E-FARM - A mobile application for predicting the plant diseases

Anita
Department of ETE,
RV College of Engineering
Bengaluru,India
anita.ldc20@rvce.edu.in

K.Viswavardhan Reddy
Dept. of ETE
RV College of Engineering
Bengaluru, India
viswavardhank@rvce.edu.in

***Abstract*— As you all know agriculture is major backbone of our country. However, due to the lack of proper observation of each leaf by the farmer with his/her physical presence, every time farmer is ending up with heavy losses. So, it is essential to build an automated system which can detect the diseases by reducing the load on the farmer, and thus helping in easy and efficient yield. So, in this paper we develop an automated system in detecting the plant leaf diseases using various machine learning (ML) algorithms such as K-nearest neighbors (KNN), support-vector machine (SVM), convolutional neural network (CNN) and visual geometry group (VGG)16. The plant leaf diseases dataset was called from Kaggle website with 2411 images of size 256*256.Later the images were denoised, where the denoising effects are resizing and removing the blur images. Furthermore, features are extraction by edge detection using Gabor filter. The dataset is divided into training and testing of 80-20 and 70-30 % respectively. Moreover, we also evaluated the performance of the above ML algorithms in-terms of accuracy, precision, recall and F1 score. From the results, it is observed that for a of 80-20% an accuracy of 39% for KNN, 76% for SVM, 98% for CNN model and 99% for VGG16 at the 10th epoch. With 70-30%, accuracy of 35% for KNN, 65% for SVM, 94.5% for CNN model and 96% for VGG 16 is observed. In both the cases, VGG16 has got highest accuracy at the 10th epoch when evaluated with confusion matrix. Finally, we also developed a mobile application by implementing VGG16 inside it, for the farmers to check the status of leaves on the go.**

Keywords— plant disease, object detection, machine learning, mobile application, Feature extraction

I. INTRODUCTION

In India farmers have a great diversity of crops. various pathogens are present in the environment which severely affect the crops and soil in which the plant is planted, there by affecting the production of crops, various disease is observed on the plants and crops. The main identification of the affected plant or crop are its leaves. The various coloured spotspatterns on the leaf are very useful in detecting the disease. The past situation for plant leaf disease detection taken direct eye observation, recall the particular set of disease as per the climate, season etc. India is a cultivated country and about 80% of the population depends upon on agriculture.[1] Farmers have large range of dissimilarity for selecting various acceptable crops and finding the suitable herbicides and pesticides for plant. Plant leaf disease leads to the reduction in both the quality and quantity of agricultural products. The diseased plant leaf Refer to the studies of visually observable patterns on the plants. Health of plant leaf and disease on plant leaf plays an important role in successful cultivation of crops in the farm

In[1]proposed that, a deep learning object identification model was used to create a smartphone application for identifying and categorizing plant diseases. To achieve reliable and effective detection, the suggested smartphone app combines a Faster R-CNN object detector with an Inception-v2 backbone network. Experiments on photos of grape disease showed that the suggested application may operate purely on a mobile without connected to a server and yet obtain an accuracy of 97.9%.A free software library of 36258 photos divided into 61 types of healthy and diseased plant leaves was used in the experiment. When [2] compared to the outcome of deploying a single CNN model, the overlaying method's prediction accuracy of 87 percent is a major improvement. This created deep learning-based image identification method. The[3] validation results demonstrate the Convolution Neural Program's feasibility with an accuracy of 94.6 percent and suggest a route for an artificial intelligence (AI) based Machine - learning resolution to this complicated situation. Proposed[4] The system developed LVQ algorithm is technique and CNN model for tomato leaf disease identification 500 photos of tomato leaves with four illness make up the dataset. For automated extracting features, we modelled a CNN. Its output features vector from the convolutional components as accuracy achieves accuracy as 97%. Proposed[5] Identification apples provides a new method for categorizing leaves using the CNN architecture and creates two models by modifying network depth with Google Net. based on colour or leaf damage, we evaluated the performance of each model. Even though 30 percent of total of the leaflets are damaged, the recognition rate is still over 94 of the totals.

