ISSN 2007-9737

Use of Computer Vision Techniques for Recognition of Diseases and Pests in Tomato Plants

Ernesto García-Amaro[1,*], Jair Cervantes-Canales[1],
Farid García-Lamont[1], Francisco Marcelo Lara-Viveros[2],
José Sergio Ruiz-Castilla[1], Josué Espejel Cabrera[1]

[1] Universidad Autónoma del Estado de México,
Centro Universitario UAEM Texcoco,
Mexico

[2] Centro de Investigación en Química Aplicada,
Mexico

{jcervantesc, fgarcial}@uaemex.mx,
{ernestogarciaamaro, jsergioruizc}@gmail.com
francisco.lara@ciqa.edu.mx, jec0309@hotmail.com

Abstract. Computer vision, for decades, has been involved in solving problems in everyday life, under the implementation of different computational methods, that have evolved over time. Feature extraction, along with other computer techniques, is considered a way to develop computer vision systems; currently, plays an important role, considered a complex task, allowing to obtain essential descriptors of the segmented images, differentiating particular characteristics between different classes, even when they share similarity with each other, guaranteeing the delivery of information not redundant to classification algorithms. Likewise, in this work, a computer vision system has been developed for the recognition of foliar damage caused by diseases and pests in tomato plants. The methodology implemented is based on four modules: preprocessing, segmentation, feature extraction, and classification; in the first module, the image is preprocessed of a color space RGB to L*a*b*; in the second module, the area interest was segmented, under the implementation of the algorithm principal component analysis PCA; in the third module, features are extracted from the area of interest, obtaining texture descriptors with the Haralick algorithm, and chromatic features through Contrast descriptors, Hu moments, Gabor characteristics, Fourier descriptors, and discrete cosine transform DCT; in the fourth module, the performance of the classification algorithms were tested, with the characteristics obtained from the previous stage, considering: SVM, Backpropagation, Logistic Regression, KNN, and Random Forests.

Keywords. Tomato diseases and pests, computer vision, feature extraction.

1 Introduction

Mexico, at present, plays a very important role in the export of different crops, sown both in protected environments (greenhouses) and in the open-air. The state of Veracruz, is the main producer of sugarcane and orange, Sinaloa of white maize, Chihuahua of yellow maize, and Sonora of wheat, grown under open-air agriculture; and the state of Chiapas, is the main producer of coffee, Guanajuato of broccoli, Mexico City of Christmas eve and Sinaloa of tomato, grown in protected environments; likewise, in Mexico, the area planted with tomatoes is 42 383.3 hectares, obtaining a yield of 2 860 305.19 tons of production annual [23]. Currently, the methods in agriculture have evolved, achieving an increase in the production per plant and quality of the fruit; these results have been obtained with the implementation of new automated techniques in

doi: 10.13053/CyS-28-2-3927

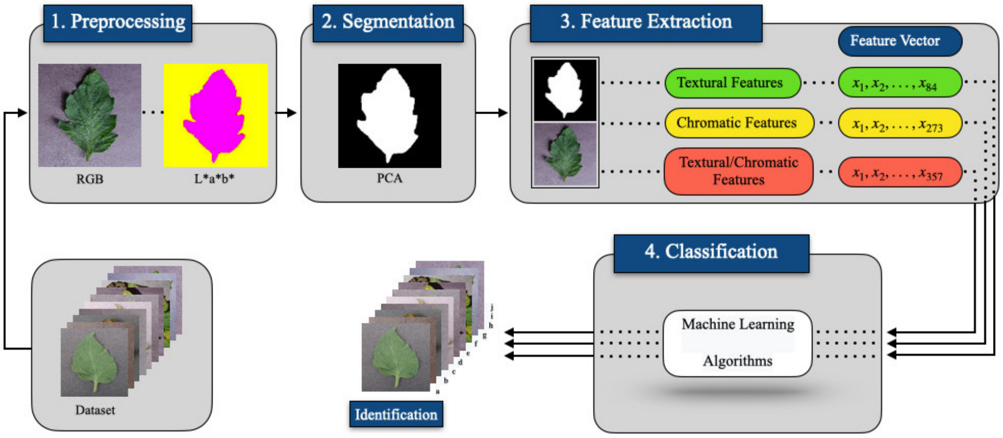


Fig. 1. Proposed method

the crops, both in the open-air and in protected environments, developing tasks in the care of planting, nutrition, growth, and harvesting of the same.

Over time, has exponentially increased the production of different crops, obtaining considerable financial income in some entities of the country, however, there are some risks in the cultivation process. Producers have reported economic declines, due to diseases y pests that have attacked tomato plants (Lycopersicum esculentum), or even, totally contaminated crops, reflecting financial losses.

Some of the most common diseases in tomato plants, the following are considered: root rot, bacterial cancer of the tomato, freckle and bacterial spot, leaf mold, gray mold, early blight, late blight, and dusty ashes [4], presented by variations of humidity, drought, temperature, residues of previous crops, wind, insects, overcast and negligence of crop operators; likewise, some of the pests more common, such as: whiteflies, leafminers, tomato psyllid, spider mites two-spotted and thrips [27], presented by variations of temperature, dust, sandy ground, humidity, inter alia; both diagnosed, through the root, stem, leaf or fruit. After the identification of an anomaly in the plant, the producer turns to experts to diagnose the disease or pest, which is considered a late detection and inaccurate; likewise, the recommended dose of a pesticide or fungicide is applied to control and/or eliminate it, generating additional expenses; in the worst-case scenario, the plants are identified with the risk that neighboring crops will be infected; therefore, contaminated plants are completely removed to prevent spread.

The main cause of loss of tomato production, is the wrong recognition of pests and diseases, since in some cases experts in the agricultural area, perform the ocular shape detection, considered an inaccurate method; for this reason, computer vision algorithms have recognized precisely foliar damage, caused by: leaf mold, late blight, early blight, bacterial spot, septoria leaf spot, target spot, tomato mosaic virus, tomato yellow leaf curl virus, spider mites two-spotted and a completely healthy class, in tomato plant leaves, avoiding the excessive or wrong application of chemical products, reducing the impact on plants and humans, in addition, contributing in the decreased loss by production, reducing financial hurt.

