ITC 1/53

**Information Technology and Control**
**Vol. 53 / No. 1 / 2024**
**pp. 19-36**
**DOI 10.5755/j01.itc.53.1.34345**

**Classification of Medicinal Plant Leaves for Types and Diseases with Hybrid Deep Learning Methods**

Received 2023/06/09 | Accepted after revision 2024/01/17

**HOW TO CITE:** Kayaalp, K. (2024). Classification of Medicinal Plant Leaves for Types and Diseases with Hybrid Deep Learning Methods. *Information Technology and Control*, 53(1), 19-36. https://doi.org/10.5755/j01.itc.53.1.34345

# Classification of Medicinal Plant Leaves for Types and Diseases with Hybrid Deep Learning Methods

**Kıyas Kayaalp**

Department of Computer Engineering, Faculty of Technology, Isparta University of Applied Sciences, Isparta, Turkey; e-mail: kiyaskayaalp@isparta.edu.tr

**Corresponding author:** kiyaskayaalp@isparta.edu.tr

Leaf images are often used to detect plant diseases because most disease symptoms appear on the leaves. Analyzes performed by experts in the laboratory environment are expensive and time consuming. Therefore, there is a need for automated plant disease detection systems that are both economical and can help diagnose early symptoms more accurately. In this study, a deep learning-based methodology is presented for the classification of leaf diseases of plants, which are very similar in color, texture, vein and shape and cannot be noticed by non-experts, which are important for traditional medicine and pharmaceutical industry. In the model development process, 7 pre-learning deep learning algorithms and an image data set created from plant leaves in ten categories were preferred. The proposed model classifies the plant type and diseased condition in the dataset. In the first step of training the model, different learning rates were tested with optimum hyperparameters. In the second part, a test accuracy rate of 98.69% was achieved with the DenseNet121 model, with increased data. At the last stage, after the edge detection processes, the test accuracy value of 67.92% was reached with the DenseNet 121 model.

KEYWORDS: CNN algorithms, edge detection, image classification, medicinal plant leaves, Multi-class detection.

## 1. Introduction

Plants are not only a source of oxygen for living things, but also a source of food. The fruits, flowers, grains, stems, and leaves of plants can be used in different areas such as food, medicine, and perfumery. In plants, leaves are the first messenger of plant diseases [2]. Plant diseases and pests can cause agricultural and ecological losses. For this reason, early detection of plant diseases is important as that of other living things. Traditionally, farmers and plant pathologists use their eyes to detect diseases and make decisions based on their experience. This is not true as many types of diseases can look the same in the early stage.

In addition, their experiences need to be passed on from generation to generation [5]. There is a need for an accurate disease detector associated with a reliable database to assist farmers, especially the young and inexperienced. Recently, it continues to be used in many areas such as the detection of diseases and the application of appropriate treatment methods in precision agriculture technology applications with computer vision and Deep learning (DL) or Machine learning (ML) algorithms [29, 36].

Computer vision, image processing, and machine learning algorithms are often used in various applications such as segmentation, classification, and pattern recognition. In recent years, DL technologies have been preferred to detect plant species and solve various agricultural problems. Different systems for agricultural product analysis based on leaf images continue to be developed [16, 22]. Numerous studies use different Convolutional Neural Networks (CNN) to identify plant diseases. According to these studies, different plant diseases affecting the same plant species or different plant species are categorized. In studies on different plant species such as sycamore and peach, when the leaf images are examined in terms of shape, it is seen that they are quite different from each other. However, in some medicinal plant leaves, it has been determined that the distinctive features of the leaves such as color, shape, texture, length, width, and veins are very similar to each other. Because of these similarities, non-experts find it difficult to distinguish between these plants and their diseases.

Alstonia Scholaris leaves are used by Indians for intestinal complaints, chronic diarrhea, and dysentery. It is also used in the treatment of cancer and tumor along with malaria [1]. Arjun leaf is often used for the treatment of cardiovascular ailments and hypertension. In addition, it has an ulcer-protective effect. It is also used to prevent bone loss and increase bone mineral density [9]. Mango peel and juice are an oily mixture of organic compounds with allergenic properties called urushoil [23]. In traditional medicine, guava leaves are used to treat diarrhea [18]. Parts of the Jamun tree such as leaves, seeds, flowers, fruits, and bark are used in diabetes, allergies, viral infections, inflammation, and stomach ulcers [14].

In this experimental study, a classification study was carried out with deep learning architectures using images of healthy and diseased leaves of Alstonia Scholaris, Arjun, Mango, Guava and Jamun trees, which are of great importance for the pharmaceutical industry and traditional medicine. The study was carried out using original, augmented and image processing datasets. The contributions and original aspects of the proposed study to the literature are given below.

- An approach is presented for the classification of plant diseases with similar physical characteristics such as color, texture, vein and shape, which are of worldwide importance and used in the treatment of many diseases, in ten different categories.
- The performance of VGG16, ResNet50V2, DenseNet121, MobileNet, Xception, InceptionV3 and EfficientNetB3 pre-learning models were evaluated in plant leaf diseases.
- Comparison of the performance ratios of the original, augmented, and obtained data sets with image processing technique is presented.
- The effect of different learning rate values on educational performance is shown.

The organization of this article: In Section 2, the importance of artificial intelligence-based plant classification in the literature and the differences in the study are explained. The DL methods used in Section 3 and the features of the dataset used for leaf category and leaf disease recognition in Section 4 are explained in detail. The details of the experimental study and the comparison of the results obtained are evaluated in Section 5. In the last step, the conclusion of the article and future studies are given.

## 2. Literature Review

In order to determine the place of the proposed study in the literature and to create its contributions, studies on plant classification and disease type detection were examined. Some of the studies conducted in the literature to detect various plant diseases using deep learning and artificial intelligence techniques are as follows. Since plant leaves carry the symptoms of plant diseases, it is possible to find out what type of disease is present in the plant by examining the leaves. Shape, texture, color, and vein characteristics are used in the classification of leaf types and diseases with deep learning methods [25]. According to Yigit et al. [38], they classified 32 different plant species using five different deep learning models. In the study, an accuracy rate of 92.53% was obtained with the SVM model.

Studies on the detection and classification of diseases from leaf images of different plant species continue in the literature. Of those, the data set, which includes ten different diseases, including 21184 images, was used by Negi et al. [19]. In the study using CNN architecture, ten diseases were classified with a 96.02% success rate. In another study conducted for ten plants, the classification of healthy and diseased leaves was made by the Sahu and Minz [26]. With the self-adaptive deer hunting optimization (SA-DHOA) suggested by the authors, weighted feature extraction is performed and classification is performed with SCNN (SVM+CNN). Citrus leaf disease was to be classified by Sujatha et al. [33] using ML and DL (InceptionV3, VGG16, VGG19). In the study in which fours diseases (Black spot, Canker, Greening, Melanose) and Healthy conditions of citrus leaf were classified, the best result was obtained with 89.5% VGG16. In order to effectively identify and categorize diseases in pomegranates, a plant with a temperate climate similar to citrus, Madhavan et al. [17] developed a framework. In the study, a 98.07% success rate was obtained for five diseases, six of which were using image processing techniques.

Studies on the detection and classification of leaf diseases of plants such as avocado, mango, guava, jamun, and arjun growing in the tropical climate zone continue. Saleem et al. [27] created a new dataset of leaf diseases of five trees. The region-based fully convolutional network (RFCN), after utilizing multiple DL architectures, produced the best performance in a study that provided a DL-based technique to identify and locate leaf disease. Kavitha Lakshmi and Savarimuthu [11] proposed a new deep learning DPD-DS framework to detect multiclass plant leaf diseases. In the Mask-RCNN-based framework, healthy and diseased leaves of apple, grape, mango, pomegranate, and pongamia pinnata plants were classified with 80.01% F1-Score. It was aimed to classify the disease by using low resolution images of plant leaf diseases with the Pham et al. [20]. ANN approach using a dataset consisting of 450 images belonging to four different classes, three of which are diseased Anthracnose, Gall Midge, Powdery Mildew, and Healthy. The proposed multi-layer perceptron (MLP) model gave better results than VGG, AlexNet, and ResNet-50. In another study with the mango plant, Prabu and Chelliah [21] used a dataset of 380 images, three diseased and healthy ones. The MobileNetV2 deep learning model for feature extraction and the DVM as the classifier were used in the study, and the classification was performed with an accuracy of 94.5%. Kour and Arora [13] presented a new method for classifying seven different plants, Guava, Jamun, Mango, Grape, Apple, Tomato, and Arjun, based on their leaf image. In the study, 95.23% segmentation and classification Accuracy was obtained with the Particle swarm optimization algorithm using 1813 leaf images. Russel and Selvaraj [24] tested their proposed six parallel CNN models in the study conducted with medicinal plants and other plant datasets. Plant Village, MepcoTropicLeaf, and the Data repository of leaf image datasets had a success rate of 98.61%, 90.86%, and 90.02%, respectively.

In the literature, it has been observed that two methods are generally used in studies on the classification of plant diseases. In the first method, plants that differ from each other in shape, texture, vein, and color such as grapes, apples, corn, and potatoes are used. Secondly, they are only studies to detect different diseases of the same plant such as mango or citrus. First, the fact that leaves have different physical properties is an easily distinguishable feature for deep learning algorithms. In second, it is an easily distinguishable feature that only the diseases are different in the detection of diseases in leaves with the same physical characteristics. However, the classification of plants with the same physical characteristics as color, texture, shape, and veins and their diseases, is a difficult process compared to other methods.

