AUC* metrics, in scenario E4. The V2 class

Table 3. Optimal hyperparameters of the SVM (support vector machine), random forest (RF), and multilayer perceptron (MLP) classifiers in E2[§] and E4[Þ] input scenarios.

Model	Scenario	Optimal hyperparameters			
		C[†]	γ [¶]	*k*[§]	
SVM	E2[Þ]	10.0	10.0	*rbf*	
	E4[¤]	1.0	10.0	*rbf*	
		nar[††]	*pmax*[¶¶]	*crit*[§§]	
RF	E2	100	20	*Gini*	
	E4	200	20	*Gini*	
		fa[ÞÞ]	α [¤¤]	*opt* [†††]	*nn* [¶¶¶]
MLP	E2	ReLU	0.001	*adam*	150
	E4	ReLU	0.001	*adam*	150

[†]C: penalty hyperparameter; [¶]γ: gamma hyperparameter; [§]*k*: kernel; [Þ]E2: upper and lower side, HSV; [¤]E4: upper side, HSV; [††]*nar*: number of trees; [¶¶]*pmax*: maximum depth; [§§]*crit*: impurity criterion; [ÞÞ]*fa*: activation function; [¤¤]α: alpha hyperparameter; [†††]*opt*: optimizer; [¶¶]*nn*: number of neurons.

Table 4. Prediction performance metrics (average and standard deviation) of the support vector machine (SVM), random forest (RF), and multilayer perceptron (MLP) classifiers for each input scenario.

Scenario	Classifiers					
	SVM[†]		RF[¶]		MLP[§Þ]	
	Acc[¶¶]	F1 *macro*[¤]	*Acc*	F1 *macro*	*Acc*	F1 *macro*
E1[Þ]	0.787+/0.010	0.787+/-0.010	0.744+/-0.010	0.744+/-0.010	0.718+/-0.016	0.719+/-0.016
E2[††]	0.812+/-0.013	0.811+-/0.013	0.812+/-0.14	0.812+/-0.14	0.74+/-0.012	0.74+/-0.012
E3[§§]	0.805+/-0.023	0.800+/-0.024	0.758+/-0.020	0.749+/-0.023	0.747+/-0.026	0.736+/-0.027
E4[ÞÞ]	0.841+/-0.020	0.837+/-0.022	0.829+/-0.019	0.823+/-0.021	0.77+/-0.026	0.763+/-0.027
E5[¤¤]	0.57+/-0.010	0.541+/-0.012	0.513+/-0.020	0.489+/-0.021	0.532+/-0.017	0.444+/-0.020
E6[†††]	0.663+/-0.013	0.637+/-0.013	0.714+/-0.013	0.698+/-0.014	0.622+/-0.023	0.586+/-0.023

[Þ]*Acc:* Average correct classification accuracy; [¤]F1 *macro*; [††]E1: upper and lower side, RGB; [¶¶]E2: upper and lower side, HSV; [§§]E3: upper side, RGB; [ÞÞ]E4: upper side, HSV; [¤¤]E5: lower side, RGB; [†††]E6: lower side, HSV.

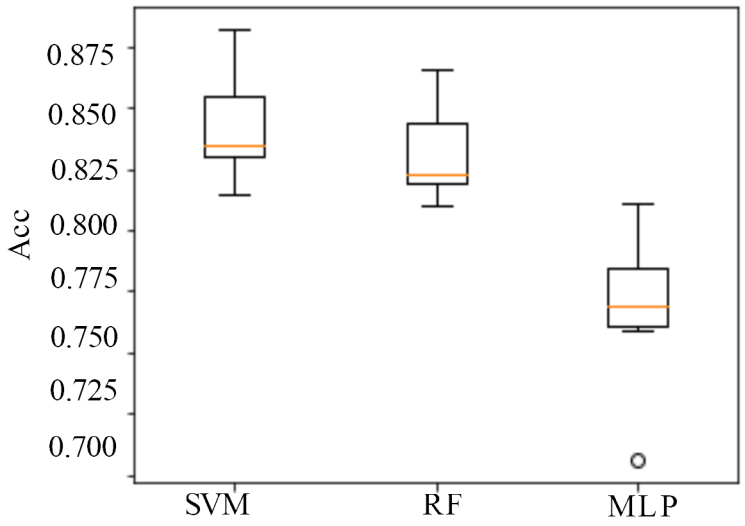

Figure 4. Comparison of the average correct classification accuracy of support vector machine (SVM), random forest (RF), and multilayer perceptron (MLP) classifiers, in input scenario E4 (upper side in HSV color space).