2 Related Work

Computer sciences, recently, have been involved in solving problems in various multidisciplinary issues, in which, the existence of living beings on planet earth becomes more stable, allowing to alert, identify or predict catastrophes that affect the environment in the one we live; in the existing literature, there are investigations with very

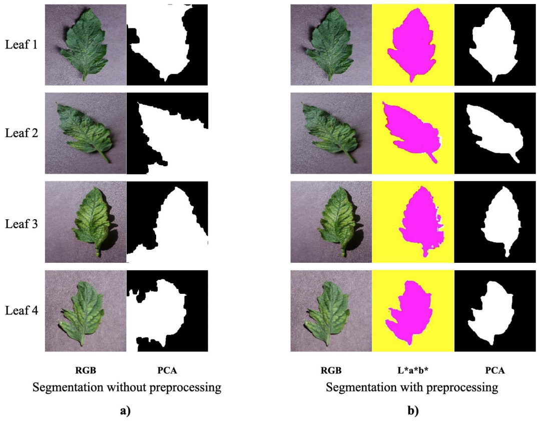

Leaf 1
Leaf 2
Leaf 3
Leaf 4
RGB
PCA
Segmentation without preprocessing
a)
RGB
L*a*b*
PCA
Segmentation with preprocessing
b)

Fig. 2. Preprocessing and segmentation

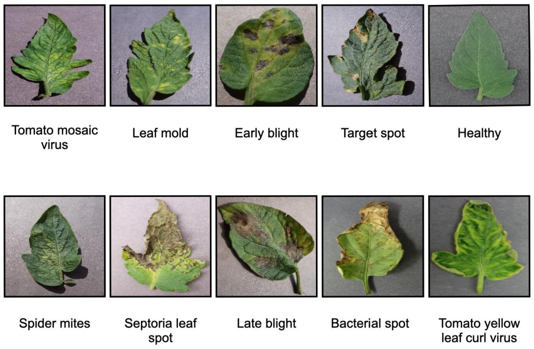


Fig. 3. Dataset plantvillage

promising results, however, the ceiling has not yet been reached and there is a great opportunity to contribute to the scientific field. The plants, in their variety of genus, are currently of great importance, since they have a fundamental role for all living beings in their entire environment.

In this part of the manuscript, the works related to this research are described, all of them focused on the agricultural area, solving issues, such as: classification and recognition of leaves, and identification of diseases and pests in plants through the leaf, implementing techniques of digital image processing, image segmentation, feature extraction, machine learning algorithms, deep learning, etc.

In the literature, exhaustive studies of works have been carried out with various methodologies, applicable to detect and classify diseases in leaves of different plants, using computer vision techniques [12]; likewise, researchers have contributed to the field of color image segmentation [18], considered a field that to this day is rigorously studied, both in controlled and uncontrolled environments, being a subject with great impact, since it influences on feature extractors and in the performance of the classification algorithms; on the other hand, under the implementation of modified fully-convolutional networks FCNs, it has been possible to segment images of plants through the leaf [36].

In previous investigations, works have been developed for the identification and classification of plants through the leaf, in [25, 40] have developed proposals methodological with deep learning techniques, specifically, convolutional neural networks CNN, comparing the performance with the architectures existing; likewise, in [7, 8, 24, 3] techniques of extraction and selection of characteristics have been implemented, considering color, shape, and texture, classifying with machine learning algorithms, obtaining favorable results for the same purpose.

In the country and in many parts of the world, the crops are affected by the unwanted arrival of pests [19] and diseases [38], both in protected environments and outdoors, likewise, this has a direct impact on production, reducing the producers financial balances; therefore, in [29] they have developed a system for the detection of diseases in different plants, using characteristics extraction techniques with Gabor wavelet transform GWT and SVM for classification; on the other hand, in [30, 28] digital image processing and machine learning methods were implemented for the recognition of diseases in tomato plant leaves.

With scientific advances and the development of new computational methods to solve problems in the field of object recognition in images, deep learning, in essence, convolutional neural networks CNN has positioned itself among the most used today, likewise, networks CNN have been evaluated for the detection of diseases and pests in tomato plants [15]; furthermore, deep learning and machine learning techniques have been merged for the same purpose [33]. In the literature, deep learning has had a great

Table 1. Dataset information

Class	Disease or pest common name	Disease or pest scientific name	Images number
a	Tomato mosaic virus	Tomato mosaic virus (ToMV)	373
b	Leaf mold	Fulvia fulva	952
c	Early blight	Alternaria solani	1000
d	Target spot	Corynespora cassiicola	1404
e	Healthy	Completely healthy leaves	1591
f	Spider mites two-spotted	Tetranychus urticae	1676
g	Septoria leaf spot	Septoria lycopersici	1771
h	Late blight	Phytophthora infestans	1908
i	Bacterial spot	Xanthomonas campestris pv. vesicatoria	2127
j	Tomato yellow leaf curl virus	Begomovirus (Fam. Geminiviridae)	5357

boost, since research has been carried out under this scheme.

With the implementation of the CNNs, has been evaluating and monitoring each proposed architecture, for the detection and recognition of diseases in tomato plant through of the leaves [2, 14, 17, 34, 37, 38, 39]; finally, and without leaving behind, in [32] a robotic system has been developed in conjunction with artificial vision techniques in greenhouses for the same purpose.

3 Materials and Methods

This section, presents the methodological proposal for this research, in which a system with four stages is exposed, preprocessing, segmentation, feature extraction, and classification; likewise, the dataset used is described. The adopted method, develops tasks such as: transformation from one color space to another, obtaining the area of interest, and the extraction of textural and chromatic features, in addition, through machine learning algorithms, has been achieved to identify foliar damage caused by diseases and pests in tomato plant leaves; contributing to the reduction of financial losses and the excessive or wrong application of chemical products in crops, decreasing their consumption in humans and plants. In Fig. 1, the implemented methodology is displayed.

3.1 Preprocessing

In stage 1 in Fig. 1 of the proposed methodology, the images of the dataset used are preprocessed, which consists of a transformation from RGB color space to L*a*b* color space.

The intensity of the different color components in RGB, determine both the tone and the brightness, in addition, it is an optimal format for the visualization of color in electronic equipment such as television and image acquisition equipment, however, it is not best suited for color image processing or segmentation, due to the high correlation between R, G and B components.

Therefore, for this research, the L*a*b* [22] color space has been used, defined by three variables: L* is the intensity, a* and b* the tonality components, the placement of this color space is similar to RGB space, but the position of the variables is different.

3.2 Segmentation

After the preprocessing of stage 1, the images have been segmented, executing the algorithm principal component analysis PCA [32], obtaining, as a result, the area of interest, which is will analyze in the next stage, determining the edges and calculating its properties, extracting textural and chromatic characteristics, and the combination of both, textural/chromatic.

