A Framework for Agriculture Plant Disease Prediction using Deep Learning Classifier

Mohammelad Baljon*
Department of Computer Engineering,
College of Computer and Information Sciences,
Majmaah University, Majmaah, Saudi Arabia, 11952

***Abstract*—The agricultural industry in Saudi Arabia suffers from the effects of vegetable diseases in the Central Province. The primary causes of death documented in this analysis were 32 fungal diseases, two viral diseases, two physiological diseases, and one parasitic disease. Because early diagnosis of plant diseases may boost the productivity and quality of agricultural operations, tomatoes, Pepper and Onion were selected for the experiment. The primary goal is to fine-tune the hyperparameters of common Machine Learning classifiers and Deep Learning architectures in order to make precise diagnoses of plant diseases. The first stage makes use of common image processing methods using ml classifiers; the input picture is median filtered, contrast increased, and the background is removed using HSV color space segmentation. After shape, texture, and color features have been extracted using feature descriptors, hyperparameter-tuned machine learning (ML) classifiers such as k-nearest neighbor, logistic regression, support vector machine, and random forest are used to determine an outcome. Finally, the proposed Deep Learning Plant Disease Detection System (DLPDS) makes use of Tuned ML models. In the second stage, potential Convolutional Neural Network (CNN) designs were evaluated using the supplied input dataset and the SGD (Stochastic Gradient Descent) optimizer. In order to increase classification accuracy, the best Convolutional Neural Network (CNN) model is fine-tuned using several optimizers. It is concluded that MCNN (Modified Convolutional Neural Network) achieved 99.5% classification accuracy and an F1 score of 1.00 for Pepper disease in the first phase module. Enhanced GoogleNet using the Adam optimizer achieved a classification accuracy of 99.5% and an F1 score of 0.997 for Pepper illnesses, which is much higher than previous models. Thus, proposed work may adapt this suggested strategy to different crops to identify and diagnose illnesses more effectively.**

Keywords*—Suggested agricultural plant disease prediction system; tuned ML models; machine learning classifiers; plant disease detection; deep learning architectures

I. Introduction

The vast boundaries of the Kingdom of Saudi Arabia include an area of over two million square kilometres, or more than 80% of the whole landmass of the Arabian Peninsula. The country's position helps to explain its mild winters and hot, dry summers. It is between 15.2 and 32.6degrees north latitude and 34.1 and 55.5degrees [1] east longitude. Outside of the south-western highlands, where it rains more often in the summer, air receives less than 100 millimetres of precipitation annually. The natural springs in the Hue region provide the vast bulk of the water utilised in local farms in order to sustain water supply and replenish aquifers, dams have been built at various locations around the country.

The agriculture industry in Saudi Arabia has received significant attention as part of the country's five-year growth goals. These initiatives aim to diversify the economy away from oil exports so that more people can eat and the quality of life can be kept high despite the population boom. The full potential of the country's agriculture sector is now being tapped upon. The quantity of land under cultivation is skyrocketed from around 435 thousand ha in 1980 to more than 1.5 million ha in 1990, [2] and a large part of that increase may be attributed to government encouragement and support. Two further agricultural academies emerged at the same time as new plant varieties were developed, the greenhouse business was launched, and massive agricultural undertakings were undertaken. In addition to meeting domestic need, the country currently exports food products such as wheat, dates, melons, poultry, fresh eggs, and milk [3]. Fig. 1 shows Saudi Arabia’s market production.

Wheat, sorghum, barley, and millet are examples of important cereal crops; tomatoes and watermelons are examples of important vegetable crops; date palms, citrus trees, and grapevines are examples of important fruit crop species. The importance of fodder crops like lucerne is also significant. More over 1,100,000 acres, or around 81% of the total cultivated land area, is devoted to these crops. In 1990, wheat was grown on an estimated total of around 744 422 acres [4], or almost 55% of the total cultivated area. About 3.5 million metric tonnes were harvested from the crop.

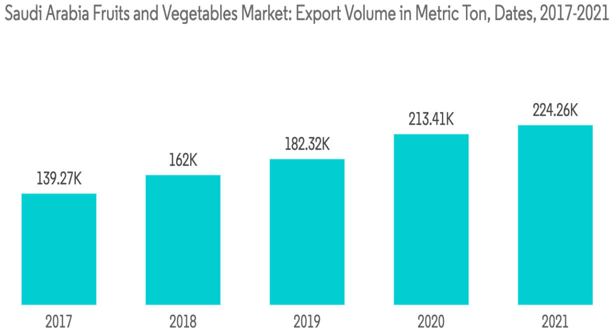


Fig. 1. Saudi Arabia vegetable market production metric.

* m.Baljon@mu.edu.sa

"Smart farming," often known as "precision agriculture," refers to the practise of using computer-based intelligent technologies in agriculture. Research in its infancy includes, but is not limited to, studies of intelligent irrigation systems, automated pesticide management, and the detection of plant diseases. Systematic methods of disease detection are crucial for the early diagnosis and prevention of certain conditions. Horticulturalists have an 88% success rate [5] in diagnosing plant diseases using standard diagnostic approaches. Despite this, it's a complicated procedure that calls for experience in the field. It may be difficult to undertake an inquiry of this sort in certain cases due to the crop's location. Several studies have proposed using Deep Learning Architectures (VGG 16, VGG 19, ResNet 50) or conventional Image Processing methods (Image filtering, contest stretching, segmentation, feature extraction, and disease classification) to classify various plant diseases. Traditional image processing techniques excel in situations when a large number of training data is unavailable. This means that several, fine-grained techniques, all fine-tuned to a high degree of precision, are required for correct sickness classification.

The structure of the paper is as follows; Section II includes study of existing methodology, Section III includes proposed methodology, Section IV includes Experimental analysis, Section V includes conclusion and future work.

II. Literature Survey

A farmer's first priority should be the detection and eradication of plant diseases. The leaves are the first part of the plant to show damage from an approaching attack, and the changes that occur in the leaves are readily apparent. However, if you try to segment a picture of a sick leaf, you'll probably wind up with a jumbled, poorly lit composite. Because of this, we'll be taking a look at and ranking the latest classification algorithms for identifying problems with tomato leaves.

Effective convolutional neural network (CNN) was the name given to the deep CNN model developed by the team working on this project [2]. After the model was built and trained, it was put to the test by identifying images of healthy and diseased tomato leaves. The U-net model and a modified U-net model were put to the test and compared with regards to how effectively they segmented leaves. Experiments using six- and ten-class classification models were also undertaken in addition to the binary-classification ones. When it comes to segmenting images of leaves, only the U-net model was able to achieve an accuracy of 98.66 per cent. In contrast, EfficientNet-B7 showed stable performance across a range of classification tasks, from binary to six-class classifications. Using segmented pictures, the average accuracy of binary classification is 99.95%, while the average accuracy of six-class classification is 99.12%.

Using an image segmentation approach optimised for super pixels, the authors of [3] created a system for automatically recognizing and classifying tomato illnesses. A color-balance technique was employed during preproduction so that the optimal threshold for each kind of image collection could be identified. To further distinguish the leaves from the background, a ground-breaking method based on a histogram of gradients and color changes was used. A pyramid of the histogram of gradients, a shape descriptor, and a grey level co-occurrence matrix (GLCM) are all components of a feature extraction approach that has been shown to be effective in distinguishing various medical presentations that are otherwise identical. The use of classifiers is pervasive in this investigation. Still, the best results towards the goal of the suggested framework were achieved by the random forest classifier, trained on a dataset of one hundred trees. When comparing the results of this study with those of many others that are quite similar to it [4,5,6,7], it was found that comparative analysis based on estimate parameters was the most accurate technique.

To classify potential tomato leaf diseases, the authors of [8] proposed a multi-class feature extraction strategy. The system is based on a deep CNN model that makes use of the attention method and the residual block. The findings show that the model is effective at picking up commonalities across diseases. Moreover, it uses the widely-used Plant Village dataset to get better results than a substantial body of prior deep learning literature. The overall positive identification percentage for this inquiry was 99.24%.

