The Fusion of CNN and MLP Algorithm as High-Performance Classification for Identification of Healthy and Unhealthy Leaves

Ira Gaba[1,*] and B Ramamurthy[1]

ARTICLE INFO

Article history:
Received: 16 January 2024
Revised: 2 May 2024
Accepted: 28 May 2024
Online: 30 June 2025

Keywords:
Pre-processing
Feature extraction and selection methods
Convolutional Neural Network
Multi-layer Perceptron

ABSTRACT

In this study, it encapsulates the results of the work carried out with the Convolutional Neural Network, Multi-layer Perceptron, and hybrid of CNN and MLP classifier for the recognition of a tea leaf. The leaves are categorized into distinct classes by analysis and identification and recognition, which benefits both the buyer and the farmer by enabling the seller to sell tea leaves based on the quality of the leaf. Nowadays there is more advancement in the field of agriculture. But it is always the latest subject to study in the field of agriculture for the analysis and to identify quality of leaves. Many AI methodologies can be used for identification and recognition and further their fusion with different techniques or methods that can be applied to address the issue and to acquire the better accuracy. In this CNN, MLP and the hybrid of CNN-MLP are employed for determining the accuracy and this can further help in classifying the leaf in different grades like best quality, average quality, and worst quality, as for the future scope. Then the feature selection algorithms are implemented based on the different selection methods such as ANOVA, information gain, feature importance, and the random forest, which will reduce the number of parameters, at the end classification is carried out for identification of the leaf.

1. INTRODUCTION

The tea leaves are globally cultivated plant that provides various varieties of tea and it differs from region to region based on the various climate changes and the presence of moisture in the soil [1]. In addition, to this the demand has also increased globally. There are herbal benefits of using tea leaves as medicines. Therefore, it became necessary to increase the yield of the tea production. However, the leaf disease can cause losses in the yield and acts as a threat to the tea leaves department. Hence, there is an immediate need for the precise and timely identification of tea disease.

Regarding AI-based tea leaf disease detection, early identification work was consisting of various independent processes i.e., feature extraction and selection, and classification. It was previously based on extraction of color and diseased spots. However, nowadays various diseases occur at different times or under different climate conditions. Therefore, traditional recognition has poor accuracy in detecting the affected areas. With the advancement in computer vision, like these days' machine learning has gained popularity thanks to deep learning.

The production of tea contributes significantly to economic growth of India. With 1400 million tons, India accounts for a big part of the world's production. To have better production of the leaves from the plant, early illness detection and diagnosis are essential. There are many factors affecting the growth of plants for example climate change, quality of the water supply, etc. Therefore, the above factors play a crucial part in the development of production, so to yield more value-added products the quality check is very important. There are several machine learning and deep learning methodologies available that must be utilized to check quality of leaves in the early stage to yield better production, this paper will through some light on the various combinations. The majority of tea plant infections develop on the primary leaf surface, guaranteeing the well-known region of concern for disease detection techniques. Several methods can be used to identify the infected area of the leaf [3].

In the proposed work GLCM, GLRLM, GLDM, and GLSZM [4] are used to calculate the textural parameters of the leaf, which are further selected by using different selection method techniques such as ANOVA, information gain, feature importance, etc. after calculation all the parameters the classification is done with the help of MLP and CNN [5]. By using the above terminology, we are able to achieve an accuracy up to 90%.

The novelty of this work is in the feature selection and feature extraction method and classifier hybrid CNN and MLP. Total number of textual parameters are being

[1]*Computer Science Department, Christ University, Bangalore, 560067, India.*
**Corresponding author:* Ira Gaba; Email: ira.gaba@res.christuniversity.in.

extracted with the help of selection methods and the number is reduced from 75 parameters to 29 parameters. Further the classification plays a vital role for this work and the concatenation of CNN and MLP is second novel work in this research work. It will help to obtain the better accuracy as compare to normal CNN and MLP, is also shown in result section.

1.1. Contribution

The following are the contributions made by this study:

- This paper will explain the way how to obtain various features like textural by using the GLCM, GLRLM, GLSZM, and GLDM.
- After that the output features are further filtered by using the Feature Selection methods, to ensure better results.
- Introduces the classification with the help of MLP and CNN combined to get the best accuracy.
- Discussion on further combinations used for the classification.

1.2. Paper Organization

As seen in figure 1, the paper is divided into several sections. Section I will provide an introduction to the topic and an overview of all the topics to be covered in this paper. Section II will provide the literature survey carried out for the proposed work. Section III will provide the overview about the tea, and how important it is to recognize the diseases at the early stage and furthermore about the grading of the tea will elaborate the need of the better quality of the leaf before sending it for the manufacturer, how the process of the proposed work is carried out, its flowchart, its methodology, and its components and all the methods carried out to achieve the best accuracy of the leaf in details. Section IV will discuss the outcomes of the proposed work. Section V will conclude the topic and its future scope. Figure 1 depicts this research project's roadmap.

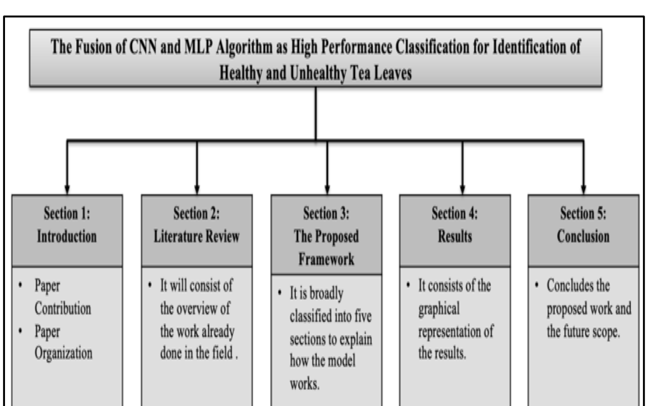


Fig. 1. Paper Roadmap.

2. LITERATURE REVIEW

The research aimed to ascertain the tea leaves quality based on a variety of leaf characteristics. Over the past few decades, the rate of consumption of tea has increased rapidly all across the world. There are many approaches have been introduced for the classification of the leaves. But for the classification mainly three steps are required preprocessing, feature extraction and classification [6]. The literature study states that numerous studies have been conducted to use machine vision to distinguish between healthy and damaged leaves. The main focus of the paper by the author [7] is to create its own dataset, and build a framework with defined parameters like shape, texture, and margin. And for the classification they have used the K-NN estimation. Many scholars are using different image processing techniques for identification and pattern recognition. The author [8] has mentioned a few methods for the identification by calculating the leaf complexity based on their dimensions. The main feature calculated in that was shape parameter and compared the internal and external shape variations. Nowadays, a wide range of tea is exported internationally, and it has a very significant role in herbal and medical use also. So before sending them to the manufacturer, it is required to identify the healthy leaves beforehand.