Table 2. Performance of algorithms for classification

Classifier	Textural	Chromatic	Textural Chromatic
KNN	74.95	82.67	84.13
Logistic Regression	73.95	83.95	86.05
Random Forests	77.91	85.12	86.63
Backpropagation	81.83	90.65	83.76
SVM	**89.40**	**93.69**	**94.46**

Due to the nature of the dataset used, which was is in an RGB color space, the segmentation stage was supported by a previous preprocessing, which helped to competently segment the images. In Fig. 2, four tests are displayed, for four different sheets; in part a), tests were made by directly segmenting in RGB color space with the PCA algorithm, doing it incorrectly; in part b), the same tests were carried out, but before the segmentation a preprocessing was applied to the images, transforming from the RGB color space to the L*a*b* color space; likewise, it is concluded that the implemented segmentation method has a better performance by applying a previous preprocessing stage.

3.3 Feature Extraction

In this section of the manuscript, the process and techniques used to extract the characteristics of each of the images in the dataset are described, considered a delicate process and a fundamental pillar for the next stage of the proposed method; the characteristics obtained in this work are invariant to scaling, rotation, and translation, which allows the classifier to recognize objects despite their size, orientation, and position.

Likewise, an analysis has been carried out with two characteristics extraction techniques, considering, textural features, chromatic features, and the combination of both, textural/chromatic features, getting descriptors with high discriminative power, representing each image through numerical values, later, in the next stage of the proposed system, the characteristic vectors obtained are evaluated with machine learning algorithms.

3.3.1 Textural Features

The texture characteristics of a leaf, are obtained from the surface, through the area of interest generated in the second stage of the proposed methodology. The textural feature extraction algorithms, look for basic repeating patterns with periodic or random structures in images.

The texture is manifested in properties such as: roughness, harshness, granulation, fineness, smoothness, among others; likewise, it is invariant to displacements, since it repeats a pattern across a surface, therefore, it is explained because the visual perception of a texture is independent of position.

In this work, Haralick feature extractors have been implemented [20], taking into account the distribution of intensity values in the region, obtaining the mean and range of the following variables: mean, median, variance, smoothness, bias, kurtosis, correlation, entropy, contrast, homogeneity, etc; calculated as follows:

$$f_1 = \sum_i \sum_j \left[p(i,j)^2 \right], \tag{1}$$

$$f_2 = \sum_{n=0}^{N_g-1} n^2 \left\{ \begin{array}{c} \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j) \\ |i-j| = n \end{array} \right\}, \tag{2}$$

$$f_3 = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} [ijp(i,j) - \mu_x \mu_y]}{\sigma_x \sigma_y}, \tag{3}$$

$$f_4 = \sum_i \sum_j (i - \mu_x)^2 p(i,j), \tag{4}$$

$$f_5 = \sum_i \sum_j \frac{1}{1 + (i-j)^2} p(i,j), \tag{5}$$

$$f_6 = \sum_{i=2}^{2N_g} i P_{x+y}(i), \tag{6}$$

$$f_7 = \sum_{i=2}^{2N_g} (i - f_8)^2 P_{x+y}(i), \tag{7}$$

$$f_8 = -\sum_{i=2}^{2N_g} P_{x+y}(i) \log\{P_{x+y}(i)\}, \tag{8}$$

$$f_9 = -\sum_i \sum_j p(i,j) \log\{p(i,j)\}, \tag{9}$$

Table 3. Performance metrics evaluated

Metric	Formula
Accuracy	$\text{Acc} = \frac{(TP+TN)}{(TP+TN+FP+FN)}$
Precision	$\text{Precision} = \frac{TP}{(TP+FP)}$
Recall	$\text{Recall} = \frac{TP}{(TP+FN)}$
F-Measure	$\text{F-Measure} = \frac{(2*precision*recall)}{(precision+recall)}$
FP Rate	$\text{FP Rate} = \frac{FP}{(FP+TN)}$
MCC	$\text{MCC} = \frac{(TP*TN)-(FP*TN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$

$$f_{10} = \sum_{i=0}^{N_g-1} (i-f_8)^2 P_{x-y}(i), \tag{10}$$

$$f_{11} = -\sum_{i=0}^{N_g-1} P_{x-y}(i) \log\{P_{x-y}(i)\}, \tag{11}$$

$$f_{12} = \frac{HXY - HXY1}{\max\{HX, HY\}}, \tag{12}$$

$$f_{13} = (1 - e^{[-2(HXY2-HXY)]})^{\frac{1}{2}}, \tag{13}$$

$$f_{14} = (\text{Second largest eigenvalue } of\ Q)^{\frac{1}{2}}. \tag{14}$$

The vector of textural characteristics obtained X_t, can be represented as: $X_t = [x_1, x_2, \ldots, x_{84}]$. Where, the numerical value of $[x_1, x_2, \ldots, x_{28}]$ are from the R component, of $[x_{29}, x_{30}, \ldots, x_{56}]$ belongs to the G component, and of $[x_{57}, x_{58}, \ldots, x_{84}]$ are from the B component. The values of the R, G, and B components are concatenated, forming the vector X_t.

3.3.2 Chromatic Features

The chromatic characteristics, provide relevant information of a portion of the image that has been segmented, the exhaustive analysis carried out by this type of techniques, is done starting from a specific color space, for example: extracting information from the primary color channels, like: red, green and blue RGB; hue, saturation, and value HSV, L*a*b*, etc. The algorithms, Contrast descriptors [13], gabor characteristics [16, 29], Hu moments [21], discrete cosine transform DCT [9, 10], and Fourier descriptors [26], were implemented for the extraction of chromatic characteristics, calculating the descriptors of all the images in the dataset. The Contrast descriptors of an image, define information about the difference in intensity between a region and its neighborhood. The smaller the difference, the lower the contrast. Contrast is defined as follows:

$$K_1 = \frac{G - G_e}{G_e}, \tag{15}$$

$$K_2 = \frac{G - G_e}{G + G_e}, \tag{16}$$

$$K_3 = ln(G/G_e), \tag{17}$$

where G and G_e denote the mean value in the region and in the neighborhood respectively.

