



## A SMART MOBILE DIAGNOSTIC SYSTEM FOR CHILI DISEASES BASED ON DEEP CONVOLUTIONAL NEURAL NETWORK

<sup>1</sup>Binzy Nazar, <sup>2</sup>Shayini R

<sup>1</sup>M-tech Student, <sup>2</sup>Assistant Professor

<sup>1</sup>Department of Electronics and Communication Engineering,

<sup>1</sup>College of Engineering Thalassery, Kannur, India

**Abstract:** The major problems that are faced by the cultivators are the diseases effecting on their crops. There is no such an authenticated and globalized technique for the detection and diagnosis of plant diseases. In this paper a Mobile diagnostic system for Chili plant was developed as an example to solve this problem. Here we take Chili plant as an example. Chili is one of the widely used as well as cultivated crop not only in our home garden, but also through out our country. India is one of the major producer of Chili crop among the whole world. With the rapid development of mobile service computing, it have an increasingly important role in our daily lives. Disease detection can be more user friendly when we utilize the mobile service computing technique for this purpose. So here we build an image dataset of 4 kinds of Chili diseases along with healthy leaves and healthy fruits. They are collected from the home garden. Then realize a Mobile diagnostic system for Chili diseases by constructing a Deep convolutional neural network (D CNN). The system realized using an Android app in our Mobile device, with which users can upload images and receive diagnostic results. Here the experimental results shows that the detection accuracy of the chili diseases exceeds 90 % and we can detect and receive diagnostic results in a few minutes using this system.

**Index Terms** - Deep Convolutional Neural Network, Treatment advices, Mobile service computing, Android app

### I. INTRODUCTION

There are so many facilities available around as for the detection and diagnosis of diseases in humans and animals. But farmers around us are suffering to detect and diagnose the diseases affecting on their crops. Since we are belonging to an agriculture-based country we have to think about a solution for this problem. In this era, the Mobile phones have become crucial part of our life. We can't think our life without them. Importance of mobile computing in the field of communication is also increasing as time progresses. We can stay connected to all sources at all times, we can interact with a variety of users via internet and we can sail our mobile computing to our individual needs. So we can make use of mobile computing Technologies for the plant disease detection.

Chili is the most favorite crop in our home garden because. It has an important role in our food menu. Chilly is one of the major item that are commonly used in spicy meals. India is world's largest producer consumer and exporter of chili Peppers. The species of Chili plants can be tolerate most climate. It is highly productive in warm and dry climate. Anthracnose, Powdery mildew, Mosaic virus and Bacterial leaf spot are the major diseases infected on Chili crop. This diseases are commonly found in all species of Chili. This diseases usually affect leaves and fruits and will affect fruit quality and damage economic benefits. It is inefficient and low accurate to recognized by the human eye.

Alone using deep learning technology. We can effectively detect these type of plant diseases. Hence the efficiency and accuracy of these system helps to save human resources. Now a days various types of deep neural networks are commonly used in daily life applications. Especially they are used for the classification and detection of images under different categories. Once we train a Deep neural network with large number of images under various categories, after complete training it can automatically detect a new image given to it and it can predict the category in which the given image belongs to.

In this paper to solve the problem of Plant disease detection the combination of deep learning and mobile computing are used. Here we develop a Mobile Diagnostic system for chili diseases based on Deep convolutional neural network as an example solution. For that

- 1) Collect real time images of Chili plant parts from home garden
- 2) Used different image processing tools in collected images for data augmentation to enhance the size of data set
- 3) Developed a multi-layered D CNN model to classify and detect types of Chili diseases along with healthy leaves and healthy fruits.
- 4) Set up front end and back end of the Mobile diagnostic system by developing an android app as user interface and laptop as server

## II. LITERATURE REVIEW

Plant disease detection can be carried out using their images. Image processing tools and Machine learning techniques are commonly used for this purpose. Usually image processing techniques like segmentation is commonly used to extract features of diseased plant part from its images. We can extract feature values of infected parts of plant parts using this technique and they can be used to train some sort of ML algorithms. Commonly supervised learning approaches like SVM classifiers are used for this purpose. Through this method, the accuracy of the detection system can be very high but the process of disease detection will be very complicated.