3. THE PROPOSED FRAMEWORK

The tea leaves play a very significant impact on the economic growth of India. There are various types of variety of tea, as shown in Figure1, but the leaves can be classified as fine leaf and coarse leaf [9]. The fine leaf is soft and light green whereas the coarse leaf is thick, dense and dark green. The fundamental characteristic of the leaves is crucial in determining their quality. The nature of the leaf to identify this is very crucial and effective if carried out in the initial stage.

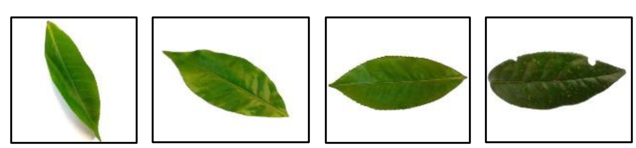


Fig. 2. The various types of tea leave.

The proposed work is carried out to evaluate the tea leaves quality by employing a range of feature selection and extraction methods followed by AI techniques that were performed ahead for the classification and to calculate the accuracy. There are following steps carried out for the proposed framework [10].

3.1 Data Collection

Data collection represents very basic step for the model, in my work, I have collected images of healthy and diseases tea leaves, which are further been classified into various categories for the grading process, which will be classified which leaves belongs to the high grade i.e., supreme quality, average grade and the worst grade. Figure 2., will show a few examples of the samples being used in the proposed work. The data set consists of 224x224 pixel images.

3.2 Image Preprocessing

The primary motive of preprocessing is to extract the correct image through the use of processing filter techniques such as Sobel, Laplacian, and canny edge detection. MSE (Mean Square Error) and SSIM (Structural Similarity Index Matric) are used to further characterize these methods [11].

3.2.1 Mean Square Error

It is typically used to assess the image's quality, as shown in Equation 1 below. Determining the degree of error and the similarity or difference between them is helpful. It is employed to ascertain the degree of noise in the input image's analytical properties [12]. We will compute the average distance between each pixel in the edge-detected image and original truth image. MSE between two pictures, f (x, y) and f' (x. y), is shown as follows.

$$MSE= \frac{1}{XY}\sum_{i=0}^{x-1}\sum_{j=0}^{y-1}[f'(i,j)-f(i,j)]^2 \qquad (1)$$

3.2.2 Structural Similarity Index Metric

It is employed to examine quality. It is further employed to determine how comparable two images are to one another. It focuses on an image's systemic information. This system is composed of three main parts: structure (s), contrast (c), and brightness (l). The method to calculate SSIM is given in equation 2 as follows.

$$SSIM(x,y) = [l(x,y)]^{\alpha} * [c(x,y)]^{\beta} * [s(x,y)]^{\gamma} \qquad (2)$$

where

$$l(x,y) = \frac{2\mu_x\mu_y+C_1}{\mu_x^2+\mu_x^2+C_1},\ c(x,y) = \frac{2\sigma_x\sigma_y+C_2}{\sigma_x^2+\sigma_x^2+C_2},\ s(x,y) = \frac{\sigma_x y+C_3}{\sigma_x\sigma_y+C_3}$$

By calculating the above method, we concluded that candy edge detection is the best method for image processing. It will provide the proper outer boundary of the leaf which will further be utilized to compute the different specified parameters in the feature extraction procedure. Following are the results captured by the filters applied to the few images of the tea leaf. Both the calculation of MSE and SSIM are shown in Figure 3.

3.3 Feature Extraction

The next crucial stage is features extraction, which is primarily concerned with figuring out the image's feature parameters. In this study, we depict the textural parameters of the input images.

The Gray Level Co-occurrence Matrix (GLCM), Gray Level Dependence Matrix (GLDM), Gray Level Run Length Matrix (GLRLM), Gray Level Size Zone Matrix (GLSZM), and Neighboring Gray Tone Difference Matrix (NGTDM) are the tools used in my work. These radiomic features are employed in the computation of an image's form parameters and texture features [13].

3.3.1. Gray Level Co-occurrence Matrix (GLCM)

The term "gray level co-occurrence matrix" refers to a second-order joint probability function of an image area in which the mask imposes constraints. Essentially, the GLCM may be used to extract 24 characteristic parameters [14]. With the aid of GLCM, the following parameters were extracted: Joint average, autocorrelation, cluster shade, contrast, correlation, difference entropy, difference variance, joint energy, and maximum probability, inverse variance, informational measure of correlation 1, inverse difference, and sum average, sum entropy, sum of squares, maximal correlation, difference average, Maximal correlation coefficient, cluster tendency, joint entropy Inverse difference moment, inverse difference moment normalized, informational measure of correlation 2, and inverse difference normalized, cluster prominence. The following figure shows the algorithm for the GLCM for tea leaf.

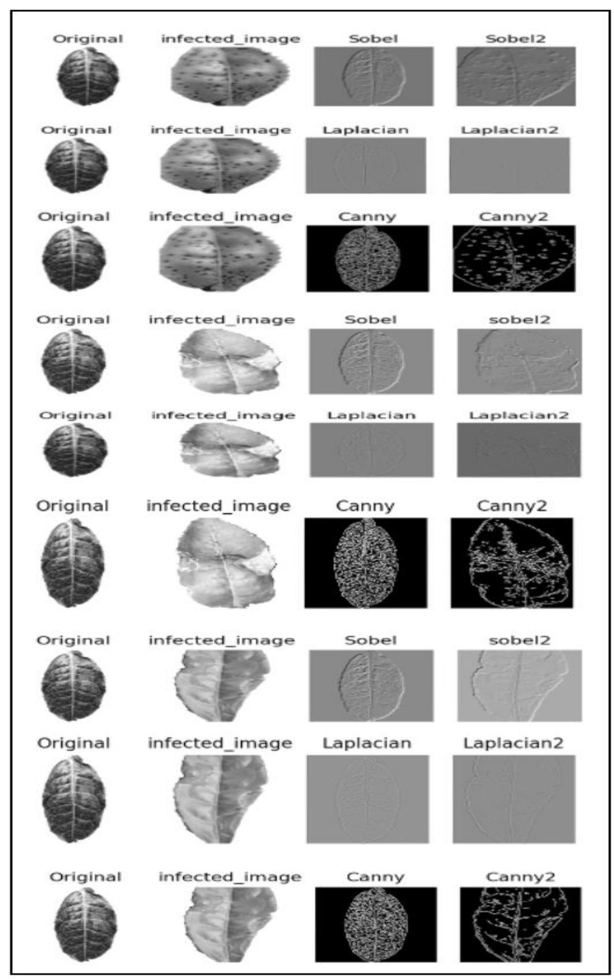


Fig. 3. using various edge detection techniques, the diseased and healthy leaves are compared in order to get the MSE and SSIM values.