The Gabor characteristics, it is considered another robust technique, used for the extraction of features in images; being a hybrid technique, composed of the nucleus of the Fourier transformation on a Gaussian function; also, the frequency resolution is more sophisticated than other techniques, since the Gaussian signal is more concentrated than the rectangular function in the frequency domain. Gabor transformation is a 2D filter, represented by the following equation:

$$G(t,w) = \int_{-\infty}^{\infty} e^{-\frac{(\tau-t)^2}{2}} e^{-jwt} x(\tau) d\tau. \tag{18}$$

On the other hand, the implementation of the seven Hu moments, they have managed to integrate information of the variable of the color of the area of interest; calculated as follows:

$$\phi_1 = \eta_{20} + \eta_{02}, \tag{19}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2, \tag{20}$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2, \tag{21}$$

$$\phi_4 = (\eta_{30} - 3\eta_{12})^2 + (\eta_{21} + \eta_{03})^2, \tag{22}$$

$$\begin{aligned}\phi_5 = {} & (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 \\ & -3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \\ & [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2],\end{aligned} \tag{23}$$

$$\begin{aligned}\phi_6 = {} & (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ & +4(\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})),\end{aligned} \tag{24}$$

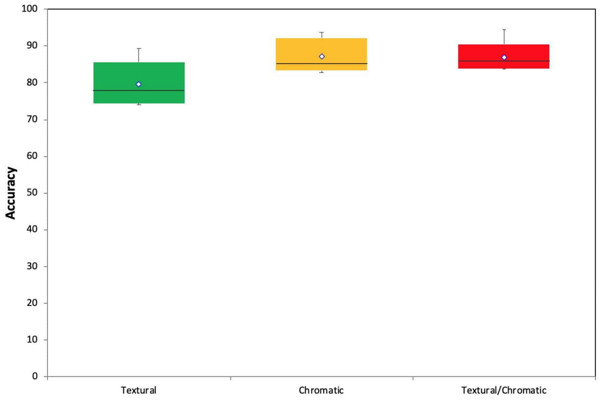


Fig. 4. Accuracy of feature extraction techniques

$$\begin{aligned}\phi_7 = (3\eta_{21}-\eta_{03})(\eta_{30}+\eta_{12})[(\eta_{30}+\eta_{12})^2\\ -3(\eta_{21}+\eta_{03})^2]-(\eta_{30}-3\eta_{12})(\eta_{21}+\eta_{03})\\ [3(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2].\end{aligned} \tag{25}$$

Likewise, the discrete cosine transform DCT, contributes to the generation of extraction of features chromatic; the DCT uses base transformations and cosine functions of different wavelengths.

A particularity about DCT in relation to the discrete Fourier transform DFT, is the limitation to the use of real coefficients. The DCT in two dimensions, is derived directly from the definition of the one-dimensional case, thus, it is calculated as follows:

$$\begin{aligned}F(u,v) = \frac{2}{\sqrt{MN}}\sum_{x=1}^{N}\sum_{y=1}^{M} I(x,y)\\ \cdot c_u \cdot \cos\left(\frac{\pi(2x+1)u}{2N}\right)\cdot c_v\cdot \cos\left(\frac{\pi(2y+1)v}{2M}\right)\\ = \frac{2c_uc_v}{\sqrt{MN}}\sum_{x=1}^{N}\sum_{y=1}^{M} I(x,y)\cdot E_u^N(x)\cdot E_v^M(y).\end{aligned} \tag{26}$$

Finally, other characteristics were obtained with the Fourier descriptors, calculated using the following equation: $d_u = |F(u)|$; where $F(u)$ is calculated for $u = 1,\ldots,N$, where N is the number of descriptors to calculate; also, for a two-dimensional function, in the case of an image $I(x,y)$ of size MxN, they are defined as:

$$F(u,v) = \frac{1}{\sqrt{MN}}\sum_{y=0}^{M-1}\sum_{x=0}^{N-1} I(x,y)\cdot e^{-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}. \tag{27}$$

After the execution of the various algorithms for extraction of chromatic characteristics, the resulting numerical vector for each image, has a length of 273, represented by the next equation: $X_t = [x_1, x_2, \ldots, x_{273}]$; where, the contrast descriptors provide 15 characteristics, considering $[x_1, x_2, \ldots, x_{15}]$; the Gabor characteristics provides 201 features, considering $[x_{16}, x_{17}, \ldots, x_{216}]$; the Hu moments add 21 characteristics, considering $[x_{217}, x_{218}, \ldots, x_{237}]$; the DCT provides 12 characteristics, considering $[x_{238}, x_{239}, \ldots, x_{249}]$; and finally, the Fourier descriptors provide 24 characteristics, considering $[x_{250}, x_{251}, \ldots, x_{273}]$, concatenated in the vector X_t.

Likewise, tests were developed combining the textural and chromatic characteristics, obtaining as a result, a numerical value of 357 characteristics for each image of the dataset. The vector of texture features $[x_1, x_2, \ldots, x_{84}]$, it has been concatenated with the vector of chromatic features $[x_1, x_2, \ldots, x_{273}]$, obtaining as a result a hybrid vector, represented by: $X_t = [x_1, x_2, \ldots, x_{357}]$.

3.4 Classification

Finally, in stage number four of the methodology, machine learning algorithms have been used to recognize ten different classes, likewise, measuring performance with, Support Vector Machines SVM, Backpropagation, K-Nearest Neighbors KNN, Random Forests, and Logistic Regression, tested with different feature extraction techniques.

In the experiments carried out, cross-validation with $k = 10$ was used to validate results, that is, 10 tests were performed with 90 % and 10 % of the data for training and testing respectively. A brief, description of the machine learning algorithms used is given below.

3.4.1 K-Nearest Neighbors KNN

The KNN algorithm, classifies a new point in the dataset, based on euclidean distance, finding the k closest distances to the object to classify. As the first instance, distances from the new point to each object in the dataset are calculated, the euclidean

Table 4. Performance by class of the tested algorithms

Class	KNN	Logistic Regression	Random Forests	Back-propagation	SVM
a	0.863	0.798	0.942	0.903	0.920
b	0.794	0.808	0.875	0.884	0.903
c	0.674	0.652	0.721	0.721	0.805
d	0.700	0.772	0.787	0.848	0.907
e	0.949	0.955	0.946	0.963	0.988
f	0.949	0.821	0.803	0.871	0.932
g	0.792	0.794	0.836	0.869	0.935
h	0.813	0.759	0.786	0.838	0.905
i	0.839	0.895	0.887	0.944	0.967
j	0.928	0.952	0.927	0.976	0.989

distance of a point a to a point b it is calculated as follows:

$$d(P_0, P_1) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \tag{28}$$

Subsequently, are located the k closest distances to the new point, finally, the class of the closest point in the dataset is assigned by majority vote. For further analysis of the KNN algorithm, refer to [1].

3.4.2 Logistic Regression

Logistic regression, is used to model the posterior class probabilities, without having to learn the conditional class densities, facilitating the classification into small training sets and less complexity.

$\pi_i = p(Y_i = 1|X_i)$ where X_i is a vector of size $1 * (p + 1)$ with the first element equal to 1, and the remaining elements, corresponding to the characteristics extracted from the leaf for the example i. The logistic regression model relates π_i with the characteristics using the function:

$$logit(\pi_i) = log(\frac{\pi_i}{1 - \pi_i}) = X_i\beta, \tag{29}$$

where $\beta = (\beta_0, \beta_1, \ldots, \beta_p)$ is the vector of regression coefficients. For a more in-depth study, refer to [5, 11].