Now a days un-supervised learning approaches like Convolutional neural networks are used for the classification and detection of plant diseases. This method is easy to understand and implement in daily life application scenarios. The main advantage is that there is no need of manual feature extraction. Through training of CNN model, it automatically extracts features of images. Various types of pre-trained neural networks like GoogLeNet, AlexNet, DenseNet etc. are using for this purpose. Similarly we can develop our own CNN model for detection and classification of plant diseases.

Wenyan Pan et al. developed an intelligent Mobile diagnostic system for Citrus diseases based on Simplified Densely connected Neural Network. In this paper they tried to developed 6 types of Citrus diseases. For that they developed a Simplified densely connected neural network by re-arranging some hidden layers of DenseNet 201. DenseNet 201 is a fully connected neural network which have 4 dense blocks. They removed 5 bottleneck layers and added batch normalization layer, global average pooling layer and soft max layer in DenseNet 201 to simplify their structure. Trained the model with their dataset and tested. They got an accuracy of 88%. The developed model uploaded in cloud space. Farmers can access the model through Wechat applet to detect the diseases by uploading the captured images. The facilities like Plant growth monitoring and telecommunication with plant disease experts also provided in this applet. They uses such a complicated model for plant disease detection and got an accuracy of 88%.

So, in this paper we tried to develop a simple Deep neural network model to detect plant diseases and a user friendly android application for its realization. Here we developed a Smart Mobile diagnostic system for Chili diseases based on Deep convolutional neural network.

## III. METHODOLOGY

The block diagram of proposed system is provided in the figure given below. We developed an Android application to set up the system in the Mobile device so that users can use the system anytime and anywhere to diagnose Chili diseases and get some information about the disease. The system mainly realizes the following things

- 1) Users can take photos of chili plant parts through mobile camera and directly upload images to the server.
- 2) Then the system feedback the disease types of Chili plant, it also displays relevant information of the disease such as disease symptoms and causes along with that this system will also propose corresponding treatment plan for the disease which will help the users to prevent and cure the disease.

Here laptop is used as the server. where users can upload the captured images of chili plant to the Android app and provide it for testing purpose. The Android app and server are connected with firebase. Firebase is a toolset to build improve and grow our app. It provides large number of services. These services are hosted in the cloud. Firebase interact with the server directly with no need to establish any middleware between our app and the server. The app is developed here using MIT app inventor. MIT app inventor is an intuitive visual programming environment that allows everyone to build fully functional app for smartphone. When we upload an image in the application the, upload flag in the 10server turns to 1 and begins a series of processes inside the server and returns back the Diagnostic result to the app by turning result flag in the server to 1