Algorithm for GLCM:

```
for i from 0 to rows -1:
  for j from 0 to cols -1:
    for angle in [0,45,90,135]:
      x, y = i, j
      x_offset, y_offset = get_offset(angle)
      if is_within_bounds(x +x_offset, y + y_offset):
      GLCM[image[i.,j],image[i + x_offset, j + y_offset]] += 1
```

3.3.2 Gray Level Size Zero Matrix (GLSZM)

it is primarily defined by the number of connected voxels that have intensity of gray level. The 16 features characteristics will be extracted by GLSZM are small area emphasis, Gray level non-uniformity, Zone variance, Large area low gray level emphasis, Low gray level zone emphasis, Gray level non-uniformity normalized, Gray level variance, Size zone non uniformity, large area emphasis, Size zero non-uniformity normalized, Zone percentage, and Small area high gray level emphasis, Zone entropy, High gray level zone emphasis, Large area high gray level emphasis, and Small area low gray level emphasis. The following figure shows the algorithm for GLSZM for the tea leaf.

Algorithm for GLSZM:

```
for i from 0 to rows -1:
  for j form 0 to cols -1:
    if min_intensity<= image[i,j] <= max_intensity:
      connected_region_size =
calculate_connected_region_size(image, i,j, min_intensity,
max_intensity)
      if min_size <= connected_region_size <= max_size:
      GLSZM[image]
```

3.3.3. Gray Level Run Length Matrix (GLRLM)

Essentially, it stated how many consecutive pixels had the same gray level value and how long they were in pixels. Additionally, it will compute the 16 feature parameters, including run length non-uniformity, Short run low gray level emphasis, and run percentage, long run low gray level emphasis, short run emphasis, long run emphasis, grey level non-uniformity normalized, gray level variance, run variance, run entropy, High gray level run emphasis, and Long run high gray level emphasis, run length non-uniformity normalized, short run high gray level emphasis low gray level run emphasis. The following figure shows the algorithm for GLRLM for the tea leaf.

Algorithm for GLRLM:

```
for i from 0 to rows -1:
  run_length = 1
  for j from 1 to cols -1:
    if image[i,j] == image[i,j-1]:
      run_length +=1
    else
    run_length = min(run_length, max_run_length)
      GLRLM [image[i, j-1], run_length -1, direction] +=1
      run_length = 1
  run_length = min(run_length, max_run_length)
  GLRLM[image[I, cols - 1], run_length -1, direction] += 1
```

3.3.4. Neighboring Gray Tone Difference Matrix (NGTDM)

It determines the difference between its gray value and the mean gray value of its neighbors over a given distance. Five feature parameters contrast, coarseness, busyness, complexity and strength are extracted using this approach. The following figure depicted the algorithm for NGTDM for the tea leaf.

Algorithm for NGTDM:

```
for i from 0 to rows -1:
  for j from 0 to cols -1:
   neighborhood_values = []
    for x from i – neighborhood_size to i + neighborhood_size:
    for y from j – neighborhood_size to j + neighborhood_size:
    if is_within_bounds(x,y):
    neighborhood_values.append(abs(image[i,j] – image[x,y]))
neighborhood_value_counts =
count_unique_values(neighborhood_values)
for value, count in neighborhood_value_counts.items():
NGTDM[image[i,j],value] += count
```

The code above uses i to indicate how many rows and j to indicate how many columns the image has.

3.3.5. Gray Level Dependence Matrix (GLDM)

In essence, it calculates an image's gray level relativity and relationship. The following are the characteristics or features parameters that this method extracted, dependence non-uniformity normalized, dependence variance, dependence entropy, and low gray level emphasis, high gray level emphasis, , small dependence low gray level emphasis, large dependence low gray level emphasis, gray level non-uniformity, gray level variance , small dependence low gray level emphasis, small dependence emphasis, large dependence emphasis, and large dependence high gray level emphasis, dependence non-uniformity. The following figure shows the algorithm for GLDM for the tea leaf.

Algorithm for GLDM:

```
for i from 0 to rows - 1:
  for j from 0 to cols – 1:
    for direction in [0, 45, 90, 135]:
      x_offset, y_offset = get_offset (direction)
      if is_within _bounds(i + x_offset, j + y_offset):
        diff = abs(image[i,j] – image[i + x_offset, j + y_offset])
        if diff <= max_diff:
        GLDM [image[i,j], diff, direction] += 1
```

In the procedure above, i stands for the number of rows in the image, and j for the number of columns. x_offset and y_offset provide the change in the x and y coordinates.

3.4 Feature Selection

The feature selection is comprised of three groups Filter, Wrapper, and Embedded [15]. The first kind of approach is the filter technique, which is employed to determine how strongly a feature and target variable are related. And on the basics of that, it will rank the features. The Wrapper method will help to determine the potential subset of features by identifying all the possible subsets of features and testing their power concerning target variable. These need the classification algorithms. The third type is the Embedded methods, these methods enclose the advantages of both the methods the wrapper and the filter method by including the features and also maintaining reasonable computational cost. Of it. In our work the features that are extracted by this step of features extraction are 75 feature parameters in total, out of which we will further imply the filters to calculate the minimum no of the parameters for better accuracy. There are several feature selection techniques used in my work, which are referred to as the Filter Method, Embedded Method, and Wrapper Method as previously mentioned [16,17].

These methods are further sub-categorized as Correlation with the target (ANOVA), Information gain, pairwise correlation, Variance threshold, Lasso regularization techniques, Random-forest importance, Backward feature elimination, and Recursive feature elimination, all these methods are employed for different features like shape, color, textual, etc. Out of all the selection methods I have used the following methods for my proposed work to reduce the above 75 parameters. These methods are Anova test, Information Gain, Feature Importance, and Random Forest. After applying the above method, we succeeded in reducing parameters to maximum of 29.

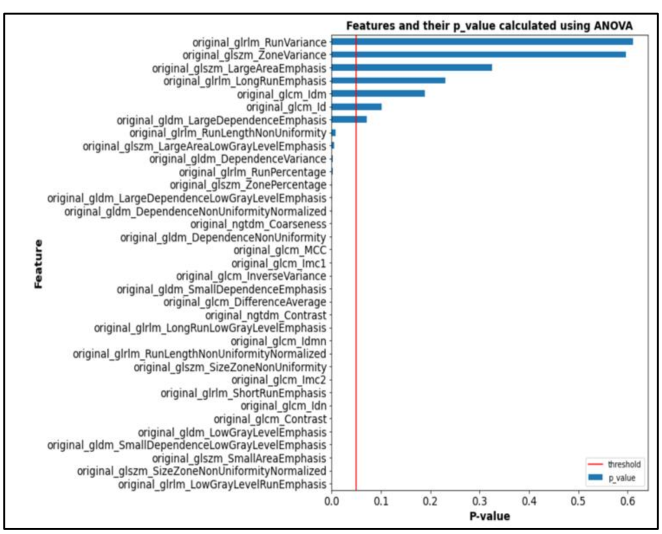


Fig. 4. The number of parameters being selected by applying the ANOVA method.