## 3. Deep Neural Networks

Today, the use of deep learning architectures has increased for object detection, recognition and classification processes in images. Widely used deep learning methods are preferred in many different fields. It is no longer used for object detection in a simple image, but even for the detection of various diseases on MR images in the health sector. In addition, remarkable studies have been carried out in many areas such as the determination of seed types in agricultural activities [37], measurement of product quality [28], classification of collected products.

We can design the deep learning model by ourselves, layer by layer, or we can use pre-developed and trained ready-made models. The fact that the performance of these models has been proven and the layer structures to be used have been defined, which facilitates the development of the model. When using the model, you only need to define the input layer and the output layers in the last section.

### 3.1. Pre-Trained CNN Models

Deep learning algorithms have different models used in many areas such as classification, detection and prediction. CNN algorithm is widely preferred in image-based operations. Researchers use different CNN architectures to train models depending on their intended use and application. Similarly, seven state-of-the-art pre-trained architectures were selected in the study to design the classification system according to the health and disease status of plants: VGG16, ResNet50V2, DenseNet121, MobileNet, Xception, InceptionV3, EfficientNetB3. Brief information about the CNN architectures used in the study is given.

#### 3.1.1. VGG16

The VGG16 architecture consists of 21 layers, 13 convolutional, 5 pooling, and 3 fully connected layers in total. The image input resolution is 224×224 pixels. The convolutional filter size is 3×3 pixels. Fully connected layers utilized for feature extraction make up the final layers [31].

#### 3.1.2. ResNet50V2

ResNet (Residual Network), in 2015 by He et al. [6] was developed to facilitate the training of networks that are significantly deeper. ResNet consists of 34 layers and compared to VGG, the number of filters is less and not too complex. In Resnet50, every 2 layers in ResNet are replaced with a 3-layer bottleneck block, and a 34-layer structure is created.

#### 3.1.3. DenseNet121

DenseNet connects its layers to all other layers in a feed-forward manner. Layers take input values from the previous layers and pass their values to the next layers. Most importantly, unlike ResNets, it never combines features via aggregation before moving to a layer; instead, it combines properties. Therefore, any layer has inputs consisting of feature maps of all previous convolutional blocks [8].

#### 3.1.4. InceptionV3

The basic model, InceptionV1, was first released as GoogLeNet in 2014. InceptionV3 is an improved version of that model. InceptionV3 consists of a large number of convolution and maximum pooling steps, and the final stage includes a fully connected neural network [34].

#### 3.1.5. MobileNet

An effective and portable CNN architecture called MobileNet is utilized in practical applications. In order to provide lighter models, MobileNet predominantly uses deeply separable convolutions as opposed to the conventional convolutions utilized in earlier architectures. Deeply separable convolution layers consist of deep convolution and point convolution [7].

#### 3.1.6. EfficientNetB3

EfficientNet is among the most efficient models achieving high accuracy in both ImagiNet and pervasive image classification transfer learning tasks. EfficientNet is an advanced neural network architecture and scaling method that evenly scales all depth/width/resolution dimensions using a composite coefficient [35]. A family of models (B0 to B7) that represent a good trade-off between efficiency and accuracy at different scales are provided by EfficientNet as a heuristic method for scaling the model. It allows the efficiency-oriented basic model (B0) to beat models of all sizes while avoiding extensive grid search of hyperparameters.

#### 3.1.7. Xception

Xception architecture, which is a powerful version of Inception architecture, is short for “Extreme Inception". It consists of 71 layers in total. There are 36 convolution layers within these layers. Between these convolution layers, there are 14 jump link layers. These layers require less computation than normal convolutional layers [3].

## 4. Material and Methods

The system architecture of the study on the deep learning-based medicinal plant leaf recognition and disease detection framework is shown in Figure 1. In this experimental study, after preprocessing and data augmentation of five different plant leaf images selected from the Mendeley dataset, the data were randomly partitioned for training, validation and testing. After the optimum hyperparameter determination process, seven different deep learning algorithms were applied. The classification success of the models was measured by classifying leaf images that had never been seen before by deep learning models trained with leaf images of healthy and diseased plants.

### 4.1. Dataset Description

The Mendeley Dataset used in the study was created by researchers from Shri Mata Vaishno Devi University in Katra, India between March and May 2019 [30, 4]. In the data set, there are 4503 leaf images of 2278 healthy and 2225 diseased leaves of 12 economically and environmentally beneficial plants named Mango, Arjun, Alstonia Scholaris, Guava, Bael, Jamun, Jatropha, Pongamia Pinnata, Basil, Pomegranate, Lemon and Chinar. The leaves of each crop are divided into two categories according to their state of health, namely healthy and diseased. In the study, five different species similar to each other in terms of color, texture, vein, and size were preferred. These are images of healthy and diseased leaves of Alstonia Scholaris, Arjun, Mango, Guava, and Jamun trees. As can be seen in Figure 2, the shapes, vein structures, and

**Figure 1**

A block representation of the suggested system for identifying and categorizing plants and their diseases

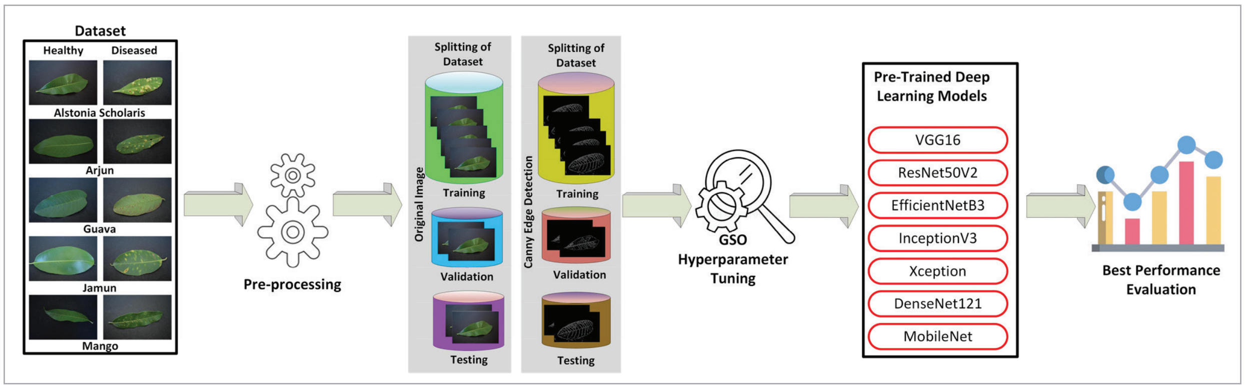


**Figure 2**

Healthy and diseased images of plants with similar characteristics in the medicical plant dataset

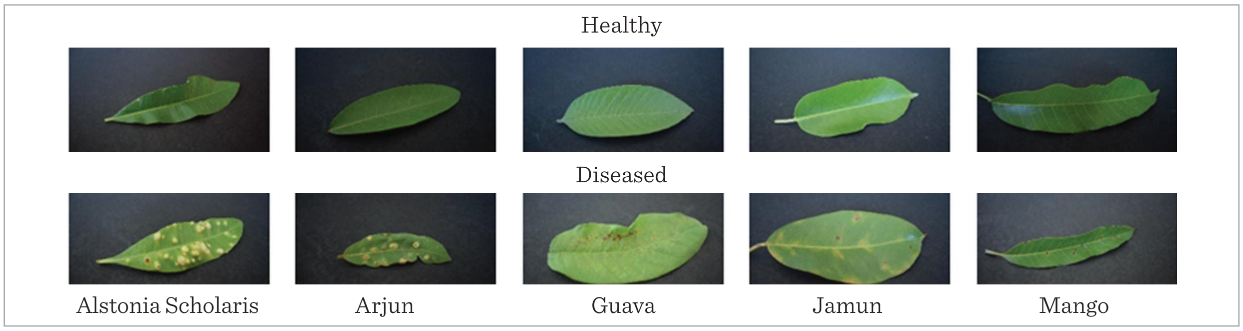


**Table 1**
Plant leaf types, diseases, and numbers used in the data set

| Plant Name | Healthy Images | Disease Category | Diseased Images | Total Images |
|---|---|---|---|---|
| Alstonia Scholaris | 179 | Leaf spot | 254 | 433 |
| Arjun | 220 | Leaf spot | 232 | 452 |
| Guava | 277 | Fungal disease | 142 | 419 |
| Jamun | 281 | Fungal disease | 344 | 625 |
| Mango | 170 | Anthracnose | 264 | 434 |
| **Total** | 1127 | | 1236 | 2363 |

colors of the leaves are so similar that they can hardly be distinguished from each other.

The names of the leaves used in the study, the number of healthy and diseased, and the disease category are given in Table 1. Out of a total of 2363 leaf images, 1127 are in the healthy category and 1236 are in the diseased category. Images are originally 4000×6000 pixels, 24-bit depth, horizontal and vertical resolutions of 96 dpi. For training deep learning algorithms in image classification, images were resized because of the high computational cost of preserving the original resolution. In all deep learning architectures used, all the images of the dataset are resized by using the non-adaptive methods of interpolation and the images were converted to 224×224 pixels.