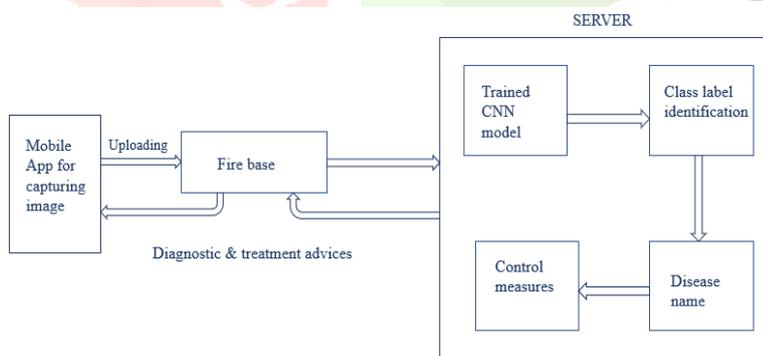


Fig. 1. Block diagram of proposed system



Fig 2. Example of diseases under consideration

### 3.1 Data set Establishment

Due to lack of Chili disease data set we collected disease data set from our home garden. The data set contains 4 types of Chili diseases along with healthy leaves and fruits, provided in TABLE 1. Powdery mildew, Bacterial leaf spot, Mosaic virus, Anthracnose are the diseases under consideration. Each category consists of corresponding images of leaves and fruits with symptoms of Chili plant and most images have their surrounding environment. Since our collected data set is not enough to train our model we use various data augmentation methods to enhance the size of the data set. Training with less number of data will prone to over fitting and hence it leads to less accuracy. Data augmentation is proved to be affective to increase the performance of Deep neural network. So, 7 image processing tools are used to augment the collected data set. They are horizontal flipping, vertical flipping, horizontal+vertical flipping, brightness & contrast enhancement, addition of noise like Guassian and Salt & pepper. The augmentation methods and their specific operations are given in the TABLE 2

TABLE I  
Augmentation Methods

METHODS	SPECIFIC OPERATIONS
Flipping	Horizontal flipping, Vertical flipping, Horizontal + vertical flipping
Increase brightness	Image intensity + 40
Increase contrast	Extract saturation channel * 1.5
Noise addition	Gaussian noise & salt and pepper noise

TABLE II  
Count of collected images

Label	Count
Healthy leaf	32
Healthy fruit	55
Anthracnose	47
Mosaic virus	75
Bacterial leaf blight	45
Powdery mildew	71

### 3.2 Deep Convolutional Neural Network

A deep CNN model is used here for the detection and classification of diseases. The architecture of the developed the CNN model is given in the figure. It consists of an input layer which is the image of size 224 \* 224. After that there will be 3 sets of convolution layers followed by max pooling layers. First set of layer consists of 30 convolutional filters each of size 5 \*5, second set have 8 filters of size 3 \*3, third set have 16 filters of size 3 \* 3, and the last set have 32 convolutional filters of size 3 \*3 .Each set of hidden layer consists of max pooling layer to down sample the convolved images with 2 \*2 kernel sized filter. After that there is a fully connected layer, and at last a classification or output layer. Here batch normalization is used and ReLU or rectified linear unit are used as an activation function. we use augmented dataset to get our model. The architecture of developed model is given here.

1) Training:- Parameters used to train the deep neural network is given in the table. network uses to test a gradient descent algorithm to train the network. To minimize the loss, this algorithm update network parameters by taking small steps in the direction of the negative gradient of the loss function. 'sgdm' that is stochastic gradient descent with momentum is the optimizer used here. The solver update the parameters using a subset of data each step, called minibatch. Each parameter update is called iteration and full pass through the entire data set is called epoch. This network undergoes 10 Epoch each with 16 iteration for training. We modied the learning rate in every 10 number of epochs by multiplying with a factor of 0.1. I took a larger learning rate in the beginning of training and gradually reduced this value during Optimisation. doing so can shorten the training time while enabling smallest steps towards the minimum of the loss as training progresses. For this purpose specified learn and write drop factor and period for the training process are selected. Here learn rate schedule is the option for dropping learn rate during training. Verbose is the indicator to display training information. L2regularization is the factor for weight decay. Momentum is the contribution of the parameter update step of the previous iteration to the current iteration of SGDM