3.4.1 ANOVA method

Also known as the Analysis of Variance this one will help to determine the ideal qualities for the necessary position, as shown in Figure 4, it depicts the number of parameters being selected. The variance is an important metric for filtering because it determines whether a feature can perform a better job according to the assigned variation or not.

3.4.2 Information Gain

One of the best feature selection filter methods is information gain, which computes the dataset's entropy reduction. In essence, it computes the dataset's entropy reduction. Gain(y,A) is the result of computing the data grouped by feature A. The representation of information gain(y,A) is:

$$gain(y,A) = entropy(y) - \sum_{C \in vals(A)} \frac{y_c}{y} entropy(yc) \quad (3)$$

where, value(A) denotes the possible rates of attribute A; y_c is the subset of y and A that has the total of c.

Moreover, the total entropy of y was the rule in the above equation, and data segregation comes next, based on feature A. The following figure 5 shows the output of it. It will further rectify the parameters.

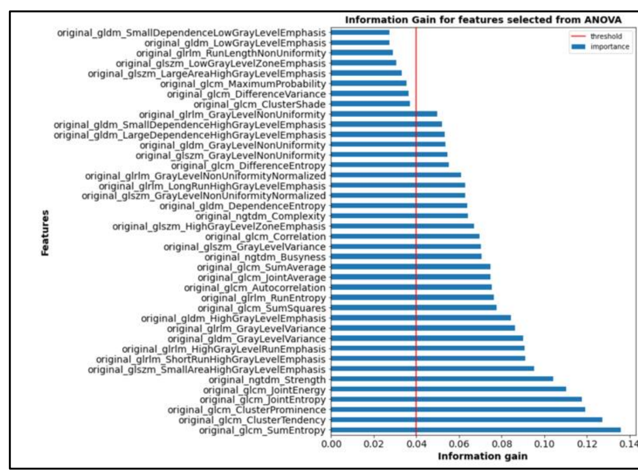


Fig. 5. Parameters chosen using the information gain approach is depicted.

The Threshold is the value that indicates the reference value of the selected feature by Information Gain. In this, we have used the value as 0.04. the following is the calculation carried out to determine each feature's value inside the given dataset.

$$S = \sqrt{\frac{\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_1)^2}{n(n-1)}} \quad (4)$$

where 'n' is number of features used in dataset, 'x' is average value of information gain, x_i is ratio of 'x' to i, and 'S' stands for standard deviation.

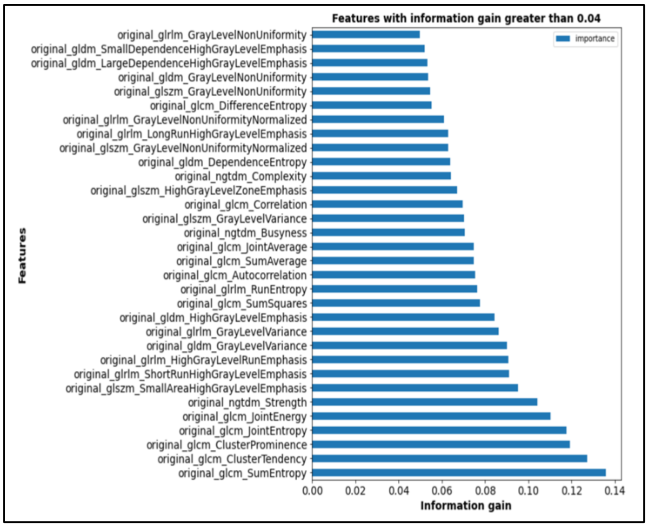


Fig. 6. The parameters being selected when the threshold value is grater then 0.04.

3.4.3 Feature Importance

The feature importance is the method that is used to further refine the parameters. The main role of the feature importance is that it will help in the dimensionality reduction and the feature selection which in return improvise the efficiency of the model. The following Figure 7, will show the selected feature.

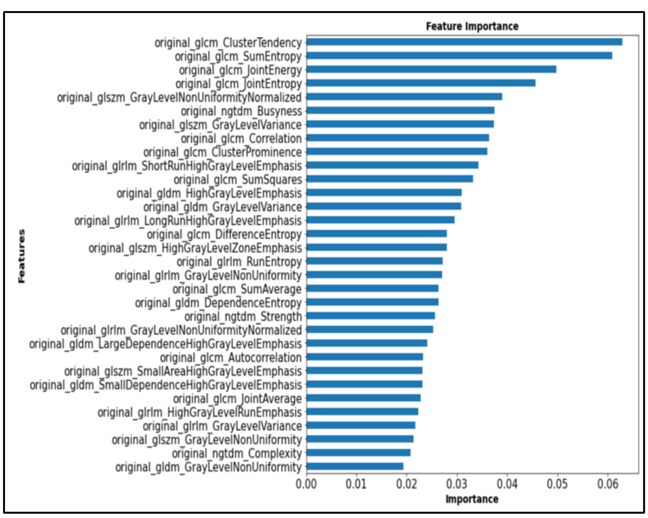


Fig. 7. The name of the parameters being selected after applying the feature importance.

3.4.4 Random Forest

It represents the decision tree method is essentially what the random forest method does. Our decision-making is aided by this tree-based machine learning system.

The final parameters being calculated after implementing all the feature selection methods are 29 in total out of 75 selected parameters. We reduced the number of parameters to 29 by using the random forest selection approach.

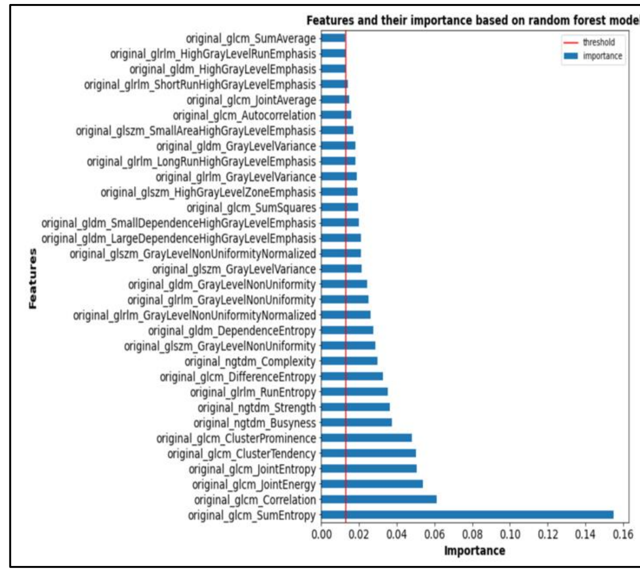


Fig. 8. The parameters selected after applying all the selection method.