## 4.2. Data Augmentation

The leaf images used in the study are the leaves of the Alstonia Scholaris, Arjun, Mango, Guava and Jamun trees in the Mendeley dataset. The leaves belonging to these five different species were divided into diseased and healthy according to their types and used as ten different classes. The study was carried out in three different sections using data sets created using original, data augmentation and image processing techniques. In the first part, the study was carried out using the data in the data set without any data augmentation. Of the dataset, 68% was allocated for training, 16% for testing and 16% for validation. 1601 images were used for training, 381 for testing, and 381 for validation (Table 2).

In addition to using more advanced techniques for data replication, some basic augmentation techniques were applied to all classes to increase the number of samples and robustness against unseen data. In the second part of the study, data augmentation was performed on the images by using horizontal rotation, horizontal and vertical scrolling, image zooming and zooming methods. After the data augmentation process, as seen in Table 2, 1000 images were obtained for both diseased and healthy species. Thus, a total of 10000 visuals were created in ten classrooms for educational purposes.

**Table 2**
Plant leaf types, diseases, and numbers used in the data set

| Plant | Train | | Train (Augmented) | | Validation / Test | |
|---|---|---|---|---|---|---|
| | Healthy | Diseased | Healthy | Diseased | Healthy | Diseased |
| Alstonia Scholaris | 121 | 168 | 1000 | 1000 | 29 | 43 |
| Arjun | 150 | 158 | 1000 | 1000 | 35 | 37 |
| Mango | 187 | 96 | 1000 | 1000 | 45 | 23 |
| Guava | 191 | 234 | 1000 | 1000 | 45 | 55 |
| Jamun | 116 | 180 | 1000 | 1000 | 27 | 42 |
| **Total** | 765 | 836 | 5000 | 5000 | 181 | 200 |

## 4.3. Edge Detection

Edges in images contain meaningful, important information and features. If an edge detector is applied to an image, the amount of data that needs to be processed is reduced and less relevant information is filtered out. In the third part of this study, a data set was created from healthy and diseased leaf images augmented using the Canny Edge Detector Algorithm (CEDA) in Python.

CEDA is an edge operator that can detect a wide variety of edges in an image, developed by JF Canny in 1986. CEDA reduces irrelevant image details within the image, allowing more important features of images to be revealed. Thus, only the outline of the image is left to create less clutter and lower error rates for machines. CEDA consists of four steps: noise reduction, density gradient calculation, non-maximum suppression, and hysteresis thresholding. Detailed descriptions for Canny edge detection are as follows:

**Step 1:** Noise Reduction

The picture is softened using a Gaussian filter to eliminate extraneous textures and details. The equation for the Gaussian Filter Kernel is provided in Equation 1. Standard convolution techniques are used to compute and apply the Gaussian filter.

$$H_{ij} = \frac{1}{2\pi\sigma^2} exp\left(-\frac{\left(i-(k+1)\right)^2+\left(j-(k+1)\right)^2}{2\sigma^2}\right); 1 \leq i,j \leq (2k+1). \tag{1}$$

**Step 2:** Calculating Intensity Gradient

To obtain the edge strength, the gradient density of the image is calculated using Sobel kernels (Gx and Gy) (Equation 2). 2 dimensional spatial gradient calculations are made by bending the image in both horizontal and vertical dimensions with 3×3 Gx and Gy cores [15].

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}. \tag{2}$$

With Equation 3, the edge gradient density is calculated. Equation 4 below can be used to determine the edge orientation once the x and y gradients are known.

$$G = \sqrt{G_x^2 + G_y^2} \tag{3}$$

$$\theta = tan^{-1}\left(\frac{G_y}{G_x}\right). \tag{4}$$

**Step 3:** Non-maximum Suppression

Each area of the image is scanned to remove any additional sets of pixels that do not make edges after the gradient direction and gradient size have been determined. By doing this, cells that might not be an edge are removed.

**Step 4:** Hysteresis Thresholding

To find out if the edges are genuine, this is done. It is necessary to have Minimum and Maximum threshold numbers. True edges are all edges with density gradients larger than the maximum value. All edges below the minimum value are considered false edges and are removed [12].

## 4.4. Hyperparameter Tuning

Hyperparameter tuning is one of the most important elements that increase the efficiency of machine learning models. During the training of the models, each classifier learns some parameters on its own, but each model has several hyperparameters that can be tuned that it cannot learn automatically. Since it is very difficult and time consuming to adjust each hyperparameter individually, this process can be done automatically with the Grid Search optimization (GSO) algorithm.

Grid Search Optimization Algorithm

```
Input: Hyper-parameters $h_1,...,h_k$,
    Itreations per stage $X=<X_1,...,X_z>$,
    Total number of stages $Z$,
    Training data per stage $D_{train}=<D_{train}^1,...,D_{train}^z>$,
    Validation data $D_{val}$,
    Validation accuracy $\lambda$
Output: Hyper parameters $h^*$
    for stage $z=1$ to $Z$ do
      for $i=1$ to $Y$
        $\lambda_i$=evaluate $\lambda(h_i,D^s_{train},D_{val})$
      end
      for $j=Y+1$ to$X_z$
        $g$=grid_search$(h_i,\lambda_i)^{j-1}_{i=1}$
        $hj$=max_args$_{h€a}a(h,g)$
        $\lambda_i$=evaluate $\lambda(h_i,D^s_{train},D_{val})$
      end
    reset $h_{1}:_{k}$=best k configs €$(h_1,...h_{Xz})$
  end
return $h^*$=max args$_{h€}(h^{X1},...,h^{Xz})\lambda_j$
```

The k best arrangement based on validation accuracy passing through the prior stage Y is first assessed on the training data for the current stage, Dtrain, during each stage z. The grid search algorithm is then introduced with these k parameters and tied for Xs-Y iterations on Dtrain, where XS is the total number of phase z repetitions. The execution of the following step is now started using the top configurations that rely on the validity accuracy. Following the run, all S organizes the computation and extracts the configuration from all of the hyperparameters considered by all stages with the exceptional verification precision. The algorithm eventually provides the configuration output with the most astounding validation accuracy near the conclusion of a substantial number of steps [10].

### 4.5. Performance Evaluation

To analyze the performances of the deep learning methods used in the study, the metrics accuracy (A), precision (P), recall (R), and F1-score (F1), whose mathematical equations are given in 5, 6, 7, and 8, respectively, are used. The metrics used are expressed mathematically as:

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \quad (5)$$

$$Precision = TP/(TP + FP) \quad (6)$$

$$Recall = TP/(TP + FN) \quad (7)$$

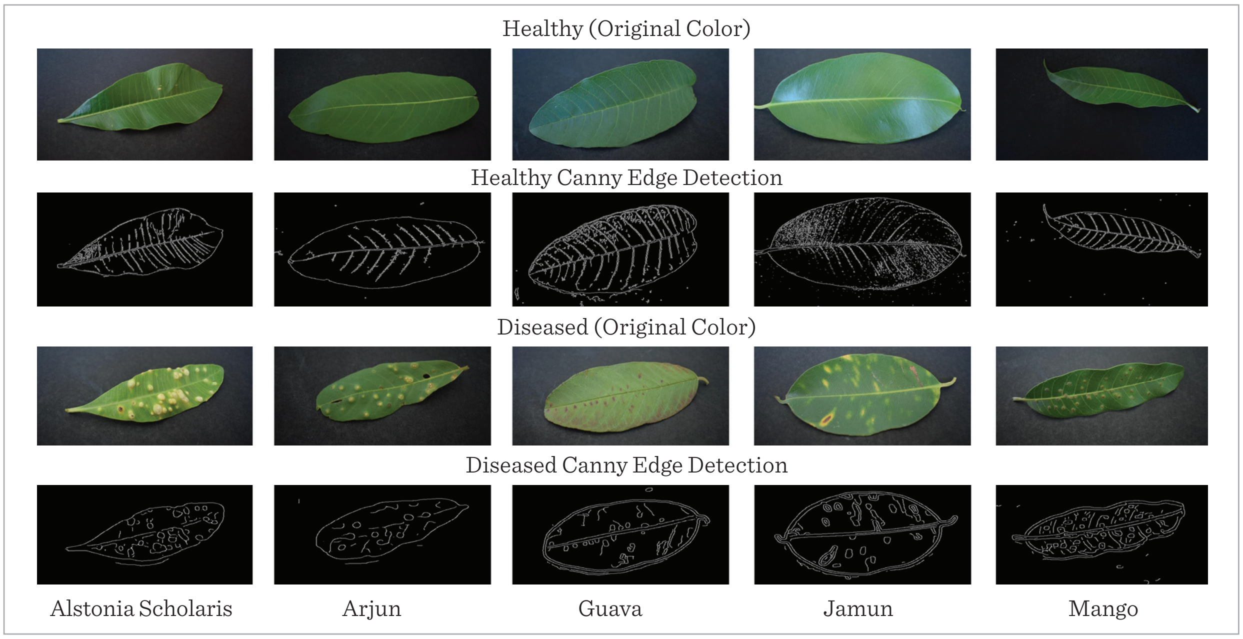
$$F1 = 2 \times ((Precision \times Recall)/(Precision + Recall)) \quad (8)$$

Here, TP (true positive) means that what is actually true is correct as a result of the estimation, TN (true negative) means that what is actually wrong is also wrong as a result of the estimation, FP (false positive) means that what should be wrong in reality is correct as a result of the estimation, FN (false negative) means that what should be true in reality is wrong as a result of the estimation. In our study, Accuracy for real data and precision, recall, and F1-score metrics for augmented data were used.