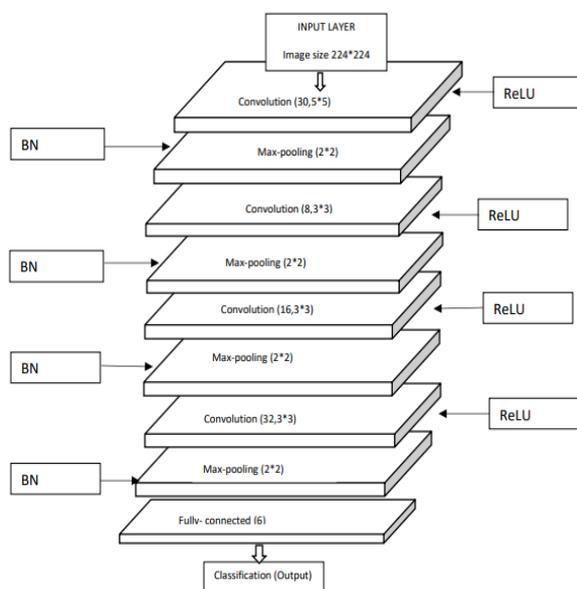


Fig. 3 Architecture of developed D CNN

TABLE III  
TRAINING PARAMETERS

PARAMETERS	SPECIFICATIONS
Algorithm or Optimizer	SGDM
Epochs	10
Iteration per Epoch	16
Initial learning rate	0.0001
Learning rate modification factor	0.1
Drop out factor	0.1
Momentum	0.9
L2Regularization	0.005
Learn rate drop period	60
Learn rate schedule	Piece wise
Verbose	TRUE
Verbose frequency	60

2) Testing:- after training we have to test the network in order to evaluate their performance for that tested the train dataset with the entire dataset. For training purpose we used 75 percentage of the augmented data set. the evaluation of CNN are carried out using confusion Matrix. A confusion matrix is a summary of predictions results on a classification problem the number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix.

That is the confusion matrix shows the ways in which our classification model is confused when it makes predictions. From this confusion Matrix we can calculate true positive, true negative, false positive and false negative values of the testing of CNN. Each row of the confusion matrix corresponds to a predicted class and each column of the matrix corresponds to actual class. We can assign event row as positive and no event as negative. We can then assign event column of predictions as true and the no event as false. The observations from confusion Matrix after training are:- true positive for correctly predicted event values false positive for incorrectly predicted event values true negative for correctly predicted no event values false negative for in correctly predicted no event values.

		Predicted	
		0	1
Actual	0	TN	FP
	1	FN	TP

Fig 4 Concept of Confusion Matrix

### 3.3 Evaluation Parameters

Using the values obtained from Confusion matrix elements( Eg. TP,TN,FP,FN) evaluation parameters of developed DCNN model are calculated using the following formula

$$\text{Accuracy} = ((T P + T N)/(T P + T N + F N + F P))100 \quad (1)$$

$$\text{Sensitivity} = (T P/(T P + F N))100 \quad (2)$$

$$\text{Specificity} = (T N/(T N + F P))100 \quad (3)$$

$$\text{Error rate} = ((F P+F N)/(T P+T N+F N+FP))100 \quad (4)$$

$$\text{Precision} = (T P/(T P + F P))100 \quad (5)$$

$$\text{False positive rate} = (F P/(T N + F P)) \quad (6)$$

$$\text{Recall} = (T P/T P + F N)100 \quad (7)$$

$$\text{F1Score} = (2T P/(2T P + F P + F N))100 \quad (8)$$

$$\text{Overall accuracy} = ((\text{total TP})/\text{Total images}) * 100 \quad (9)$$

### 3.4 Mobile Application

Here we developed an Android app for the realization of Mobile diagnostic system. Developed an app using MIT app inventor platform. It is a web application integrated development environment originally provided by Google and now maintained by Massachusetts Institute of Technology(MIT)[13]. There are 2 parts in the app development process in this platform. first one is The design editor, or designer, it is a drag and drop interface to lay out the elements of the application's user interface. The next one is the blocks editor, It is an environment in which app inventors can visually lay out the logic of their apps using color-coded blocks that snap together like puzzle pieces to describe the program[12]. There is options like TyniDB and TinyWebDB it helps us to access Cloud data base like Firebase. Firebase Cloud Messaging (FCM) is a cross-platform messaging solution that lets us reliably send messages at no cost[14].