In [9], the feasibility of separating apart the various tomato diseases is examined. To boost model performance with little computational overhead, a lightweight CNN approach was developed and shown. The computational complexity, performance, and network architecture are only few of the areas that were explored for this essay. The used dataset consists of information from one healthy person and nine patients with various diseases. According to the results, if a model that is easy to understand and computationally efficient is created, the classification accuracy might be improved. Many different CNN architectures were developed for the tomato leaf disease detection study presented in [10]. LeNet, VGGNet, ResNet50, and Xception are only few of these designs. Scientists used 14,903 images of tomato plant leaves from the Plant Village dataset to build a deep Convolutional Neural Network (CNN). Both healthy and diseased plant leaves were shown in these photographs. The data showed that among all the tested designs, the fine-tuned VGGNet de-sign provides the best classification (99.25% accuracy) and achieves the lowest loss. This is true despite the fact that there is a significant time commitment associated with training and substantial financial outlay for the necessary technology

Grape plant leaf diseases are no match for the deep transfer learning-based model developed by the authors of [11]. In order to separate out the most crucial features, the authors built a fully connected layer. After that, redundant data was removed from the feature extractor vector using the variance method. With the use of images from the Plant Village dataset, the Efficient Net B7 deep architecture was retrained in this case. Then, logistic regression was utilized to further refine the collected data. By using this method, we were able to improve our categorization accuracy to 99.7%.

Using a dataset of 3,000 images of tomato leaves using the Google Collaborative Net-work (CN) model, the authors of [12] were able to accurately identify and classify nine different diseases and one healthy leaf class. Images are first preprocessed, then regions are separated, and finally the findings are presented in this novel approach. The images are then processed further, which entails tuning the CNN model's hyper-parameters. The input picture is then analysed by the CNN, which pulls out features like as colors, borders, and textures. Prediction accuracy for the built categorization model is 98.49%.

According to the study published in [13], a CNN model with three convolutional layers, one max-pooling layer, and a filter count that can be adjusted between one and ten was developed. The authors of the Plant Village dataset used augmentation techniques to rectify the imbalance they discovered between the number of photos in each class. This model has an average accuracy of 91.2% and needs nearly 1.5 MB of storage space, whereas the pretrained model needs 100 MB.

The authors of [14] employed a transfer learning technique to construct a deep learning model that needed less training data, less computational resources, and less time to train. The scientists employed five different deep network topologies—MobileNet, Resnet50, Xception, Densenet121 Xception, and Shuffle Net—to extract characteristics. The authors experimented with many educational methods and tempos. With a classification accuracy of 97.10%, the DenseNet Xception easily triumphed over the competition.

According to [15], a deep residual network is built to identify tomato leaf diseases. To improve the remaining thick network, the authors chose to alter its structure. The model might be easily modified into a classification model with a 95% accuracy rate. The authors of [16] provide a system that can automatically detect and classify leaf diseases. To lessen the load on the computer's resources, the input photographs must be downscaled before the background can be removed and the photos separated. In order to extract features, the researchers in this work used two distinct deep learning models: VGG19 and Alex Net. Then, an ECOC-based SVM classifier was used to determine the identities of these characteristics. The results showed that VGG19 and Alex Net both achieved 98.8% and 98.9% classification accuracy, respectively.

In order to categorize the wide variety of diseases that might afflict soybean plants, the authors of [17] use multilayer perceptron deep learning and support vector machine techniques. In all, 19 diseases were correctly classified by the SVM. There were 683 examples in the dataset used; 643 were correctly classified while the remaining 40 had incorrect labels, for a classification accuracy rate of 94.14 per cent.

As was seen in the prior section of this article, the classification accuracies of various deep learning and machine learning algorithms created and contested for the detection of tomato leaf diseases vary widely. In addition, a few techniques have been created. In Table I, we can see a comparison of the algorithms utilized the diseases that may be identified, and the accuracy of the various classification systems that have been covered thus far.

TABLE I. A COMPARISON BETWEEN THE RECENTLY DEVELOPED CROP CLASSIFICATION SYSTEM

[2]	CNN model	3 disease classes	RF: 97%
[3]	Segmentation-based CNN	8 disease classes	99.23%
[4]	Light weight CNN	8 disease classes	99.25%
[5]	VGGNet- CNN	8 disease classes	VGGNet: 99.25%
[6]	Hybrid CNN (Hy-CNN)		
[7]	RNN Model	8 disease classes	97%
[8]	SVM model	8 disease classes	98.39%
[9]	ResNet-50 + SeNet	8 disease classes	97.8%
[15]	Restructured residual dense network	8 disease classes	95%
[16]	VGG19, Alex Net		91.2%
[17]	Deep learning, SVM		Densenet121 Xception

III. PROPOSED MATERIALS AND METHODS

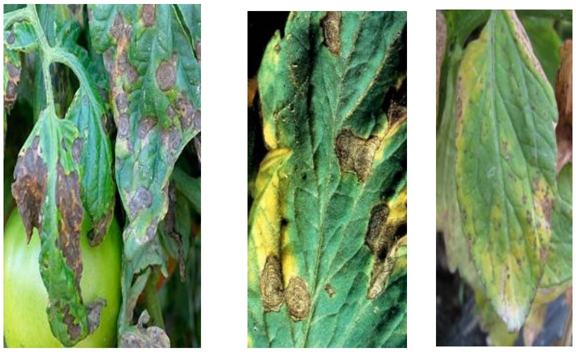
A. Proposed First Phase DLPDS

The sequential method is used for the first phase of layer creation. This is due to the fact that building the model in layers is easiest using the sequential approach. To implement Softmax's input shape of (28, 28, 1), we utilise the add () function with the parameters conv2D, kernel _size, activation 'Elu,' and model layers. These are the fundamental building blocks of a convolutional neural network.

This procedure, which starts with the input picture and continues through the Conv2D layers and the flatten layers [18] connects the convolution and the other dense layers. Machine learning (ML) algorithms have become widely employed as artificial intelligence (AI) has advanced because ML succeeded in achieving emerges and cost-effective solutions to exploration of harvest yield [19]. The best result, which might be any of the values 0 through nine, is selected by setting the node in output to ten.

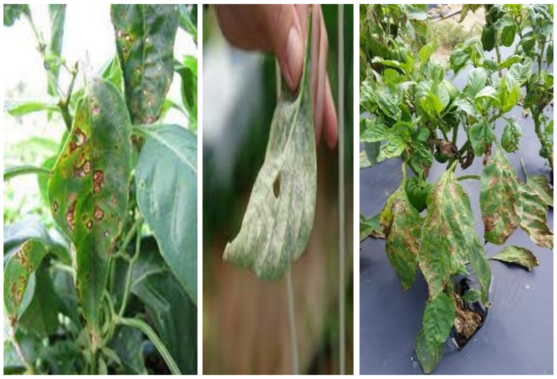
Each value is given a probability according to the Softmax activation, and the highest probability value is used as the forecast to determine whether the model needs more training to improve.

The agriculture sector makes use of validity measures to assess the performance of the disease segmentation technique for leaf blight. Blight, which affects tomatoes, is caused by the proteobacterium Xanthomonas Axonopodisis [20]. There are few illnesses as damaging as this one. Similarly, if a pepper plant is infected with blight, its yield might drop by 27.57.36 per cent. Fig. 2 depict a tomato leaf that has been infected with the blight disease.



(a) (b) (c)

Fig. 2. Tomato leaf image.



(a) (b) (c)

Fig. 3. Pepper leaf images.

Fig. 3 shows the pepper leaf images of the proposed work. Pepper leaf pictures, as well as unaffected input tomato, are taken into consideration for the input image, which is indicated by the letter I, and the impacted area of interest (ROI) is written as follows,

$$I = \sum n_{x=1} S_x \tag{1}$$

In Eq. (1), n is represented as number of regions in input image, n number of regions are represented in terms of $S_1 + S_2 + \cdots \ldots \ldots .Sn.$

Let us take two different regions then S_i not equal to Sj for alli $\neq$ j. The name for this kind of property is a disjoint property. A method for segmenting the intensity of histograms based on indices has been developed as a means of enhancing the segmentation and classification results produced by the aforementioned technique. Fig. 4 shows the proposed work block diagram.

1) Denoising in input image: This section explains the process of noise removal, since the provided input may have a possibility of having undesirable signals, which are referred to as noise. Denoising is a method that removes undesired signals from a picture while still preserving important information. Denoising is an essential step in the preprocessing of the picture, as it helps increase the accuracy of the final product. The median filter was used to eliminate the sounds. The following is a representation of the three-by-three matrix that serves as the input to the median filter procedure. Fig. 5 shows the pixel selection and analysis of proposed work.

Let M be the input matrix, and then apply the sorting procedure to M in order to sort all of its values, and finally, get the median of M. In the previous illustration, the picture was smoothed down when the values of the surrounding neighborhoods were replaced with the median value of 214. After that, an adjustment was made to the image's contrast. The procedure described above is used to the process of pixel representation for the input picture in order to remove noise from the image and get the specific area of the leaf image that was damaged.

It can be seen in Fig. 6, that the noise has been reduced from the leaf photos. The median filter is applied to the photos of the three distinct illnesses, including foiled, rot, and rust, that have impacted the leaves.