Traditional image processing approaches utilize handcrafted features (e.g. colors, textures, shapes) extracted from training images and machine learning techniques (e.g. SVM, Neural Networks) to classify diseased leaf images. Although such approaches are simple and do not require large amount of training samples,[2] they only work well in controlled imaging environment and the limited expressive power of handcrafted features kept the model from further generalizing for real world applications [3]. The recent success of deep learning model, in particular, deep convolutional neural networks (CNNs) in object detection and recognition has attracted much attention in the development of robust plant disease recognition systems [4-7] and have reported remarkable performances. A comprehensive review of plant disease detection and classification using CNN models is given in [8]. Most of the existing plant disease classification methods classify an input image as either diseased or non-diseased, and assign a label for the type of disease for the diseased cases. Such approaches have several drawbacks. First, they do not identify the sizes and locations of the infected regions, which can be important information for locating the infectious areas and identifying the condition of the disease. Secondly, images taken from the field usually contain complex backgrounds which will affect the classification accuracy if the entire image is used as an input to the CNN network. And not giving highest accuracy as transfer learning

From the state of the art, we can observe that plant leaf detection used convolutional neural network, SVM and KNN algorithms which didn't give the required accuracy. Hence, in this, automated plant diseases prediction model is designed to classify the plant diseases. Steps involved in the work are: loading of input datasets, pre-processing the images by removing blur images, detecting the diseases' part etc. Further, the output of pre- processing stage is given as input to the feature extraction stage. Over here, we consider various features which helps in detecting the plant diseases and yielding in best accuracy. The whole work is based on the plant species such as apple, grapes, and tomato leaves. Also, an android application for the sake of farmers to check the status of the crops or plants is also developed. The work is carried keeping the requirements of small-scale farmer. The classifier and web application will need access to both smartphone and internet, which as previously said are still expanding to remote areas. Our proposed model is capable of scanning through thousands of leaf images in order to identify if a plant has a certain type of disease or not, based on its leaf image. The proposed model achieves a great accuracy of detection among 19 different disease classes in a short period of time and it doesn't require a long time in training either.

Section II deals with the system architecture of implementation and mobile application development. Also, steps involved in classification are discussed. Section III deals with implementation of the VGG model and mobile application, as well as algorithm. Results, discussion and comparison analysis of various ML algorithms were discussed in section IV. Section V deals with the conclusion and feature scope followed by references.

II. METHODOLOGY

A. *Data set*

The Data is collected from Kaggle It mainly consist of plant leaf images of healthy, and disease Infected leaves. The extracted input images and preprocessing and feature selection phase. The Benchmark. The dataset includes 1928 data samples for training and 483 for testing. The architecture used for the detection process as shown in table 1.

Table1:esign specification of plant disease module

Parameter	Value
Image Dataset (Plant village)	2411
Training set images	1928
Testing set images	483
Learning Rate	1e-06

First given the input (dataset) to the intermediate Representation model that is ML layers for the preprocessing and feature extraction and CNN model classification for training and CNN model and IR model that operate on mobile device in layer 1 and after the model we deploy in cloud server in and created android platform the user interface is depicted in Layer 2 and created as an android application to make it easy for system user (as shown in Layer 3) to interact with system as shown in figure 1

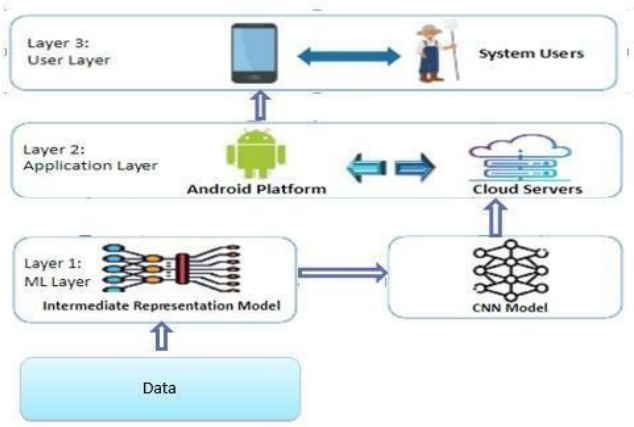

Figure 1: System architecture

By giving input images denoising the image and preprocessing by resizing all images 256*256 and feature is extracted by using the Gabor filter such as and classify the algorithms such as SVM, KNN, CNN and VGG 16 algorithm and give to the testing and training 80% and 20% display the accuracy as shown in Block diagram figure 2