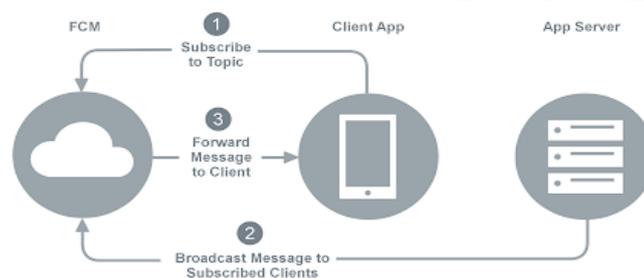


Fig 5. Concept of Firebase Cloud Messaging

## IV. RESULTS AND DISCUSSION

This paper proposes a Mobile diagnostic system for Chili diseases based on Deep convolutional neural network. For that, collected 325 images of 4 types of chili plants from home garden. Augmented images in dataset in order to increase the size of the data set. Then developed a Deep neural network for classification and detection of diseases. Trained the model with augmented data set. The performance of developed model is evaluated using confusion matrix after testing. Then the developed mode uploaded to the server. Lap topis used as the server. Then a Mobile application developed using MIT app inventor for uploading images of plant parts and receiving treatment advices. The server and app are connected with a firebase platform. The experiments were conducted in the Desk top PC intel core i3-7020U 7th Gen Processor, 2.30 GHz, 4 GB RAM. The soft wares used for the implementation are MATLAB2019 PYTHON 3.95. MATLAB is used for the development and working of D CNN and PYTHON used to establish firebase platform to connect server and Mobile app. The main steps involve in the development of Mobile diagnostic system are as follows

### 4.1 Training and Testing of D CNN

We use 75 percentage of the augmented dataset for training purposes and the entire image dataset for testing purpose to evaluate the performance of developed deep convolutional neural network. Training progress of deep convolutional neural network is given in the figure(). We can say that as accuracy of training process increases error is decreasing. In case of accuracy plot, x axis denote the epochs and y-axis denote the accuracy and In case of loss plot, x-axis denotes number of epochs and y- axis denotes the error rate. It takes 10 epochs for the completion of the Training process and each epoch undergoes 16iterations. it takes 20 minutes and 9 seconds for the completion of training process and got an highest accuracy of 92.24 percentage. Testing accuracy of each class are calculated based on confusion Matrix elements. classification accuracy of each category are provided in figure, it can be observed that the accuracy of each category is exceeds 95 percentage using this network, this shows that the performance of the develop model is very good for the disease detection.24 There are evaluation parameters like recall , F1 score etc that we are used to evaluate the developed model from figure. we can see that the obtained evaluation parameters shows a better result that indicates the developed model shows a good performance in plant disease detection

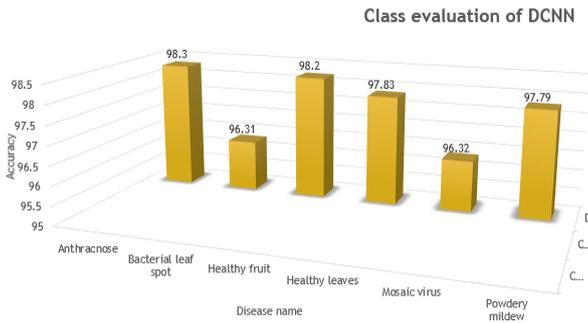


Fig 6. Class evaluation plot of developed DCNN

CLASS	ACCURACY	SENSITIVITY	SPECIFICITY	ERR	PRECISION	FPR	RECALL	F1_SCORE
'1'	98.326	96.019	98.74	1.6738	93.182	1.26	1800	94.579
'2'	96.314	80.378	99.156	3.6865	94.444	0.84353	8400	86.845
'3'	98.208	94.151	99.115	1.7919	95.962	0.88533	3200	95.048
'4'	97.83	91.632	98.451	2.1698	85.547	1.5494	2100	88.485
'5'	96.325	94.66	96.749	3.6745	88.102	3.251	3400	91.264
'6'	97.799	94.046	98.699	2.2012	94.542	1.3014	3500	94.293

Fig 7. Evaluation parameters of D CNN

### 4.2 Mobile Diagnostic System

Here the connection between server and app is established with the help of PYTHON 3.95version software by calling firebase package. The MATLAB Engine API for Python allows you to call MATLAB as a computational engine from Python. Its working is explained in the flow chart given below. When we upload an image to app it will pass to fire base cloud platform. This fire base platform is connected with python software in the server. It turns the upload flag in the server turns 1 from 0. Then the python allows us to call matlab engine, where the developed D CNN is working. Then the DCNN detect the category in which uploaded image belongs to. Then corresponding diagnostic results are fetched and this file is passed to python program as a value of the variable, then the result flag turned to 1 from 0.when the result flag turns 1 upload flag automatically turns to zero, which indicates that the server is now ready to accept the new input image .This detected value or diagnostic results send back to the android app through FCM platform. The screen shot of android app part and server part of the developed diagnostic system is given below