## 5. Results and Discussion

In this section, classification analysis was performed with VGG16, ResNet50V2, DenseNet121, MobileNet, Xception, EfficientNetB3, and InceptionV3 models created by transfer learning for the classification of medicinal plant images. In the study, a computer equipped with Intel I9 3.56 GHz Processor, 32 GB Ram, and 11 GB Nvidia Graphics card was used. The study was carried out in Jupyter Notebook using the Python programming language on the Anaconda platform.

**Figure 3**
Healthy and diseased leaf images obtained with CEDA


Healthy (Original Color)
Healthy Canny Edge Detection
Diseased (Original Color)
Diseased Canny Edge Detection
Alstonia Scholaris
Arjun
Guava
Jamun
Mango

CEDA was used in Python Spyder to segment the leaf veins in the images. Edge detection was tested on images of healthy and diseased leaves, and examples of the results obtained are presented in Figure 3. The main and secondary veins are clearly visible on healthy leaves. In diseased leaves, although the veins were not clear, a mixture of diseased areas and veins was obtained.

In the study, pre-trained models with good classification success were used to create new models using the transfer learning method. By altering the layers of the current architectures, the transfer learning technique enables the model to be successfully trained. By altering the layers of preexisting architectures, models can be effectively trained using the transfer learning technique [32]. In order to obtain high accuracy in the deep learning algorithms used in the study, the GSO technique was tested in the selection of the optimum hyperparameters in the ranges shown in Table 3 and optimum values were obtained.

**Table 3**
Tested hyperparameters and their values

| Hyperparameter | Range value | Best value |
|---|---|---|
| Dropout | 0.1, 0.2, 0.3, 0.4, 0.5 | 0.2, 0.4 |
| Batch size | 32, 64, 128, 256 | 64 |
| Epochs | 30, 50, 60, 90 | 50 |
| Learning rate | 0.001, 0.003, 0.01, 0.0001, 0.00001 | 0.001, 0.0001 |

**Table 4**
Experimental setup with many study-related characteristics

| Parameters | Values |
|---|---|
| Deep learning models | M1=VGG16, M2=ResNet50V2, M3=DenseNet121, M4=Xception, M5=InceptionV3, M6=MobileNet, M7=EfficientNetB3 |
| Input Image Dimensions | 224×224×3 pixels |
| Activation Function | ReLU |
| Optimizer | Adam |
| Loss Function | Categorical cross-entropy |
| Batch Size | 64 |
| Learning Rates | L1 = 0.001, L2 = 0.0001 |
| Epoch | 50 |
| Number of Categories | 10 |

In the study, as shown in Table 4, using seven different deep learning methods, ten different categories of five different plants (Alstonia Scholaris, Arjun, Mango, Guava and Jamun) were classified. Images of 224×224×3 pixels were given to the input of the models, Adam optimizer was used as the optimizer, Softmax was used as the classifier, and activation function ReLU was used. Table 4 provides additional information about the various experimental parameters used in this experimental study.

## 5.1. Comparision Between DL Architecture

In the first part of the study, the images collected from ten different classes and categories of the data set before the data augmentation process were divided into training (68%), test (16%) and validation (16%) sets. A total of 2363 images were used, of which 1601 images were used for training, 381 images for testing, and 381 images for validation.

In the proposed framework, seven different pre-learning models with different feature extraction strategies, namely VGG16, ResNet50V2, DenseNet121, MobileNet, Xception, InceptionV3, and EfficientNetB3, were selected. Some layers needed to be updated in order to train with leaf images using the weights of the pre-trained models used in the research. Input for the entire link layer is 4096 in all models after the input layer values are set. The input of the full link layer has been modified to ten in all models due to the study's dataset having ten classes; as output layers, Flatten, Dense and Dropout layers in VGG16 model, GlobalAveragePooling2D and Dense layers in ResNet50V2, DenseNet121, InceptionV3, Xception and MobileNet models, GlobalAveragePooling2D, BatchNormalization and Dropout layers in EfficientNetB3 model. In addition, the parameter value is set to 0.2 in the dropout layer. Softmax function is used for classification in the last layer of the model. Adam optimizer is preferred as the optimizer function. The loss function chosen is the categorical cross-entropy. The epoch value is set to 50 and the stack size is 64. Models were tested separately, using values of 0.001 and 0.0001 as learning rates.

The accuracy values obtained from the training, validation and testing data for two different learning values and all models are given in Table 5. In the data used by the training values, the best results were obtained with DenseNet121 with a learning rate of 0.0001 and with the VGG16 model with a learning rate of 0.001. In

**Table 5**
Train and validation accuracy of deep learning models for the original dataset

| Model | Training Accuracy (%) | | Validation Accuracy (%) | | Test Accuracy (%) | |
|---|---|---|---|---|---|---|
| | Learning rate | | | | | |
| | 0.0001 | 0.001 | 0.0001 | 0.001 | 0.0001 | 0.001 |
| VGG16 | 72.46 | 86.92 | 70.32 | 84.44 | 70.02 | 83.28 |
| ResNet50V2 | 79.11 | 69.64 | 76.48 | 67.78 | 77.03 | 66.23 |
| DenseNet121 | 86.08 | 84.86 | 82.78 | 80.93 | 80.87 | 81.02 |
| Xception | 80.66 | 79.66 | 74.98 | 78.04 | 75.02 | 76.08 |
| InceptionV3 | 72.39 | 73.39 | 69.66 | 69.52 | 67.27 | 69.13 |
| MobileNet | 82.88 | 70.38 | 80.38 | 66.46 | 80.56 | 67.61 |
| EfficientNetB3 | 74.11 | 60.81 | 78.79 | 60.73 | 75.23 | 60.55 |

the validation and test values, DenseNet121 at 0.0001 learning rate and VGG16 model at 0.001 learning rate achieved the best results.

In the training data set, very close results were obtained in the MobileNet and DenseNet121 models for 0.0001 learning rate, and in VGG16 and DenseNet121 models for 0.001 learning rate. The DenseNet121 model gave quality results for both learning rates. In the validation and test datasets, the MobileNet and DenseNet121 models for 0.0001 learning rate, DenseNet121 and VGG16 models for 0.001 learning rate were the most successful models.

## 5.2. Effects of Data Augmentation Techniques

In the second part of the study, a total of 10762 images were obtained for ten different categories belonging to five different healthy and diseased plant classes by using horizontal rotation, horizontal and vertical scrolling and image zooming methods. The images of healthy and diseased leaves of each plant for training were rounded out to 1000, so that 10000 images were used for training, 381 images for validation, and 381 images for testing.

In this part of the experimental study, seven different deep learning models with different feature extraction strategies used in the first part were used. First, in the preprocessing step, the input image values are sized to 224×224 pixels. For transfer learning, VGG16, ResNet50V2, DenseNet121, InceptionV3, Xception, MobileNet, EfficientNetB3 models were added as output layer as -1 for axis, 0.99 for momentum and 0.001 for epsilon in BatchNormalization layer. Then, 256 value and ReLU function were used for the added dense layer. Moreover, in the dropout layer the parameter value is set to 0.4. The softmax function was used as the activation function of the model. Adam was chosen as the optimizer function. Categorical crossentropy was used as the loss function. Batch size is set to 64, learning rate is set to 0.001, and epoch value is set to 50. The graphical representation of the accuracy and loss values of the training and validation sets obtained after training the models with augmented data is given in Figure 4.

The train, validation and test accuracy values obtained from all models after the data augmentation process, and the tested and faulty image values are given in Table 6. Among the deep learning models studied, the DenseNet121 model showed the highest educational success. Model 381 has 99.99% training success, 98.69% test success and only five error values in 381 test images. The best result in validation performance was obtained from the EfficientNetB3 model with a rate of 98.89%. After this model, the best test performance values were obtained from EfficientNetB3 and Xception models. The VGG16 model was the model with the lowest accuracy. In studies with data augmentation, the success of the DenseNet121 model stands out, as in studies with original data.

The confusion matrix of the DenseNet121 model, which has the highest classification accuracy as a result of the models tested after the training with data augmentation, which is the second stage of the study, is given in Figure 5.