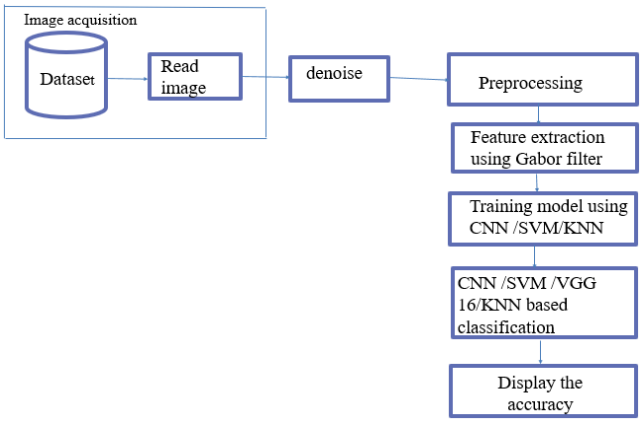

Figure 2: Block diagram of plant disease detection using different algorithm

Pre –Processing – The pre-processing the images includes noise elimination process and de-blurring process The leaf images is an RGB(red,green,blue)images with size of 256*256 pixels.The RGB image is an converted into gray scale then normalized is carried out using Z parameter as shown in figure 3

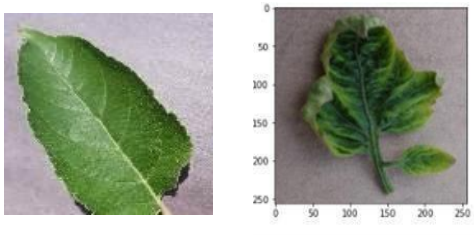

Figure 3: Resizing leaf

Feature extraction: In feature extraction process features like shape, colour, texture and edges are extracted that helps in

classification of the object followed by the detection step. texture -based feature extraction process for this first conversion process is caried out RGB to gray scale. In this the features are extraction method used is the Gabor filter. Gabor filter itself is a sinusoidal function modulated by the gaussian function. This method is often used to detect edges, lines and shapes as shown in figure 4

Figure 4: Feature extraction

Testing and Training: classifying the algorithms is given to 80% of training and 20% of testing the proposed plant disease predictor model achieved an accuracy. VGG16 model achieved better performance when evaluated with the confusion matrix of other models. finally build the application using VGG16 for the sake of farmers to check the status of the crops or plants

B. System architecture using Flask framework

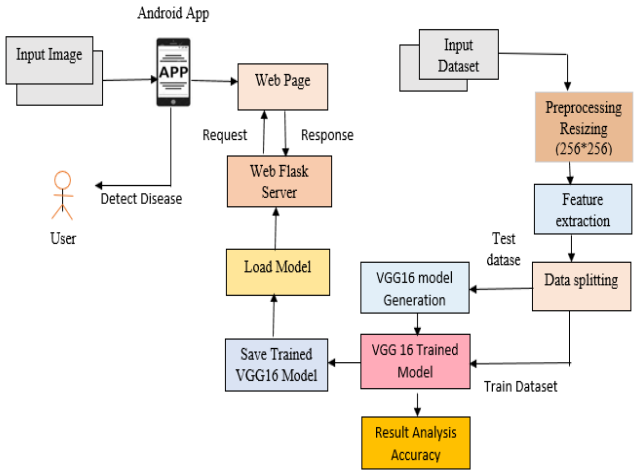

Figure 5 System architecture using Flask framework

First take the dataset and sent to the preprocessing and all the images are resized and then extracted the features and convert into the numerical form of conversion then give for data splitting and trained the model then classify the accuracy. And save the trained model and save the trained VGG 16 model and deploy that model to the web flask server default URL route as the first step in establishing the Flask server. Flask has GET and POST methods When we send a request to the server using the GET technique, we do not transfer any files. In addition to requesting the server's response, the POST method will transmit files to it also request the result our data will be used. When we first access the web, it will only fetch web pages using the GET technique. After that input an image before uploading it. Because we upload the data, it will process it using the POST method and determine which disease is present in the leaf image as shown in figure 5 when it's finished, a new page appears with descriptions of the leaf disease and it's from it then this URL id will take and design a mobile using eclipse when we first access the web, it will only fetch web pages using the GET technique. After that input an image before uploading it. Because we upload the data, it will process it using the POST method and determine which disease is present in the leaf image as When it's finished, a new page appears with descriptions of the leaf disease and it's from it then this URL id will take and design a mobile using eclipse