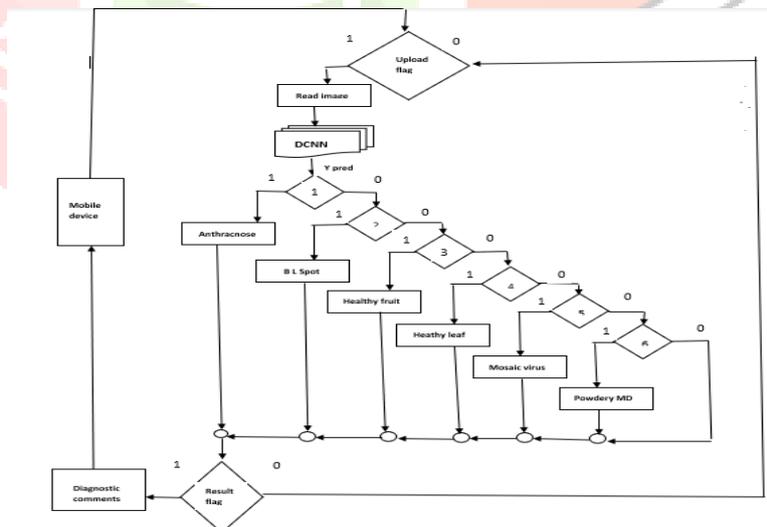


Fig 8 Flow chart of working of Mobile diagnostic system

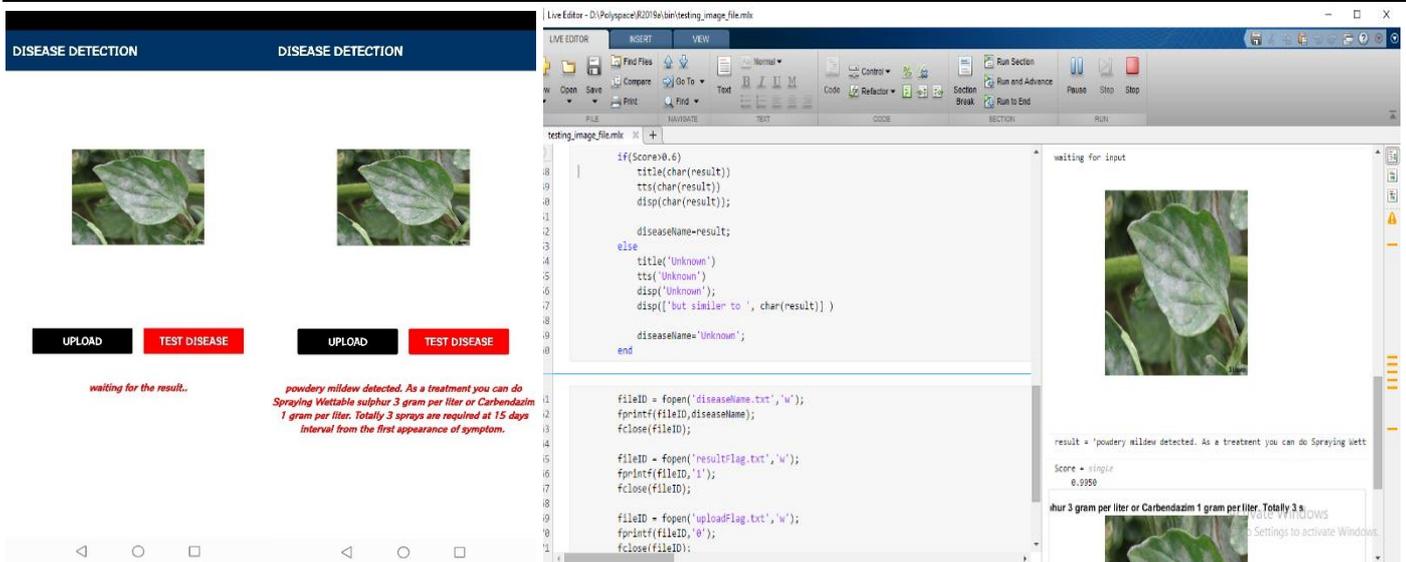


Fig 9. Mobile Application part and Server part of the Mobile diagnostic system for Chili plant

