A Fast and Accurate Method for Classifying Tomato Plant Health Status Using Machine Learning and Image Processing

Hasan Ulutas[1, *], **Veysel Aslantas**[2]
[1]*Department of Computer Engineering, Yozgat Bozok University,*
Ataturk Road 7.km, 66200 Yozgat, Turkey
[2]*Department of Computer Engineering, Erciyes University,*
Meliksah District, 38200 Yozgat, Turkey
[*]*hasan.ulutas@bozok.edu.tr, aslantas@erciyes.edu.tr*

***Abstract*—Agriculture is crucial to economic growth and development, and maintaining high-quality, disease-free plants is crucial to its success. Early detection of plant diseases, which can be caused by environmental factors, fungi, bacteria, and viruses, is essential to implement appropriate treatments. Tomatoes, which are one of the most vital food crops, are susceptible to diseases that can result in significant economic losses in agriculture.**

This study introduces a method to evaluate the health of tomato leaf using image processing techniques and machine learning algorithms. A dataset of 1,778 images of healthy and infected tomato leaves was collected from tomato planting areas in the Turkish provinces of Samsun and Mersin. Sixteen advanced machine learning algorithms were used for classification, and the optimal hyper parameters for each algorithm were determined using a grid search approach. The classifiers were executed on Jetson Nano and TX2 embedded systems.

The experimental results indicate that the Random Forest classifier outperformed other algorithms, achieving approximately 99 % accuracy in detecting and classifying the health status of tomato leaves. The proposed system enables faster and more accurate detection, allowing farmers to classify plants as infected or healthy, ultimately improving decision-making on treatment and pest management strategies.

***Index Terms*—Classification algorithms; Image processing; Smart agriculture; Machine learning algorithms.**

I. Introduction

Plants are critical to the environment and humanity. Without them, sustaining the ecology of the Earth would be impossible. They are widely used in various fields, including energy, industry, food, and medicine. Plant infections and diseases significantly affect crop quality and quantity. This situation has detrimental effects on the economies of nations where agriculture is the primary source of income [1]. Early detection, diagnosis, and management of crop infections are vital to reduce crop damage and maximise crop production, quality, and quantity. According to research in this field, there are approximately 500,000 plant species worldwide. New species have been discovered as a result of research by plant experts, and the number of existing plant species is increasing day by day. However, certain plant species are threatened with extinction due to seasonal conditions and environmental pollution. Therefore, research in this field is essential to protect plants and discover new plant species [2]–[4].

Numerous diseases affect plants due to adverse environmental and seasonal conditions. Each year, these diseases result in significant productivity losses and economic impacts. Consequently, early detection of plant diseases and timely administration of appropriate actions are of crucial importance [5]. Experts in this field are responsible for identifying plant species and diseases. However, these processes are vital and challenging. To ensure the sensitivity and reliability of the identification results, visual examinations are typically conducted first, followed by laboratory examinations. However, these conventional methods require lengthy, tedious, and complex processes. For example, numerous biological tests and microscopic examinations must be performed to identify the species of thousands of plants. Extensive analysis is required, especially considering the similar characteristics of plants within the same family [3], [4].

Traditionally, plant species and diseases have been classified using conventional methods. Due to the shortcomings of these methods, computers have become inevitable substitutes in this area. Advancements in computer vision, machine learning, and deep learning can be used for accurate, rapid, and early identification of a substantial number of plant diseases to address the aforementioned problems in modern farming. These computerised approaches that use image processing techniques provide quick and precise solutions to the problems considered [6].

To diagnose non-invasively plant diseases and automate the entire process, leaf images are subjected to digital image processing techniques. Imaging technology has become increasingly important due to advancements in computer

Manuscript received 16 December, 2022; accepted 4 March, 2023.

technology, and numerous studies and applications for object analysis have been conducted. In this context, data analysis previously performed by individuals can be performed automatically and more easily using image processing techniques [2], [3]. Wang, Li, Ma, and Li [7] proposed a hybrid system based on image identification for the detection of grapes and wheat. First, image clipping, image balancing, and k-average clustering algorithms were used to divide diseased plant images. Then, by employing shape, colour, and texture-based methods, feature vectors were extracted from the images of the diseased plant. Finally, the performance of attributes derived from the backpropagation network was calculated. According to the experimental results reported, a 100 % accuracy score was obtained for the detection of wheat and grape diseases. Singh and Misra [8] utilised segmentation in conjunction with a genetic algorithm to identify leaf damage caused by the spread of several diseases.

Arivazhagan, Shebiah, Ananthi, and Varthini [9] presented a four-stage methodology for detecting plant diseases in their study. After applying colour conversion to the RGB image and masking the green pixels, the segmentation procedure was performed by setting a threshold value. The segmented images were then used to extract tissue attributes. After all, the performance of the suggested model was evaluated using the Support Vector Machine (SVM) classifier. The suggested model was validated using a database of around 500 diseased plant leaves, yielding a 94.74 % accuracy score. Bashir [10] developed a hybrid system for the detection of Malus domestica disease using colour analysis based on clustering methods on k-average. The combination of features of the proposed system has been shown to be extremely effective in detecting diseases and improving performance.

Munisami, Ramsurn, Kishnah, and Pudaruth [11] proposed a recognition system for classifying plant species based on leaf images. The classifications were made using the k-Nearest Neighbors (k-NN) method based on morphological and colour characteristics. The proposed system was evaluated using the Folio dataset. In experimental studies, an accuracy score of 87.3 % was reported.

Nguyen Thanh Le, Apopei, and Alameh [12] suggested a multiclass plant recognition system based on the Local Binary Pattern (LBP) approach and an SVM classifier to extract the textural characteristics of the leaf. The best classification performance of 91.85 % was obtained when variable parameters such as neighbor count and radius were used for the LBP approach.

Herdiyeni and Santoni [13] proposed a combined system for plant recognition based on the characteristics of texture, shape, and colour in their study. To extract the properties of the leaf texture, statistical colour moments were used to distinguish the leaf colour using the proposed local binary cover variant. Following the collection of these attribute parameters, the classification performance was calculated employing the Probabilistic Neural Network (PNN) method. The proposed system was evaluated using a dataset containing 2448 sheet images classified into 51 classes. According to the experimental results obtained, the proposed system achieved a 72.18 % accuracy score.

Wang, Liang, and Guo [14] developed a new algorithm based on double-scale separation and local binary pattern methods to extract distinguished properties from plant leaf images while minimising noise distortion. The k-NN method was used to improve the classification performance. Experimental studies on the Flavia and ICL datasets demonstrated 99.25 % and 98.03 % accuracy, respectively.

Elhariri, El-Bendary, and Hassanien [15] proposed a system based on a combination of colour characteristics, vascular properties, shape characteristics, and tissue properties. Different types of plants were classified using Random Forests (RF) and Linear Differential Analysis (LDA) algorithms. In experimental studies using a dataset of 340 sheet images, the LDA classification achieved the highest accuracy score of 92.65 %.

Lee, Chung, and Hong [16] suggested a plant recognition system based on the vein and shape of the plant leaf in their study. A Fast Fourier Transformation (FFT) was performed in the proposed approach using the distances between the pixels on the leaf boundary curve and the centre point. In addition to that, geometric and vascular characteristics were extracted using statistical equations and then all characteristics were combined. The Flavia dataset was used to verify the validity of the proposed system. The experimental results indicate that the recommended leaf recognition system achieved a 97.19 % accuracy score.

Mahdikhanlou and Ebrahimnezhad [17] proposed an approach based on the Minimum Axis of Inertia and Central Edge Length methods, which employ the boundary curves of the leaf shape. The PNN classifier was used to calculate the individual and hybrid performance of the attributes derived from these methods. According to the results obtained in the experimental studies, 82.05 % accuracy scores were obtained for the Swedish leaf dataset, and 80.10 % accuracy scores were obtained for the Flavia leaf data. Today, there are many datasets on plant species, and numerous research studies are conducted to classify plant species using these datasets. These studies were, on average, successful by more than 85 %.

In this study, a new classification approach is proposed that classifies tomato leaves according to their health status using machine learning. The primary contributions of this study are as follows:

– Comprehensive Dataset Collection: The authors collected a rich dataset comprising 1,778 images of healthy and diseased tomato leaves from various tomato planting regions in Turkey, establishing a strong foundation for this study.

– Robust Image Processing Pipeline: The study introduces a reliable image processing pipeline that encompasses pre-processing, segmentation, and feature extraction, enabling the derivation of meaningful attributes for training machine learning models.

– Hyperparameter Optimisation Strategy: By employing a grid search method, the authors optimised the hyperparameters for various machine learning classifiers, resulting in improved performance and accuracy.

– Thorough Algorithm Evaluation: The study systematically evaluated the performance of 16 machine learning algorithms on two embedded systems (Jetson Nano and Jetson TX2) using different datasets, offering valuable information about their respective advantages and disadvantages.

– Identification of the Best Classifier: This study identifies the Random Forest classifier as the most effective algorithm for this task, achieving an impressive accuracy rate of approximately 99 %.

– Development of an Efficient Detection System: The proposed system enables a rapid and objective evaluation of tomato leaf health, providing a practical solution for farmers to better manage their crops and minimise possible crop loss. Detailed Embedded Systems Analysis: The authors offer an in-depth comparison of the training and testing times for machine learning algorithms on the two embedded systems, providing valuable guidance to users in choosing the most appropriate platform for their specific needs.

– A Clear Roadmap for Future Research: The study highlights several promising directions for future research, including expanding the dataset to encompass more diseases and crops, using advanced image sources, such as aerial photographs, and exploring hybrid algorithms to further enhance the performance of diagnostic systems.

These scientific contributions underscore the value of this study in developing a highly accurate, efficient, and user-friendly system for diagnosing the health status of tomato leaves, ultimately benefiting farmers and the agricultural industry as a whole.