C. Flowchart

This is the Flow chart for VGG16 first data loading input image given to model then preprocessing the image and all size are converted into 256*256 and given test size as 0.2 for 80%training and 20% testing. and feature is extracted using Gabor filter and image classification given for the testing and training with the epocs of 10 and given the accuracy in Vgg 16 is 99% and CNN is 98% and, in the report, analysis given the all the parameter values like precision, recall, f1-ratio as individual and confusion matrix are displayed for different disease and graph are displayed with accuracy and validation loss and save the model in the form of h5 figure 6

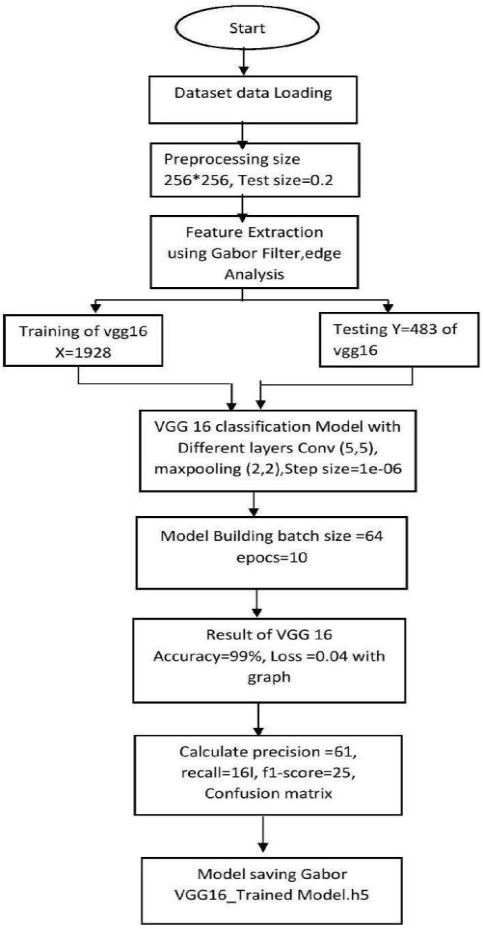

Figure 6: Flow chart of VGG 16

The plant disease detector GUI model is designed to perform comparative analysis of SVM model, KNN model, CNN model and VGG 16 model for plant village dataset. Tinter library is used to create the GUI and is present in all the standard python distribution the main screen consists of the

plant disease classification and prediction and provide the options to select input. When user selects input image, computation process carries down in the backend for all Four models and display the output screen with status and output data in terms of percentage

III. IMPLEMENTATION

A. *Pseudocode*

Algorithm: Pseudocode for Vgg 16 algorithm

Import os, cv2, NumPy, Matplotlib, pickle Libraries
If Data Loading
X=Data[x] Data normalization,
Y=data [y] labels
If Data Preprocessing
{
Data Spitting of X and Y Test and Train with test size=0.2, random state=42 Training data X = 1928 with labels Y= 483 and Test data X test. with size 256*256
}
Feature extraction
Data Visualization with Different disease using path using Gabor filter Dirlist and for range i, Plot show with fig size (12,8)
Model building
{
Model with different layers with activation=relu, input shape=256*256 as Conv (5,5), maxpooling (2,2), Dropout (0.25), dropout (0.25) Reduce the step size with decay=1e-06
Model. summary ()
Model building with batch size=64, epocs=10, validation data
History of accuracy and loss plot graphs
model prediction with batch size=32
}
Result analysis
{
Result Analysis CNN accuracy=99%, are obtained andprecision=100, recall=100, f1-score=100,
Confusion matrix of CNN with labels and
fig size= (15,8)
Model Saving in json with weight by .h5 form
GaborVGG 16_Trainedmodel.h5 as saved
}

B. *Mathematical modeling of Vgg 16:*

Activation Function: Used Two activation Functions were used for our model training where the SoftMax activation and the ReLU function. The ReLU function was used at the fully connected layers, where the ReLU or "Rectified Linear Unit" is one of the popular activation functions used in Neural Networks and specifically in Convolutional Neural Networks and is defined as in (1):

$$y=\max(0, x) \quad (1)$$

SoftMax activation function is used for the output layers and this activation function is a type of logistic regression that is able of normalizing the inputted vector to a new vector where its probability distribution is equal to 1 and is defined in (2)

$$\sigma(a_i) = \frac{e^z}{\sum_{j=1}^{k} e^{z_i}} \quad (2)$$

Loss Functions used in the machine learning domain, the cost functions tend to optimize the model in the training procedure and the aim of the training procedure is to minimize the loss function and the model obtained is better as much as we tend to minimize this and is defined in (3)

$$\text{Loss}=-\sum_{i=1}^{output\ size} y_i log\hat{y}i \quad (3)$$

C. *Implementation of Flask*

Steps to ML model deploy in Flask web server

Importing libraries
Load the machine learning model
Build functions to pre-process and to
predict the image Initialize the flask
object
Set the route and the function that returns
something to the user's browser
Run and test the API