The following table shows the final count and the names of the parameters being finalized after applying all selection methods. These parameters can be further used for the classification method to differentiate between healthy and unhealthy leaves. These can be further used for future work for grading the leaves into the high-quality, medium-quality and worst-quality leaves.

Table1. The following are the parameters selected after feature selection

GLCM	GLRLM	GLSZM	GLDM	NGTDM
SE	RE	GLN	DE	Busyness
Correlation	GLN	GLV	LDHGLE	Strength
Joint Energy	GLNN	GLNN	SDHGLE	Complexity
Joint Entropy	GLV	HGLZE	GLV	-
Cluster Tendency	LRHGLE	SAHGL E	-	-
Cluster Prominence	SRHGLE	-	-	-
Difference Entropy	-	-	-	-
Autocorrection	-	-	-	-
Joint Average	-	-	-	-
Sum Squares	-	-	-	-

3.5 Classification

The classification is the final stage for calculating the accuracy and to check the quality of an image. The final procedure to determine if the leaves are healthy or unhealthy is classification. The input image is identified using a variety of classifiers. Convolutional neural networks (CNNs) [18] and multilayer perceptron (MLPs) [36] are employed in my suggested study and they are further concatenated to produce better results. The following is the description of each method used separately for the classification [19].

3.5.1 Convolutional Neural Network (CNN)

The neural network made up of neurons is called the CNN [20]. Neuron is processing unit of neural network. The concept is to supply the input; after which it will be transformed mathematically to generate an output.

The relationship of that is shown in the following equation:

$$f(b+\sum_{i=1}^{n}(x_i * w_i)) \quad (5)$$

The activation function, denoted by f(.), comes in a variety of forms including Sigmoid and ReLU. The basic formula for computing, which basically represents the neuron model, is as follows: x_i is the input signal, n is the number of signals, w_i is the input signal's weight value, b is the bias, and y is the neuron's output.

3.5.1.1 Convolutional layer

This layer is most crucial layer of the model, plays most significant function in CNN. This layer is crucial since it allows for the application of several filters, which further alter the image and start its feature maps. The mapping between input and hidden layers is another component of feature maps. While the top layer captures abstract information, common features like texture, lines, and edges are predominantly recovered by the final convolution layer.

This will be transmitted to the convolutional layer. The weight matrix represents the local area of the connection feature map, and the convolution kernel uses sliding to apply convolution operations to the feature map progressively.

3.5.1.2 Pooling Layer

The feature map dimensions are optimized by the usage of the pooling layer. In essence, it will result in decreasing parameters, which will lower the network's computing performance even further. The pooling layer primarily compiles the features in the region of the feature map created by a convolution layer. We incorporated max pooling into the model, which reduces the number of pixels in the output of the preceding convolutional layer, hence reducing the dimensionality of the image.

In order to reduce the load on the convolutional layer and further reduce the number of connections in the layer, input image is primarily down-sampled and reduced using the pooling layer. Several forms of pooling may be applied, including average and maximum pooling. We have to employ the maximum pooling in our study. The output of the max pooling will include the most useful characteristics of the feature map as it typically chooses the maximum element from a section of the feature map [21,30].

Advantages of the pooling layers:

- Translation invariance: they benefit from being able to make the feature maps invariant. This indicates that an object's location is unaffected.
- Dimensionality reduction: The primary benefit of the pooling layer is its ability to assist in lowering the feature maps' spatial dimensions. Along with lowering the number of parameters, this also helps to lower the computation's overall cost and lessen overfitting.
- Feature selection: it will also help in the selection of/ the features which are further important.

3.5.1.3 ReLU layer

This layer of the rectified linear units. In actuality, this layer of neurons utilizes a non-linearity function or non-saturating loss function. Except for zero, the ReLU activation function is differentiable at all places. We only take into account the maximum of the function for values larger than zero. The maximum for the positive integer is taken into account, and all negative values default to zero.

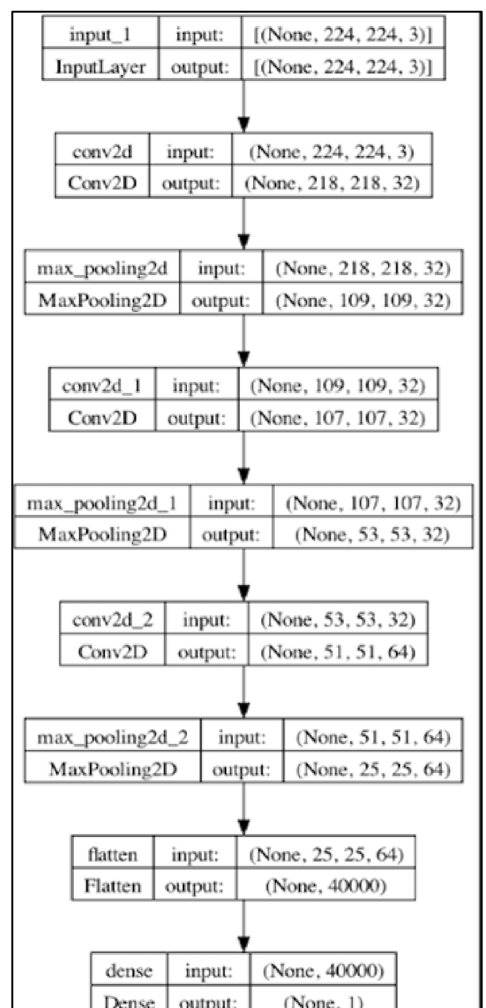


Fig. 8. The flowchart of the CNN classifier.

3.5.1.4. Flatten layer

By using flattening, all of the output arrays from pooled feature maps of any form may be made into a single, long, continuous linear vector that retains all of the data. The main function performed by the flatten layer is to convert multi-dimensional arrays to one-dimensional arrays.

In the proposed work the images were passed through the CNN for the classification of the images which gave an accuracy of 84%. The following figure shows the flowchart for the CNN classifier as it will be implemented on the images.

3.5.2 Multilayer Perceptron (MLP)

The input layer, hidden layer and output layer are additional components of Multilayer Perceptron [28,29]. It further can have a single or multiple number of neurons, which are responsible for the transmissions of signal from one layer to the other layer. Figure 7, shows the structure of the MLP. The output value vectors for the hidden unit H = F (W_h H + B_h) and the output unit Y = F (W_y H + B_y) are denoted by the letters H and Y, respectively. In this case, X is input value matrix, and the bias matrices $B_h\ and\ B_y$ are the weight matrices between layers W_h and W_y.

The MLP is also used to pass the numerical values to it. In our proposed work we have passed the calculated values of the feature selection parameter to MLP and was capable of offering 75% accuracy. The result of the images that were classified using the MLP is displayed in the following figure.