The manuscript is divided into five sections. Section II outlines the phases of detection and categorisation of the health status of tomato plants. General steps, such as image acquisition, data pre-processing, image segmentation, feature extraction, and classification of health status, are briefly described. In addition, various algorithms and techniques commonly used in conjunction with automatic plant health status detection and classification using machine learning in embedded systems are also explained. Section III presents the machine learning techniques with various criteria and k-fold results in detail. In Section IV, several challenges and unanswered questions regarding machine learning and disease detection are discussed for the detection and classification of plant health status, as well as future research prospects. The performance of these frameworks largely depends on the size of the dataset and the classifiers used. Section V presents concluding remarks that summarise information available and the issues identified in this investigation.

The introduction should present the case for the study, highlighting only the essential background and not including the findings or conclusions. It should not be a review of the subject area, but it should finish with a clear statement of the question being addressed.

II. Materials and Methods

Plant diseases and infections can be detected in numerous agricultural areas using imaging techniques. Several studies have employed image processing techniques for the identification of plant diseases, and scientists continue to research new methods to monitor plant diseases and develop practical tools for use in the field [18], [19].

This study also uses image processing techniques, such as resizing, smoothing, thresholding, and segmentation. The basic procedures of the proposed leaf classification system are given in Fig. 1. The proposed technique for classifying the health status of leaves consists of four stages: dataset construction, pre-processing and segmentation, feature extraction, and classification. Each stage and its respective operations are explained in detail below.

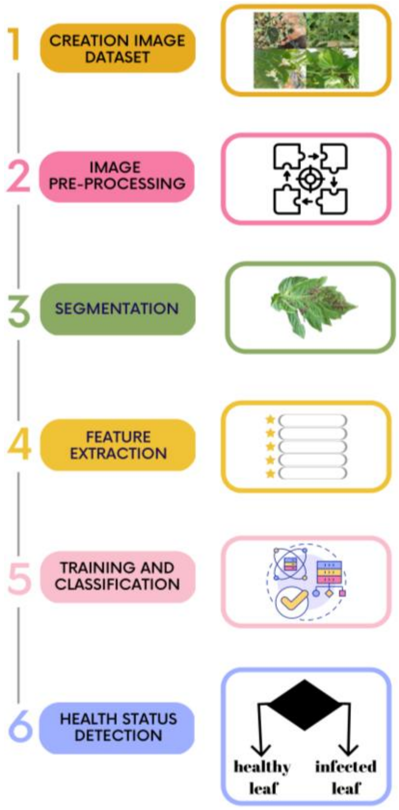


Fig. 1. The basic procedures of the proposed leaf classification system.

A. Dataset

In this work, studies were carried out on the health status of tomato plants. Images were collected from tomato planting areas in the provinces of Samsun (Carsamba Plain, Gungor Farm) and Mersin (Silifke Cay Farm) of Turkey. To build the dataset, images of tomato plants were captured using a Redmi Note 9 pro AI quad camera and a Samsung A51 quad rear camera. The Redmi Note 9 Pro is equipped with a 64-megapixel primary camera with an f/1.89 focal length, an 8-megapixel ultrawide-angle camera with an f/2.2 focal length, a 2-megapixel depth camera with an f/2.4 focal length, and a 5-megapixel macro camera with an f/2.4 focal length. The Samsung A51 is equipped with a 48-megapixel primary camera with an f/2.0 focal length, a 12-megapixel ultrawide-angle camera with an f/2.2 focal length, a 5-megapixel depth camera with an f/2.2 focal length, and a 5-megapixel macro camera with an f/2.4 focal length. These systems are capable of shooting in 4K at 30 frames per second.

A total of 1778 images of healthy and infected leaves were captured. Some of the images from the constructed dataset are shown in Fig. 2. The dataset consists primarily of two types of images: one that contains healthy leaf images and another that contains infected leaf images. Sample images of healthy and infected leaves are illustrated in Fig. 3.

Fig. 2. Sample images from our dataset.

The images collected were divided into two groups: training and test datasets. On each set, pre-processing, segmentation, feature extraction, and classification processes were performed. Feature extraction computations on the training dataset produce attributes that will be used for the training process to classify the leaves as healthy or infected. The trained model was then tested using the test dataset.

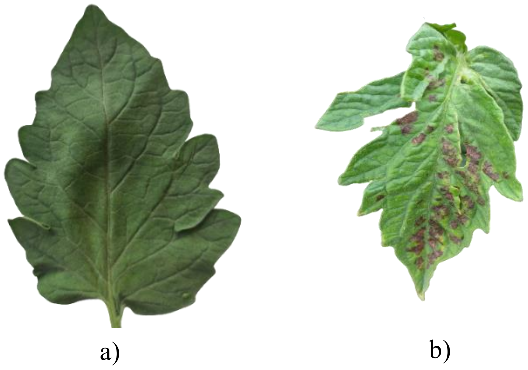


Fig. 3. Leaf images: a) healthy leaf and b) infected leaf.

B. Pre-Processing

As stated by Gibert, Sànchez-Marrè, and J. Izquierdo [20], real data often contains noise, uncertainty, errors, redundancies, or irrelevant information. Therefore, pre-processing is crucial in any data analysis process. This is because the collected images are pre-processed to bring them to the same benchmarking standard specifications to which various machine learning models can be applied. Several pre-processing methods are involved in this stage: noise removal, distortion removal, colour space conversion, image resizing and cropping, smoothing, enhancement, etc. [21]. The primary stage of pre-processing is resizing the input images. Mostly, the image's initial size is so large that it requires additional processing time. Therefore, each image was down sampled to 275×185 pixels for computational efficiency. Before proceeding to the next processing phases of image segmentation and feature extraction, noise and distortion must be eliminated in the image [22]. Otherwise, these problems could have a detrimental impact on system performance. A Gaussian blur filter was employed to eliminate the noise, and bilinear spatial transform and interpolation methods were used for the distortion problem. All the RGB images were transformed to the HSL format. Only the H (hue) component was taken into account, as it contains the necessary information for the problem considered in this work [23], [24].

C. Segmentation

Segmentation is the process of separating an image into its constituent parts. In this study, the segmentation process consisted of two stages. In the first stage, the images were segmented into leaf and background areas. In the second stage, the leaf area obtained in the previous stage was segmented into healthy and infected regions. These regions provide valuable information for learning and categorisation [6]. The mean shift clustering algorithm and the OTSU method, which are widely used in the literature, are preferred for the segmentation process.

– *Mean shift clustering*

The mean shift algorithm is a sophisticated and versatile technique for clustering-based segmentation [25]. It is a centroid-based method for determining the centres of each group and cluster. It calculates the associated peak for each individual point. The algorithm then repositions the window to its mean position and repeats this process until convergence occurs. At each iteration, the window will be replaced or shifted to a section of the dataset that is more densely populated than the previous section until the peak is reached, at which point the data are evenly distributed. This feature of the mean shift algorithm was used to segment colour images to extract the infected portion of the leaf.

– *Otsu method*

Green pixels were masked because they represent the healthy region of a leaf. The green pixel masking phase was divided into two sections: Otsu's method was used to find the different threshold value that minimises the intraclass variance of the black-and-white pixels thresholded [26]. At this stage, the green pixels were first masked, and then the threshold value was used. Mostly green pixels were masked in the following way: If the pixel value of the green component was less than the pre-calculated threshold value, the red, green, and blue components of this pixel were all set to zero.

D. Feature Extraction

Feature extraction is the process of converting the information gathered from the segmented regions into a set of features. Shape, colour, and texture are the global feature descriptors extracted from the input images. The extraction of features is crucial for many classification tasks, such as the identification of plant health status. This stage of leaf health status detection was used to extract features from the dataset used to categorise and identify healthy and infected plants. In the proposed method, the classifier employs the characteristics such as leaf perimeter, leaf area, and infected area obtained through the feature extraction process to train and test the dataset to classify the leaf health status. Each feature is discussed in the subsections as follows: Leaf perimeter and leaf area: The Gaussian and mean shift filters were applied to the resulting image. The reason for applying these filters was that the "centroids" in samples with smooth density could easily be identified. After applying the mean shift filter, canny edge detection was employed to calculate the perimeter of the leaf. Subsequently, the perimeter and area of the leaf were calculated. Infected area: The infected area was calculated from the infected region determined by mean shift clustering.

E. Health Status Classification of Tomato Leaves

Classification involves the discovery and categorisation of input data into different possible classes. At this stage, the leaves are categorised as healthy or infected. In the detection of health status and classification of leaves, it is vital to select the suitable classifiers based on the nature of the problem. In this study, sixteen sixteen machine learning algorithms were employed on the experimental dataset as the k-Nearest Neighbors Classifier (No. 1), Logistic Regression (No. 2), Decision Tree Classifier (No. 3), Extra Trees Classifier (No. 4), Random Forest Classifier (No. 5), Gradient Boosting Classifier (No. 6), Light Gradient Boosting Machine Classifier (No. 7), Support Vector Classifier (No. 8), Multi-layer Perceptron Classifier (No. 9), Gaussian Naïve Bayes (No. 10), Multinomial Naïve Bayes (No. 11), Bernoulli Naïve Bayes (No. 12), eXtreme Gradient Boosting Classifier (No. 13), AdaBoost Classifier (No. 14), Linear Discriminant Analysis (No. 15), Quadratic Discriminant Analysis (No. 16).

F. Embedded System

Developments of a computer vision algorithms and the study of machine learning approaches rely not only on techniques used, but also on modern parallel computing architectures that enable computations effectively. The hardware industry has begun to concentrate on embedded platforms, particularly portable systems with high precision and low latency. In addition to the commonly used personal computers, the research was conducted on hardware cards such as Jetson Nano and Jetson TX2, which are frequently used in the literature. The hardware portability of the study was demonstrated by comparing the results (operation time) of these cards. Jetson Development Kits' features are given in Table I.

TABLE I. JETSON DEVELOPMENT KITS FEATURES.

Hardware features	Jetson Nano	Jetson TX2
CPU	4-core ARM A57 @ 1.43 GHz	4-core ARM CortexA57 @ 2 GHz, 2-core Denver2 @ 2 GHz
GPU	128-core Maxwell @ 921 MHz	256-core Pascal @ 1.3 GHz
Memory	4 GB LPDDR4, 25.6 GB/s	8 GB 128-bit LPDDR4, 59.7 GB/s
Power	5 W or 10 W	7.5 W or 15 W

G. Steps of Proposed Algorithm

The proposed technique, outlined in Fig. 1, comprises four stages: dataset construction, pre-processing and segmentation, feature extraction, and classification. Each stage and its respective operations are explained below.