3.4.3 Random Forests

Random Forests, is an algorithm composed of decision tree classifiers, each tree depends on the values of a random vector con with sampling independently and with the same distribution for all trees in the forest. Generalization error for forests converges to a limit, as the number of trees in the forest increases.

When a model is generalized and fails, depends on the strength of individual trees in the forest and the correlation between them. By randomly selecting features to divide each node, error rates occur that compare favorably with the Adaboost algorithm, but are more robust with respect to noise.

In [6], the Random Forests algorithm is described, specifying the characterization of precision, the use of random characteristics, the selecting random entries, the linear combination inputs, the Adaboost algorithm operation, the effects of output noise, the weak data inputs, the random forests for regression, theorems, and equations that lead to the execution of the Random Forests classifier.

3.4.4 Backpropagation

Artificial neural networks ANN, nowadays, try to imitate the learning process and solution of the human brain, this is achieved with the implementation of computational methods applied to different areas.

Humans, to solve problems of daily life, take prior knowledge, acquired from the experience of some specific area, likewise, artificial neural networks, collect information on solved problems to build models or systems that can make decisions automatically.

The multiple connections between neurons, form an adaptive system, the weights of which are updated using a particular learning algorithm. One of the most used algorithms and the one that was implemented in this work, was the algorithm of backpropagation BP; which in general, performs the learning and classification process in four points, initialization of weights, forward spread, backward spread, and the updating of weights.

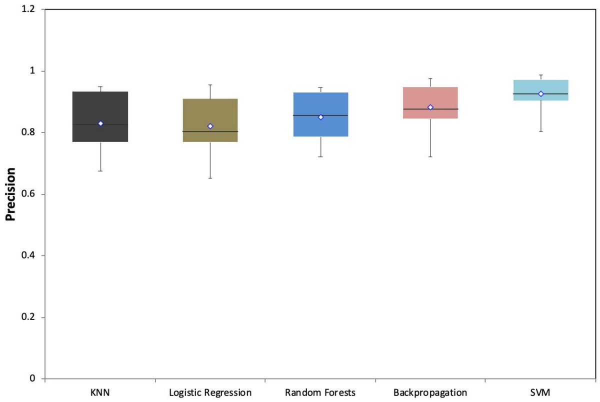


Fig. 5. Graphics of boxplots by algorithm

To carry out the learning process, backpropagation algorithm iteratively changes weights between neurons, minimizing the quadratic error between the desired output and that obtained with the current weights.

Each of the training set examples $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ are used to adjust the weights in the network. By being presented an example, the signal is propagated forward in the network until the output is obtained. The output of the j-th hidden unit is calculated as:

$$o_{nj}^{h} = f_j^{h}(net_{nj}^{h}) = \frac{1}{1 + exp(-net_{nj}^{h})}, \tag{30}$$

where $net_{nj}^{h} = \sum w_{ji}^{h} x_{ni} + \theta_j^{h}$. w_{ji}^{h} is the weight of the connection of the i-th input neuron to the j-th hidden neuron. θ_j^{h} and f_j^{h} represent the bias and the activation function of the j-th hidden neuron. So, the output of the k-th neuron is represented by:

$$o_{nk}^{o} = f_k^{o}(net_{nj}^{o}) = \frac{1}{1 + exp(-net_{nk}^{o})}, \tag{31}$$

where the superscripts h and o they refer to the quantities in the hidden and output layers respectively. For a more in-depth study of the algorithm, refer to [31].

3.4.5 Support Vector Machines SVM

The main characteristics that identify the SVM algorithm, are the use of kernels when working in non-linear sets, the absence of local minima, depends on a small subset of data and the discriminative power of the model constructed by optimizing the separability margin between the classes.

SVM is a linear classifier, in other words, it classifies between two data sets through the construction of a line that separates two classes. When this is not possible, a function called *Kernels* is used, which transforms the input space to a highly dimensional space, where the sets can be linearly separated after the transformation.

However, the choice of a function is restricted to those that satisfy the Mercer conditions. Training an SVM allows solving a quadratic programming problem, as shown below:

$$\max_{\alpha_i} -\frac{1}{2}\Sigma_{i,j=1}^{l}\alpha_i y_i \alpha_j y_j \mathbf{K}\langle x_i \cdot x_j\rangle + \Sigma_{i=1}^{l}\alpha_i. \tag{32}$$

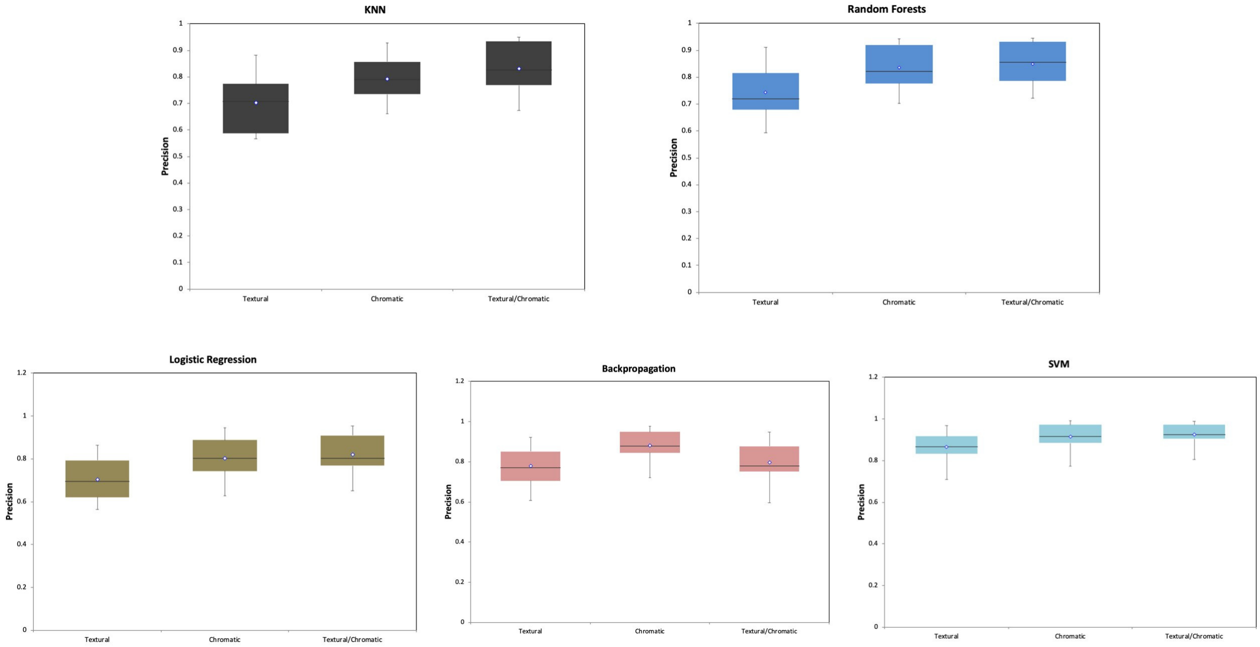
subject to: $\Sigma_{i=1}^{l}\alpha_i y_i = 0,\ C \geq \alpha_i \geq 0, i = 1, 2, \ldots, l,$
where $C > 0, \alpha_i = [\alpha_1, \alpha_2, \ldots, \alpha_l]^T, \alpha_i \geq 0, i = 1, 2, \ldots, l$ are coefficients that correspond to x_i, y_i with α_i nonzero which are called Support Vectors SV. For a more in-depth study of the algorithm, refer to [35].