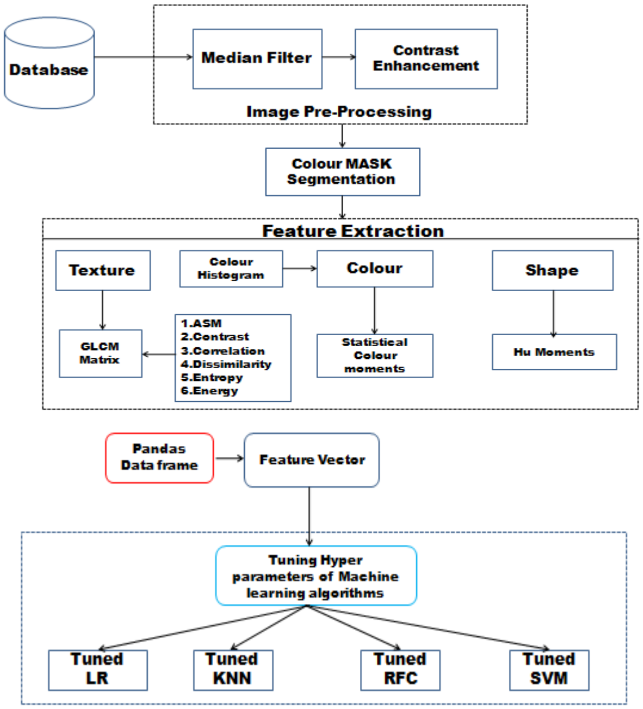
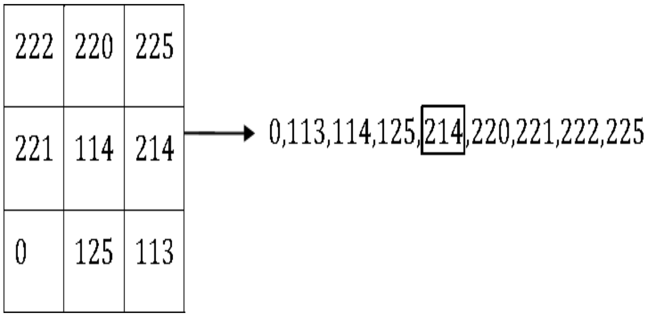


Fig. 4. Proposed first phase DLPD.



222	220	225
221	114	214
0	125	113

→ 0,113,114,125,$\boxed{214}$,220,221,222,225

Fig. 5. Pixel selection.

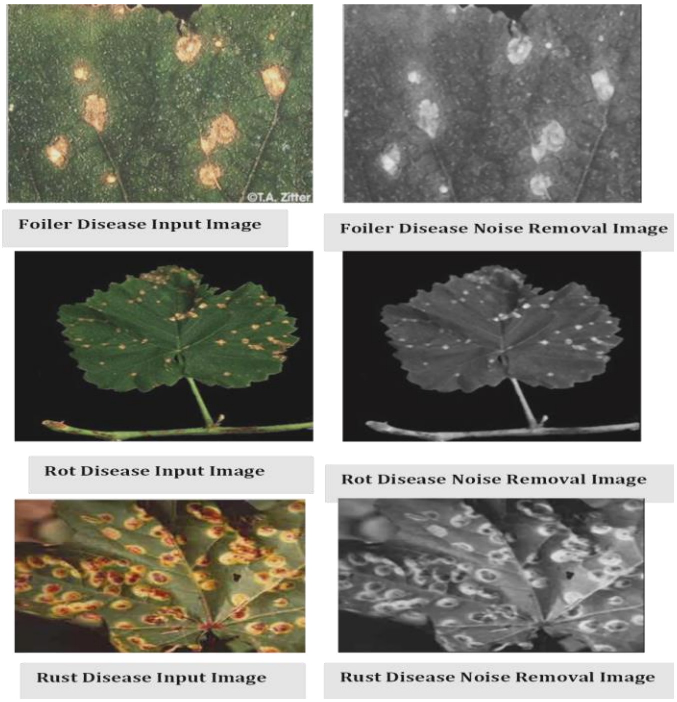

©T.A. Zitter
Foiler Disease Input Image
Foiler Disease Noise Removal Image
Rot Disease Input Image
Rot Disease Noise Removal Image
Rust Disease Input Image
Rust Disease Noise Removal Image

Fig. 6. Pre-processed leaf images.

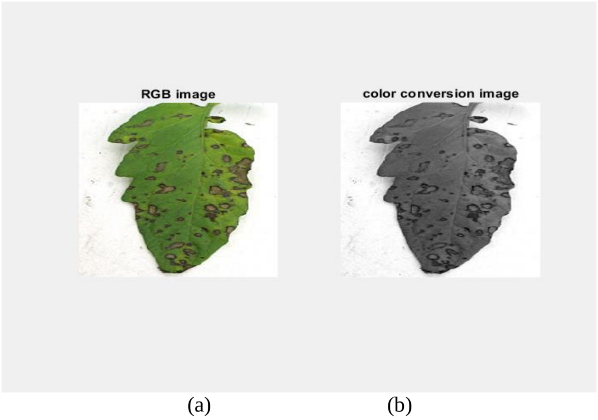


Fig. 7. (a). The original version, (b) Color conversion image.

Fig. 7 shows the original and color conversion results of provided dataset.

B. HSV Color Space Segmentation

The use of the Thresholding approach, the ROI may be extracted from the picture. Determine a value for the threshold, denoted by "T," depending on the highest and lowest points of the histogram. It is possible to employ the local threshold method, the global threshold technique, or the optimum threshold technique depending on the application. The area that is of interest is known as the ROI, and it may either be a portion of the picture or the whole image. It depends on the application, but often it involves combining the OTSU threshold approach with the canny edge detection operator and dividing the jujube leaf into a number of regions. The function f is used to depict the picture of a Pepper leaf after it has been translated into a digital format using MATLAB (x, y). By selecting the picture's threshold value, which is symbolized by the letter T, you can see that the image has been segmented. If the pixel intensity value is more than T, then it assigns value as 1; else, it takes the value as 0 for the pixel.

The formula for thresholding the leaf disease on Pepper is represented as follows, as in

$$\text{Threshold } = \begin{cases} 1 & \text{if } f(x, y) > T \\ 0 & \text{if } f(x, y) < T \end{cases} \quad (2)$$

Choose the T value but the result varies based on the selected domain. Same T value could not give accurate ROI for all kinds of input image. Implementation of the image segmentation technique used to separate the disease affected part from the background image. In this scenario first choose the seed point from the image. By using the seed point separate the diseased part. Choosing seed point is one of the biggest challenges in region growing method and wrong selection of seed point may lead to over segmentation. The study [21] implemented region growing method in cucumber downy mildew disease and get more accurate segmentation result compared with Otsu and K-means algorithm. Foliar disease of the leaf can be detected by using proposed region growing method and produced better result compared with existing algorithms. In this paper new algorithm is introduced based on the indices of histogram. During the analyze process, Pepper leaf images taken as input with the size of 256×256 as shown in Fig. 8.

Table II shows the feature metrics of the images and their values.

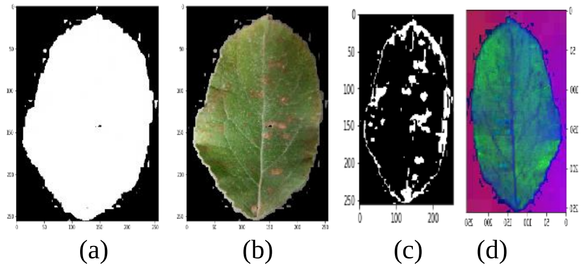


Fig. 8. (a) Graph cut algorithm (b) The original version, (c) Otsu thresholding, (d) HSV Color space segmentation.

TABLE II. COLOR MASKS FOR TOMATO, ONION, PEPPER LEAVES

S.no	Crop	Upper Green	Lower Green	Upper Brown	Lower Brown	Upper Yellow	Lower Yellow
1	Tomato	[86,255,255]	[36, 0, 0]	[30, 255, 200]	[8, 60, 20]	[40, 255, 255]	[21, 39, 64]
2	Onion	[86,255,255]	[36,0,0]	[30, 255, 200]	[8, 60, 20]	[255, 255, 255]	[3,3,3]
3	Pepper	[86,255,255]	[36,0,0]	[30, 255, 200]	[8, 60, 20]	[255, 255, 255]	[21,30,5]

C. Feature Extraction

1) Color: Extracting in all plant disease detection systems, color characteristics are essential since ill regions have different colors than healthy leaf photos; in this work, color features are recovered by measuring the color histogram and statistical color moments. The features of the HSV color histogram are extracted in the following stages: To begin, the cv2.COLOR RGB2HSV function converts the input image from the RGB color space model to the HSV color space model. Quantizing the HSV color model has the effect of lowering both the cost of computation and the size of the feature. At this part of the process, each channel's H, S, and V components are quantized using eight different bins. The frequency distribution of quantized HSV values for each pixel in an image is shown by a histogram, which is created for each quantized picture. This histogram is specific to the image in question. In addition, statistical color moments are generated by first isolating the R, G, and B components of the input image, then computing the mean and standard deviation for each channel separately. This process is repeated for all three channels. In conclusion, color moments provide a feature vector that is six-dimensional.

