A

PROJECT REPORT

ON

Disease Prediction In Cotton Plant Using Drone

Submitted by

Sanat Dhobi (15IT027)
UjasThummar (15IT033)
Ravi Vaniya (15IT036)
Meet Patel (15IT071)

For Partial Fulfillment of the Requirements for Bachelor of Technology in Information Technology

Guided by

Dr. Zankhana Shah

April, 2019

Information Technology Department

Birla Vishvakarma Mahavidyalaya Engineering College

(An Autonomous Institution)

Vallabh Vidyanagar – 388120

Gujarat, INDIA

Birla Vishvakarma Mahavidyalaya Engineering College

(An Autonomous Institution)

Information Technology Department

AY: 2018-19, Semester II

CERTIFICATE

This is to certify that the projectwork entitled **Disease Prediction In Cotton Plant Using Drone** has been successfully carried out by **Sanat Dhobi (15IT027), Ujas Thummar (15IT033), Ravi Vaniya (15IT036), Meet Patel (15IT071)**for the subject **IT442-Minor Project II** during the academic year 2018-19, Semester-IIfor partial fulfilment of Bachelor of Technology in Information Technology. The work carried out during the semester is satisfactory.

Dr.Zankhana Shah
IT Department
BVM

Dr. Keyur Brahmbhatt
Head, IT Department
BVM

Acknowledgement

We are grateful to the Information Technology department of BVM Engineering College for giving us an opportunity for doing this work. We would like to express our sincere gratitude to Dr.Indrajit Patel, Principal B.V.M Engineering College, Dr.Keyur Brahmbhatt, Head of IT Department and Dr.Zankhana Shah, Faculty guide. We are indebted to all those who provided reviews or suggestions for improving the report.

Last but not the least, we would like to sincerely thank to our beloved Guru HDH Mahant Swami Maharaj without him we would never have been able to prepare this report.

Sanat Dhobi

UjasThummar

Ravi Vaniya

Meet Patel

Abstract

Agriculture is the mainstay of the Indian economy. Almost 70% people depend on it & shares major part of theGDP. Diseases in crops mostly on the leaves affects on the reduction of both quality and quantity of agriculturalproducts. Perception of human eye is not so much stronger so as to observe minute variation in the infectedpart of leaf. It is very difficult to monitor the plant diseases manually. It requires tremendous amount of work, expertize in the plant diseases, and also require the excessive processing time.Hence, image processing is used for the detection of plant diseases.Disease detection involves the steps like image acquisition, imagepre-processing, image segmentation, feature extraction andclassification. In this report, we are providing mobile application solution to automatically detect and classify plant leafdiseases. This approach will enhance productivity of crops. The proposed system works in two phases: the firstphase deals with training data sets. This includes, trainingboth healthy and as well as diseased data sets. The secondphase deals with monitoring the crop and identifying theDisease. Experimentalresult showed that classification performance by a simple Neural Network taking feature set is showing better accuracy. The presentwork proposes a methodology for detecting cotton leaf disease early, using image processing techniques and artificialneural network (ANN).

Table of Content

List of Figures

List of Tables

List of Abbreviations

OpenCV - Open source computer vision

GLCM - Gray-Level Co-Occurrence Matrix

GUI - Graphical User Interface

FCM - Fuzzy c-means

PNN - Probabilistic Neural Network

HSV - Hue, Saturation, Value

ANN - Artificial Neural Network

RGB - Red, Green, Blue

HSI - Hyper Spectral Image

AI - Artificial Intelligence

JSON - JavaScript Object Notation

ReLU - Rectified Linear Unit

API - Application Programming Interface

NN - Neural Network

CSV - Comma Separated Value

WAMP - Window, Apache, MySQL and PHP server

e.g. - For example (*exempli gratia)*

Fig. - Figure

No. - Number

et al - And others

Chapter – 1

Introduction

Introduction

Agriculture gave birth to civilization. India is an agrarian country and its economy largely based upon crop production. Agriculture is the backbone of every economy. In a country like India which has ever increasing demand for food due to the raising population, advances in the agriculture sector are required to meet the need. The agriculture sector needs a huge up-gradation in order to survive the changing conditions of the Indian economy. For optimum yield, the crop should be healthy, therefore some highly technical method is needed for periodic monitoring of crop. Crop disease is one of the major factors which indirectly influence the significant reduction of both the quality and quantity of agricultural products. A number of varieties of pesticides are available to control diseases and increase production. But finding the most current disease, appropriate and effective pesticide to control the infected disease is difficult and requires experts advise which is time-consuming and expensive.