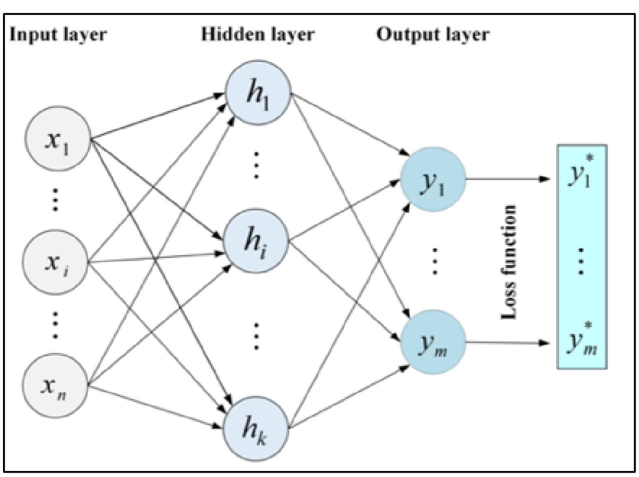


Fig. 9. the above figure shows the basic layout of the MLP.

3.5.3 Proposed CNN – MLP model

The hybrid model of CNN and MLP [22,23] will be able to take both the input the image and the numerical data. The images are being sent to the CNN and the numerical values obtained by the feature selection are input to the MLP. Further, they both are being concatenated to get better accuracy. The main idea to hybrid the CNN and MLP is to obtain the better accuracy for the leaf to segregate it into healthy and unhealthy leaves [24]. We can get improved accuracy and a better performance ratio using this hybrid technique.

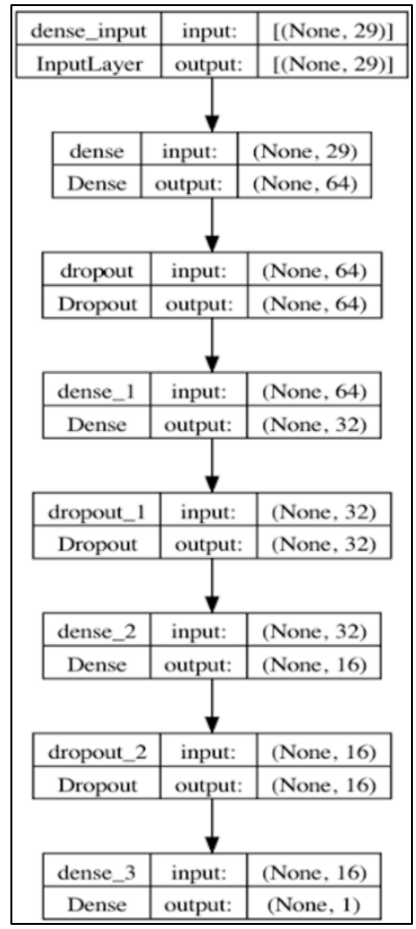


Fig. 10. Flow chat of the MLP classifier.

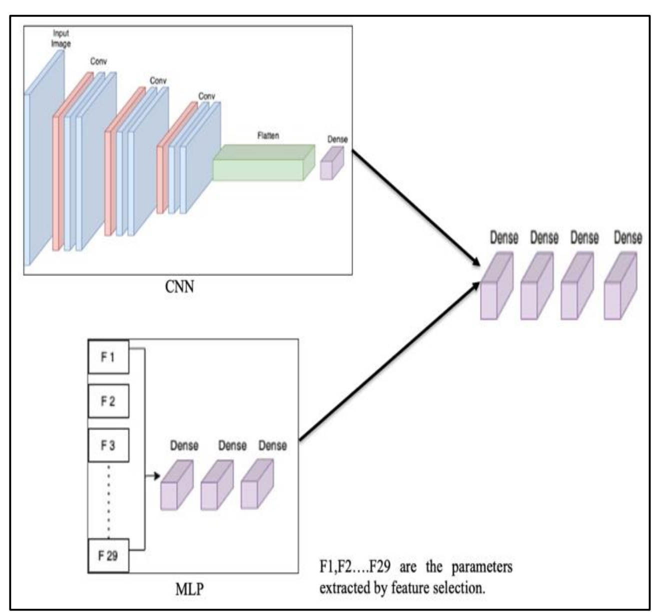


Fig. 11. The layout for hybrid CNN and MLP.

The Rectified Linear Unit (ReLU) is used as the activation function for each neuron in the input and hidden layers of CNN in our suggested study. The first layer of the MLP receives the numerical and categorical data as a one-dimensional array. The CNN model also consists of three Max-pooling layers and three convolutional layers. There are two individual outputs from the CNN and the MLP which will further be concatenated to form the one single input for the further dense layer. The new concatenated single-value input was processed as an initial input followed by the dense layers.

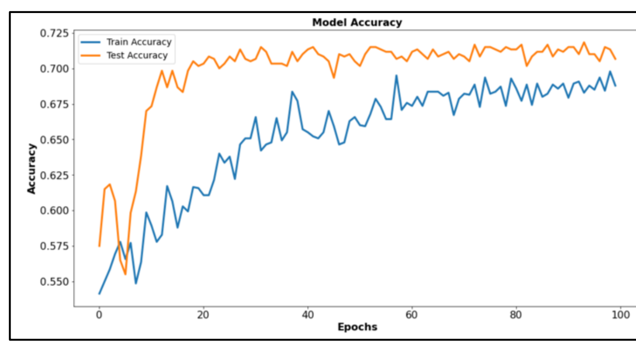
Table 2: It shows the parameters at each layer

Layer (Type)	Output Shape	Parameters	Connected To
input_2	224,224,3	0	[]
conv2d	218,218,32	4736	input_2[0][0]
max_pooling2d	109.109,32	0	conv2d[0][0]
dropout	109,109,32	0	max_pooling2d[0][0]
conv2d_1	107,107,32	9248	dropout [0][0]
max_pooling2d_1	53,53,32	0	conv2d_1[0][0]
dropout_1	53,53,32	0	max_pooling2d_1[0][0]
conv2d_2	51,51,64	18496	dropout_1[0][0]
max_pooling2d_2	25,25,64	0	conv2d_2[0][0]
dropout_2	25,25,64	0	max_pooling2d_2[0][0]
input_1	29	0	[]
conv2d_3	23,23,32	18464	dropout_2[0][0]
dense	64	1920	input_1[0][0]
max_pooling2d_3	11,11,32	0	conv2d_3[0][0]
dense_1	32	2080	dense[0][0]
dropout_3	11,11,32	0	max_pooling2d_3[0][0]
dense_2	16	528	dense_1[0][0]
flatten	3872	0	dropout_3[0][0]
concatenate	3888	0	dense_2[0][0] flatten[0][0]
dense_3	64	248896	concatenate[0][0]
dense_4	32	2080	dense_3[0][0]
dense_5	16	528	dense_4[0][0]
dense_6	1	17	dense_5[0][0]

The following Table 2 will illustrate the values of input and output of our proposed model of CNN and MLP [25,26]. To obtain this we are using the Keras functional API, in this we can to develop multiple inputs and outputs. Therefore, the overall sum up is that we can concatenate the two inputs to obtain a single output to get better accuracy. In this the input for MLP was numerical values obtained from the parameters from the feature selection and the input for the CNN was the images, which were concatenated. The below table shows the parameters obtained by the proposed architecture [27]. The following table displays the values of the parameters being calculated.