2) Shape: An essential component of the quantification process for image objects is the extraction of shape information. In this work, form attributes are determined by calculating shape moments and locating shape edges. For the purpose of computing Hu moments, the function Hu moments found inside the Open – CV Python module is utilized. Converting the RGB image to grayscale is the first step that must be taken before attempting to calculate the Hu moments, which need just a single channel. Calculating the first 124 moments of the original picture is the responsibility of the CV2.moment module. After that, the real moments are input into the CV2 method, and the first six Hu moments are generated using the results of that. In order to get a shape feature vector, the Hu moments method and the array flattening method are used.

3) Texture: There are four distinct varieties of texture measurements that may be used in image processing, and they are structural, statistical, model-based, and transform-based. Statistical approaches may be used in this situation since the size of the texture is about equivalent to the size of the pixels. For instance, GLCM is an example of a statistical method that may be used to measure the textural features of an image, which can then result in various shades of grey. In order to generate the grey level co-occurrence matrix, fourteen Hara lick texture characteristics, such as ASM, Correlation, Contrast, Entropy, Homogeneity, and Dissimilarity, in addition to eight additional features, were extracted. This process was carried out. This results in a thirteen-dimensional feature vector, with the fourteenth feature being omitted due to the substantial increase in the amount of processing it requires. Using manually chosen feature descriptors, the shape, color, and texture characteristics of the panda's data frame are concatenated together to generate a 532-dimensional feature vector.

D. ML Classifier Tuning

The numerous layers that make up the CNN are capable of performing a wide variety of tasks, some of which include convolution, max pooling, activation, and a fully connected network (Fig. 9). Every CNN input image that is sent on to the Convolution layers includes filters, receptive fields, stride, padding, pooling, and ReLu stacks in some form or another. The receptive field of a CNN is the part of the network that monitors the activity of any filter that responds to each individual pixel. As further stack layers are added to the convolutional layers, it behaves in a linear manner throughout the process.

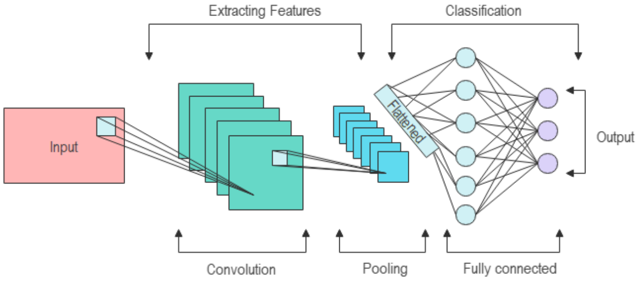


Fig. 9. Architecture of CNN.

When there is a negative number, the activation level that follows after it clips off to zero. This happens because negative numbers are less than zero. While the ReLu activation function converges more rapidly than the sigmoid activation function, it is also saturated in the negative region, which causes the gradient to be zero. This is because the ReLu activation function is a Convolutional Neural Network.

The very next pooling layer happens in between the convolutional layers, and as a consequence, it reduces the amount of data sampled by the network and gets rid of the other value data that is not relevant. After passing through the pooling layer, the dataset is shrunk down to a more manageable size, and the process of sending the dataset through the pooling layer is repeated until the desired output is produced. The matrix size will be decreased from 4x4 to 3x3 as a direct consequence of the maximum pooling layer, and then it will be decreased even more from there. The next layer is likewise fully connected, and it comprises neurons that have a full connection to all of the activations that were produced using a matrix multiplication. This layer is referred to as "totally linked."

Inside of a CNN network with several layers, the image is first warped and then converted into pixels. Following that, an activation of a neuron is performed for each location, and the results are compiled inside of the feature map. If the receptive field is moved one pixel further away from the activation layer, then the field plane will overlap with the previous activation by an amount that is equal to the field plane width minus one input value. This will occur if the receptive field is moved. After this, the fully connected layers will proceed to categories the many classes that have been instructed as binary values.

The supervised machine learning algorithms L.R., KNN, SVM, and R.F. are used in the initial step of the DLPDS process. In most cases, machine learning models are made up of two parameters: the default Model Parameter, and the hyperparameter.

Algorithm 1: Training and Testing phase

Step 1: Initialize the bias and weights to a random value.
Step 2: Input the training input vector and its targets.
Step 3: Compute the output of the hidden layer. The net input of the hidden layer is computed as follows,

$$y_k = \Sigma w_{ik} x_i + b_k$$

where b_k is the bias of the hidden layer, W_{ik} is the weights between the input and the hidden node and x_i is the input vector.
The output of the hidden layer is given by,

$$\mathrm{P_k} = \mathrm{f}(y_\mathrm{k})$$

where $f(0)$ is the activation function of the neuron.
Step 4: Compute the output of the output layer. The net input of the output layer is computed as follows,

$$z_1 = \sum w_{jk} p_k + b_i$$

where b_1 is the bias of the output layer, w_{jk} is the weights between the hidden and the output node and p_k is the output vector from the hidden neuron.
The output of the output layer is given by.

$$y_1 = \mathrm{f}(\mathrm{Z_j})$$

where $f(0)$ is the activation function of the neuron.
Step 5: Compute the error at the output layer and at the hidden layer.
The error at the output layer is given by,

$$E_j = (t_j - y_j) f'(z_j)$$

The error at the hidden layer is given by.

$$E_k = \sum \delta_j w_{jk} f'(y_k)$$

Step 6: Update and renew the weights and bias with the learning rate α for output layer which is given by,

$$\Delta \mathrm{w}_{jk} = a \mathrm{E}_j \mathrm{x}_i$$
$$\Delta \mathrm{b}_k = -a \mathrm{E}_k$$
$$\mathrm{w_k}(\text{ new }) = \mathrm{w}_{ik} + \Delta \mathrm{w}_{ik}$$
$$\mathrm{b_k}(\text{ new }) = \mathrm{b}_j + \Delta \mathrm{b}_k$$

Update and renew the weights and bias with the learning rate $\boldsymbol{\alpha}$ for hidden which is given by,

$$\Delta w_{ik} = a E_k y_k$$
$$\Delta b_{kk} = -\alpha E_j$$
$$\Delta w_{jk}(\text{ new}) = w_{jk} + \Delta w_{jk}$$
$$b_j(\text{ new }) = b_j + \Delta b_j$$

Step 7: Repeat step 2 to step 6 until the stopping criteria is reached.

5.2.2 Testing Process
Step 1: Feed the unknown data vector X
Step 2: Compute the output of the hidden layer. The net input of the hidden layer is computed as follows,

$$y_k = \Sigma w_{ik} x_i + b_k$$

The output of the hidden layer is given by,

$$\mathrm{P_k} = \mathrm{f}(y_\mathrm{k})$$

Step 3: Compute the output of the output layer. The net input of the output layer is computed as follows.

$$Z_1 = \Sigma w_{jk} P_k + b_i$$

The output of the output layer is given by.

$$y_j = f(Z_j)$$

where f(0) is the activation function of the neuron.

Step 4: Obtain the classification result from the output neuron in the output layer.

The practitioner may modify the hyperparameters to improve the classification results. Grid search and Manual search are used in this study for the Tomato, Tomato, and Pepper datasets because they provide results more quickly than other techniques. Fig. 10 shows the proposed methodology block diagram stage 2.