1. Load the input image.
2. Pre-processing of the image:

a) The image was resized to a smaller size (275×185 pixels) using bilinear interpolation to reduce computational complexity;

b) A Gaussian blur filter is applied to the resized image to reduce noise;

c) The colour space of the filtered image is converted from RGB to HSL.

3. The hue (H) component is extracted from the HSL image, as it contains the necessary information for segmentation.
4. Otsu thresholding is performed on the H component to determine the optimal threshold value that separates the leaf from the background.
5. Create a binary mask by setting all pixels with hue values greater than the Otsu threshold to 1 (indicating a leaf) and all other pixels to 0 (indicating a background).
6. Mean shift clustering is performed on the H component to further refine the segmentation. Update the binary mask by merging the results of mean shift clustering.
7. Morphological operations (opening and closing) are performed on the binary mask to remove any small holes or artefacts within the segmented leaf region.
8. The binary mask is applied to the original RGB image to extract the segmented leaf region.
9. The segmented leaf image is converted into the HSL colour space and the H component is extracted again.
10. Otsu thresholding is performed on the H component of the segmented leaf image to determine the optimal threshold value that separates the healthy and diseased areas of the leaf.
11. Create a binary mask for the diseased area by setting all pixels with hue values greater than the Otsu threshold to 1 (indicating diseased) and all other pixels to 0 (indicating healthy).
12. Mean shift clustering is performed on the H component of the segmented leaf image to further refine the segmentation between healthy and diseased areas.
13. The binary mask for the diseased area is updated by merging the results of mean shift clustering.
14. The diseased area binary mask is applied to the segmented leaf image to extract the diseased region of the leaf.
15. Calculation of the leaf perimeter:

a) Apply edge detection (canny edge detection) to the binary mask of the segmented leaf to determine the boundaries;

b) Calculate the total length of the detected edges, which represents the perimeter of the leaf.

16. Compute leaf area:

a) The total number of pixels with a value of 1 is counted in the binary mask of the segmented leaf;

b) Multiply the pixel count by the actual area represented by a single pixel (calculated in Step 2a);

c) The result of the multiplication is the area of the leaf.

17. Calculate the diseased area.

a) Count the total number of pixels with a value of one in the binary mask of the diseased region;

b) Multiply the pixel count by the actual area represented by a single pixel (calculated in step 2a);

c) The result of the multiplication is the diseased area.

18. Use features for processing in the sixteen machine learning classification algorithms on two embedded systems.

III. Experimental Results

This section presents the experimental results of our proposed method for the detection and classification of infected and healthy leaf images of tomato plants. The basic procedures of the proposed leaf classification system are given in Fig. 4.

The infected symptom areas of the images of tomato leaves were first segmented from the normal leaf and background, as described previously. The segmented images were then sent to machine learning algorithms for classification purposes. Based on the statistics of accurate detections (also referred to as true positives), misdetection (often referred to as false negatives), true negatives, and false positives, the performance of various techniques is assessed using metrics such as accuracy, recall, precision and F1-score. All the measures considered are common performance indicators in machine learning.

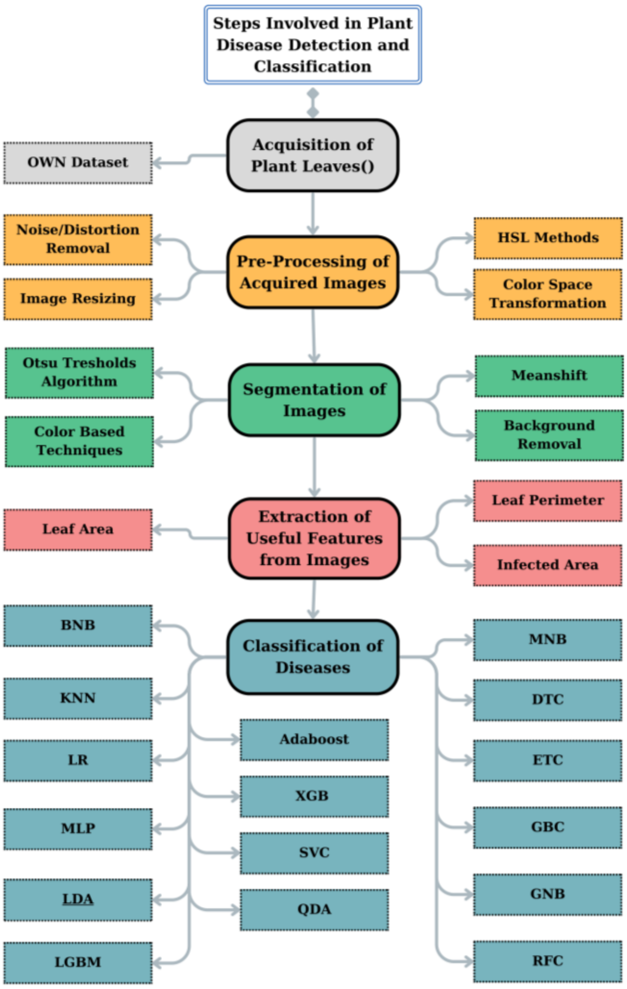


Fig. 4. Basic procedures of the proposed leaf classification system.

A. Evaluation Metrics

The evaluation of the proposed system has been done employing the system outcomes. A classification system returns four outcomes: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). Using these outcomes, the following measures are calculated.

Accuracy is a measure of correctness. It is defined as the number of samples correctly classified divided by the total number of samples [27]

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}. \tag{1}$$

Recall is computed as the number of true positives divided by all positives

$$Recall = \frac{TP}{TP+FN}. \tag{2}$$

Precision is computed as the number of true positives divided by the precited positives

$$Precision = \frac{TP}{TP+FP}. \tag{3}$$

F1-Score is a measure that combines precision and recall. It is calculated as the harmonic mean of the Precision and Recall

$$F1-Score = 2\times\frac{Precision\times Recall}{Precision+Recall}. \tag{4}$$

B. Hyperparameter Optimisation

Hyperparameters in machine learning are the parameters that cannot be modified during training. Therefore, optimal parameters must be established prior to training. Using trial-and-error methods to calculate these parameters is a very time-consuming process. Consequently, optimisation algorithms are used to find the optimal values for these parameters. There is more than one hyperparameter optimisation method in the literature. The most commonly used methods are grid search, random search, and evolutionary algorithms. In the study, the grid search method was used as a hyperparameter optimisation.

Grid search is the simplest and most widely used search algorithm for hyperparameter optimisation. It is advantageous in terms of easier parallelisation and flexible resource allocation. The search on the grid leads to the most accurate predictions as long as sufficient resources are given, and the user can always find the most suitable combination [28].

Some classification model hyperparameters have an infinite number of possible values. Consequently, it is necessary to establish a search interval for these parameters. The classification models were trained with values in the specified ranges and the best results obtained were used as the hyperparameters of the model [29]. The most significant drawback of the grid search method is its long computation time. To save time, the computations can be performed on a subset of the dataset. Through this way, the initial range of each parameter can be estimated [30].

The grid search method was applied to all classification algorithms employed, and trainings were conducted with the optimal parameters obtained. The complexity of the model can be balanced for overfitting and underfitting via hyperparameter optimisation. With the constraints imposed by hyperparameters, the overfitting problem caused by the flexibility of the models can be resolved. Table II displays the best parameters obtained by the grid search method for the algorithms used in this study.

TABLE II. BEST CLASSIFICATION MODEL PARAMETER VALUES BY GRID SEARCH.

Model	**Search space for hyperparameters**	**Best parameter values**
No. 1	'n_neighbors': np.arange(1, 50)	'n_neighbors': 3
No. 2	"C": np.logspace(-3, 3, 7), "penalty": ["l1", "l2"]	'C': 10.0, 'penalty': 'l2'
No. 3	'max_features': ['auto', 'sqrt', 'log2'], 'ccp_alpha': [0.1, 0.01, 0.001], 'max_depth': [5, 6, 7, 8, 9], 'criterion' :['gini', 'entropy'	'ccp_alpha': 0.01, 'criterion': 'entropy', 'max_depth': 8, 'max_features': 'auto'
No.4	'n_estimators': [1, 500], 'criterion': ['gini', 'entropy', 'log_loss'], 'max_depth': [4, 5, 6, 7, 8]	'criterion': 'entropy', 'max_depth': 8, 'n_estimators': 500
No. 5	'n_estimators': [200, 500], 'max_features': ['auto', 'sqrt', 'log2'], 'max_depth': [4, 5, 6, 7, 8]	'max_depth': 8, 'max_features': 'auto', 'n_estimators': 200
No. 6	'n_estimators': range(20, 81, 10)	'n_estimators': 20
No. 7	'n_estimators': [400, 700, 1000], 'colsample_bytree': [0.7, 0.8], 'max_depth': [15, 20, 25], 'num_leaves': [50, 100, 200], 'reg_alpha': [1.1, 1.2, 1.3], 'reg_lambda': [1.1, 1.2, 1.3], 'min_split_gain': [0.3, 0.4], 'subsample': [0.7, 0.8, 0.9], 'subsample_freq': [20]	'colsample_bytree': 0.7, 'max_depth': 15, 'min_split_gain': 0.3, 'n_estimators': 400, 'num_leaves': 50, 'reg_alpha': 1.1, 'reg_lambda': 1.1, 'subsample': 0.7, 'subsample_freq': 20
No. 8	'C': [0.1, 1, 10, 100, 1000], 'gamma': [1, 0.1, 0.01, 0.001, 0.0001], 'kernel': ['rbf']	'C': 1000, 'gamma': 0.0001, 'kernel': 'rbf'
No. 9	'solver': ['lbfgs'], 'max_iter': [1000, 1100, 1200, 1300, 1400, 1500, 1600, 1700, 1800, 1900, 2000], 'alpha': 10.0 ** -np.arange(1, 10), 'hidden_layer_sizes': np.arange(10, 15), 'random_state': [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]	'alpha': 1e-05, 'hidden_layer_sizes': 14, 'max_iter': 1000, 'random_state': 7, 'solver': 'lbfgs'
No. 10	'var_smoothing': np.logspace(0, -9, num = 100)	'var_smoothing': 1.2328467394420635e-09
No. 11	'alpha': [0.01, 0.1, 0.5, 1.0, 10.0]	'alpha': 0.01
No. 12	'alpha': [0.01, 0.1, 0.5, 1.0, 10.0]	'alpha': 0.01
No. 13	'n_estimators': [400, 700, 1000], 'colsample_bytree': [0.7, 0.8], 'max_depth': [15, 20, 25], 'reg_alpha': [1.1, 1.2, 1.3], 'reg_lambda': [1.1, 1.2, 1.3], 'subsample': [0.7, 0.8, 0.9]	'colsample_bytree': 0.7, 'max_depth': 15, 'n_estimators': 1000, 'reg_alpha': 1.1, 'reg_lambda': 1.1, 'subsample': 0.9
No. 14	'n_estimators': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 20, 30]	'n_estimators': 9
No. 15	'solver': 'svd', 'lsqr', 'eigen'	'solver': 'svd'
No. 16	'reg_param': [0.1, 0.2, 0.3, 0.4, 0.5]	'reg_param': 0.5