**Figure 4**

Accuracy and loss graph for deep learning models after data augmentation (a)VGG16, (b)ResNet50V2, (c)DenseNet121, (d)Xception, (e)InceptionV3, (f)MobileNet, (g) EfficientNetB3

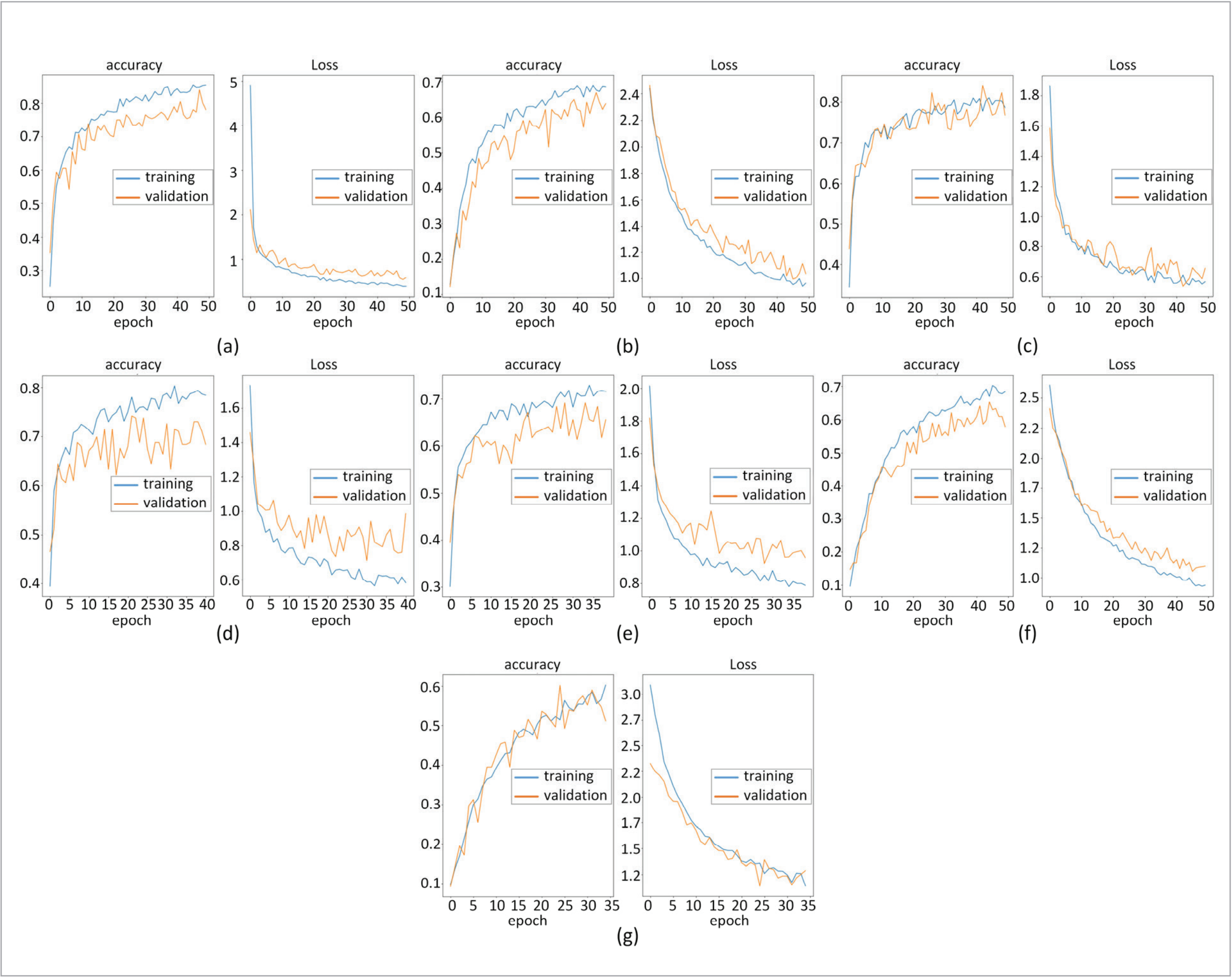


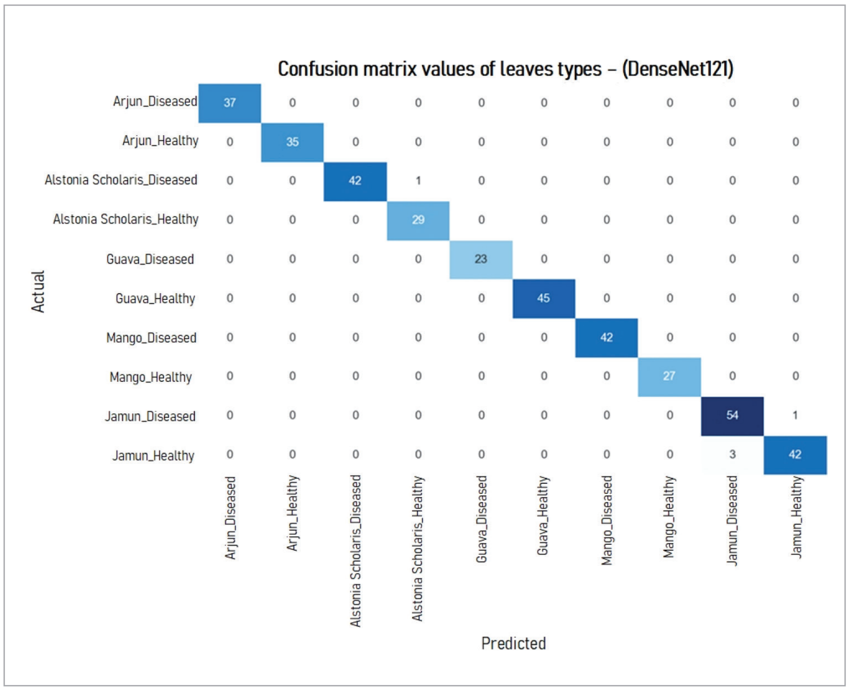
**Table 6**

Training accuracies and test/fault values of deep learning models for the augmented data set

| Model | Train Accuracy (%) | Validation Accuracy (%) | Test Accuracy (%) | Test / Fault |
|---|---|---|---|---|
| VGG16 | 98.27 | 95.67 | 95.01 | 381 / 19 |
| ResNet50V2 | 99.98 | 97.50 | 97.90 | 381 / 8 |
| DenseNet121 | 99.99 | 98.28 | 98.69 | 381 / 5 |
| Xception | 99.98 | 98.75 | 98.16 | 381 / 7 |
| InceptionV3 | 99.97 | 98.15 | 97.38 | 381 / 10 |
| MobileNet | 99.86 | 98.00 | 97.64 | 381 / 9 |
| EfficientNetB3 | 99.98 | 98.89 | 98.43 | 381 / 6 |

**Figure 5**

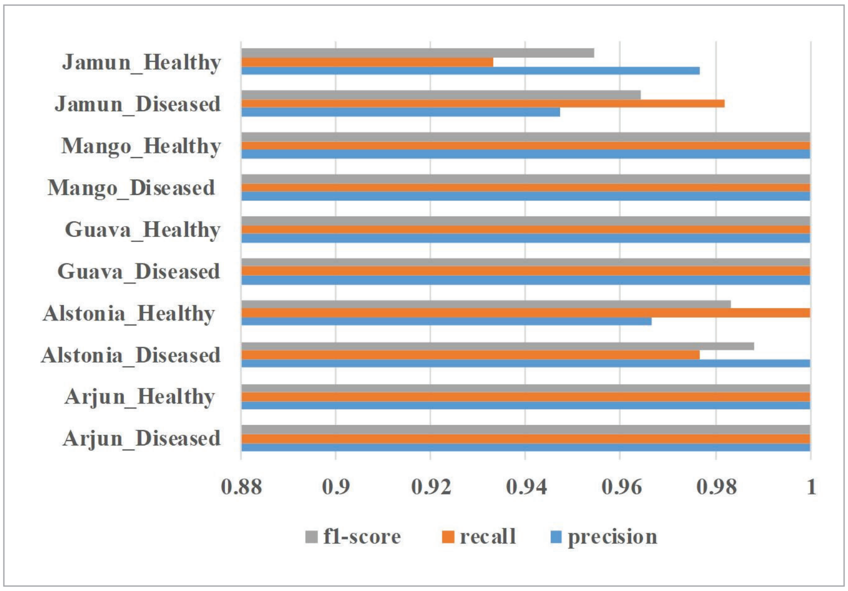
Confusion matrix for DenseNet121 on a test batch of unobserved images for data augmentation



| | Arjun_Diseased | Arjun_Healthy | Alstonia Scholaris_Diseased | Alstonia Scholaris_Healthy | Guava_Diseased | Guava_Healthy | Mango_Diseased | Mango_Healthy | Jamun_Diseased | Jamun_Healthy |
|---|---|---|---|---|---|---|---|---|---|---|
| Arjun_Diseased | 37 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Arjun_Healthy | 0 | 35 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Alstonia Scholaris_Diseased | 0 | 0 | 42 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Alstonia Scholaris_Healthy | 0 | 0 | 0 | 29 | 0 | 0 | 0 | 0 | 0 | 0 |
| Guava_Diseased | 0 | 0 | 0 | 0 | 23 | 0 | 0 | 0 | 0 | 0 |
| Guava_Healthy | 0 | 0 | 0 | 0 | 0 | 45 | 0 | 0 | 0 | 0 |
| Mango_Diseased | 0 | 0 | 0 | 0 | 0 | 0 | 42 | 0 | 0 | 0 |
| Mango_Healthy | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 27 | 0 | 0 |
| Jamun_Diseased | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 54 | 1 |
| Jamun_Healthy | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 42 |

**Figure 6**

Performance evaluation of the DenseNet121 model with data augmentation



The performance evaluation values (F1-Score, Recall and Precision) of the DenseNet121 model, in which the best performance value was obtained after the data increase, are presented in Figure 6 for each category.