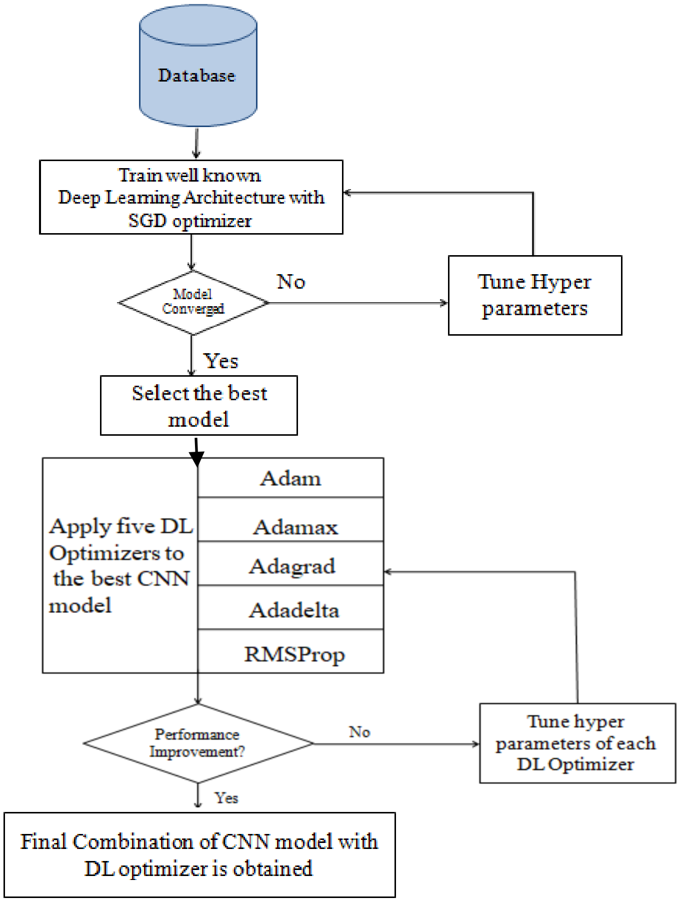


Fig. 10. Proposed second phase DLPDS.

E. Deep Learning Optimizers

First, the SGD optimizer is put to use in order to train the previously stated well-known architectures on the basis of validation accuracy, after which the best CNN model is selected. In conclusion, the Six DL Optimizers, whose features are outlined in Table III, are fine-tuned in order to improve the classification accuracy.

TABLE III. PARAMETERS OF DEEP LEARNING

gamma	alpha	value	Accuracy %		
			Tomato	Pepper	Onion
Scale	0.1-0.3	alpha	0.803333	0.9862	0.866667
		alpha	0.783667	0.9921	0.838333
		alpha	0.711667	0.9752	0.838333
	0.4-0.6	alpha	0.832667	0.9923	0.821667
		alpha	0.809667	0.9775	0.729
		alpha	0.741333	0.9921	0.884667
	0.7-1.0	alpha	0.881333	0.9712	0.871333
		alpha	0.866667	0.9923	0.729
		alpha	0.803333	0.9862	0.866667

IV. EXPERIMENTAL ANALYSIS

For the purpose of this study, ten thousand photographs spanning fourteen distinct categories were taken in the area of agriculture and sourced from the EPPO Global Database, both of which are open to the public. In addition, there are eight other categories that make up the Classification of Tomato plant diseases. In this study, we utilize datasets including both pepper and tomato. In both the pepper and tomato, onion datasets, the pictures are divided into training and testing sets in an 80:20 arrangements. There are a total of 5932 photos in the pepper and tomato collection shown in Table IV.

A. Tuning ML Classifiers

1) LR Tuning: C value [100, 10, 1.0, and 0.1], solver ['newton-cg,' 'lbfgs,' and 'liblinear,'] and penalty ['l2'] are the most important parameters that are extracted from the default Python specification to tailor the performance of LR. The settings shown above have been applied to these parameters in order to modify the Tomato, Pepper, and Tomato characteristics shown in Table IV.

TABLE IV. LR-TUNING RESULTS

Sigma Value	Penalty	Solver	Accuracy %		
			Tomato	Pepper	Onion
100	elasticnet	saga	0.936333	0.937667	0.938667
		sag	0.936333	0.939000	0.937333
		liblinear	0.945333	0.929000	0.974667
10		lbfgs	0.874561	0.894561	0.925333
		newton-cg	0.861234	0.929000	0.925333
		lbfgs	0.797531	0.974333	0.895333
1.0		liblinear	0.971667	0.945333	0.970667
		sag	0.974000	0.974333	0.974000

2) SVM Tuning: The kernels- ['linear', 'poly', and 'rbf'] parameter, the C value- [10, 1.0, and 0.1] parameter, and the gamma-scale parameter are the ones that were chosen from the default Python SVM classifier specification. Table V demonstrates that an SVM with a C value of 1.0 and a kernel of poly achieves an accuracy of 88.13% for tomato features, 97.12% for Pepper features, and 87.13% for onion leaf features.

Table V shows the existing tuning values of the SVM classifier.

TABLE V. SVM-TUNING RESULTS

ML classifier	Para meters	Tomato								
		Healthy	Early blight	Leaf mold	Septoria leaf spot	Target spot	Bacterial blight	Late blight	Spider millets	Accuracy %
TLR	P	0.96	0.79	0.84	0.80	0.81	0.94	0.81	0.89	85.50
	R	0.95	0.81	0.91	0.75	0.88	0.91	0.75	0.88	
	F1	0.96	0.80	0.88	0.77	0.84	0.93	0.78	0.88	
TSVM	P	0.92	0.82	0.88	0.81	0.79	0.93	0.82	0.87	85.62
	R	0.92	0.81	0.94	0.76	0.86	0.92	0.81	0.83	
	F1	0.92	0.81	0.91	0.79	0.82	0.93	0.82	0.85	
TKNN	P	0.92	0.80	0.82	0.84	0.72	0.93	0.90	0.73	82.87
	R	0.86	0.74	0.95	0.74	0.79	0.84	0.63	0.82	
	F1	0.85	0.75	0.85	0.76	0.74	0.88	0.72	0.80	
MCNN	P	0.97	0.85	0.94	0.92	0.77	0.97	0.87	0.92	90.12
	R	0.98	0.88	0.96	0.88	0.90	0.93	0.80	0.88	
	F1	0.98	0.86	0.95	0.90	0.82	0.94	0.84	0.90	

3) KNN Tuning: The parameters n neighbors [3, 4], metric ['Euclidean', 'Manhattan'], and weights ['uniform,' 'distance'] were chosen from the KNN default python specification. According to Table VI, using n neighbors =4 with metric = Manhattan, weights = Distance results in an accuracy of 82.75% for tomato, 90.12% for tomato, while using n neighbors =3 with Euclidean metric and distance as weight results in an accuracy of 99.21% for Pepper leaf characteristics.

4) RF Tuning: When it comes to tuning, the parameters max features ['sqrt', 'log2'], and n estimators [10, 100, and 1000] were chosen from the RF default Python specification. According to Table VII, using max features = sqrt and n estimators =1000 results in an accuracy of 95% for tomato and 99.50% for Pepper. Yet, using the same max features and the same number of estimators results in an accuracy of 90.12%.

TABLE VI. KNN-TUNING RESULTS

Max-Features	metric	weights	Accuracy %		
			Tomato	Pepper	Onion
10	sqrt	estimators	0.7975	0.9	0.9862
		estimators	0.81	0.891	0.9921
	log2	estimators	0.8062	0.9051	0.9752
		estimators	0.82	0.8937	0.9923
100	sqrt	estimators	0.7762	0.8811	0.9775
		estimators	0.8087	0.8957	0.9921
	log2	estimators	0.8025	0.8912	0.9712
		estimators	0.8275	0.9012	0.9923

B. First Phase DLPDS with Tuned Models

1) Tomato diseases: For the purpose of categorization, a total of 4000 different photos of tomato leaves are taken, 500 of which are healthy and 3500 of which display one of seven different disorders. For the objectives of experimentation, the 50:50 technique has been selected. The training and testing photos consist of 250 healthy and 1750 sick images.

TABLE VII. RF-TUNING RESULTS

max-features	estimators	Accuracy %		
		Tomato	Pepper	Onion
sqrt	10	82.50	91.00	99.12
	100	90.12	93.00	99.21
	1000	89.50	95.00	99.50
log2	10	82.00	91.00	98.96
	100	88.00	92.23	99.34
	1000	88.62	93.50	99.37

SHAPE and TEXTURE are put to use for the purpose of distinguishing sick samples and healthy samples using 10-fold cross-validation (a). According to Table VIII, the Tuned Random Forest Classifier has an overall accuracy of 90.12%. Higher values of performance measures include Precision equal to 0.97 for both healthy tomatoes and tomatoes with bacterial blight, recall equal to 0.98 and 0.96, and F1 score equal to 0.98 and 0.95 for both healthy tomatoes and healthy tomatoes with leaf mold shown in Table VIII.