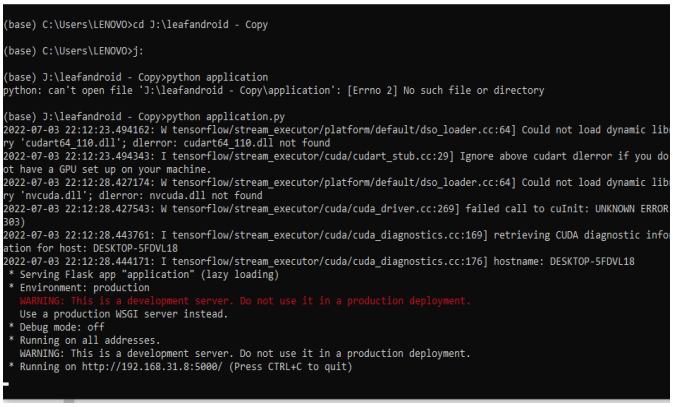

Figure 7: Flask framework deployment

The access the API by address http://192.168.31.8:5000/, where 192.168.31.8 is your local address, and 5000 is the port number to access the API. If it is a success, it will show like this on the web as shown in figure 7 and click on ip address and it connect to server and open and predict the leaf it is healthy or disease one and using eclipse the application is design

IV. RESULTS AND DISCUSSION

Preprocessing: The dataset that has been obtained is not of same size (scaling invariance and translation invariance) and not clear, so we first resized them into 256*256 as shown in figure 8.

Figure 8 :Resized image

Feature extraction: Gabor filter is a linear filter that is employed in a variety of image processing tasks, including edge detection, texture analysis, and feature extraction technique on the ROI of leaf image and the extracted features are in gray color using Gabor filter and edge detection as shown in figure 10

Figure 9: Feature extraction

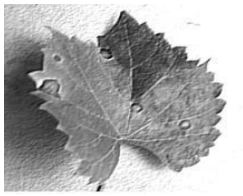

Figure 10: Feature extraction using Gabor filter

A. Results of CNN algorithm

The experimental analysis for the image classification based on plant diseases was carried out for 10 epochs for the VGG 16 model. Figure 11 displays the plot between epochs and accuracy for determining accuracy and loss of the model. From the figure it is clear that as the number of epochs increases, the accuracy of the model is also increasing as well the loss is decreasing with respect increase in epochs. From the results, it is clearly understood that performance of the CNN model achieved an accuracy of 99% at the 10th epoch as shown in figure 11

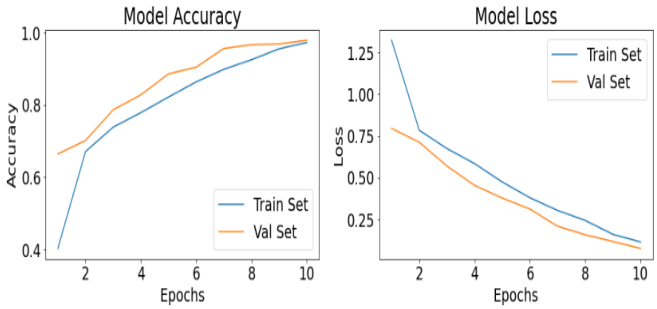

Figure 11: Accuracy curve and validation loss curve for CNN

	precision	recall	f1-score	support
0	0.99	0.94	0.97	127
1	1.00	1.00	1.00	106
2	0.94	1.00	0.97	130
3	1.00	0.97	0.99	120
accuracy			0.98	483
macro avg	0.98	0.98	0.98	483
weighted avg	0.98	0.98	0.98	483

Figure 12: Report analysis of CNN algorithm

The report analysis also contains precision, recall, f1-score has 99%,94%,97% as shown in figure 12