### 5.3. Effects of Edge Detection Technique

In the third part of the study, 10762 images obtained by data augmentation were divided into leaf vein segments using the CEDA code in Python. 10000 images were used for training new data, 381 images for validation and 381 images for testing. Since the results obtained in the second part of the experimental study were more successful than the first part, the transfer learning structure used in the second part was preferred in this part as well. The graphical representation of the accuracy and loss values of the training and validation sets of the segment images obtained by edge detection is given in Figure 7.

The train, verification and test accuracy values obtained from all models after the edge detection process, and the tested and faulty image values are given in Table 7. Among the deep learning models examined, the DenseNet121 model showed the highest educational success. Model 381 has 80.86% training, 75.64% validation and 67.92% test success values in 381 test images. After this model, the best test perfor-

**Figure 7**

Accuracy and loss graph for deep learning models after edge detection (a)VGG16, (b)ResNet50V2, (c)DenseNet121, (d) Xception, (e)InceptionV3, (f)MobileNet, (g)EfficientNetB3

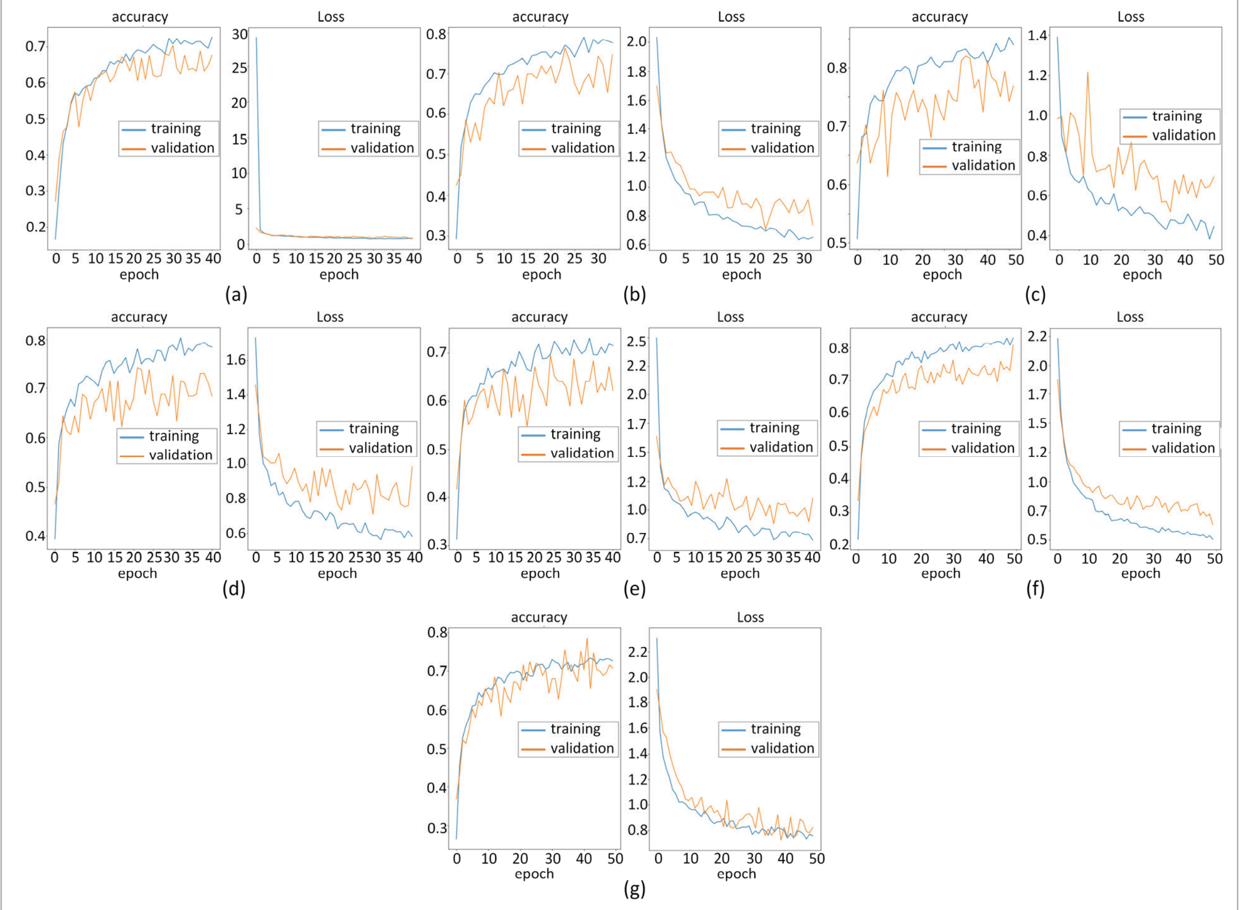


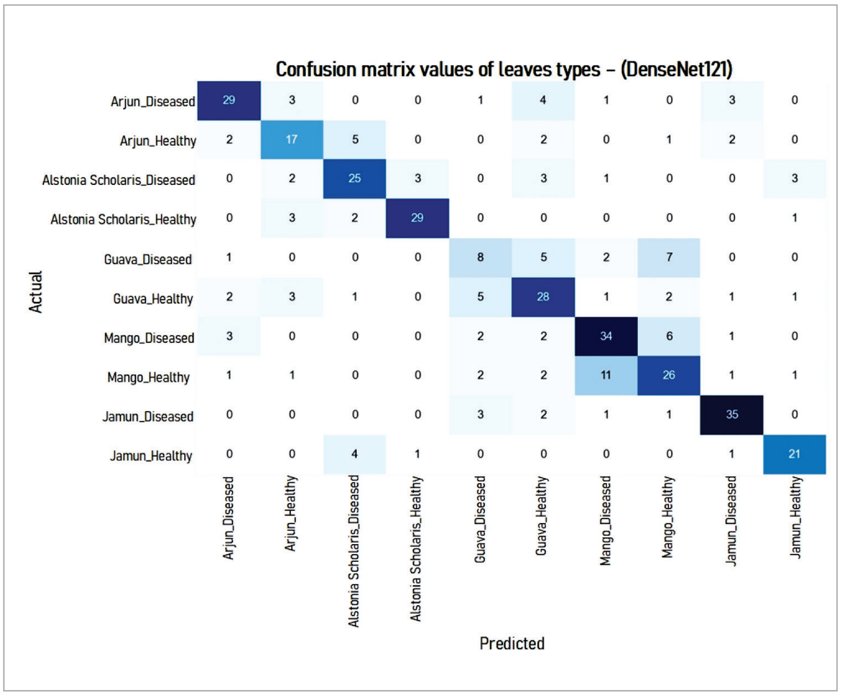
**Table 7**

Training accuracies and test/fault values of deep learning models for the edge detection data set

| Model | Train Accuracy (%) | Validation Accuracy (%) | Test Accuracy (%) | Test / Fault |
|---|---|---|---|---|
| VGG16 | 75.63 | 68.84 | 63.39 | 381 / 202 |
| ResNet50V2 | 78.34 | 71.90 | 65.37 | 381 / 130 |
| DenseNet121 | 80.86 | 75.64 | 67.92 | 381 / 119 |
| Xception | 78.81 | 72.96 | 67.55 | 381 / 127 |
| InceptionV3 | 77.09 | 71.55 | 65.02 | 381 / 136 |
| MobileNet | 76.72 | 69.44 | 67.74 | 381 / 123 |
| EfficientNetB3 | 79.41 | 73.06 | 64.20 | 381 / 142 |

**Figure 8**

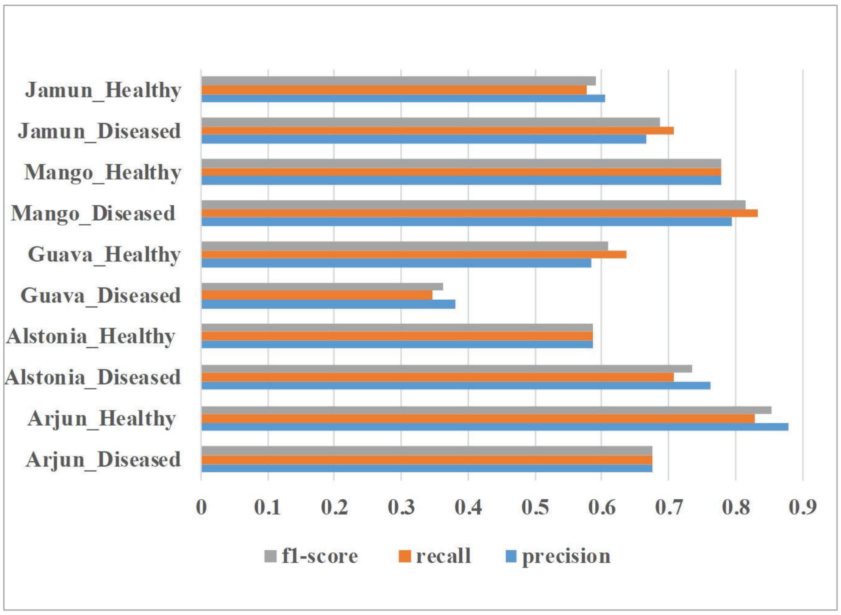
Confusion matrix for DenseNet121 after CEDA