3.5 Dataset

The images used in this investigation, belong to the Plantvillage dataset [14, 33, 34, 38], which has been acquired through an Internet repository of free environment; considering ten different classes, eight diseases (class a, b, c, d, g, h, i, and j), one pest (class f) and one completely healthy class (class e), the images are in an RGB color space, with dimensions of 256x256 pixels, see Table 1, and visually relate it to Fig. 3.

4 Results and Discussions

In this section of the manuscript, the metrics used are defined and the experimental results of the tests developed are analyzed and discussed. The results obtained are visualized with tables, confusion matrices, and boxplots, through exit percentages of the performance of classifiers, for each algorithm used, accuracy and precisions by class are reported.


KNN

Random Forests

Logistic Regression

Backpropagation

SVM

Fig. 6. Performance of used algorithms

4.1 Performance Metrics

Accuracy, Precision, Recall, F-Measure, FP Rate, MCC, are the metrics evaluated for the experimental results presented in this work, defined in Table 3.

4.2 Experimental Results

Table 2, shows the results of the algorithms used, evaluating performance against the feature extraction techniques mentioned in the proposed method.

The algorithm that obtained the lowest percentage of correctly classified instances, was KNN, with 82.67% for the chromatic characteristics, and for the combination of both, textural/chromatic an 84.13% respectively, nevertheless, for the test with textural features, has outperformed the algorithm Logistic Regression with 74.95%.

The Logistic Regression algorithm, was the second with lower results, outperforming to classifier KNN in tests with chromatic features and the combination of both, textural/chromatic; obtaining an 83.95% of accuracy for chromatic characteristics, and for the combination of both, textural/chromatic an 86.05%. KNN classifier and Logistic Regression had very similar behavior in their performance, however, the third-best algorithm, was Random Forests, obtaining 77.91% for textural features, for chromatic features, and textural/chromatic hybrid characteristics, it is has exceeded 85% respectively.

One of the classifiers with the best performance for this research, was the Backpropagation learning algorithm for artificial neural networks; with an accuracy percentage of 81.83% for textural characteristics, for chromatic characteristics, a 90.65% was obtained, and finally for hybrid, textural/chromatic characteristics it had the lowest performance than its counterparts, with an 83.76%.

The best performance for the proposed system, was obtained by the SVM algorithm, achieving an accuracy of 89.40% for textural features, for chromatic characteristics, 93.69% was obtained, finally, for hybrid, textural/chromatic features, was demonstrated a 94.46% respectively.

Table 5. Confusion matrix for backpropagation

	a	b	c	d	e	f	g	h	i	j
a	344	4	1	2	0	10	9	1	0	2
b	8	818	22	5	3	6	32	37	4	17
c	5	16	682	44	3	27	49	113	30	31
d	5	7	33	1158	24	103	27	20	16	11
e	1	2	0	24	1544	10	4	5	1	0
f	11	4	23	73	7	1506	17	21	1	13
g	6	23	20	23	8	13	1586	61	20	11
h	1	40	113	24	12	19	62	1602	20	15
i	0	2	23	4	3	2	22	26	2015	30
j	0	9	29	8	0	33	18	25	28	5207

In Table 2, for each classifier tested, the best results have been achieved based on the extraction of the hybrid, textural/chromatic features, except with the backpropagation learning algorithm, since the best performance has been obtained with characteristics chromatic.

In the boxplots of Fig. 4, it is notable, that the performance of the hybrid, textural/chromatic characteristics (red boxplot) considerably surpasses the textural features (green boxplot), and slightly the chromatic features (yellow boxplot).

In Fig. 4, 5, and 6, the results of each of the tests carried out with the classification algorithms and feature extraction methods are displayed, plotted using box plots, where the data distribution is analyzed, considering the median, value minimum, maximum, and intermediate. In Table 4, the best performance is reflected for each of the tested algorithms, showing the precisions by class, likewise, are graphed in Fig. 5.

The best two precisions of the tested algorithms exceed 0.94%; obtaining 0.942% for class (a) and 0.946% for class (e) with Random Forests; for class (e) and (f) 0.949% was achieved with KNN; for class (j) a 0.952% and 0.955% were obtained for class (e) with Logistic Regression; with the Backpropagation algorithm, 0.963% was achieved for class (e) and 0.976% for class (j); finally, for class (e) a 0.988% was obtained and for class (j) a 0.989% with SVM, see Table 4.

In Fig. 5, the boxplot of the algorithm KNN and Logistic Regression, have the largest data ranges, so the values are more dispersed or separated from their counterparts; likewise, the precision data ranges of the Random Forests and Backpropagation algorithm are moderately more concentrated, that the KNN and Logistic Regression algorithm; finally, the algorithm with the best performance was the SVM classifier, since the data is more concentrated compared to the other tests.

In Table 4, are plotted the precisions by class, obtained from the experimentation with the machine learning algorithms front to the features extraction methods. For each algorithm used, tests were carried out with textural features, chromatic features, and textural/chromatic hybrid features.

From the experimentation developed, the algorithm that showed the lowest performance was KNN, in addition, the precisions obtained are more dispersed in comparison with the rest of the classifiers. However, the algorithm with the best results was SVM, since the data is more concentrated than those of its counterpart.

In Table 5 and 6, confusion matrices are shown two, considering the performance of the two best classifiers, highlighting the backpropagation algorithm and SVM, likewise, an analysis of confusion between the ten classes is made. For an understanding of the confusion matrices, the nomenclature is as follows, where: a=Tomato mosaic virus, b=Leaf mold, c=Early blight, d=Target spot, e=Healthy, f=Spider mites, g=Septoria leaf spot, h=Late blight, i=Bacterial spot, and j=Tomato yellow leaf curl virus.