4. RESULTS

The result column discusses the model accuracy as well as model loss of the proposed work and the different classifiers CNN, MLP, and the hybrid of CNN -MLP. The result section will be able to explain the loss and accuracy curves under the classifiers CNN, MLP, and the proposed CNN-MLP. The curve of the loss drops drastically and then it decreases slowly. In the accuracy curves also there a change in the graph that varies from model to model, below figure 12 will show the results of the MLP individual classifier implemented on the leaves images. It works on the parameters we extracted by using the feature selection methods.



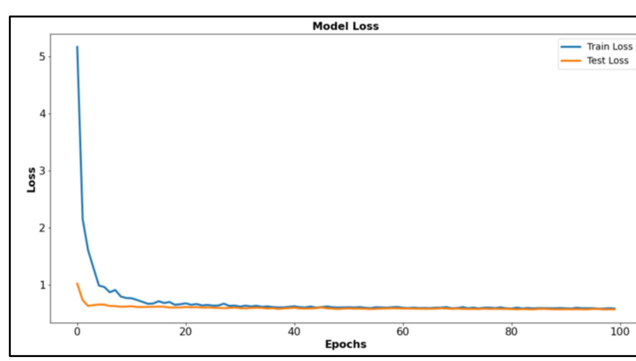


Fig. 12. Accuracy(up) and Loss(down) throughout the procedure of MLP.

The accuracy and loss of classification for CNN is shown in the below figure 13.

The primary concept underlying this proposed work is that the suggested CNN and MLP approach provide the better accuracy as compare to the individual work carried out by the CNN and MLP, as displayed in figure 14.

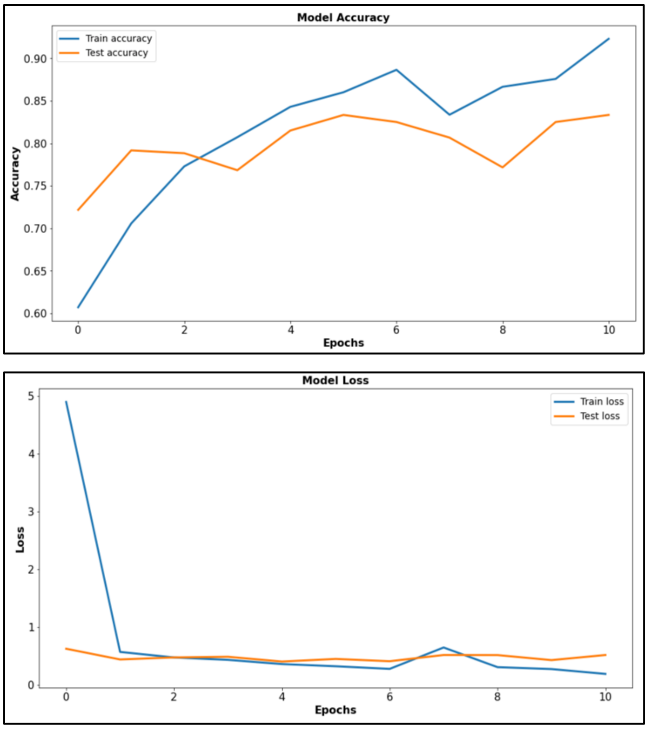


Fig. 13. Accuracy(up) and Loss(down) throughout the CNN process.

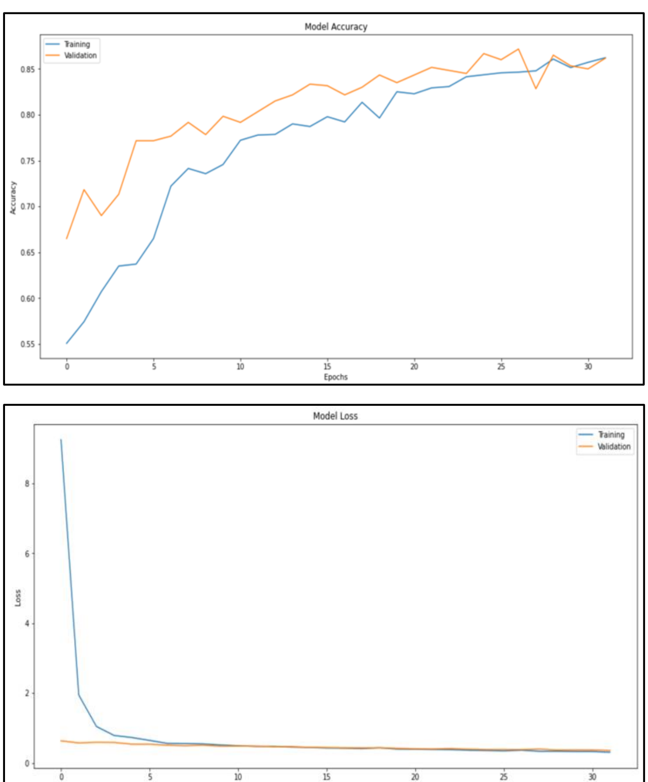


Fig. 14. Accuracy(up) and Loss(down) throughout the procedure of CNN- MLP.

The graph is basically representing the accuracy of trained and test model and loss accuracy of the test and trained model for all the CNN, MLP and the CNN-MLP. The graph is drawn with the help of epochs and the accuracy and the epochs and the loss.

5. CONCLUSION

The standard CNN, MLP, and hybrid CNN-MLP approaches are suggested in this work using various feature extraction and feature selection methods. The hybrid approach of CNN-MLP is more accurate as compare to the individual ones, it provides the accuracy rate up to 89%. The hybrid approach is more effective and reliable. The new task undertaken in this paper is to find the parameters by using the feature extraction and the feature selection out of 75 parameters it was reduced up to 29 parameters which was further passed in the MLP architecture to get the better accuracy. The CNN -MLP is the best technique out of all the three used up and it was shown in the above section also. By using this we were able to classify the healthy leaf and disease leaves separately. For future work these healthy leaves can be used for calculating the quality of the leave. There are various grades assigned to the leaf whether it belongs to the supreme quality, average quality or worst quality. In the proposed work the main novelty was in the feature selection and the hybrid of the CNN and MLP which actually increase the accuracy of the work.

REFERENCES

[1] Al-Hiary, H. 2011. Fast and accurate detection and classification of plant diseases. *Machine Learning* 14(5): 31–38. https://www.researchgate.net/publication/253511537. doi:10.5120/2183-2754.