(Oro Azteca variety) was the best classified, with an average *F*1 of 87.9 %. V3 (Oro San Juan variety) obtained the lowest average *F*1 (74.8 %). *F*1 (*P* and *S* harmonic mean) is a more conservative estimator of classifier performance than *P* and *S*, with regard to the identification of each target class (Table 5).

Table 5. Performance metrics (average prediction) of the support vector machine (SVM) classifier in scenario E4, based on cross-validation (k = 10).

Class	*P*[¤]	*S*[††]	*F1*[¶¶]
V1[†]	0.855+/-0.013	0.880+/-0.017	0.867+/-0.009
V2[¶]	0.911+/-0.025	0.849+/-0.041	0.879+/-0.020
V3[§]	0.766+/-0.068	0.732+/-0.065	0.748+/-0.062
V4[Þ]	0.816+/-0.064	0.860+/-0.060	0.837+/-0.055

Peach varieties: [†]V1, CP-03-06; [¶]V2, Oro Azteca; [§]V3, Oro San Juan; and [Þ]V4, Cardenal. [¤]*P*: Accuracy. [††]*S*: Sensitivity. [¶¶]*F*1: *F*1 Metric.

The SVM classifier obtained its best performance in the identification of the V2 target class (in terms of the area under the ROC curve) with *AUC* (95 %) and the lowest performance in the identification of V3 with *AUC* (87 %) (Figure 5). Both metrics (*F*1 and *AUC*) have the same the SVM classification ability.

In the SVM confusion matrix of in E4 scenario (Table 6), the classifier predicts V2 with greater accuracy (*P* = 91.1 %), when the classifier confuses V2 with V1 (FP = 27), with V3 (FP = 13), and with V4 (FP = 22); likewise, SVM identified V3 with lower *P* = 76.6 % and V3 with V1 (FP = 149), V2 (FP = 43), and V4 (FP = 41). These results show that color is not enough to discriminate varieties with a high performance (> 90 %), as a consequence of the color characteristics of the upper side of the leaf, which is very similar among peach varieties.

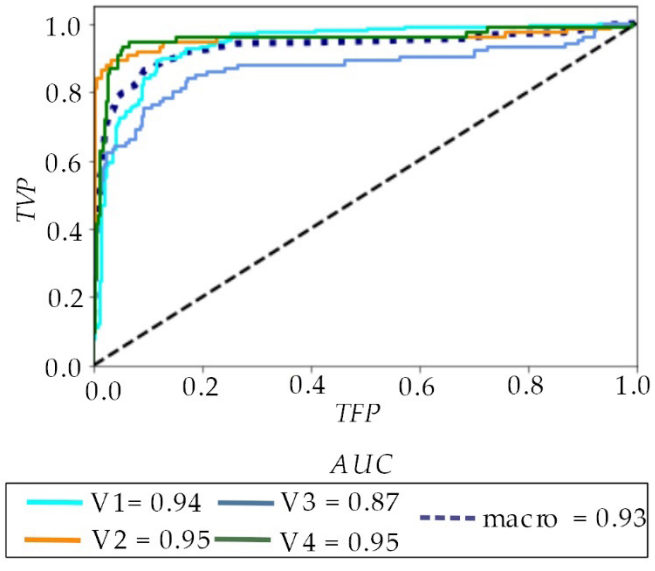

Figure 5. Area under the ROC curve (*AUC*) of the support vector machine (SVM) classifier obtained by CV cross-validation (*k* = 10) in the tenth iteration, scenario E4; *TVP*: true positive rate; *TFP*: false positive rate.

Table 6. Confusion matrix of peach varieties (classes), predicted *vs*. observed, support vector machine (SVM) classifier, cross-validation (CV), scenario E4 (upper side and HSV color space), k = 10 partitions (total dataset).

Class Observed	Predicted			
	V1	V2	V3	V4
V1[†]	1647	27	149	47
V2[¶]	40	629	43	28
V3[§]	208	13	762	57
V4[Þ]	32	22	41	585

Peach varieties: [†]V1, CP-03-06; [¶]V2, Oro Azteca; [§]V3, Oro San Juan; [Þ]V4, Cardenal.

The models that achieved the highest performance in this research were trained with data transformed to the HSV color space, in accordance with the research carried out by Dubey and Jalal (2012) and the results obtained by Łuszczkiewicz-Piątek (2014) and García-Mateos *et al.* (2015) regarding the improvement in the performance of the models in which the HSV color space is used.