C. Performance Comparison of Classification Algorithms on Embedded Systems

In this work, the features obtained as described in Section II-D were used for training and testing the classification algorithms using the k-fold cross-validation method. The k-fold cross-validation divides the data to be tested into k subdata. k-1 layers are used for training and the last layer is used for testing. The cross-validation process is repeated k times, separating the different folds used in each evaluation. Each data point is necessarily included once in the test sequence and k-1 times in the training sequence. Thus, all the data to be classified is tested and a result is obtained about the entire dataset. In this study, k-fold cross-validation was applied to the dataset. The values for k were used as 5 and 10. The results obtained are illustrated in Figs. 5–8. The classification algorithms were also trained and tested with two different datasets, which were obtained by dividing the original dataset. In the first (DATASET-1), 80 % of the dataset was used for training, and the rest was used for testing. These two different sets were tested on embedded systems by using various classification algorithms. The results obtained with these datasets are given in Table III and Figs. 9–14. In the second (DATASET-2), 70 % of the dataset was used for training and the rest was used for testing. The results obtained with these datasets are given in Table IV and Figs. 15–20.

Table III and Table IV list a comparison of the accuracy of several machine learning models. As seen from Table III and Table IV and Figs. 9 and 10, comparing the results of the classifiers obtained on the Jetson TX2 and Jetson Nano platforms, it is evident that the performance of the machine learning algorithms can vary significantly between embedded systems. The Random Forest classifier consistently achieved the highest accuracy on both platforms, demonstrating its robustness and suitability for tomato leaf health classification. However, differences in training and testing times across classifiers highlight the importance of considering the trade-offs between accuracy and computational efficiency when deploying machine learning models on embedded systems.

TABLE III. RESULTS OF ALL CLASSIFIERS OBTAINED BY NANO.

Algorithms	**Accuracy**	**Train time (s)**	**Test time (s)**
RFC	**0.997207**	1.51395	0.11995
GBC	0.994413	0.12806	0.01018
DTC	0.988827	0.01704	0.00904
LGBM	0.988827	1.83732	0.01032
BNB	0.988827	0.01367	0.00875
XGB	0.988827	7.48146	0.02182
AdaBoost	0.988827	0.09281	0.01878
ETC	0.977654	2.75736	0.29196
LR	0.938547	0.03125	0.00833
MLP	0.927374	6.53921	0.01129
SVC	0.916201	2.47568	0.04011
QDA	0.891061	0.01152	0.00936
GNB	0.868715	0.01214	0.00789
KNN	0.837989	0.01318	0.06802
MNB	0.762570	0.01279	0.00838
LDA	0.745810	0.01465	0.00767

Some classifiers, such as Gradient Boosting Classifier and Decision Tree Classifier, displayed faster training times on the Jetson Nano compared to the Jetson TX2, despite the former being a less powerful platform. This finding underscores the potential of optimising algorithm implementations to achieve better performance on resource-constrained platforms. Ultimately, these comparisons provide valuable information for researchers and practitioners seeking to develop and deploy accurate and efficient plant health classification systems for embedded devices. The confusion matrix and the classification report of the Random Forest classifier on Jetson Nano and Jetson TX2 are given in Figs. 11 and 12. As seen from the figures, the precision, recall, and F1-scores have achieved high values.

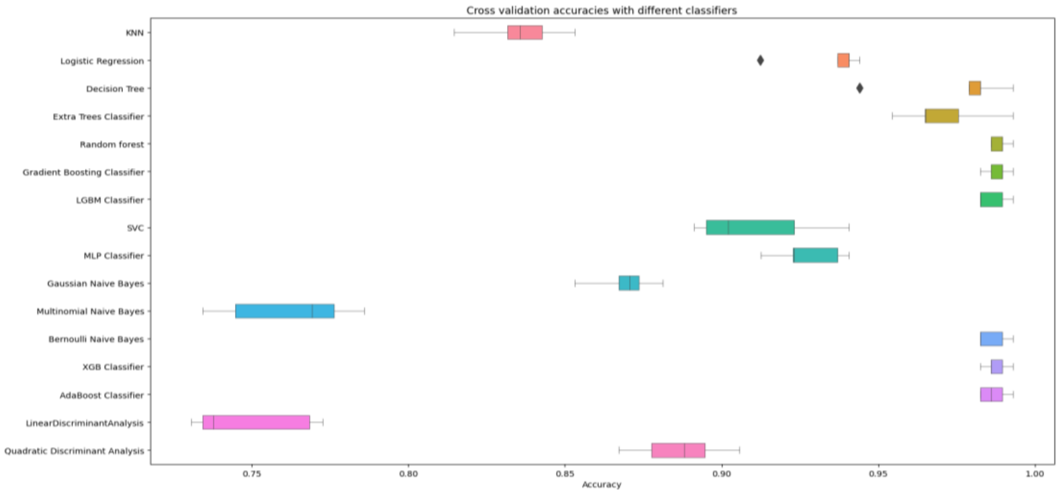


Fig. 5. Box and whisker plot of the cross-validation result for k = 5 on the Nano.

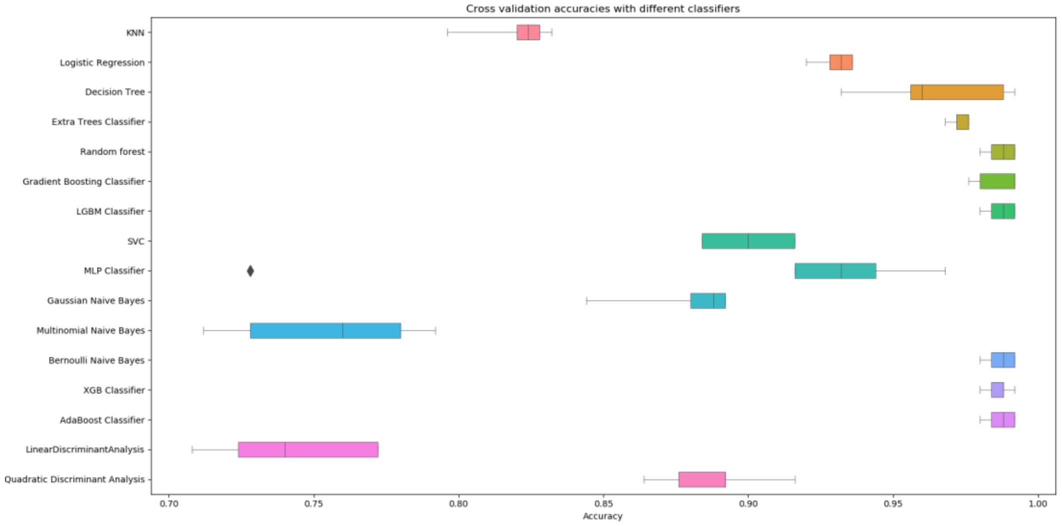


Fig. 6. Box and whisker plot of the cross-validation result for k = 5 on the TX2.

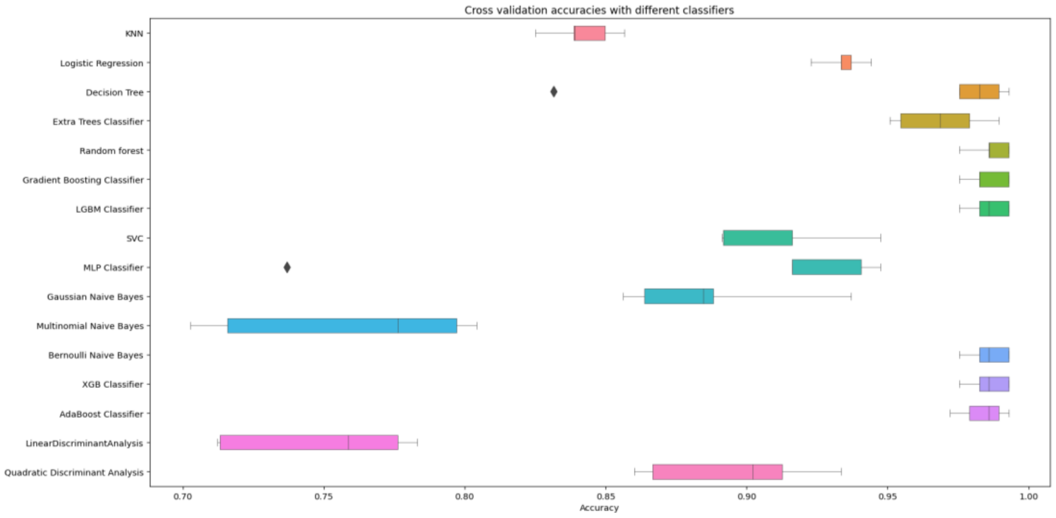


Fig. 7. Box and whisker plot of the cross-validation result for k = 10 on the Nano.