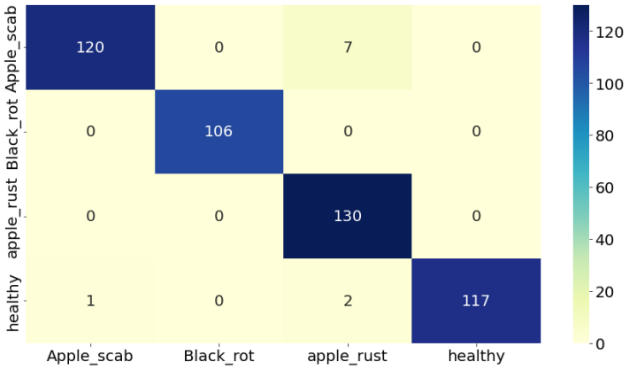

Figure 13 :Confusion matrix For CNN

The confusion matrix are in between predicted value and actual values for four different diseases.in this out of 121 detected 120 images one loss, black rot actual value is predictive value are same there is no loss, and apple rust out of 139 images 130 are detected and 9 are not recognized, and healthy leaves are out of 120 images 117 are detected .by calculating, true positive values have come 120, true negative value 355, false negative is 7 and false positive values are 1. by using this calculate accuracy, and f1 ratio and selectivity as shown figure 13.

B. Results of Comparison between SVM & KNN Algorithm

Support vector machine achieved an accuracy of 76% and the KNN achieved an accuracy of 36%. From the graph shown in figure 14, SVM achieved a highest accuracy when compared with KNN.

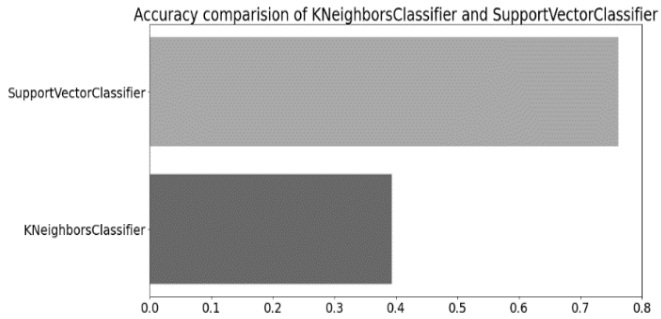

Figure 14: Comparison of SVM and KNN

The report Analysis contains the accuracy as 76 and by calculating the precision, recall, f1-score has 80%,55%,65% as shown in figure 15

	precision	recall	f1-score	support
0	0.80	0.55	0.65	127
1	0.74	0.82	0.78	106
2	0.73	0.89	0.80	130
3	0.80	0.79	0.79	120
accuracy			0.76	483
macro avg	0.77	0.76	0.76	483
weighted avg	0.77	0.76	0.76	483

Figure 1: Report analysis of SVM algorithm

C. *Results of VGG 16 Algorithm*

The experimental analysis for the image classification based on plant diseases was carried out for 10 epochs for the VGG 16 model. Figure 11 displays the plot between epochs and accuracy for determining accuracy and loss of the model. From the figure it is clear that as the number of epochs increases, the accuracy of the model is also increasing as well the loss is decreasing with respect to increase in epochs. From the results, it is clearly understood that performance of the CNN model achieved an accuracy of 99% at the 10th epoch as shown in figure 11

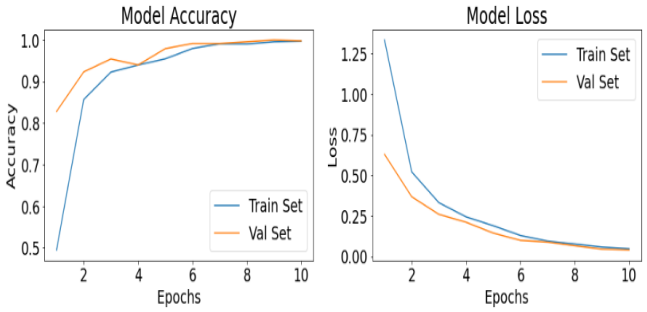

Figure 16 Accuracy curve and validation loss curve for VGG 16

The maximum validation loss for CNN was achieved at 1st with a loss of 69% as shown in figure 17

	precision	recall	f1-score	support
0	1.00	1.00	1.00	127
1	0.99	1.00	1.00	106
2	1.00	1.00	1.00	130
3	1.00	0.99	1.00	120
accuracy			1.00	483
macro avg	1.00	1.00	1.00	483
weighted avg	1.00	1.00	1.00	483

Figure 17: Report Analysis of VGG 16

The report Analysis contains the accuracy as 100% and by calculating the precision, recall, f1-score has 100%,100%,100% as shown in Figure 17