The matrix of the Table 5, has been built from the tests performed with the Backpropagation algorithm front to chromatic features, the analysis by class is the following: for class a, the model has confused more with the class (f); for the class b, the confusion highest was with the class (h); for the class c, the confusion highest was with the class (h); for the class d, the confusion highest was with the class (f); for the class e, the confusion highest was with the class (d); for the class f, the confusion highest was with the class (d); for the class g, the confusion highest was with the class (h); for the class h, the confusion highest was with the class (c); for the class i, the confusion highest was with the class (j); and for the class j, the confusion highest was with the class (f). The matrix of the Table 6, has been built from the tests performed with the algorithm SVM, front to chromatic features, the analysis by class is the following: for class a,

Table 6. Confusion matrix for SVM

	a	b	c	d	e	f	g	h	i	j
a	352	6	0	1	0	6	5	2	0	1
b	5	877	14	2	0	3	19	25	2	5
c	6	19	783	34	1	9	17	103	13	15
d	2	1	24	1270	3	69	11	10	7	7
e	0	1	3	15	1563	1	3	5	0	0
f	8	4	14	75	1	1552	4	10	0	8
g	13	24	20	20	0	4	1639	32	9	10
h	4	35	114	9	7	13	35	1665	14	12
i	0	2	20	6	2	0	5	17	2054	21
j	0	10	21	8	0	14	7	12	26	5259

the model has confused more with the class (b,f); for the class b, the confusion highest was with the class (h); for the class c, the confusion highest was with the class (h); for the class d, the confusion highest was with the class (f); for the class e, the confusion highest was with the class (d); for the class f, the confusion highest was with the class (d); for the class g, the confusion highest was with the class (h); for the class h, the confusion highest was with the class (c); for the class i, the confusion highest was with the class (j); and for the class j, the confusion highest was with the class (i). In most of the experimental tests, the class that showed the most confusion and the one that the models assigned as correct, was class h, belonging to late blight disease.

5 Conclusions

In this manuscript, work was developed based on features extraction techniques and machine learning, for the recognition of foliar damage caused by pests and diseases that affect tomato plants. After preprocessing and image segmentation, the proposed system extracts textural features, chromatic features, and the features hybrid, textural/chromatic, finally, automatic learning algorithms evaluate the obtained descriptors.

Derived from the tests in the preprocessing and segmentation stage, it is verified that the implemented segmentation method, has a better performance by applying a previous preprocessing stage; likewise, of the three characteristics extraction methods implemented in this research, the one that obtained the best descriptors, directly impacting on the performance of the classifiers, were the features hybrid, textural/chromatic; furthermore, the best classifier was SVM; therefore, it was shown, that by applying the image color space transformation of input, the segmentation PCA method, the conjunction of textural/chromatic feature extraction, and the SVM classification process, the system has achieved a performance favorably.

References

1. **Aha, D. W., Kibler, D., Albert, M. K. (1991).** Instance-based learning algorithms. Machine Learning, Vol. 6, pp. 37–66. DOI: 10.1007/bf00153759.

2. **Ahmad, I., Hamid, M., Yousaf, S., Shah, S. T., Ahmad, M. O. (2020).** Optimizing pretrained convolutional neural networks for tomato leaf disease detection. Complexity, Vol. 2020, pp. 1–6. DOI: 10.1155/2020/8812019.

3. **Ayala-Niño, D., Ruíz-Castilla, J. S., Arévalo-Zenteno, M. D., Jalili, L. D. (2019).** Complex leaves classification with features extractor. Intelligent Computing Theories and Application, Springer International Publishing, pp. 758–769. DOI: 10.1007/978-3-030-26969-2_72.

4. **Blancard, D. (2011).** Enfermedades del tomate. Mundi-Prensa.

5. **Borges, J. S., Bioucas-Dias, J. M., Marcal, A. R. (2011).** Bayesian hyperspectral image segmentation with discriminative class learning. IEEE Transactions on Geoscience and Remote Sensing, Vol. 49, No. 6, pp. 2151–2164. DOI: 10.1109/tgrs.2010.2097268.

6. **Breiman, L. (2001).** Random forests. Machine learning, Vol. 45, pp. 5–32. DOI: 10.1023/a:1010933404324.

7. **Cervantes, J., Garcia-Lamont, F., Rodriguez-Mazahua, L., Zarco-Hidalgo, A., Ruiz-Castilla, J. S. (2018).** Complex identification of plants from leaves.

Intelligent Computing Methodologies: 14th International Conference, ICIC 2018, Springer International Publishing, pp. 376–387. DOI: 10.1007/978-3-319-95957-3_41.

8. **Cervantes, J., Taltempa, J., García-Lamont, F., Ruiz-Castilla, J. S., Yee-Rendon, A., Jalili, L. D. (2017).** Análisis comparativo de las técnicas utilizadas en un sistema de reconocimiento de hojas de planta. Revista Iberoamericana de Automática e Informática Industrial RIAI, Vol. 14, No. 1, pp. 104–114. DOI: 10.1016/j.riai.2016.09.005.

9. **Cuevas, E., Zaldívar, D., Pérez, M. (2016).** Procesamiento digital de imágenes con MATLAB & Simulink. Ra-Ma.

10. **Dabbaghchian, S., Ghaemmaghami, M. P., Aghagolzadeh, A. (2010).** Feature extraction using discrete cosine transform and discrimination power analysis with a face recognition technology. Pattern Recognition, Vol. 43, No. 4, pp. 1431–1440. DOI: 10.1016/j.patcog.2009.11.001.

11. **Dempster, A. P., Laird, N. M., Rubin, D. B. (1977).** Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society: Series B (Methodological), Vol. 39, No. 1, pp. 1–22. DOI: 10.1111/j.2517-6161.1977.tb01600.x.

12. **Dhingra, G., Kumar, V., Joshi, H. D. (2017).** Study of digital image processing techniques for leaf disease detection and classification. Multimedia Tools and Applications, Vol. 77, No. 15, pp. 19951–20000. DOI: 10.1007/s11042-017-5445-8.

13. **Donis-González, I. R., Guyer, D. E., Pease, A. (2016).** Postharvest noninvasive classification of tough-fibrous asparagus using computed tomography images. Postharvest Biology and Technology, Vol. 121, pp. 27–35. DOI: 10.1016/j.postharvbio.2016.07.012.

14. **Durmus, H., Gunes, E. O., Kirci, M. (2017).** Disease detection on the leaves of the tomato plants by using deep learning. 2017 6th International Conference on Agro-Geoinformatics, IEEE, pp. 1–5. DOI: 10.1109/agro-geoinformatics.2017.8047016.

15. **Fuentes, A., Yoon, S., Kim, S. C., Park, D. S. (2017).** A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. Sensors, Vol. 17, No. 9, pp. 2022. DOI: 10.3390/s17092022.

16. **Gabor, D. (1946).** Theory of communication: Journal of the institute of electrical engineers. Vol. 93, pp. 429–457.