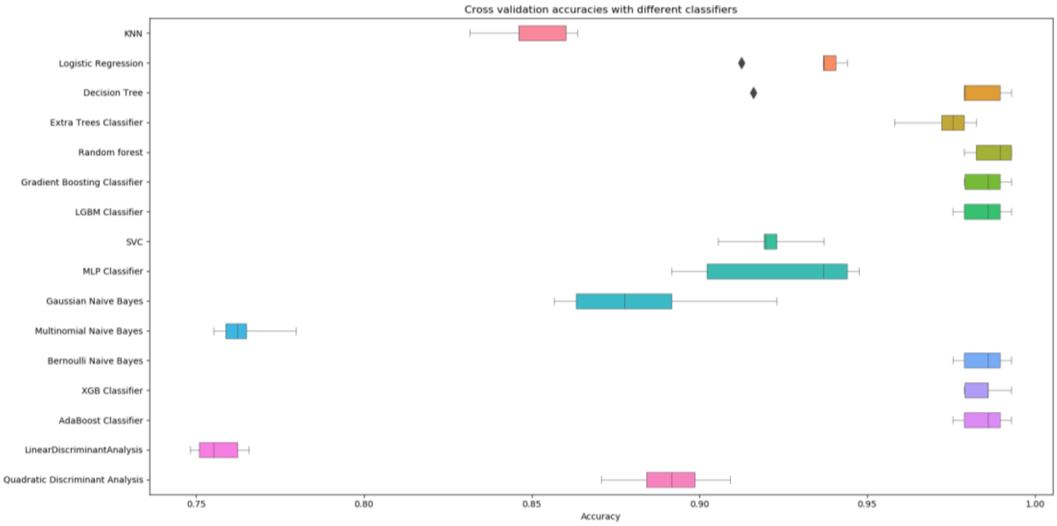


Fig. 8. Box and whisker plot of the cross-validation result for k = 10 on the TX2.

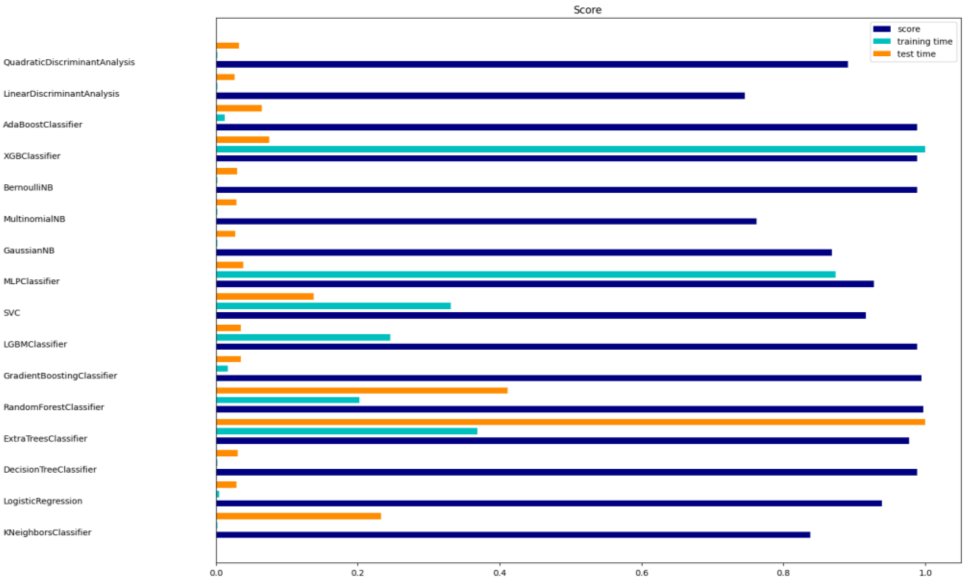


Fig. 9. Accuracy, train time, and test time of all classifiers obtained by Nano using DATASET-1.

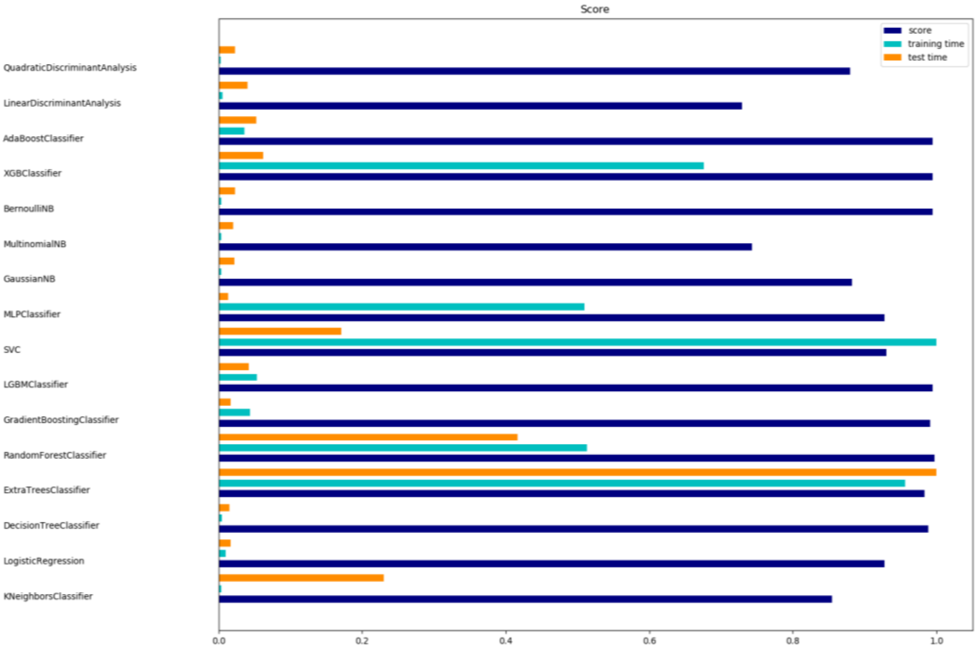
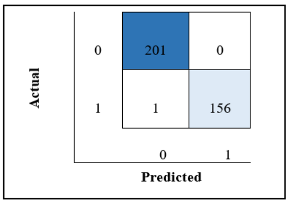


Fig. 10. Accuracy, train time, and test time of all classifiers obtained by TX2 using DATASET-1.

TABLE IV. RESULTS OF ALL CLASSIFIERS OBTAINED BY TX2.

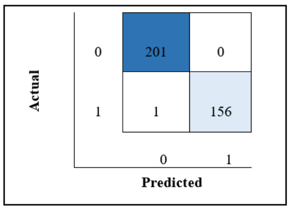
Algorithms	**Accuracy**	**Train time (s)**	**Test time (s)**
RFC	**0.997207**	1.00662	0.08148
LGBM	0.994413	0.10397	0.00823
BNB	0.994413	0.00743	0.00439
XGB	0.994413	1.32664	0.01215
AdaBoost	0.994413	0.07110	0.01028
GBC	0.991620	0.08622	0.00322
DTC	0.988827	0.00934	0.00292
ETC	0.983240	1.87615	0.19575
SVC	0.930168	1.96245	0.03339
LR	0.927374	0.01922	0.00324
MLP	0.927374	1.00008	0.00265
GNB	0.882682	0.00679	0.00425
QDA	0.879888	0.00624	0.00451
KNN	0.854749	0.00758	0.04512
MNB	0.743017	0.00704	0.00404
LDA	0.729050	0.00997	0.00782



```
              precision    recall    f1-score    support
           0       1.00      1.00        1.00        201
           1       1.00      0.99        1.00        157

    Accuracy                             1.00        358
   Macro avg       1.00      1.00        1.00        358
Weighted avg       1.00      1.00        1.00        358
```

Fig. 11. Confusion matrix and classification report of RFC on Nano using DATASET-1.



```
              precision    recall    f1-score    support
           0       1.00      1.00        1.00        201
           1       1.00      0.99        1.00        157

    Accuracy                             1.00        358
   Macro avg       1.00      1.00        1.00        358
Weighted avg       1.00      1.00        1.00        358
```

Fig. 12. Confusion matrix and classification report of RFC on TX2 using DATASET-1.

Figures 13 and 14 depict the plot of the multiclass Receiver Operating Characteristic (ROC) curve. A ROC curve is a graphical representation of the diagnostic capacity of a binary classifier system as its discrimination threshold changes. The ROC curve is derived by plotting the True Positive Rate (TPR) versus the False Positive Rate (FPR) at various threshold settings. The x and y coordinates represent true positives and false positives, respectively, and define the ROC space. The curves (Figs. 13 and 14) demonstrate the diagnostic ability of various classifiers on tomato health status at the discriminating threshold [31]. A perfect classifier should have a true positive rate of 1 and a false positive rate of 0. On the basis of the ROC curve, the area under the ROC curve (AUC) can be calculated to characterise the performance of a classification model. When Figs. 13 and 14 are examined, it is seen that the AUC of the Random Forest classifier model is greater than the other classifiers. It gets a 1.00 AUC accuracy rate using both Jetson Nano and Jetson TX2.

When comparing the results in Tables V and VI, which represent the performance of various classifiers on the Jetson Nano and Jetson TX2 platforms with a 70 %–30 % data split, several noteworthy points can be observed.

As shown in Table V and Fig. 15, the Random Forest classifier classified the health status of tomato leaves with the highest accuracy of 99.81 %. Table V and Fig. 15 reveal that Bernoulli Naïve Bayes is superior to other algorithms with respect to training and testing times. Table VI and Fig. 16 show that Bernoulli Naïve Bayes is better than other algorithms when training and testing times are taken into account and highest level of accuracy, at approximately 98.88 %.

The classifier accuracy rankings were consistent between the two platforms. For example, the best performing classifiers, such as RFC, GBC, LGBM, and BNB, maintain high rankings on both Jetson Nano and Jetson TX2. This similarity indicates that these classifiers are robust and perform well on different hardware platforms.

There were noticeable variations in training and testing times between the Jetson Nano and Jetson TX2 platforms. For example, classifiers such as XGB and MLP have considerably shorter training times on Jetson TX2 than on Jetson Nano. This can be attributed to the differences in hardware specifications and computational capabilities between the two platforms, with Jetson TX2 being more powerful than the Jetson Nano.