The confusion Matrix are in between predicted value and actual values for four different diseases. by calculating true positive value will be 127, true negative value 356, false negative is 0 and false positive value are 0 by using this calculate accuracy, and f1 ratio and selectivity as shown in figure 18

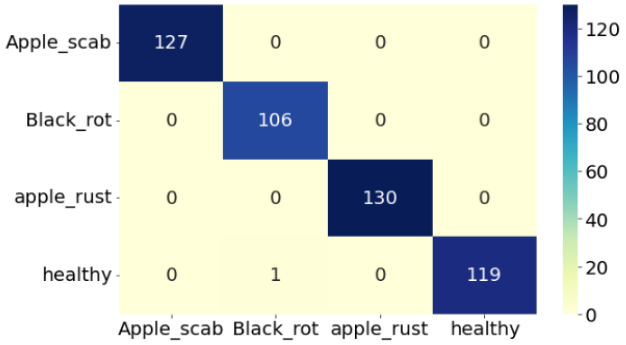

Figure 18 :Confusion matrix For CNN

The confusion matrix for All four Model was computed and the parameters such as Sensitivity, Precision and F1-Score Were evaluated for All models 2 illustrate the performance matrix for plant diseases dataset

From the experimental analysis it was observed that performance parameters such as sensitivity, precision and F1-Score were all with 100% which is better when compared with CNN, SVM, KNN.

Table 2: Performance matrix for plant disease detection dataset

SINO	model	Epoch	Accuracy	Precision	Recall	F1-score
1	KNN	10	39	80	16	25
2	SVM	10	76	80	55	65
3	CNN	10	98	99	94	96
4	VGG	10	99	100	100	100

The confusion matrix are in between predicted value and Actual value in this confusion matrix between 12 types of diseases as shown in figure 19

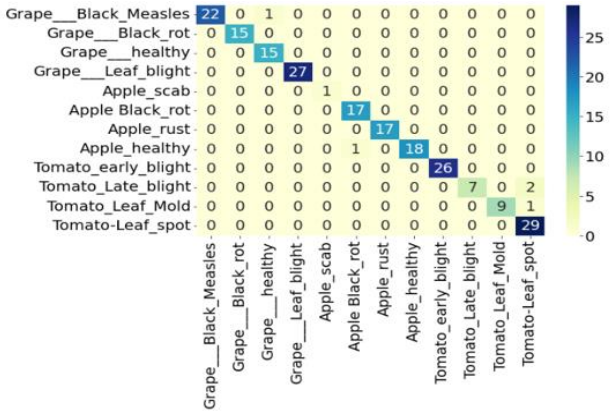

Figure19 Confusion matrix for different plant

Table 3: Comparison between previous and current technology

Techniques	**Plants**	**Accuracy**
FRCNN	grapes	95.7%
CNN	Tomato, potato, grapes	87%
CNN and LVQ	Apple, corn, potato	97%
Google net	Apple, corn, Tomato, grapes	94%
VGG 16	Apple, grapes, Tomato	99.7%

D. Results of Mobile Application

As we discussed in above methodology, the outcomes are illustrated in figure 18, which shows the leaf disease detection in the developed mobile application. At initial stage, the home login page will ask for the user credentials i.e., username and password.

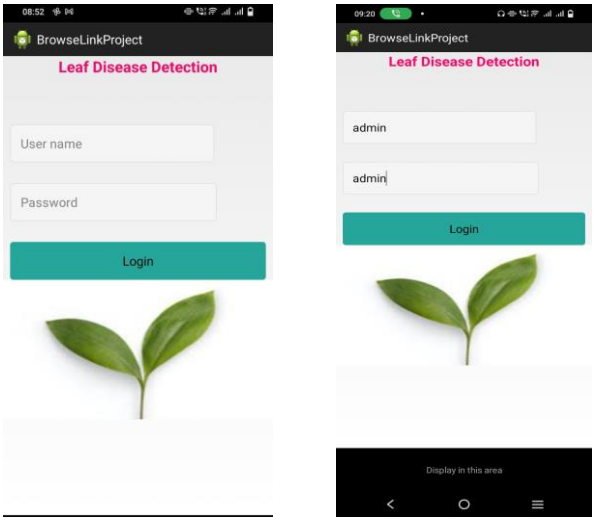

Figure 18 login page

After entering the login button, application will interface with the web application i.e., server. It gives option to upload the leaf image. Once, you upload the image, the developed application will detect, whether the leaf is healthy or diseased. Figure 19 displays the button to reveal the plant disease.