The classifiers that were trained with color data from the lower side obtained a very low performance; regardless of the color space, a maximum *Acc* of 71.4 % and a minimum *Acc* of 51.3 % were obtained. The chromatic data of the lower side of the leaf did not contribute to the improvement of the performance of the models, in scenarios E1, E2, E5, and E6 (Table 4).

A limitation of color-based variety classification is data reduction after repeated pixel samples are removed; in the image samples, 99.5 % of the total pixels obtained were removed. Even with this limitation, SVM obtained an average *Acc* of 84.1 %. As suggested by Caglayan *et al.* (2013), the combination of color with shape descriptors improves the performance of classification models.

CONCLUSIONS

The support vector machine (SVM) —a supervised learning classifier— had a better prediction performance in the identification of four peaches varieties than the random forest (RF) and multilayer perceptron (MLP) classifiers. The peach varieties in question were CP-03-06 (V1), Oro Azteca (V2), Oro San Juan (V3), and Cardenal (V4). The best performance of the classifiers was obtained in input scenario E4, which consisted of using images of the upper side of the leaves transformed to the HSV color space. In this scenario, SVM achieved an average overall classification accuracy (*Acc*) of 84.1 % and an average *F*1 *macro* of 83.7 %. In terms of the ability of the classifier to identify a specific variety, SVM ranked higher than the Oro Azteca variety, with an average score *F*1 of 87.9 % and an area under the ROC curve of 95 %.

The transformation of digital images to the HSV color space was more suitable than the RGB color space for the improvement of the prediction performance of the SVM,

RF, and MLP classifiers. Likewise, the use of the color of the upper side of the leaf was more relevant than the color of the lower side or its joint use. Consequently, the three classifiers used in this research achieved their best performance with the upper side of the leaf data transformed to HSV color space. The automatic identification of characteristics of interest in plants based on artificial intelligence techniques has a great potential application in agriculture. This research shows how it can be applied in the identification of peach varieties.

ACKNOWLEDGMENTS

This research was funded by Estancias de Investigación Especializadas COMECYT 2021, classification code EESP2021-007.

REFERENCES

Ayala ND, Cervantes JC, García FL, Ayala de la Vega J, Calderón GZ. 2020. A hybrid convolutional neural network for complex leaves identification. *In*: Huang DS, Bevilacqua V, Hussain A. (eds.); Intelligent Computing Theories and Application. ICIC 2020. Lecture Notes in Computer Science, Springer: Cham, Switzerland 12463: 291–301. https://doi.org/10.1007/978-3-030-60799-9_25

Boser BE, Guyon IM, Vapnik VN. 1992. A training algorithm for optimal margin classifiers. *In*: Proceedings of the Fifth Annual Workshop on Computational Learning Theory. ACM: Pittsburgh, PA, USA 144–152. https://dx.doi.org/10.1145/130385.130401

Breiman L. 1996. Bagging predictors. Machine Learning 24: 123–140. https://dx.doi.org/10.1023/A:1018054314350

Breiman L.1999. Pasting small votes for classification in large databases and on-line. Machine Learning 36: 85–103. https://dx.doi.org/10.1023/A:1007563306331

Breiman L. 2001. Random forests. Machine Learning 45: 5–32. https://dx.doi.org/10.1023/A:1010933404324

Caglayan A., Guclu O, Can AB. 2013. A plant recognition approach using shape and color features in leaf Images. *In*: Petrosino, A. (eds.) Image Analysis and Processing ICIAP 2013. Lecture Notes in Computer Science, Springer: Berlin, Germany 8157: 161–170. https://dx.doi.org/10.1007/978-3-642-41184-7_17

Cervantes J, Garcia FL, Rodriguez LM, Zarco AH, Ruiz JSC, 2018. Complex Identification of Plants from Leaves. *In*: Huang DS, Gromiha M, Han K, Hussain A. (eds.) Intelligent Computing Methodologies. ICIC 2018. Lecture Notes in Computer Science, Springer: Cham, Switzerland 10956: 376–387. https://dx.doi.org/10.1007/978-3-319-95957-3_41

Dubey SR, Jalal AS. 2012. Detection and classification of apple fruit diseases using complete local binary patterns. *In:* Third International Conference on Computer and Communication Technology 346–351. https://dx.doi.org/10.1109/ICCCT.2012.76

Evgeniou T, Pontil M. 2001. Support vector machines: theory and applications. *In*: Paliouras G, Karkaletsis V, Spyropoulos CD. (eds.) Machine Learning and Its Applications. ACAI 1999. Lecture Notes in Computer Science, Springer: Berlin, Germany 2049: 249–257. https://dx.doi.org/10.1007/3-540-44673-7_12