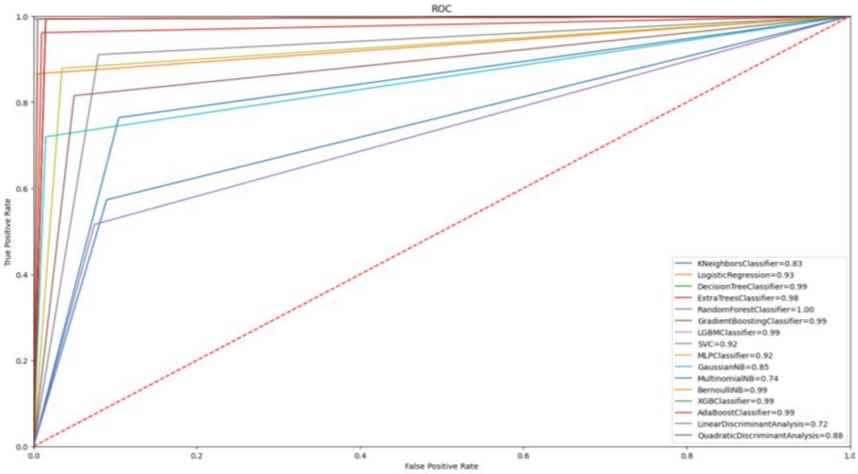


Fig. 13. ROC curve of all classifiers obtained by Nano using DATASET-1.

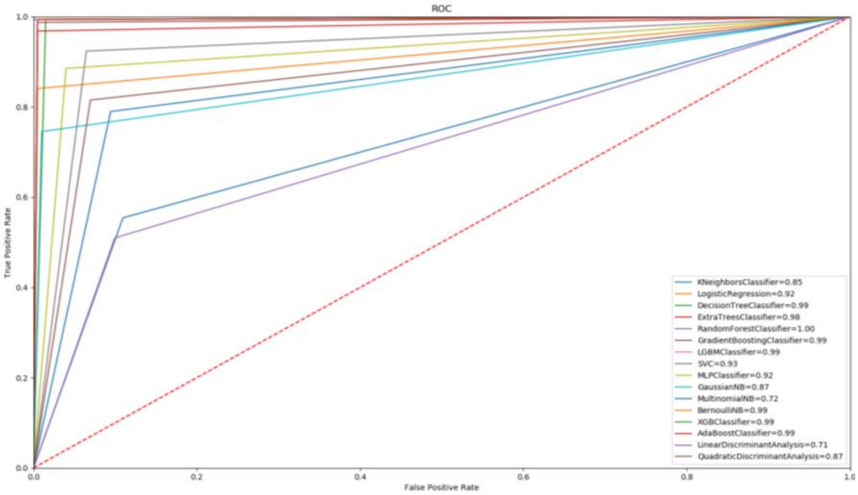


Fig. 14. ROC curve of all classifiers obtained by TX2 using DATASET-1.

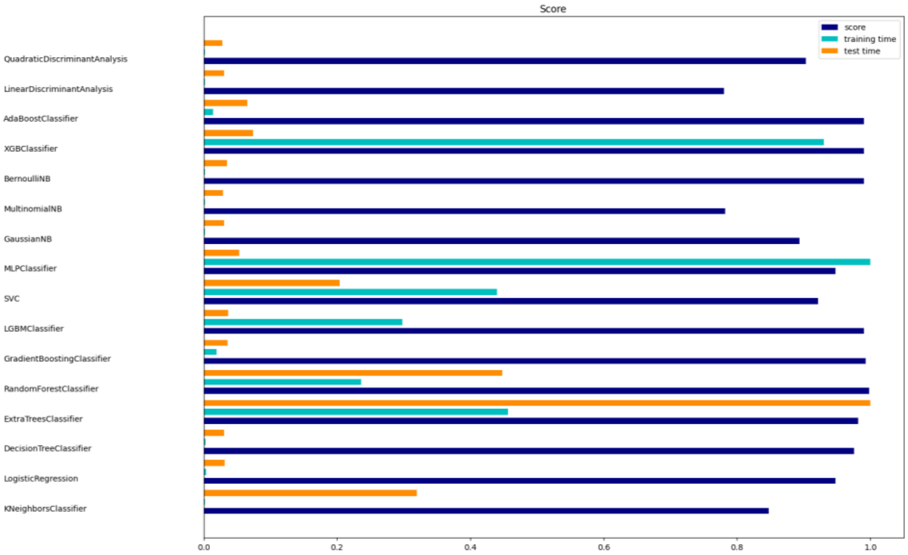


Fig. 15. Accuracy, train time, and test time of all classifiers obtained by Nano using DATASET-2.

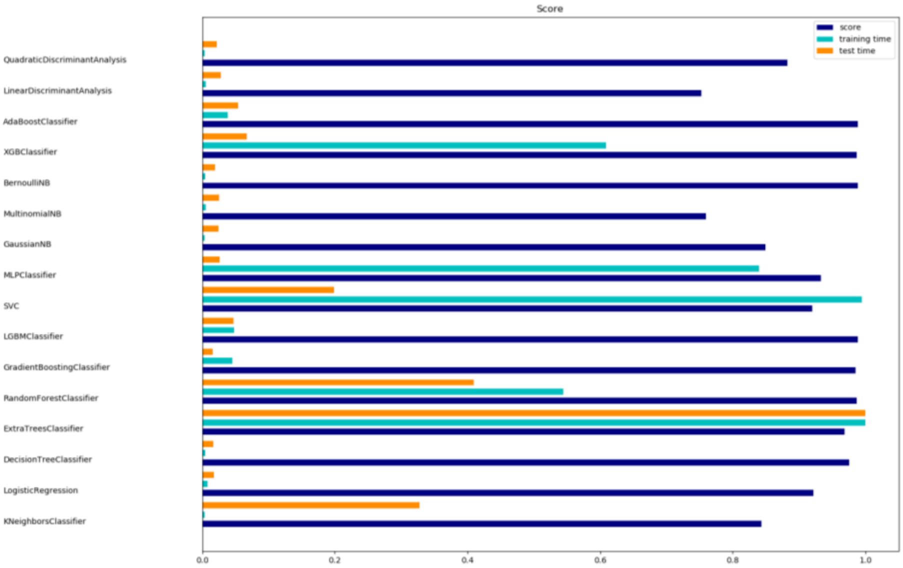


Fig. 16. Accuracy, train time, and test time of all classifiers obtained by TX2 using DATASET-2.

Some classifiers exhibit platform-specific performance variations. For instance, the accuracy of the BNB classifier is slightly higher on the Jetson TX2 compared to the Jetson Nano, while the accuracy of the ETC classifier is slightly lower on the Jetson TX2. These differences may be due to platform-specific optimisations or variations in the manner in which classifiers handle hardware resources.

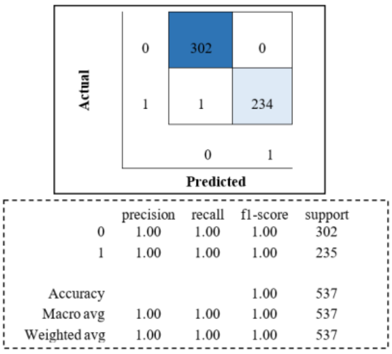
Despite the differences in training and testing times between the Jetson Nano and the Jetson TX2, the overall performance trends remained similar. Classifiers that perform well on one platform generally perform well on the other, which suggests that the choice of classifier is more critical than the choice of hardware for this particular classification problem.

TABLE V. RESULTS OF ALL CLASSIFIERS OBTAINED BY NANO.

Algorithms	Accuracy	Train time (s)	Test time (s)
RFC	**0.998138**	1.46166	0.12521
GBC	0.992551	0.11960	0.01003
LGBM	0.990689	1.84280	0.01032
BNB	0.990689	0.01323	0.00983
XGB	0.990689	5.75926	0.02077
AdaBoost	0.990689	0.08865	0.01824
ETC	0.981378	2.82431	0.27969
DTC	0.975791	0.01643	0.00866
LR	0.947858	0.02485	0.00878
MLP	0.947858	6.18953	0.01482
SVC	0.921788	2.72523	0.05703
QDA	0.903166	0.01175	0.00776
GNB	0.893855	0.01207	0.00862
KNN	0.8473	0.01292	0.08945
MNB	0.782123	0.01271	0.00818
LDA	0.780261	0.01467	0.00858

TABLE VI. RESULTS OF ALL CLASSIFIERS OBTAINED BY TX2.

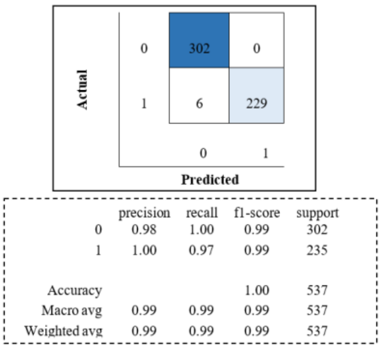
Algorithms	Accuracy	Train time (s)	Test time (s)
BNB	**0.988827**	0.008196	0.003992
LGBM	0.988827	0.089010	0.009876
AdaBoost	0.988827	0.070751	0.011156
XGB	0.986965	1.116255	0.013910
RFC	0.986965	0.998648	0.084946
GBC	0.985102	0.082891	0.003294
DTC	0.975791	0.008404	0.003521
ETC	0.968343	1.833711	0.207395
MLP	0.932961	1.540555	0.005402
LR	0.921788	0.014691	0.003696
SVC	0.919926	1.823378	0.041209
QDA	0.882682	0.006061	0.004485
GNB	0.849162	0.006788	0.005023
KNN	0.843575	0.006053	0.068001
MNB	0.759777	0.009299	0.005317
LDA	0.752328	0.009828	0.005898



```
              precision   recall   f1-score   support
        0       1.00       1.00      1.00       302
        1       1.00       1.00      1.00       235

   Accuracy                          1.00       537
  Macro avg     1.00       1.00      1.00       537
Weighted avg    1.00       1.00      1.00       537
```

Fig. 17. Confusion matrix and classification report of RFC on Nano using DATASET-2.



```
              precision   recall   f1-score   support
        0       0.98       1.00      0.99       302
        1       1.00       0.97      0.99       235

   Accuracy                          1.00       537
  Macro avg     0.99       0.99      0.99       537
Weighted avg    0.99       0.99      0.99       537
```

Fig. 18. Confusion matrix and classification report of BNB on TX2 using DATASET-2.