TABLE VIII. PERFORMANCE METRICS BASED ON ACCURACY

Sigma Value	Penalty	Solver	Accuracy %		
			Tomato	Pepper	Onion
100	elasticnet	saga	0.936333	0.937667	0.938667
		sag	0.936333	0.939000	0.937333
		liblinear	0.945333	0.929000	0.974667
10		lbfgs	0.874561	0.894561	0.925333
		newton-cg	0.861234	0.929000	0.925333
		lbfgs	0.797531	0.974333	0.895333
1.0		liblinear	0.971667	0.945333	0.970667
		sag	0.974000	0.974333	0.974000

Moreover, there are two classifications that are used for categorizing tomato leaf diseases, and each class has a selection of one thousand photos. The majority of the illnesses that might affect tomato seem to be bacterial infections, according to the visual examination. Hence, any and all photographs of a sick tomato plant are saved under the heading "tomato diseases," whereas any and all images of a healthy tomato plant are saved under the heading "healthy tomato" shown in Table IX.

The tomato train and the test dataset are both developed by the use of the 50:50 approach. Validation of the tomato test picture dataset was accomplished with the help of the tuned four ML models. MCNN delivers 95% overall accuracy with Precision= 0.96, F1 score= 0.95 for bacterial illness characteristics, and Recall =0.96 for Healthy leaf features. The experimental findings are produced by ten-fold cross-validation, as shown in Fig. 11(b), and from Table IX, MCNN gives 95% overall accuracy.

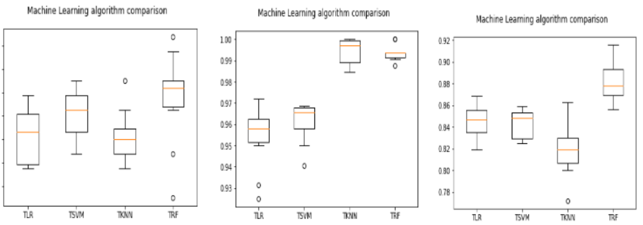
TABLE IX. PERFORMANCE METRICS OF TOMATO LEAVES

ML classifier	Parameters	Tomato		
		Bacterial Disease	Healthy	Accuracy %
TLR	P	0.90	0.87	88.50
	R	0.86	0.91	
	F1	0.88	0.89	
TSVM	P	0.95	0.91	92.50
	R	0.90	0.95	
	F1	0.92	0.93	
TKNN	P	0.94	0.87	90.50
	R	0.86	0.95	
	F1	0.90	0.91	
TRF	P	0.96	0.94	95
	R	0.94	0.96	
	F1	0.95	0.95	

2) Pepper diseases: There are four categories that make up the Pepper leaf disease categorization system, and each category has a thousand pictures. A total of four thousand photographs are amassed, and then, applying the 50:50 technique, the picture datasets for the train and test are partitioned into healthy images, class 500 images, and test images, respectively. There are 1500 photos that fall under the heading of the sick image, which may be used for training and testing purposes. The test picture dataset was verified by using the extracted characteristics of Pepper leaves as well as the results of ten-fold cross-validation, which are shown in Fig. 11(c). Table X shows that the majority of the Tuned models got improved classification results, with TKNN (Tuned K Nearest Neighbor) reaching an accuracy of 99.25% and MCNN achieving a greater accuracy of 99.50.

TABLE X. PERFORMANCE METRICS OF PEPPER LEAVES

ML classifier	Parameters	Pepper				
		Healthy	Cercospora_leaf_spot	Common rust	Northern_Leaf_Blight	Accuracy %
TLR	P	1.00	0.95	1.00	0.90	96.25
	R	1.00	0.90	1.00	0.96	
	F1	1.00	0.93	1.00	0.93	
TSVM	P	1.00	0.97	1.00	0.92	97.12
	R	1.00	0.92	1.00	0.97	
	F1	1.00	0.94	1.00	0.94	
KNN	P	1.00	0.97	1.00	1.00	99.25
	R	1.00	1.00	1.00	0.97	
	F1	1.00	0.99	1.00	0.98	
TRF	P	1.00	0.99	1.00	0.99	99.50
	R	1.00	0.99	1.00	0.99	
	F1	1.00	0.99	1.00	0.99	



(a) (b) (c)

Fig. 11. Ten- fold cross-validation results of (a) Onion, (b) Tomato, (c) Pepper.

C. Second Phase DLPDS

The Tomato, Tomato, and Pepper leaf pictures were used to train the chosen DL Architectures and the experimental findings were represented by validation, training accuracy/loss, Precision, F1 Score, and Recall. The model or optimizer that achieves the highest possible Validation accuracy and F1 score is taken into consideration to be the best option for the Second Phase DLPDS that is being suggested. The accuracy and loss measurements in the training and validation stages need a combined total of 20 epochs in order to converge.

1) Performance of pretrained models: Accuracy, sensitivity, selectivity, kappa coefficient, and mean square error are some of the several performance metrics that are taken into consideration. The following is a representation of the notations that are used in the computation of the metrics: TP - true positive, TN - true negative, FN - false negative, TP - positive, TN - negative, FN - false negative. The following is the calculation for the performance metrics:

- Accuracy: It is the measure of how well the classifier correctly identifies whether the leaf is healthy or diseased.

$$\text{Accuracy } = \frac{\text{True Positive+True Negative}}{\text{Positive+Negative}} \quad (3)$$

- Sensitivity: It is also known as recall and it represent the measure of proposition of diseased leaf correctly identified as such.

$$\text{Sensitivity} = \frac{\text{True Positive}}{\text{True Positive}+False\ Negative} \quad (4)$$

- Specificity: It is also termed by True Negative Rate, and it represents the proportion of healthy leaf correctly identified.

$$\text{Specificity} = \frac{True \text{ Negative}}{\text{False } Positive\ +True \text{ Negative}} \quad (5)$$

- Kappa Coefficient: The kappa coefficient is used to determine how closely the original values and the graded values are related to one another. A kappa score of one shows that all respondents are in complete agreement, while a value of 0 indicates that there is no consensus. Here is how it is figured out,

$$\text{Kappa } = \frac{z\sum_{j=1}^{n} m_{j,j} - \Sigma_{j=1}^{p}(G_j \mathrm{C}_j)}{\mathrm{z}^2 - \sum_{j=1}^{n}(\mathrm{G}_j \mathrm{C}_j)} \quad (6)$$

where j is the number of the class, Z is the total number of graded values that are compared to the original values, m (i,j) is the number of values belonging to the truth class j that have been classified as class j, Cj is the total number of expected values belonging to class j, and Gj is the total number of truth values belonging to class j. where j is the number of the class, Z is the total number of graded values that are compared

During the second phase, RT is used to pinpoint the precise location of the leaf illness as well as determine its degree of severity shown in Table XI.

TABLE XI. COMPARISON OF PROPOSED TRAINING FUNCTION AND TRAINBR

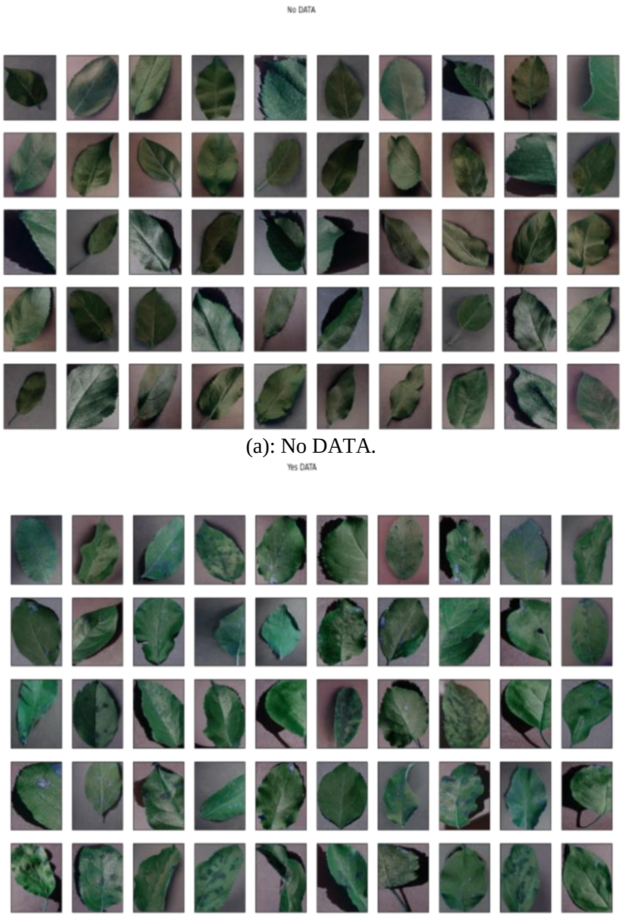
Sl.No	Training Function	MSE	Accuracy	Kappa Coefficient
1	Trainbr	0.03	93.4	0.892
2	Proposed Method	0.02	95.64	0.928

The categorized picture is then put through RT in order to pinpoint the area that is affected by the illness. If the picture is determined to be a disease, a morphological operation is performed on it, coupled with the determination of an appropriate threshold, in order to differentiate the sick area from the backdrop. The segmented output is created by applying a morphological opening operation with a square structural element to the identified picture. This results in the output being segmented. The RT is applied to the sick leaf that has been segmented. The results of the morphological operation are first segmented, and then the RT is applied to the results of that segmentation. The RT output provides evidence that the disease node is present in the leaf. The radon transform is useful for pinpointing exactly where the sick area is located in the body.