Fawcett T. 2006. An introduction to ROC analysis. Pattern Recognition Letters 27 (8): 861–874. https://dx.doi.org/10.1016/j.patrec.2005.10.010

García-Mateos G, Hernández-Hernández JL, Escarabajal-Henarejos D, Jaén-Terrones S, Molina-Martínez JM. 2015. Study and comparison of color models for automatic image analysis in irrigation management applications. Agricultural Water Management 151 (31): 158–166. https://dx.doi.org/10.1016/j.agwat.2014.08.010

Grinblat GL, Uzal LC, Larese MG, Granitto PM. 2016. Deep learning for plant identification using vein morphological patterns. Computers and Electronics in Agriculture 127: 418–424. https://dx.doi.org/10.1016/j.compag.2016.07.003

Ho Y, Wookey S. 2020. The real-world-weight cross-entropy loss function: modeling the costs of mislabeling. IEEE Access 8: 4806-4813. https://dx.doi.org/10.1109/ACCESS.2019.2962617

Kingma DP, Ba J. 2014. Adam: a method for stochastic optimization. arXiv: 1412.6980.

Larese, MG, Bayá AE, Craviotto RM, Arango MR, Gallo C, Granitto PM. 2014. Multiscale recognition of legume varieties based on leaf venation images. Expert Systems with Applications 41 (10): 4638–4647. https://dx.doi.org/10.1016/j.eswa.2014.01.029

Lee SH, Chan CS, Mayo SJ, Remagnino P. 2017. How deep learning extracts and learns leaf features for plant classification. Pattern Recognition 71: 1–13. https://dx.doi.org/10.1016/j.patcog.2017.05.015

Liu N, Kan J. 2016. Improved deep belief networks and multi-feature fusion for leaf identification. Neurocomputing 216: 460–467. https://dx.doi.org/10.1016/j.neucom.2016.08.005

Łuszczkiewicz-Piątek M. 2014. Which color space should be chosen for robust color image retrieval based on mixture modeling? *In*: S. Choras, R. (eds.) Image Processing and Communications Challenges 5. Advances in Intelligent Systems and Computing. Springer: Heidelberg, Germany 233: 55–64. https://dx.doi.org/10.1007/978-3-319-01622-1_7

Rumelhart D, Hinton GE, Williams R. 1986. Learning representations by back-propagating errors. Nature 323: 533–536. https://doi.org/10.1038/323533a0

SADER (Secretaría de Agricultura y Desarrollo Rural). 2018. Michoacán primer lugar en producción de durazno a nivel nacional. https://www.gob.mx/agricultura/michoacan/articulos/michoacan-primer-lugar-en-produccion-de-durazno-a-nivel-nacional?idiom=es (Retrieved: September 2021)

Savas C, Dovis F. 2019. The impact of different kernel functions on the performance of scintillation detection based on support vector machines. Sensors 19 (23). https://doi.org/10.3390/s19235219

SIAP (Servicio de información Agroalimentaria y Pesquera). 2020. Datos Abiertos Estadística de Producción Agrícola. http://infosiap.siap.gob.mx/gobmx/datosAbiertos.php (Retrieved: August 2021).

UPOV (Unión internacional para la protección de las obtenciones vegetales). 2022. Durazno/melocotonero: directrices para la ejecución del examen de la distinción, la homogeneidad y la estabilidad. https://www.upov.int//test_guidelines/es/list.jsp (Retrieved: April 2022)

Wäldchen J, Mäder P. 2018. Plant species identification using computer vision techniques: a systematic literature review. Archives of Computational Methods in Engineering 25:507–543. https://dx.doi.org/10.1007/s11831-016-9206-z

Wang Z, Sun X, Zhang Y, Ying Z, Ma Y. 2016. Leaf recognition based on PCNN. Neural Computing and Applications 27: 899–908. https://dx.doi.org/10.1007/s00521-015-1904-1

Wu SG, Bao FS, Xu EY, Wang Y, Chang Y, Xiang Q. 2007. A leaf recognition for plant classification using probabilistic neural network. *In*: IEEE International Symposium on Signal Processing and Information Technology 11–16. https://dx.doi.org/10.1109/ISSPIT.2007.4458016

Xia C, Lee J, Li Y, Song Y, Chung B, Tae-Soo Chon. 2013. Plant leaf detection using modified active shape models. Biosystems Engineering 116 (1): 23–35. https://dx.doi.org/10.1016/j.biosystemseng.2013.06.003