Random Forest and Bernoulli Naïve Bayes classifiers were executed on Jetson Nano and Jetson TX2. The ROC curve for each model is given in Figs. 19 and 20.

Analysis of the performance and generalisation capabilities of the best performing classifiers on both the Jetson Nano and Jetson TX2 platforms reveals that they are particularly effective at handling the tomato leaf health classification problem. The classifiers showed remarkable stability across both data splits and platforms, maintaining high accuracy rates with minimal variations. The increased size of the test sets suggests that the classifiers effectively learnt the underlying patterns in the data and can generalise well to unseen images. This suggests that they may be well suited for this classification task, particularly when computational resources are limited.

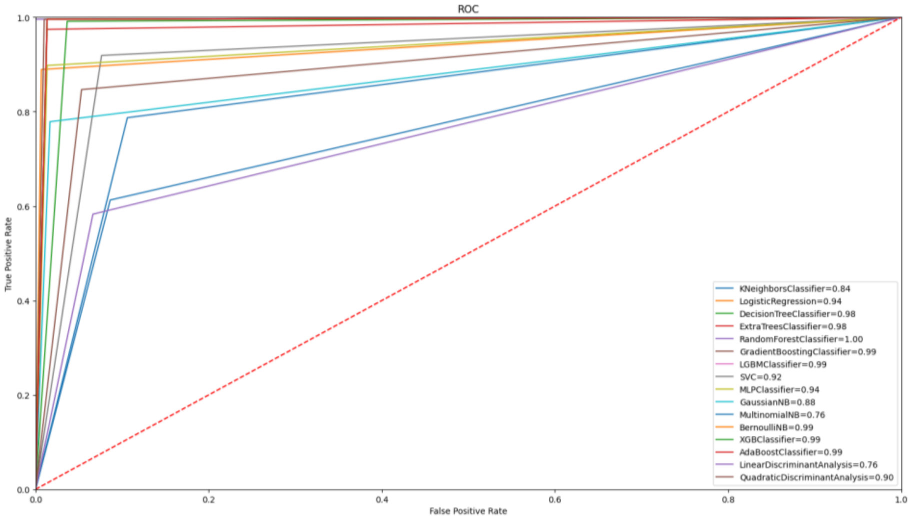


Fig. 19. ROC curve of all classifiers obtained by Nano using DATASET-2.

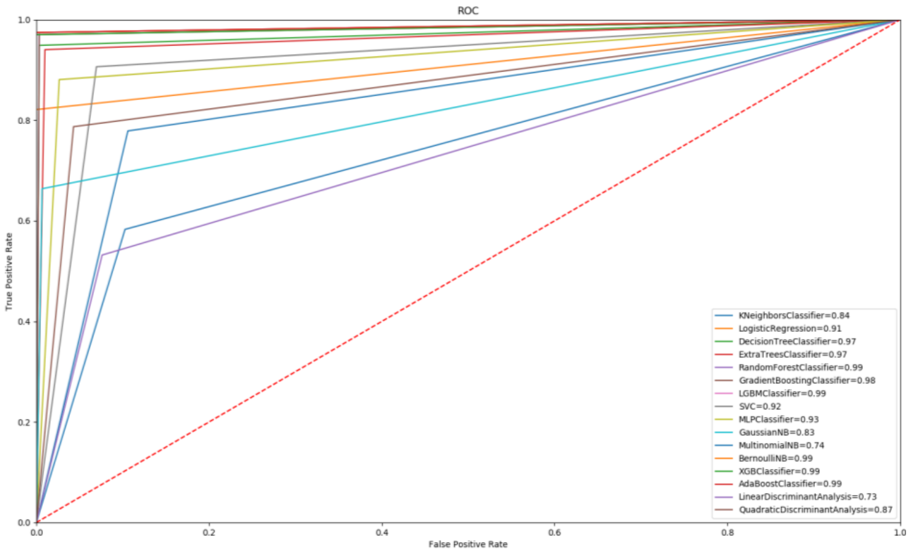


Fig. 20. ROC curve of all classifiers obtained by TX2 using DATASET-2.

Since the datasets used in different published research are so dissimilar, it is difficult to compare the performance of different prediction models. However, we have highlighted some of the most recent research in Table VII to conduct a comparative analysis with our own findings.

TABLE VII. RESULTS OF DIFFERENT STUDIES.

No.	Method	Dataset	Performance (accuracy)
1	Multi class SVM [32]	Detection and classification of diseases of grape plant	96.60 %
2	SVM [33]	Identification of unhealthy tomato leaves (800 images)	99.83 %
3	k-NN [34]	Plant disease detection (40 images x3)	97.00 %
4	SVM [35]	1120 images of tomato leaves collected from Plant Village Dataset	22.00 %
	Random Forest classifier [35]		95.20 %
5	Random Forest classifier [36]	320 images with 80 images of each class (Apple Fruit)	80.00 %
6	ResNet [37]	Citrus leaf diseases	95.83 %
	AlexNet [37]		97.92 %
7	SE-ResNet50 [38]	Plant Village dataset (tomato)	96.81 %
8	YOLO-Dense [39]	Self-made tomato anomaly dataset	96.41 %
9	IBSA_Net [40]	Plant Village dataset Data augmentation (tomato)	94.60 %
10	Ensemble fine-tuned CNN optimised with PSO [31]	Plant Village dataset (tomato)	99.60 %
11	**Random Forest classifier**	**Constructed dataset with this study**	**99.81 %**
	Bernoulli Naïve Bayes		**98.88 %**

IV. DISCUSSION

This study presented an advanced diagnostic system capable of accurately determining the health status of the leaves of tomato plants. Using various machine learning classifiers, the system was executed on Jetson Nano and Jetson TX2 embedded systems to achieve precise identification. The system's performance, including accuracy, F1-Score, and ROC accuracy rates, was assessed and compared with other studies in the literature. Additionally, the testing and training times of the machine learning algorithms on different embedded systems were analyzed and presented comparatively.

The experimental results demonstrated that the proposed model is highly effective in identifying the health status of tomato leaves. The superior performance of the Random Forest classifier showcases its potential for real-world applications. However, there are several avenues for improvement and future research:

– Enriching the dataset: To enhance the system's applicability, future studies can expand the dataset by incorporating a wider variety of diseases and crops. By including more images, the network can better identify and classify a broader range of plant diseases and species.

– Smartphone integration: With the growing ubiquity and improved quality of smartphone cameras, developing a system for accurate diagnosis using mobile devices could provide a more accessible and user-friendly experience for farmers and agricultural professionals.

– Leveraging alternative data sources: Training models on additional data sources, such as panoramic land area views, aerial photographs, and images of various disease stages, could potentially boost the system's performance and enable more in-depth analysis.

– Edge computing and IoT integration: Integrating the proposed system with edge computing devices and Internet of Things (IoT) technologies could lead to more efficient and real-time monitoring of crop health. This would enable farmers and agricultural professionals to make data-driven decisions about crop management, disease prevention, and resource allocation, ultimately increasing the overall efficiency and sustainability of agricultural practises.

– Investigating ensemble learning and transfer learning techniques: Future research could explore the use of ensemble learning methods, which combine multiple machine learning models, to improve the overall performance and reliability of the system. Transfer learning techniques could also be employed to leverage pre-trained models and adapt them to the specific task of plant disease detection and classification. These approaches may lead to improved system performance and a deeper understanding of the complex patterns associated with plant diseases.

V. CONCLUSIONS

Tomato plants are a crucial food crop, and diseases that affect them can lead to significant economic losses in agriculture. This study presented a prototype for detecting the health status of tomato plant leaves. A dataset was created using 1,778 images of healthy and diseased tomato leaves, collected from tomato planting areas in the Turkish provinces of Samsun and Mersin. Images were divided into training and testing datasets, and pre-processing, segmentation, feature extraction, and classification were performed on each set.

Sixteen machine learning algorithms were used for classification, including Logistic Regression, Linear Discriminant Analysis, Decision Tree Classifier, Multi-Layer Perceptron Classifier, Multinomial Naïve Bayes, Gaussian Naïve Bayes, Gradient Boosting Classifier, Quadratic Discriminant Analysis, Bernoulli Naïve Bayes, AdaBoost Classifier, Light Gradient Boosting Machine Classifier, eXtreme Gradient Boosting Classifier, Support Vector Classifier, K-Nearest Neighbors Classifier, Random Forest Classifier, and Extra Trees Classifier. These classifiers were executed on Jetson Nano and Jetson TX2 platforms. The experimental results showed that the Random Forest Classifier outperformed other algorithms, achieving approximately 99 % accuracy in classifying the health status of tomato leaves.

The proposed system provides a faster and more objective method to detect plant health, assisting farmers in making informed decisions about the use of pesticides, reducing costs and minimising environmental impact. Future research will focus on recognising plant diseases, developing new hybrid algorithms such as neural networks, and evaluating the performance of these hybrid algorithms in the detection and classification of plant deseases.

ACKNOWLEDGMENT

A significant part of this article includes data from the Doctorate thesis data of Hasan ULUTAŞ.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] D. Al Bashish, M. Braik, and S. Bani-Ahmad, "Detection and classification of leaf diseases using K-means-based segmentation and neural-networks-based classification", *Information Technology Journal*, vol. 10, no. 2, pp. 267–275, 2011. DOI: 10.3923/itj.2011.267.275.

[2] P. Nijalingappa and V. J. Madhumathi, "Plant identification system using its leaf features", in *Proc. of 2015 International Conference on Applied and Theoretical Computing and Communication Technology (ICATccT)*, 2015. DOI: 10.1109/icatcct.2015.7456906.

[3] K. Okarma *et al.*, "A Visual Feedback for Water-Flow Monitoring in Recirculating Aquaculture Systems", *Applied Sciences*, vol. 12, no. 20, p. 10598, 2022. DOI: 10.3390/app122010598.