The presence of disease on the plant is mainly reflected by symptoms on leaves. So there is a need for an automatic, accurate and less expensive Machine Vision System for detection of diseases from the image.

1.1 Brief overview of the work

Our project is based on disease prediction in the cotton plant, so the foremost requirement for the project is cotton leaves. The image of the cotton leaves will be uploaded by android application to the server. On the server side some techniques like preprocessing, feature extraction, classification will be done by the model on the image. The predicted result will be displayed on the application by the notification module.

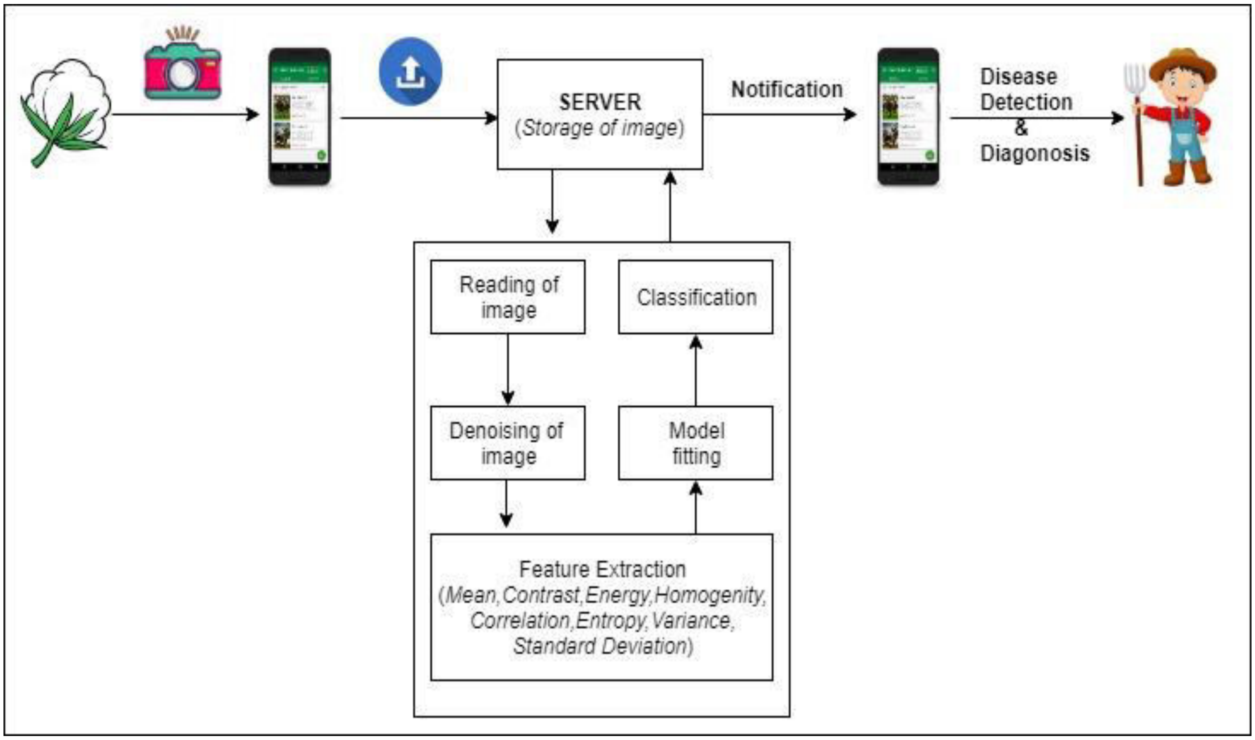

Fig. 1.1 Proposed Methodology

1.2 Project Objective

- To detect disease in the cotton plant by examining leaf of the plant and provide support for disease and its control measures.

1.3 Project Scope

- The proposed system will be able to detect four common diseases (Alternaria, Anthracnose, Bacterial Blight, Cercospora) found in cotton plant.
- The proposed system will only detect diseases caused by fungi and bacteria in cotton leaf.

1.4 Project Modules

1.4.1. Capturing and uploading of image: Using Drone or Mobile Phone's camera capture the image of the cotton leaf plant and by mobile application upload the image to the server.

1.4.2. Preprocessing of image: The aim of pre-processing of the image is an improvement of the image data that suppresses unwanted distortions or enhance some image features important for further processing. Different preprocessing techniques will be used. e.g. De-noising, RGB to Gray conversion, Gaussian Blur, Canny Edge Detection, Segmentation, Histogram Equalization.

1.4.3. Feature extraction from image: After preprocessing of the image some of the features considered for the classification process. e.g. mean, standard-deviation, variance, energy, entropy, homogeneity, contrast, correlation.