[2] Alpaydin, E. 2017. *Introduction to Machine Learning*. MIT Press, London. ISBN: 978-0-262-01243-0. https://mitpress.mit.edu/9780262043793/introduction-to-machine-learning

[3] Myburgh, J., Mouton, C., Davel, M. 2020. Tracking translation invariance in CNN's. In: *Artificial Intelligence Research*, Springer. 1342: 282–295. https://www.researchgate.net/publication/350853256. doi:10.1007/978-3-030-66151-9_18.

[4] Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R. 2012. Improving neural networks by preventing co-adaptation of feature detectors. https://www.researchgate.net/publication/228102719. doi:10.48550/arXiv.1207.0580.

[5] Gershgorn, D. 2018. The inside story of how AI got good enough to dominate Silicon Valley.

[6] Ji, S., Xu, W., Yang, M., Yu, K. 2013. 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 35(1): 221–231. https://ieeexplore.ieee.org/document/6165309. doi:10.1109/PESGM.2016.7741877.

[7] Huang, J., Zhou, W., Zhang, Q., Li, H., Li, W. 2018. Video-based Sign Language Recognition without Temporal Segmentation. https://www.researchgate.net/publication/322819128. doi:10.1609/aaai.v32i1.11903.

[8] Anand, R., Veni, S., Aravinth, J. 2016. An application of image processing techniques for detection of diseases on brinjal leaves using K-means clustering. In: *5th Int. Conf. on Recent Trends in Information Technology*. https://ieeexplore.ieee.org/document/7569531. doi:10.1109/ICRTIT.2016.7569531.

[9] Agrawal, S., Jalal, A.S., Khan, A. 2018. Plant identification using leaf image analysis. In: *Int. Conf. on Internet of Things and Connected Technologies*. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3170725. doi:10.2139/ssrn.3170725.

[10] Le, T.L., Duong, D.N., Nguyen, V.-T., Hai, V., Hoang, V.-N., Nguyen, T.-N. 2015. Complex background leaf-based plant identification based on interactive segmentation. In: *Proc. 2nd Int. Workshop on Environmental Multimedia Retrieval*, pp. 3–8. doi:10.1145/2764873.2764877.

[11] Pawar, P., Turkar, V., Patil, P. 2016. Cucumber disease detection using artificial neural network. In: *Int. Conf. on Inventive Computation Technologies*. doi:10.1109/INVENTIVE.2016.7830151.

[12] Xiao, Z., Yan, Q., Amit, Y. 2020. An out of distribution detection score for variational auto-encoder. *Advances in Neural Information Processing Systems*. https://proceedings.nips.cc/paper/2020/file/eddea82ad2755b24c4e168c5fc2ebd40-Paper.pdf.

[13] Gondara, L. 2016. Medical image denoising using convolutional denoising autoencoders. In: *IEEE ICDMW*, pp. 241–246. doi:10.48550/arXiv.1608.04667.

[14] Bashish, D.A., Braik, M.S., Bani-Ahmad, S. 2011. Detection and classification of leaf diseases using K-means and neural networks. *Information Technology Journal* 10: 267–275. doi:10.3923/itj.2011.267.275.

[15] Nijalingappa, P., Madhumathi, V.J. 2015. Plant identification system using its leaf features. In: *Proc. Int. Conf. on Applied and Theoretical Computing and Communication Technology (iCATccT)*, IEEE, pp. 338–343. doi:10.1109/ICATCCT.2015.7456906.

[16] Zheng, J.G., Vaishnavi, V. 2011. A multidimensional perceptual map approach to project prioritization. *AIS Trans. Human-Computer Interaction* 3(2): 82–103. doi:10.17705/1thci.00028.

[17] He, K., Zhang, X., Ren, S., Sun, J. 2016. Deep residual learning for image recognition. In: *IEEE CVPR*, pp. 770–778. doi:10.1109/CVPR.2016.90.

[18] Turkoglu, M., Hanbay, D. 2015. Classification of grape varieties based on leaf recognition using SVM. In: *Proc. Signal Processing and Communications Applications Conf.*, pp. 2674–2677. doi:10.1109/SIU.2015.7130439.

[19] Malarvizhi, C., Balamurugan, P. 2019. Qualitative analysis of edge detection techniques on cervical spine images. *Int. J. Image and Vision Processing*. doi:10.21917/ijivp.2019.0282.

[20] Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., Stefanovic, D. 2016. Deep neural networks for recognition of plant diseases by leaf classification. *Comput. Intell. Neurosci.* 2016: 3289801. doi:10.1088/1757-899X/288/1/012058.

[21] Zhang, S., Wu, X., You, Z.-H., Zhang, L. 2017. Cucumber disease recognition using sparse representation. *Comput. Electron. Agric.* 134: 135–141.

[22] Karmokar, B.C., Ullah, M.S., Siddiquee, M.K., Alam, K.M.R. 2015. Tea leaf disease recognition using neural network ensemble. *Int. J. Comput. Appl.* 114: 27–30. doi:10.5120/20071-1993.

[23] Keith, L., Ko, W.-H., Sato, D.M. 2006. Identification guide for diseases of tea. *Plant Disease* 2006: 1–4.

[24] Tax, N., Verenich, I., La Rosa, M., Dumas, M. 2017. Predictive business process monitoring with LSTM. In: *CAiSE*, LNCS 10253: 477–492. doi:10.1007/978-3-319-59536-8_30.

[25] Choi, E., Bahadori, M.T., Schuetz, A., Stewart, W.F., Sun, J. 2016. Predicting clinical events via recurrent neural networks. In: *Proc. Machine Learning for Healthcare*, pp. 301–318.

[26] Schmidhuber, J. 2015. Deep learning in neural networks: An overview. *Neural Networks* 61: 85–117. doi:10.10106/j.neunet.2014.09.003.

[27] Zoph, B., Le, Q.V. 2016. Neural architecture search with reinforcement learning. https://arxiv.org/abs/1611.01578. doi:10.48550/arXiv.1611.01578.

[28] Maitra, D.S., Bhattacharya, U., Parui, S.K. 2015. CNN-based approach for handwritten character recognition of multiple scripts. In: *13th Int. Conf. on Document Analysis and Recognition (ICDAR)*, pp. 1021–1025. doi:10.1109/ICDAR.2015.7333916.

[29] Nabian, M.A., Meidani, H. 2017. Deep learning for accelerated reliability analysis of infrastructure networks. *Comput.-Aided Civil Infrastructure Eng.* 33(6): 443–458. doi:10.1111/mice.12359.

[30] Khan, A.I., Shah, J.L., Bhat, M.M. 2020. Coronet: A deep neural network for detection and diagnosis of COVID-19. *Comput. Methods Programs Biomed.* 196: 105581. doi:10.1016/j.cmpb.2020.105581.