| Actual \ Predicted | Arjun_Diseased | Arjun_Healthy | Alstonia Scholaris_Diseased | Alstonia Scholaris_Healthy | Guava_Diseased | Guava_Healthy | Mango_Diseased | Mango_Healthy | Jamun_Diseased | Jamun_Healthy |
|---|---|---|---|---|---|---|---|---|---|---|
| Arjun_Diseased | 29 | 3 | 0 | 0 | 1 | 4 | 1 | 0 | 3 | 0 |
| Arjun_Healthy | 2 | 17 | 5 | 0 | 0 | 2 | 0 | 1 | 2 | 0 |
| Alstonia Scholaris_Diseased | 0 | 2 | 25 | 3 | 0 | 3 | 1 | 0 | 0 | 3 |
| Alstonia Scholaris_Healthy | 0 | 3 | 2 | 29 | 0 | 0 | 0 | 0 | 0 | 1 |
| Guava_Diseased | 1 | 0 | 0 | 0 | 8 | 5 | 2 | 7 | 0 | 0 |
| Guava_Healthy | 2 | 3 | 1 | 0 | 5 | 28 | 1 | 2 | 1 | 1 |
| Mango_Diseased | 3 | 0 | 0 | 0 | 2 | 2 | 34 | 6 | 1 | 0 |
| Mango_Healthy | 1 | 1 | 0 | 0 | 2 | 2 | 11 | 26 | 1 | 1 |
| Jamun_Diseased | 0 | 0 | 0 | 0 | 3 | 2 | 1 | 1 | 35 | 0 |
| Jamun_Healthy | 0 | 0 | 4 | 1 | 0 | 0 | 0 | 0 | 1 | 21 |

**Figure 9**

Performance evaluation of DenseNet121 model with CEDA



mance values were obtained from MobileNet and Xception models. The VGG16 model was the model with the lowest accuracy. In studies with edge detection, the success of the DenseNet121 model stands out, as in studies with original and data augmentation data.

As a result of the models tested after the training with CEDA, which is the third stage of the study, the highest classification was obtained in the DenseNet121 algorithm. The confusion matrix of the model with the highest accuracy is given in Figure 8.

The performance evaluation values of the DenseNet121 model, in which the best performance value is obtained after edge detection, is presented in Figure 9 for each category.

### 5.4. Discussion

Finally, the proposed method is discussed and the importance of each processing step is shown. Figure 1 shows the proposed framework for describing and classifying plants and their diseases. In the first stage of the proposed study, it was seen that better results were obtained with a test learning rate of 0.001 in the results obtained from different learning methods with the original data. In the second stage of data augmentation, the DenseNet121 model with the best test accuracy was obtained. It was determined that the test success rate obtained from the original data increased by 18.50% with the data increase. In the third stage, the expected test success rate could not be obtained after the edge detection process.

**Table 8**

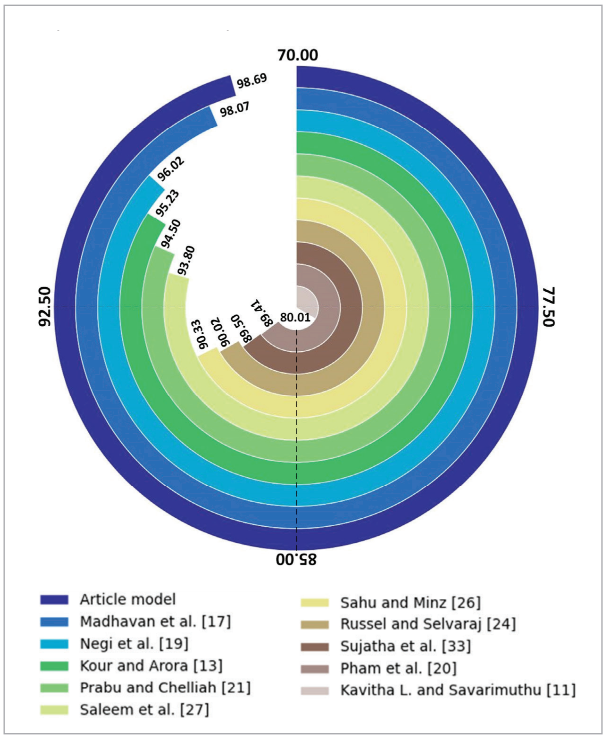
A summary of studies on the classification of plant leaves and their diseases in the literature

| References | Techniques used | No. of categories | No. of images | Accuracy (%) |
|---|---|---|---|---|
| Negi et al. [19] | CNN | 10 | 21184 | 96.02 |
| Sahu and Minz [26] | SVM+CNN | 10 | 1269 | 90.33 |
| Sujatha et al. [33] | Inception-V3, VGG16, VGG19 | 4 | 609 | 89.5 |
| Madhavan et al. [17] | Multi-class SVM | 5 | 201 | 98.07 |
| Saleem et al. [27] | Faster RCNN ResNet-50, RFCN ResNet-101, EfficientNet, and RetinaNet | 20 | 3545 | 93.80 |
| Kavitha Lakshmi and Savarimuthu [11] | Mask-RCNN | 7 | 3953 | 80.01 |
| Pham et al. [20] | Multi-layer perceptron (MLP), VGG16, AlexNet, ResNet-50 | 4 | 450 | 89.41 |
| Prabu and Chelliah [21] | MobileNetV2, SVM | 4 | 380 | 94.5 |
| Kour and Arora [13] | Particle Swarm Optimization Based Support Vector Machine (P-SVM) | 7 | 1813 | 95.23 |
| Russel and Selvaraj [24] | Unified model with six parallel CNNs | 24 | 4503 | 90.02 |
| Article model | VGG16, ResNet50V2, DenseNet121, MobileNet, Xception, InceptionV3, EfficientNetB3 | 10 | 2363 | 98.69 |

Finally, the accuracy of the proposed method and its differences with similar studies in the literature are given in Table 8. In the literature, studies have been carried out on the classification of plant leaves with different colors, textures, veins and sizes. For example, Negi et al. [19] tried to detect leaf diseases using the leaves of nine different plants such as Apple, Corn, Grape, and the leaves of ten different plants such as Sahu and Minz [26], Apple, Pepper and Tomato. Since the physical properties of the leaves used in this and similar studies are very different from each other, disease detection is easier than plants with the same physical properties [25]. Sujatha et al. [33], Madhavan et al. [17], Pham et al. [20] and Prabu and Chelliah [21] examined the diseases of only one plant and stated that they worked in maximum five different categories. However, when the number of categories (classification) increases, the classification process becomes 2-3 times more difficult. Although leaves with similar physical properties were studied in this experimental study [4] and in ten different categories, a higher success rate was obtained than the studies given in Table 8. In addition, although leaves with different physical properties (Basil, Jatropha, and Chinar etc.) are used, which makes classification easier

**Figure 10**

Comparative analysis results of the study with similar studies in the literature



in the study conducted by Russel and Selvaraj [24], the study is 9.63% more successful. Moreover, it was observed that the highest classification success rate was obtained among the classification studies conducted with ten categories (Figure 10).

## 6. Conclusions

In this study, which consists of three parts, the test performance of deep learning models obtained with a small number of images, the test performance obtained when using more images by increasing the data, and the test performance obtained when using images obtained after image processing techniques are tried to be compared. In all three sections, settings close to each other were used for all learning models. Input size, activation function, loss function, batch value and learning rate value for images are used the same in all models. Only the optimizer and epoch values have been changed.

As a result of training the models using a small number of images without data augmentation: When the Learning Rate value is set to 0.0001, the most successful model was the DenseNet121 model with 80.87%, and the most successful model was the VGG16 model with 83.28% when the Learning Rate value was set to 0.001. As a result of training the models by increasing the number of images by increasing the data, DenseNet121 was the most successful of the trainings with a test accuracy of 98.69%. Among the 381 images tested, the DenseNet121 model had the least errors with five false image values. The model with the most incorrect image values was the VGG16 model with 19 incorrect image values. After the edge detection of leaf images with CEDA, the best test performance was obtained with the DenseNet121 model with 67.92% in the trainings made as a result of training the models. It is seen that the classification success rate obtained in the edge detection process for leaf images with similar color, texture and vein structure is lower than the original images. Looking at the whole study, the DenseNet121 model achieved high educational success in both data augmentation and non-data augmentation studies. In the future, a more advanced data set will be created with different disease categories for each medicinal plant and studies will be carried out to determine the diseased area with deep learning methods.

### Funding

The author received no specific funding for this study.

### Conflicts Of Interest

The author declares to not have any conflicts of interest regarding reporting on the present study.

### Data Availability Statement

Data sharing not applicable to this article as no datasets were generated during the current study.

## References

1. Baliga, M. S. Alstonia Scholaris Linn R Br in the Treatment and Prevention of Cancer: Past, Present, and Future. Integrative Cancer Therapies, 2010, 9(3), 261-269. https://doi.org/10.1177/1534735410376068
2. Basnet, B., Bang, J. The State-of-the-Art of Knowledge-Intensive Agriculture: A Review on Applied Sensing Systems and Data Analytics. Journal of Sensors, 2018, 1-13. https://doi.org/10.1155/2018/3528296
3. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, 1251-1258. https://doi.org/10.1109/CVPR.2017.195
4. Chouhan, S. S., Singh, U. P., Kaul, A., Jain, S. A Data Repository of Leaf Images: Practice towards Plant Conservation with Plant Pathology. In 2019 4th International Conference on Information Systems and Computer Networks (ISCON), 2019, 700-707. IEEE. https://doi.org/10.1109/ISCON47742.2019.9036158
5. Fu, L., Gao, F., Wu, J., Li, R., Karkee, M., Zhang, Q. Application of Consumer RGB-D Cameras for Fruit Detection and Localization in Field: A Critical Review. Computers and Electronics in Agriculture, 2020, 177, 105687. https://doi.org/10.1016/j.compag.2020.105687
6. He, K., Zhang, X., Ren, S., Sun, J. Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, 770-780. https://doi.org/10.1109/CVPR.2016.90

7. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H. Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications. ArXiv Preprint, 2017. ArXiv:1704.04861.

8. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K. Q. Densely Connected Convolutional Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, 4700-4708. https://doi.org/10.1109/CVPR.2017.243

9. Jain, S., Yadav, P. P., Gill, V., Vasudeva, N., Singla, N. Terminalia Arjuna, a Sacred Medicinal Plant: Phytochemical and Pharmacological Profile. Phytochemistry Reviews, 2009, 8(2), 491-502. https://doi.org/10.1007/s11101-009-9134-8

10. Kaur, S., Aggarwal, H., Rani, R. Hyper-Parameter Optimization of Deep Learning Model for Prediction of Parkinson's Disease. Machine Vision and Applications, 2020, 31, 1-15. https://doi.org/10.1007/s00138-020-01078-1

11. Kavitha Lakshmi, R., Savarimuthu, N. DPD-DS for Plant Disease Detection Based on Instance Segmentation. Journal of Ambient Intelligence and Humanized Computing, 2021, 1-11. https://doi.org/10.1007/s12652-021-03440-1

12. Kieu, S. T. H., Bade, A., Hijazi, M. H. A., Kolivand, H. COVID-19 Detection Using Integration of Deep Learning Classifiers and Contrast-Enhanced Canny Edge Detected X-Ray Images. It Professional, 2021, 23(4), 51-56. https://doi.org/10.1109/MITP.2021.3052205

13. Kour, V. P., Arora, S. Particle Swarm Optimization Based Support Vector Machine (P-SVM) for the Segmentation and Classification of Plants. IEEE Access, 2019, 7, 29374-29385. https://doi.org/10.1109/ACCESS.2019.2901900

14. Kumar, S., Sharma, S., Kumar, V., Sharma, A., Kaur, R., Saini, R. Jamun (Syzygium Cumini (L.) Skeels): The Conventional Underutilized Multifunctional Plant-an Exotic Gleam into Its Food and Functional Significance. Industrial Crops and Products, 2023, 191, 115873. https://doi.org/10.1016/j.indcrop.2022.115873

15. Li, F., Du, X., Zhang, L., Liu, A. Image Feature Fusion Method Based on Edge Detection. Information Technology and Control, 2023, 52(1), 5-24. https://doi.org/10.5755/j01.itc.52.1.31549

16. Ling, X., Zhao, Y., Gong, L., Liu, C., Wang, T. Dual-Arm Cooperation and Implementing for Robotic Harvesting Tomato Using Binocular Vision. Robotics and Autonomous Systems, 2019, 114, 134-143. https://doi.org/10.1016/j.robot.2019.01.019

17. Madhavan, M. V., Thanh, D. N. H., Khamparia, A., Pande, S., Malik, R., Gupta, D. Recognition and Classification of Pomegranate Leaves Diseases by Image Processing and Machine Learning Techniques. Computers, Materials & Continua, 2021, 66(3), 2939-2955. https://doi.org/10.32604/cmc.2021.012466

18. Mukhtar, H. M., Ansari, S. H., Bhat, Z. A., Naved, T., Singh, P. Antidiabetic Activity of an Ethanol Extract Obtained from the Stem Bark of Psidium Guajava (Myrtaceae). Die Pharmazie-An International Journal of Pharmaceutical Sciences, 2006, 61(8), 725-727.

19. Negi, A., Kumar, K., Chauhan, P. Deep Neural Network-Based Multi-Class Image Classification for Plant Diseases. Agricultural Informatics, Wiley, 2021, 117-129. https://doi.org/10.1002/9781119769231.ch6

20. Pham, T. N., Van Tran, L., Dao, S. V. T. Early Disease Classification of Mango Leaves Using Feed-Forward Neural Network and Hybrid Metaheuristic Feature Selection. IEEE Access, 2020, 8, 189960-189973. https://doi.org/10.1109/ACCESS.2020.3031914

21. Prabu, M., Chelliah, B. J. Mango Leaf Disease Identification and Classification Using a CNN Architecture Optimized by Crossover-Based Levy Flight Distribution Algorithm. Neural Computing and Applications, 2022, 34(9), 7311-7324. https://doi.org/10.1007/s00521-021-06726-9

22. Qin, F., Liu, D., Sun, B., Ruan, L., Ma, Z., Wang, H. Identification of Alfalfa Leaf Diseases Using Image Recognition Technology. PLOS ONE, 2016, 11(12), e0168274. https://doi.org/10.1371/journal.pone.0168274

23. Rocha Ribeiro, S. M., Queiroz, J. H., Lopes Ribeiro de Queiroz, M. E., Campos, F. M., Pinheiro Sant'Ana, H. M. Antioxidant in Mango (Mangifera Indica L.) Pulp. Plant Foods for Human Nutrition, 2007, 62(1), 13-17. https://doi.org/10.1007/s11130-006-0035-3

24. Russel, N. S., Selvaraj, A. Leaf Species and Disease Classification Using Multiscale Parallel Deep CNN Architecture. Neural Computing and Applications, 2022, 34(21), 19217-19237. https://doi.org/10.1007/s00521-022-07521-w

25. Sachar, S., Kumar, A. Survey of Feature Extraction and Classification Techniques to Identify Plant through Leaves. Expert Systems with Applications, 2021, 167, 114181. https://doi.org/10.1016/j.eswa.2020.114181

26. Sahu, K., Minz, S. Self-adaptive-deer Hunting Optimization-based Optimal Weighted Features and Hybrid Classifier for Automated Disease Detection in Plant Leaves. Expert Systems, 2022, 39(7), e12982. https://doi.org/10.1111/exsy.12982

27. Saleem, M. H., Potgieter, J., Arif, K. M. A. A Performance-Optimized Deep Learning-Based Plant Disease Detection Approach for Horticultural Crops of New Zealand. IEEE Access, 2022, 10, 89798-89822. https://doi.org/10.1109/ACCESS.2022.3201104

28. Salvucci, G., Pallottino, F., De Laurentiis, L., Del Frate, F., Manganiello, R., Tocci, F., Vasta, S., Figorilli, S., Bassotti, B., Violino, S., Ortenzi, L., Antonucci, F. Fast Olive Quality Assessment through RGB Images and Advanced Convolutional Neural Network Modeling. European Food Research and Technology, 2022, 248(5), 1395-1405. https://doi.org/10.1007/s00217-022-03971-7

29. SepúLveda, D., Fernández, R., Navas, E., Armada, M., González-De-Santos, P. Robotic Aubergine Harvesting Using Dual-Arm Manipulation. IEEE Access, 2020, 8, 121889-121904. https://doi.org/10.1109/ACCESS.2020.3006919

30. Siddharth, S. C., Ajay, K., Uday, P. S. A Database of Leaf Images: Practice towards Plant Conservation with Plant Pathology. Mendeley Data, Madhav Institute of Technology & Science, 2019.

31. Simonyan, K., Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. ArXiv Preprint, 2014. ArXiv:1409.1556. https://doi.org/10.48550/arXiv.1409.1556

32. Singh, D., Taspinar, Y. S., Kursun, R., Cinar, I., Koklu, M., Ozkan, I. A., Lee, H. N. Classification and Analysis of Pistachio Species with Pre-Trained Deep Learning Models. Electronics, 2022, 11(7), 981. https://doi.org/10.3390/electronics11070981

33. Sujatha, R., Chatterjee, J. M., Jhanjhi, N. Z., Brohi, S. N. Performance of Deep Learning vs Machine Learning in Plant Leaf Disease Detection. Microprocessors and Microsystems, 2021, 80, 103615. https://doi.org/10.1016/j.micpro.2020.103615

34. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. Rethinking the Inception Architecture for Computer Vision. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, 2818-2826. https://doi.org/10.1109/CVPR.2016.308

35. Tan, M., Le, Q. Efficientnet: Rethinking Model Scaling for Convolutional Neural Networks. International Conference on Machine Learning, PMLR, 2019, 6105-6114.

36. Tang, Y., Chen, M., Wang, C., Luo, L., Li, J., Lian, G., Zou, X. Recognition and Localization Methods for Vision-Based Fruit Picking Robots: A Review. Frontiers in Plant Science, 2020, 11, 510. https://doi.org/10.3389/fpls.2020.00510

37. Yasar, A. Benchmarking Analysis of CNN Models for Bread Wheat Varieties. European Food Research and Technology, 2022. https://doi.org/10.1007/s00217-022-04172-y

38. Yigit, E., Sabanci, K., Toktas, A., Kayabasi, A. A Study on Visual Features of Leaves in Plant Identification Using Artificial Intelligence Techniques. Computers and Electronics in Agriculture, 2019, 156, 369-377. https://doi.org/10.1016/j.compag.2018.11.036