17. **Gadekallu, T. R., Rajput, D. S., Reddy, M. P. K., Lakshmanna, K., Bhattacharya, S., Singh, S., Jolfaei, A., Alazab, M. (2020).** A novel PCA–whale optimization-based deep neural network model for classification of tomato plant diseases using GPU. Journal of Real-Time Image Processing, Vol. 18. DOI: 10.1007/s11554-020-00987-8.

18. **Garcia-Lamont, F., Cervantes, J., López, A., Rodriguez, L. (2018).** Segmentation of images by color features: A survey. Neurocomputing, Vol. 292, pp. 1–27. DOI: 10.1016/j.neucom.2018.01.091.

19. **Gutierrez, A., Ansuategi, A., Susperregi, L., Tubío, C., Rankić, I., Lenža, L. (2019).** A benchmarking of learning strategies for pest detection and identification on tomato plants for autonomous scouting robots using internal databases. Journal of Sensors, Vol. 2019, pp. 1–15. DOI: 10.1155/2019/5219471.

20. **Haralick, R. M., Shanmugam, K., Dinstein, I. H. (1973).** Textural features for image classification. IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-3, No. 6, pp. 610–621. DOI: 10.1109/tsmc.1973.4309314.

21. **Hu, M. K. (1962).** Visual pattern recognition by moment invariants. IRE Transactions on Information Theory, Vol. 8, No. 2, pp. 179–187. DOI: 10.1109/tit.1962.1057692.

22. **Huang, R., Sang, N., Luo, D., Tang, Q. (2011).** Image segmentation via coherent clustering in L$*$a$*$b$*$ color space. Pattern Recognition Letters, Vol. 32, No. 7,

pp. 891–902. DOI: 10.1016/j.patrec.2011.01.013.

23. **INEGI (2019).** Encuesta nacional agropecuaria 2019. Report ENA. 2019, Instituto Nacional de Estadística y Geografía, www.inegi.org.mx/temas/agricultura/.

24. **Jalili, L. D., Morales, A., Cervantes, J., Ruiz-Castilla, J. S. (2016).** Improving the performance of leaves identification by features selection with genetic algorithms. Workshop on engineering applications, Springer International Publishing, pp. 103–114. DOI: 10.1007/978-3-319-50880-1_10.

25. **Jiao, Z., Zhang, L., Yuan, C. A., Qin, X., Shang, L. (2019).** Plant leaf recognition based on conditional generative adversarial nets. Intelligent Computing Theories and Application, Springer International Publishing, pp. 312–319. DOI: 10.1007/978-3-030-26763-6_30.

26. **Mingqiang, Y., Kidiyo, K., Joseph, R. (2008).** A survey of shape feature extraction techniques. Pattern Recognition Techniques, Technology and Applications, Vol. 15, No. 7, pp. 43–90. DOI: 10.5772/6237.

27. **Miyao, G. (2016).** Tomato: UC IPM pest management guidelines. UC ANR Publication 3470. https://www2.ipm.ucanr.edu/agriculture/tomato/.

28. **Mokhtar, U., Ali, M. A., Hassenian, A. E., Hefny, H. (2015).** Tomato leaves diseases detection approach based on support vector machines. 2015 11th International Computer Engineering Conference (ICENCO), IEEE, pp. 246–250. DOI: 10.1109/icenco.2015.7416356.

29. **Prasad, S., Kumar, P., Hazra, R., Kumar, A. (2012).** Plant leaf disease detection using gabor wavelet transform. Swarm, Evolutionary, and Memetic Computing, Springer Berlin Heidelberg, pp. 372–379. DOI: 10.1007/978-3-642-35380-2_44.

30. **Raza, S. E. A., Prince, G., Clarkson, J. P., Rajpoot, N. M. (2015).** Automatic detection of diseased tomato plants using thermal and stereo visible light images. PLOS ONE, Vol. 10, No. 4, pp. e0123262. DOI: 10.1371/journal.pone.0123262.

31. **Rumelhart, D. E., Hinton, G. E., Williams, R. J. (1986).** Learning representations by back-propagating errors. Nature, Vol. 323, No. 6088, pp. 533–536. DOI: 10.1038/323533a0.

32. **Schor, N., Bechar, A., Ignat, T., Dombrovsky, A., Elad, Y., Berman, S. (2016).** Robotic disease detection in greenhouses: Combined detection of powdery mildew and tomato spotted wilt virus. IEEE Robotics and Automation Letters, Vol. 1, No. 1, pp. 354–360. DOI: 10.1109/lra.2016.2518214.

33. **Shijie, J., Peiyi, J., Siping, H. (2017).** Automatic detection of tomato diseases and pests based on leaf images. 2017 Chinese Automation Congress (CAC), IEEE, pp. 2537–2510. DOI: 10.1109/cac.2017.8243388.

34. **Suryawati, E., Sustika, R., Yuwana, R. S., Subekti, A., Pardede, H. F. (2018).** Deep structured convolutional neural network for tomato diseases detection. 2018 International Conference on Advanced Computer Science and Information Systems (ICACSIS), IEEE, pp. 385–390. DOI: 10.1109/icacsis.2018.8618169.

35. **Vapnik, V. N. (1999).** An overview of statistical learning theory. IEEE Transactions on Neural Networks, Vol. 10, No. 5, pp. 988–999. DOI: 10.1109/72.788640.

36. **Wang, X., Wang, Z., Zhang, S. W. (2019).** Segmenting crop disease leaf image by modified fully-convolutional networks. Intelligent Computing Theories and Application, Springer International Publishing, pp. 646–652. DOI: 10.1007/978-3-030-26763-6_62.

37. **Wu, Q., Chen, Y., Meng, J. (2020).** DCGAN-based data augmentation for tomato leaf disease identification. IEEE Access,

Vol. 8, pp. 98716–98728. DOI: 10.1109/access.2020.2997001.

38. **Zhang, K., Wu, Q., Liu, A., Meng, X. (2018).** Can deep learning identify tomato leaf disease?. Advances in Multimedia, Vol. 2018, pp. 1–10. DOI: 10.1155/2018/6710865.

39. **Zhang, Y., Song, C., Zhang, D. (2020).** Deep learning-based object detection improvement for tomato disease. IEEE Access, Vol. 8, pp. 56607–56614. DOI: 10.1109/access.2020.2982456.

40. **Zheng, Y., Yuan, C. A., Shang, L., Huang, Z. K. (2019).** Leaf recognition based on capsule network. Intelligent Computing Theories and Application, Springer International Publishing, pp. 320–325. DOI: 10.1007/978-3-030-26763-6_31.

Article received on 03/04/2021; accepted on 23/04/2024.
Corresponding author is Ernesto García-Amaro.