1.4.4. Classification of diseases present in image: Implement of the neural network for the classification of diseases of the cotton plant.

1.4.5. Notification: After detection of disease message will be sent which contain identified disease in the plant by notification in the mobile application to the end-user.

1.5 Project Hardware/Software requirements

Following are required tools and software for project implementation

1. Text editor – Jupyter notebook, Android Studio
2. Drone or Mobile Phone
3. Server -Wamp server
4. Browser-Any Web Browse
5. Libraries-Keras, OpenCV, Scikit-image, TensorFlow

Chapter-2

System Analysis

2.1 Comparison of Existing Application with your Project with merits and demerits:

The proposed system by Prakash Manikar et al in paper [1] perform leaf disease detection and classification using image processing techniques. It includes feature extraction using GLCM and image segmentation using K-means clustering. It also provide graphical user interface to detect disease but GUI can only be accessed via desktop application.

In the [2] paper proposed system to identify diseases in cotton plant using spatial FCM and PNN classification. The proposed system also helps in classifying the pest in cotton plant. In pre-processing noise removal is performed using median filter. The system did not provide any diagnosis measure to control disease.

In the [3] paper Anuradha Bagdage, proposed an approach to detect disease in crops like what and cotton. Here it uses machine learning techniques to perform classification. Classification is performed using canny edge detection algorithm. Although, the proposed system did not provide any method to perform pre-process and feature extraction method. The method did not include any result analysis.

In the [4] ReenaTijare et al proposed system include two diseases alternaria and fungi. The images are pre-processed by cropping the image, gray-scaling and threshold. Cotton feature like HSV features extracted from the capture of segment and ANN. Experimental accuracy is around 80%.But this system did not provide a user interface to interact a user with the system.

Sachin khirade et al uses image processing techniques for detection of disease in any plant in [5]. Here image is captured through only mobile camera, classification by k-means clustering and ANN. Here feature extraction is performed using colour co-occurrence method where RGB image converted into HSI translation. But it didn’t performed any pre-processing step and implementation is done in MATLAB software.

2.2 Literature Review:

In India, there is a drastic change in Agriculture. There is a huge gap between technology and farmers in India. Many start-ups have emerged to bridge this gap between technology and farmers. Food demand is exponentially increasing due to a rise in population. It is often heard that pests and diseases attack crops and therefore food gradually reduces due to these attacks. By 2050, the earth's population is expected to grow by 9.7 billion. Therefore, a clear graph of rise in food demand is visible. Feeding a growing world population asks for continuous increases in food production, but arable land remains a limited resource. To cope up with these challenges, we proposed A.I.-based model that help in early detection of disease in cotton plant.

First, the cotton leaf image is captured with the help of drone or mobile application then; image is uploaded to the server (Here WAMP Server installed in computer). Internally image is redirected to the classification model which returns resultant disease predicted. The intended users get the notification in his/her mobile application.

Pre-processing: An image captured by drone may not be always of good quality so pre-processing of the image is required. Using of Gaussian Blur de-noising method is used. In image processing, a Gaussian blur (also known as Gaussian smoothing) is the result of blurring an image by a Gaussian function. The Gaussian blur is a type of image-blurring filters that uses a Gaussian function (which also expresses the normal distribution in statistics) for calculating the transformation to apply to each pixel in the image. The formula of a Gaussian function in two dimensions is,

$$G(x) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Where *x* is the distance from the origin in the horizontal axis, *y* is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution. When applied in two dimensions, this formula produces a surface whose contours are concentric circles with a Gaussian distribution from the centre point. Values from this distribution are used to build a convolution matrix which is applied to the original image. This convolution process is illustrated visually in the figure below. Each pixel's new value is set to a weighted average of that pixel's neighbourhood. The original pixel's value receives the heaviest weight (having the highest Gaussian value) and neighbouring pixels receive smaller weights as their distance to the original pixel increases. This results in a blur that preserves boundaries and edges better than other, more uniform blurring filters [6].

Fig. 2.1 Before Gaussian blur **Fig. 2.2 After Gaussian blur**

Feature extraction: Feature extraction is a dimensionality reduction process, where an initial set of raw variables is reduced to more manageable groups (features) for processing, while still accurately and completely describing the original data set. Here we have used the Gray-Level Co-occurrence Matrix (GLCM). Total eight texture based features are extracted namely, Contrast, energy, homogeneity, correlation, entropy, mean, variance and standard deviation. A GLCM contains information about the positions of pixels having similar gray level values. A co-occurrence matrix is a two-dimensional array, C in which both the rows and the columns represent a set of possible image values. A GLCM $C_\mu(i,j)$ is defined by first specifying a displacement vector $\mu=(\mu x,\mu y)$ and counting all pairs of pixels separated by μ having gray levels *i* and *j*.

$$C_\mu(i,j) = n_{ij}$$

Where n_{ij} is the number of occurrences of the pixel values (i,j) lying at distance μ in the image. The co-occurrence matrix **Cμ** has dimension $\boldsymbol{n \cdot n}$, where *n* is the number of gray levels in the image.