## V. CONCLUSION AND FUTURE WORKS

The purpose of this project is to destroy the barrier between plant growers and experts and also bother the common people about the crop diseases and to promote people for home vegetable gardening. Chili diseases are widely spread and rapidly transmitted, which will lead to economic loss if not effectively treated well. This system will help farmers for correct management of the crop cultivation. In this paper we construct a model to identify Chili diseases using deeply connected networks and got an accuracy of 90.24%. Which is a good detection model for disease detection. At the same time Android app is used to realize automatic identification of Chili diseases and provide proper disease prevention and treatment suggestions for crop growers. Here firebase cloud messaging platform is used to communicate between server and the android app.

Next we will continue to expand the data set for different crop category and further optimize our method to achieve higher accuracy. In future, better efficiency can be obtained by applying advanced classification algorithms. So this system can be beneficial in different agricultural sectors to increase crop productivity

## REFERENCES

- [1] S. R. Dubey, S. Jalal, "Principle of disease classification using color, texture and shape features from images", Signal, Image and Video Processing, Springer (2015).
- [2] Pen Jian, Yuehan Chen, Bin Liu, Dongjian He, Chunquan Liang, "Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach supported Improved Convolutional Neural Networks", IEEE Transaction, Volume 7, May 17, 2019, Digital Object identifier 10.1109/ACCESS.2019.291492
- [3] Qiufeng Wu, Yiping Chun, Jun Meng, "DCGAN- Based Data Augmentation for Tomato plant disease Identification" IEEE Transaction, Volume 8, June 5, 2020. Digital Object Identifier 10.1109/ACCESS.2020.299700
- [4] Uday Pratap Singh, Siddharth Singh Chouhan, Sukirty Jain, Sanjeev Jain, "Multilayer Convolution Neural Network for the Classification of Mango Leaves Infected by Anthracnose Disease", IEEE transaction, 2017, doi: 10.1109/ACCESS.2017
- [5] Wenyan Pan, Jiaohua Qin, Xuyu Xiang, Yan Wu, Yun Tan, Lingyun Xiang "A Smart Mobile Diagnosis System for Citrus Diseases supported Densely Connected Convolutional Networks", IEEE Transaction, volume 7, July 18, 2019, doi: 10.1109/ACCESS.2019.2924973
- [6] Taghi M. Khoshgoftar, "A survey on Image Data Augmentation for Deep Learning", Springer open, Journal of big data (July 2019)
- [7] Konstantinos P. Ferentinos, "Deep learning models for disease detection and diagnosis," Computers and Electronics in Agriculture, vol. 145, pp. 311–318, 2018. doi: 10.1016/j.compag.2018.01.009.
- [8] Evan W. Patton, Michael Tissenbaum, Farzeen Harunan, MIT App Inventor: Objectives, Design, and Development, in Computational Thinking Education pp 31-49— Cite as, SpringerLink
- [9] Romin Irani's Blog, Tutorial : MIT App Inventor + Firebase, <https://rominirani.com/tutorial-mit-app-inventor-firebase-4be95051c325>
- [10] Understanding Confusion Matrix Sarang Narkhede May 9, 2018 <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd6>
- [11] Firebase cloud messaging, <https://firebase.google.com/products/cloudmessaging>
- [12] Raj Kiran, Jameel Akhtar, Pardeep Kumar and Meena Shekhar, "Anthracnose-of-chilli-status-diagnosis-and-management", <https://www.intechopen.com>
- [13] Mosaic Virus, Proven methods for identifying and preventing mosaic virus in home and market gardens, <https://www.planetnatural.com/garden-advice/>