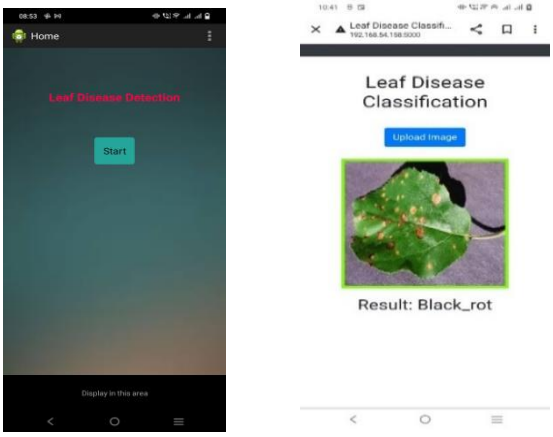

Figure 19 :Plant disease detection in application

V.CONCLUSION

The paper has presented a smart plant leaf disease detection model and implemented with a graphical user interface to support 4 algorithms: SVM, KNN, CNN, and VGG 16 for plant village image dataset The dataset for the image classification mainly consisted of 12 types of plant diseases with 2411 images. The input images are preprocessed and used for data augmentation. The accuracy of KNN is 39% and that SVM model is 76% and CNN model is 98% at 10^{th} epoch, and that of VGG-16 model is 99% at 10^{th} epoch. Since VGG 16 model involves large number of layers and it is pretrained model So compared to other model this is the highest accuracy So build the model with VGG16 so proposed mobile app is created to function independently on a smartphone.

Future Scope involves to include the detection of additional agricultural diseases, and it will also be tested in a field scenario. Add new features to the programmer to improve usability, such as disease treatment. The system might also be developed utilizing a different dataset or image of a different plant species. Prepare for It may be usable in different countries if many languages are supported.

REFERENCES

[1] H. F. Ng, C. -Y. Lin, J. H. Chuah, H. K. Tan and K. H. Leung, "Plant Disease Detection Mobile Application Development using Deep Learning," *2021 International Conference on Computer & Information Sciences (ICCOINS)*, 2021, pp. 34-38, doi: 10.1109/ICCOINS49721.2021.9497190.

[2] X. Guan, "A Novel Method of Plant Leaf Disease Detection Based on Deep Learning and Convolutional Neural Network," *2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP)*, 2021, pp. 816-819, doi: 10.1109/ICSP51882.2021.9408806.

[3]. Jeon, Wang-Su, and Sang-Yong Rhee. "Plant leaf recognition using a convolution neural network." *International Journal of Fuzzy Logic ad Intelligent Systems* 17.1 (2017): 26-34.

[4]. Liu, X., Min, W., Mei, S., Wang, L., & Jiang, S. (2021). Plant disease recognition: A large-scale benchmark dataset and a visual region and loss reweighting approach. *IEEE Transactions on Image Processing*, *30*, 2003-2015.

[5]. asim, M. A., & Al-Tuwaijari, J. M. (2020, April). Plant leaf diseases detection and classification using image processing and deep learning techniques. In *2020 International Conference on Computer Science and Software Engineering (CSASE)* (pp. 259-265).

[6] H. F. Ng, C. -Y. Lin, J. H. Chuah, H. K. Tan and K. H. Leung, "Plant Disease Detection Mobile Application Development using Deep Learning," *2021 International Conference on Computer & Information Sciences (ICCOINS)*, 2021, pp. 34-38, doi: 10.1109/ICCOINS49721.2021.9497190

[7] Sardogan, M., Tuncer, A., & Ozen, Y. (2018, September). Plant leaf disease detection and classification based on CNN with LVQ algorithm. In *2018 3rd International Conference on Computer Science and Engineering (UBMK)* (pp. 382-383

[8] Wang, G., Sun, Y., & Wang, J. (2017). Automatic image-based plant disease severity estimation using deep learning. *Computational intelligence and neuroscience*, *2017*.

[1] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[2] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[3] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper not being published.

Figure 9 Feature extraction

We suggest that you use a text box to insert a graphic (which is ideally a 300 dpi TIFF or EPS file, with all fonts embedded) because, in an MSW document, this method is somewhat more stable than directly inserting a picture.

To have non-visible rules on your frame, use the MSWord “Format” pull-down menu, select Text Box > Colors and Lines to choose No Fill and No Line.