There is no indication of either underfitting or overfitting in the experimental findings of the pre-trained models using the Pepper picture dataset. From Fig. 12(b) Enhanced GoogleNet was able to achieve the highest possible level of Validation accuracy by making use of the idea behind the Inception module. The Improved GoogleNet and SGD optimizer, together with a comparative study of the several models shown in Table XII, led to the best possible F1 score of 0.993.

TABLE XII. PERFORMANCE OF PRE-TRAINED MODELS

Parameters	**MobileNetV2**	**Improved GoogleNet**
Training Accuracy	98.99	98.99
Validation Accuracy	97.73	99.92
Training Loss	0.220	0.021
Validation Loss	0.285	0.002
Precision	0.982	0.991
F1 Score	0.986	0.993



(a): No DATA.

(b) Yes DATA.

Fig. 12. Training and testing data.

TABLE XIII. ANALYSIS OF SEGMENTATION

Raw Input Image	K-Means Clustering	Fuzzy Logic	Region-Based Segmentation	HSV Color Segmentation
Tomato blight disease	0.684	0.82	0.8843	0.8368
Tomato Leaf Spot	0.66	0.856	0.801	0.8465
Tomato Powdery Mildew	0.68	0.861	0.8154	0.8624
Pepper blight disease	0.803	0.865	0.8266	0.868

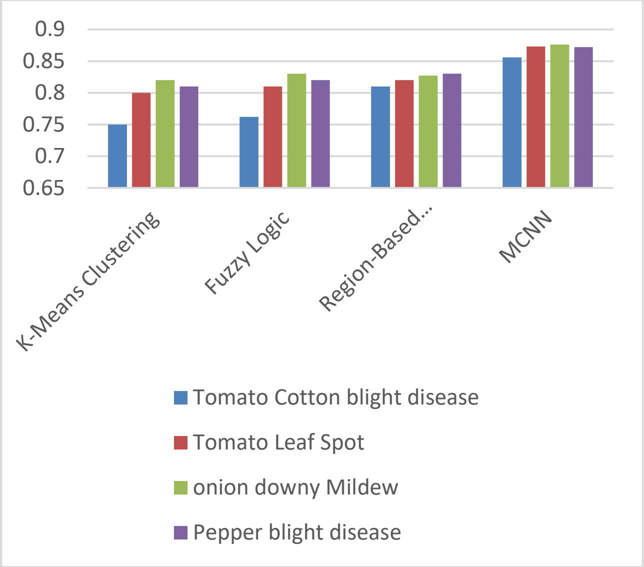


Fig. 13. DSC Performance of segmentation techniques.

Fig. 13 makes it abundantly evident that the indices-based intensity histogram segmentation technique assures a somewhat superior performance value for various plant leaves, such as those affected by tomato blight disease, tomato spot, tomato powdery mildew, and Pepper blight disease, respectively. The divided components each provide important information that may be utilized to examine the characteristics and facts connected to the condition. Table XIII presents the results of the mutual information calculation for the segmented area.

TABLE XIV. MUTUAL INFORMATION FOR SEGMENTATION RESULT

Raw Input Image	K-Means Clustering	Fuzzy Logic	Region-Based Segmentation	MCNN
Tomato blight disease	0.67	0.75	0.83	0.85
Tomato Leaf Spot	0.68	0.78	0.86	0.86
Tomato Powdery Mildew	0.73	0.80	0.88	0.868
Pepper blight disease	0.76	0.83	0.82	0.878

According to what is shown in Table XIV. It is abundantly obvious that the indices that are based on the intensity of the histogram include various information about impacted diseases in an efficient way. Indices Based Intensity Histogram Segmentation approach segment region contain significant information because it assures the largest mutual information value. This can be understood since the technique for indices-based intensity histogram segmentation segments the histogram based on its intensity. Fig. XIV is a graphical depiction of the linked data for the mutual information value, and it was created using Table XIV.

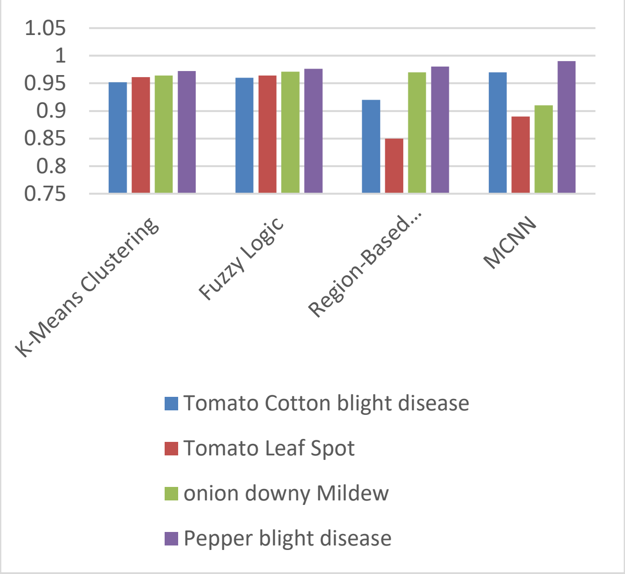


Fig. 14. Mutual information for different segmentation methods.

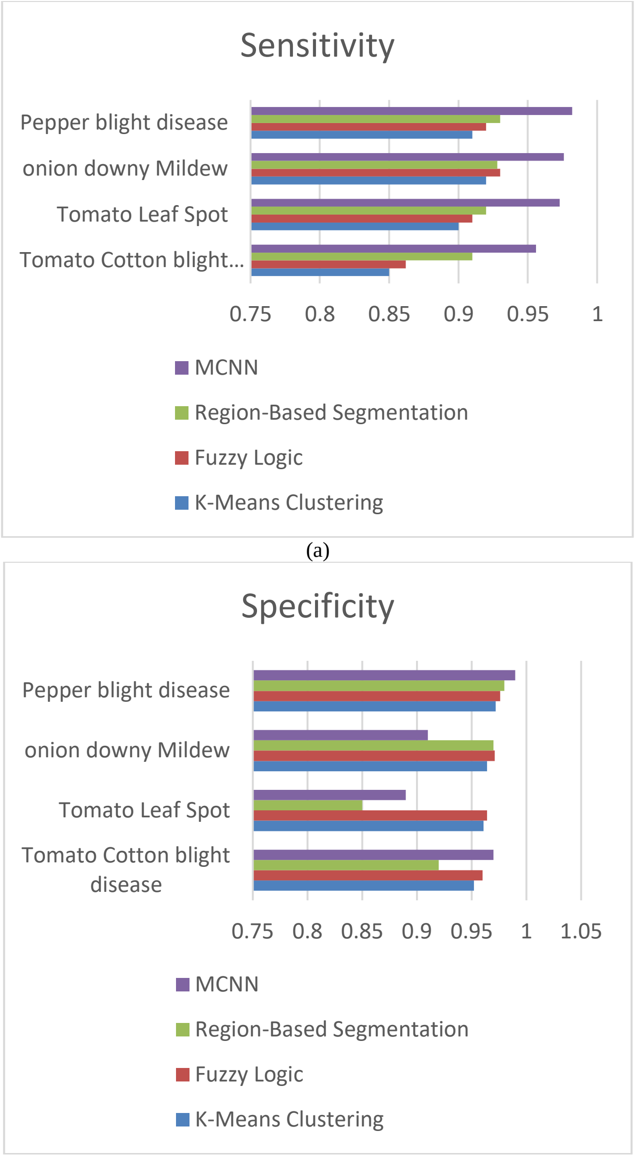
According to Fig. XIV, a technique to segmenting intensity histograms that is based on indices assures a high mutual information value for various plant leaves, such as those affected by tomato blight disease, tomato spot, tomato powdery mildew, and Pepper blight disease. The divided components each provide important information that may be utilized to examine the characteristics and facts connected to the condition. Effective retrieval of disease-related information from the segmented area is accomplished, and the accuracy of the segmented region's diagnostic performance is evaluated by means of the sensitivity and specificity metrics that are shown in Table XIV.