Training of model: This is a supervised machine learning model where the class label is already known. During training, every image present in a dataset is passed through a neural network model for classification. The trained model is stored in H5 format and the entire related configuration in JSON file. An **Epoch** is a hyper-parameter which is defined before training a model. One epoch is when an entire dataset is passed both forward and backward through the neural network only once.

Classification model: An artificial Perceptron is created for several hidden layers. In machine learning, the perceptron is an algorithm for supervised learning of binary classifiers. It is a type of linear classifier, i.e. a classification algorithm that makes its predictions based on a linear predictor function combining a set of weights with the feature vector. Linear classifier defined that the training data should be classified into corresponding categories i.e. if we are applying classification for the two categories then all the training data must be lie in these two categories. Binary classifier defines that there should be only two categories for classification [7].

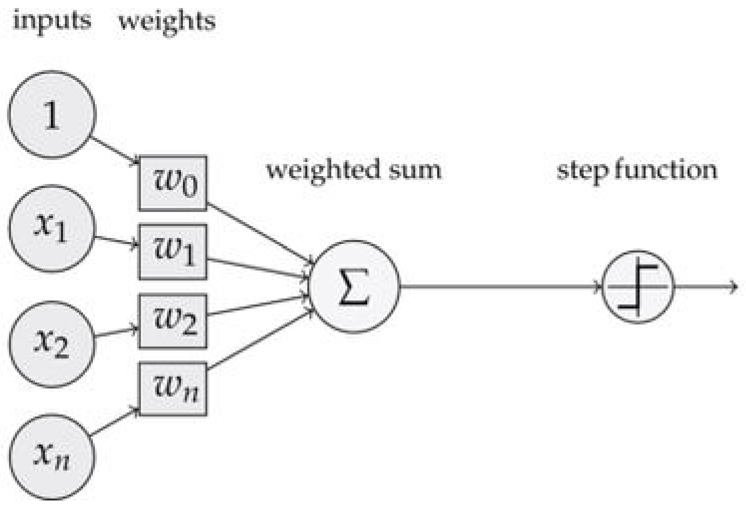

Fig. 2.3 A simple perceptron

Following are the major components of a Perceptron

Input- All the feature becomes the input for a perceptron. We denote the input of a perceptron by [x_1, x_2, x_3, ..], here x represent the feature value and n represent the total number of features. We also have special kind of input called the BIAS. In the image, we have described the value of bias as w_0.

Weights- Weights are the values that are computed over the time of training the model. Initial we start the value of weights with some initial value and these values get updated for each training error. We represent the weights for perceptron by [w_1,w_2,w_3,.. w_n].

BIAS- A bias neuron allows a classifier to shift the decision boundary left or right. In an algebraic term, the bias neuron allows a classifier to translate its decision boundary. To translation is to “move every point a constant distance in a specified direction”. BIAS helps to training the model faster and with better quality.

Weighted Summation- Weighted Summation is the sum of value that we get after the multiplication of each weight [w_n] associated the each feature value [x_n]. We represent the weighted Summation by $\sum w_i x_i$ for all i => [1 to n].

Step/Activation Function- the role of activation functions is make neural networks non-linear. For linearly classification of example, it becomes necessary to make the perceptron as linear as possible.

Output- The weighted Summation is passed to the step/activation function and whatever value we get after computation is our predicted output.

Hence, the basic Perceptron algorithm is used for binary classification and the entire training example should lie in these categories. During each Epochs feature extraction is done and accuracy is measured. In each hidden layer, there will be stage-by-stage feature extraction. Each neuron present in the hidden layer is fully connected to each other. Where every edge connecting each neuron have some initial weight which changes during each iteration of Epoch. Adam optimizer is used for weight initialization. Each hidden layer output will be input to the corresponding next hidden layer. Linear activation function i.e., ReLU is used for classification [8].