[4] C. Uluturk and A. Ugur, "Recognition of leaves based on morphological features derived from two half-regions", in *Proc. of 2012 International Symposium on Innovations in Intelligent Systems and Applications*, 2012. DOI: 10.1109/inista.2012.6247030.

[5] J. Liu and X. Wang, "Plant diseases and pests detection based on deep learning: A review", *Plant Methods*, vol. 17, art. no. 22, 2021. DOI: 10.1186/s13007-021-00722-9.

[6] J. A. Wani, S. Sharma, M. Muzamil, S. Ahmed, S. Sharma, and S. Singh, "Machine learning and deep learning based computational techniques in automatic agricultural diseases detection: Methodologies, applications, and challenges. archives of computational methods in engineering", vol. 29, no. 1, pp. 641–677, 2021. DOI: 10.1007/s11831-021-09588-5.

[7] H. Wang, G. Li, Z. Ma, and X. Li, "Image recognition of plant diseases based on backpropagation networks", in *Proc. of 2012 5th International Congress on Image and Signal Processing*, 2012. DOI: 10.1109/cisp.2012.6469998.

[8] V. Singh and A. Misra, "Detection of plant leaf diseases using image segmentation and soft computing techniques", *Information Processing in Agriculture*, vol. 4, no. 1, pp. 41–49, 2017. DOI: 10.1016/j.inpa.2016.10.005.

[9] S. Arivazhagan, R. N. Shebiah, S. Ananthi, and S. V. Varthini, "Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features", *Agricultural Engineering International: CIGR Journal*, vol. 15, no. 1, pp. 211–217, 2013.

[10] S. Bashir, "Remote area plant disease detection using image processing", *IOSR Journal of Electronics and Communication Engineering*, vol. 2, no. 6, pp. 31–34, 2012. DOI: 10.9790/2834-0263134.

[11] T. Munisami, M. Ramsurn, S. Kishnah, and S. Pudaruth, "Plant leaf recognition using shape features and colour histogram with K-nearest neighbour classifiers", *Procedia Computer Science*, vol. 58, pp. 740–747, 2015. DOI: 10.1016/j.procs.2015.08.095.

[12] V. Nguyen Thanh Le, B. Apopei, and K. Alameh, "Effective plant discrimination based on the combination of local binary pattern operators and multiclass support vector machine methods", *Information Processing in Agriculture*, vol. 6, no. 1, pp. 116–131, 2019. DOI: 10.1016/j.inpa.2018.08.002.

[13] Y. Herdiyeni and M. M. Santoni, "Combination of morphological, local binary pattern variance and color moments features for indonesian medicinal plants identification", in *Proc. of 2012 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 2012, pp. 255–259. DOI: 10.1109/ICACSIS.2013.6761601.

[14] X. Wang, J. Liang, and F. Guo, "Feature extraction algorithm based on dual-scale decomposition and local binary descriptors for plant leaf recognition", *Digital Signal Processing*, vol. 34, pp. 101–107, 2014. DOI: 10.1016/j.dsp.2014.08.005.

[15] E. Elhariri, N. El-Bendary, and A. E. Hassanien, "Plant classification system based on leaf features", in *Proc. of 2014 9th International Conference on Computer Engineering & Systems (ICCES)*, 2014. DOI: 10.1109/icces.2014.7030971.

[16] K. B. Lee, K. W. Chung, and K. S. Hong, "An implementation of leaf recognition system based on leaf contour and centroid for plant classification", *Lecture Notes in Electrical Engineering*, vol. 109–116, 2012. DOI: 10.1007/978-94-007-5857-5_12.

[17] K. Mahdikhanlou and H. Ebrahimnezhad, "Plant leaf classification using centroid distance and axis of least inertia method", in *Proc. of 2014 22nd Iranian Conference on Electrical Engineering (ICEE)*, 2014, pp. 1690–1694. DOI: 10.1109/IranianCEE.2014.6999810.

[18] M. M. Ali, N. A. Bachik, N. A. Muhadi, T. N. Tuan Yusof, and C. Gomes, "Non-destructive techniques of detecting plant diseases: A review", *Physiological and Molecular Plant Pathology*, vol. 108, art. 101426, 2019. DOI: 10.1016/j.pmpp.2019.101426.

[19] Z. Iqbal, M. A. Khan, M. Sharif, J. H. Shahur, M. H. Rehman, and K. Javed, "An automated detection and classification of citrus plant diseases using image processing techniques: A review", *Computers and Electronics in Agriculture*, vol. 153, pp. 12–32, 2018. DOI: 10.1016/j.compag.2018.07.032.

[20] K. Gibert, M. Sànchez-Marrè, and J. Izquierdo, "A survey on pre-processing techniques: Relevant issues in the context of environmental data mining", *AI Communications*, vol. 29, no. 6, pp. 627–663, 2016. DOI: 10.3233/aic-160710.

[21] M. E. Sahin, "Image processing and machine learning-based bone fracture detection and classification using X-ray images", *Int. J. Imaging Syst. Technol.*, pp. 1–13, 2023. DOI: 10.1002/ima.22849.

[22] Q. Yao, Z. Guan, Y. Zhou, J. Tang, Y. Hu, and B. Yang, "Application of support vector machine for detecting rice diseases using shape and color texture features", in *Proc. of 2009 International Conference on Engineering Computation*, 2009. DOI: 10.1109/icec.2009.73.

[23] H. Al Hiary, S. Bani Ahmad, M. Reyalat, M. Braik, and Z. ALRahamneh, "Fast and accurate detection and classification of plant diseases", *International Journal of Computer Applications*, vol. 17, no. 1, pp. 31–38, 2011. DOI: 10.5120/2183-2754.

[24] S. W. Zhang, Y. J. Shang, and L. Wang, "Plant disease recognition based on plant leaf image", *J. Anim. Plant Sci.*, vol. 25, no. 3, pp. 42–45, 2015.

[25] A. Alorf, "K-means, mean shift, and SLIC clustering algorithms: A comparison of performance in color-based skin segmentation", M.S. thesis, University of Pittsburgh, 2017.

[26] N. Otsu, "A threshold selection method from gray-level histograms", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979. DOI: 10.1109/tsmc.1979.4310076.

[27] M. Emin Sahin, "Deep learning-based approach for detecting COVID-19 in chest X-rays", *Biomedical Signal Processing and Control*, vol. 78, art. 103977, 2022. DOI: 10.1016/j.bspc.2022.103977.

[28] R. Valarmathi and T. Sheela, "Heart disease prediction using hyper parameter optimization (HPO) tuning", *Biomedical Signal Processing and Control*, vol. 70, art. 103033, 2021. DOI: 10.1016/j.bspc.2021.103033.

[29] R. S. Arslan, "FG-droid: Grouping based feature size reduction for android malware detection", *PeerJ Computer Science*, vol. 8, p. e1043, 2022. DOI: 10.7717/peerj-cs.1043.

[30] R. S. Arslan, H. Ulutaş, A. S. Köksal, M. Bakır, and B. Çiftçi, "Automated sleep scoring system using multi-channel data and machine learning", *Computers in Biology and Medicine*, vol. 146, art. 105653, 2022. DOI: 10.1016/j.compbiomed.2022.105653.

[31] H. Ulutaş and V. Aslantaş, "Design of efficient methods for the detection of tomato leaf disease utilizing proposed ensemble CNN model, *Electronics*, vol. 12, no. 4, p. 827, 2023. DOI: 0.3390/electronics12040827.

[32] H. Waghmare, R. Kokare, and Y. Dandawate, "Detection and classification of diseases of grape plant using opposite colour Local Binary Pattern feature and machine learning for automated Decision Support System", in *Proc. of 2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN)*, 2016. DOI: 10.1109/spin.2016.7566749.

[33] U. Mokhtar, N. El Bendary, A. Hassenian, E. Emary, M. A. Mahmoud, H. Hefny, and M. F. Tolba, "SVM-based detection of tomato leaves diseases", *Advances in Intelligent Systems and Computing*, pp. 641–652, 2015. DOI: 10.1007/978-3-319-11310-4_55.

[34] J. Singh and H. Kaur, "Plant disease detection based on region-based segmentation and KNN classifier", in *Proc. of the International Conference on ISMAC in Computational Vision and Bio-Engineering 2018 (ISMAC-CVB)*, pp. 1667–1675, 2019. DOI: 10.1007/978-3-030-00665-5_154.

[35] M. Govardhan and V. M B, "Diagnosis of tomato plant diseases using Random Forest", in *Proc. of 2019 Global Conference for Advancement in Technology (GCAT)*, 2019. DOI: 10.1109/gcat47503.2019.8978431.

[36] B. J. Samajpati and S. D. Degadwala, "Hybrid approach for apple fruit diseases detection and classification using random forest classifier", in *Proc. of 2016 International Conference on Communication and Signal Processing (ICCSP)*, 2016. DOI: 10.1109/iccsp.2016.7754302.

[37] A. R. Luaibi, T. M. Salman, and A. H. Miry, "Detection of citrus leaf diseases using a deep learning technique", *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 2, p. 1719, 2021. DOI: 10.11591/ijece.v11i2.pp1719-1727.

[38] S. Zhao, Y. Peng, J. Liu, and S. Wu, "Tomato leaf disease diagnosis based on improved convolution neural network by attention module", *Agriculture*, vol. 11, no. 7, p. 651, 2021. DOI: 10.3390/agriculture11070651.

[39] X. Wang and J. Liu, "Tomato anomalies detection in Greenhouse Scenarios based on YOLO-Dense", *Frontiers in Plant Science*, vol. 12, 2021. DOI: 10.3389/fpls.2021.634103.

[40] R. Zhang, Y. Wang, P. Jiang, J. Peng, and H. Chen, "IBSA_Net: A network for tomato leaf disease identification based on transfer learning with small samples", *Applied Sciences*, vol. 13, no. 7, p. 4348, 2023. DOI: 10.3390/app13074348.

This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution 4.0 (CC BY 4.0) license (http://creativecommons.org/licenses/by/4.0/).