Fig. 15(a) and (b) shows that using a strategy that segments intensity histograms based on indices assures a high sensitivity and specificity value for various plant leaves, such as those affected by tomato blight disease, tomato spot, tomato powdery mildew, and Pepper blight disease. The segmented sections include valuable information that can be used to assess the disease-related characteristics and information that can be used to obtain disease-related information with the maximum possible accuracy, as seen in Table XV.

The results shown in the preceding Table XV make it abundantly evident that the indices based on the intensity histogram segmented area include a high degree of accuracy (88.78%) about the afflicted illness portion in an efficient way.

2) Discussion: The experimental results show that the enhanced versions of GoogleNet and MobileNetV2 using SGD Optimizer with ImageNet weights achieve a significantly more acceptable degree of validation accuracy. These two models are compared using five different optimizers and the same amount of epochs in an effort to improve classification accuracy. Based on the data in Table XI, we may make the following inferences:

Training the pre-trained models with various optimizers led to notable improvements in validation, training accuracy/loss, F1 Score, precision, and recall.



(a)

(b)

Fig. 15. (a) Sensitivity and (b) Specificity.

TABLE XV. ACCURACY FOR SEGMENTATION RESULT

Raw Input Image	K-Means Clustering	Fuzzy Logic	Region-Based Segmentation	MCNN
Tomato blight disease	82.1	83.5	86.34	87.31
Tomato Leaf Spot	82.8	83.88	85.134	87.543
Tomato Powdery Mildew	83.1	84.21	86.43	88.46
Pepper blight disease	83.78	84.72	86.83	88.78

The best validation accuracy may be achieved with the use of optimizers such as Adam, SGD, and Adadelta.

- Enhanced GoogleNet with Adam Optimizer achieved a 99.95% success rate and an F1 score of 0.997, demonstrating the efficacy of the proposed tuning strategy. The results of this experiment are also significantly improved upon when compared to those of previous studies. Therefore, the given technique may be used to a wide range of plant diseases.
- The F1 score and validation accuracy for MobileNet were both improved by using the Adam and Adadelta optimizers.

However, it is possible to see a decrease in performance after switching from SGD to RMSProb and Adamax for improved GoogleNet and MobileNetV2. This is the case despite the fact that these algorithms are designed to improve performance.

V. Conclusion and Future Work

Plant disease and pest attack is a major threat to the farmers and the farming industries. It impacts the economy majorly by destroying the plants and production quality. An automatic plant disease detection system is a key factor in the growth and benefit of farm production. This research is carried out based on the image-based automatic disease detection system which includes various image processing and CNN techniques. The proposed methods are developed to enhance the detection system and classify diseases. Training, validation, and testing were conducted on various publicly available datasets. The datasets consist of images captured under different lighting conditions, resolution, position, and complex background to train the system with all possible complexities to avoid the misclassification rate. When compared to ML models, the F1 Score and Accuracy of the Enhanced GoogleNet and MobileNetV2 models were shown to be superior. The best results for differentiating between photos of diseased and healthy tomato leaves were achieved using the DL model and optimizer combination known as Improved GoogleNet with the Adam optimizer, which achieved a validation accuracy of 99.5% and an F1 score of 0.997%. Furthermore, two distinct DL architectures were tweaked with five distinct DL Optimizers to enhance classification accuracy. The future research in the automatic plant disease and pest detection method focuses mainly on increasing the efficiency of the system by reducing the computational time. It also further focuses on treatment and prevention methods based on the severity of the impact. The entire system should be implemented in real-time using mobile and web applications, where the test images can be taken using any device including mobile cameras, drone cameras, satellite images and sensor images. The real-time applications store and retrieve all the information based on cloud services. Which enables the farmers to operate the farming portal to detect the disease or pest attack based on the captured images and based on the impact and severity of the disease the portal provides a solution to the farmer by prescribing the procedure for the treatment and prevention techniques like cross farming, seedling selection, ground detection, canopy estimation, water level detection, phenotype, and genotype evaluation to avoid the further occurrence of the same disease or pest attack.

Acknowledgment

The author would like to thank Deanship of Scientific Research at Majmaah University for supporting this work under Project Number No. R-2023-536.

References

[1] A.S.Deshapande, S.G.Girardi, K.G.Karibasappa and S.D.Desai, “Fungal disease detection in Pepper leaves using Haar wavelet features, "Information and Communication Technology for Intelligent Systems,vol 106, pp.275-286, 2019.

[2] X.Zhang, Y.Qiao, F.Meng, C.Fan and M.Zhang, "Identification of Pepper leaf diseases using improved deep convolutional neural networks," IEEE Access, vol.6, pp.30370–30377,2018.

[3] K.Golhani, S.K.Balasundram,G.Vadamalai and B.Pradhan, "A review of neural networks in plant disease detection using hyperspectral data airborne imaging spectrometer for applications," Information processing in Agriculture, vol. 5,no3,pp. 354-371, 2018.

[4] J.G.ArnalBarbedo, "Plant disease identification from individual lesions and spots using deep learning,"Biosystems Engineering, vol.180, pp.96 - 107,2019.

[5] Z. Xihai, S. Yue Qiao, V. FanfengMeng, C. Chengguo Fan and A. Mingming Zhan, "Identification of Onion leaf diseases using improved deep convolutional neural networks,"IEEE Access,vol 6,pp. 30370 – 30377, 2018.

[6] P.Sethy, S.Gouda, N.Barpanda and A.Rath, "Detection of white ear-head of rice crop using image processing and machine learning techniques," Smart Computing Paradigms: New Progresses and Challenges, vol.1,pp. 87-95, 2020.

[7] P. Patrick Wspanialy and C. MedhatMoussa, "A detection and severity estimation system for generic diseases of tomato greenhouse plants," Computers and Electronics in Agriculture, vol.178, 105701, 2020.

[8] X.E.Pantazi, D.Moshou and A.A.Tamouridou,"Automated leaf disease detection in different crop species through image features analysis and One-Class Classifiers," Computers and Electronics in Agriculture, vol.156, pp.96–104,2019.

[9] J.Chen, H.Shao and C.Hu,“Image segmentation based on mathematical morphological operator," Colorimetry and Image Processing,doi:10.5772/intechopen.72603,2018.

[10] P.Xu, G.Wu, Y.Guo, X.Chen, H.Yang et al.,"Automatic wheat leaf rust detection and grading diagnosis via embedded image processing system," Procedia Computer Science, vol.107, pp.836–841,2017.

[11] A.Cruz, Y.Ampatzidis, R.Pierro and A.Panattoni,“Detection of grapevine yellows symptoms in vitis viniferal with artificial intelligence," Computers and Electronics in Agriculture, vol.157, pp.63–76,2018.

[12] E. Proposed worknjie Liu, P. Yongjun Zhang, B. Haisheng Fan, Z. Yongjie Zou and Q. YongbinQin, "Detection of late blight in potato leaves based on multifeature and svm classifier," Journal of Physics: Conference Series, vol.1518, no.012045,2020.

[13] E.Alehegn, “Ethiopian Onion diseases recognition and classification using support vector machine,”International Journal of Computational Vision and Robotics,vol. 9,no.1, pp.90 – 109. 2019.

[14] M.Sharif, M.Khan and M.Javed, "Detection and classification of citrus diseases in agriculture based on optimized proposed workighted segmentation and feature selection," Computers and Electronics in Agriculture,vol.150, pp.220–234,2018.

[15] S. Solemane Coulibalya, V. Bernard Kamsu-Foguema, C. DantoumaKamissokob and T.DaoudaTraore. “Deep neural networks with transfer learning in millet crop images,” Computers in Industry, vol.108, pp. 115–120,2019.

[16] K.Bashir, M.Rehman and M.Bari"Detection and classification of rice disease an automated approach using textural features,"Mehran University Research Journal of Engineering & Technology, vol. 38, no. 1, pp.239-250,2019.

[17] J.Pujari, R.Yakkundimath and A.Byadgi, "Classification of fungal disease symptoms affected on cereals using color texture features," International Journal of Signal Processing, Image Processing and Pattern Recognition,vol.6, no.6, pp.321-330, 2013.

[18] A.Das, R.Dutta, S.Das and S.Sengupta, “Feature selection using graph-based clustering for rice disease prediction," Computational Intelligence in Pattern Recognition,pp. 589–598,2020.

[19] Baljon, M.; Sharma, S.K. Rainfall Prediction Rate in Saudi Arabia Using Improved Machine Learning Techniques. Water 2023, 15, 826.

[20] A.Chatterjee, S.Roy and S.Das, "Feature selection using rough set theory from infected rice plant images, “Computational, pp. 417–427,2020

[21] R.Yadav, Y.Rana and S.Nagpal, "Plant leaf disease detection and classification using particle swarm optimization, “Machine, pp 294-306,2019.