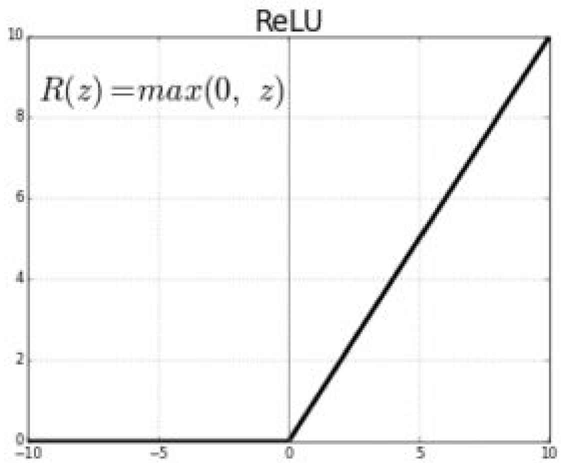

Fig. 2.4 ReLU graph

ReLU works as follows:

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$$

The following sketch is a neural network of a classification model.

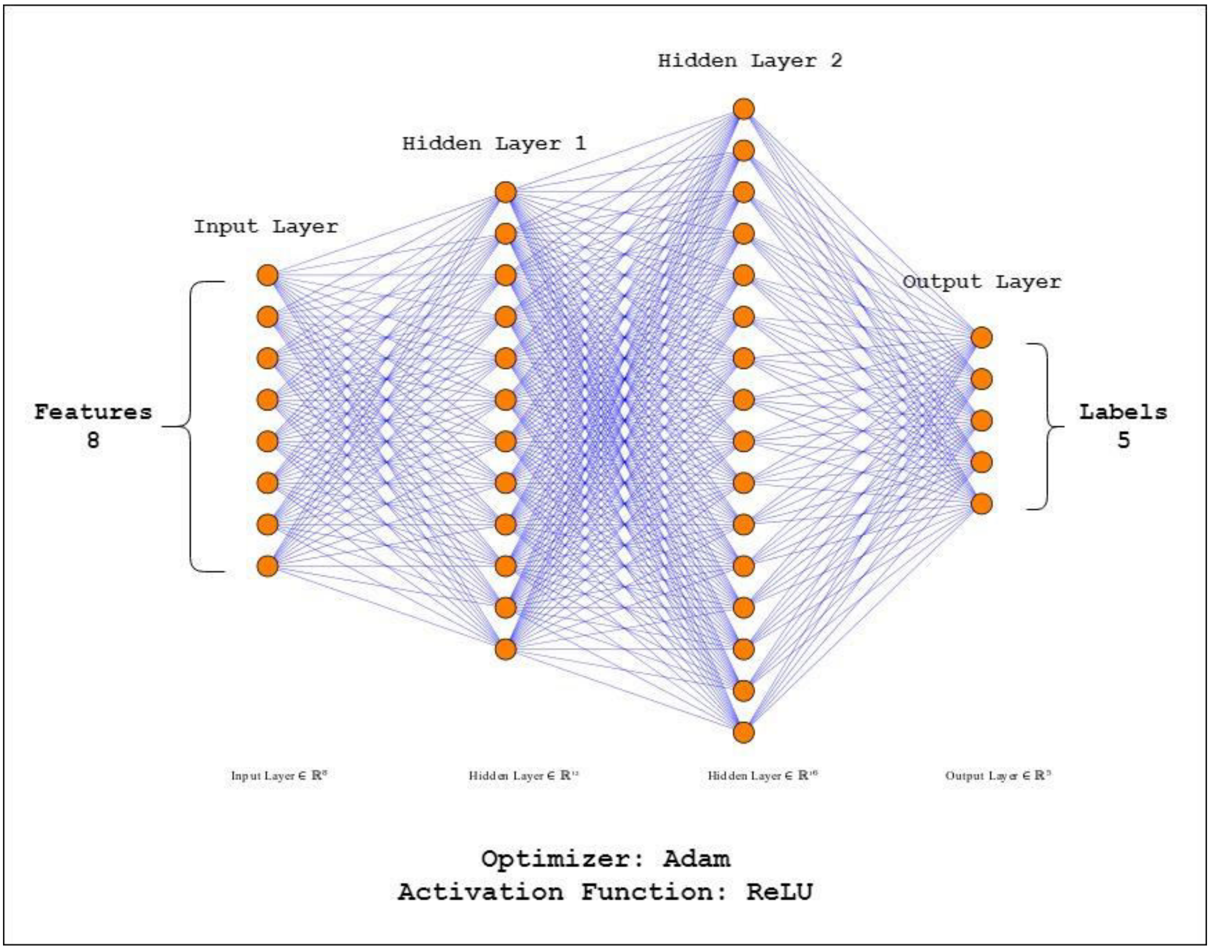

Fig. 2.5 Proposed system neural network

2.3Project Feasibility Study:

2.3.1 Technical Feasibility:

In this project, technically it is possible to detect disease by examining the leaf of a plant. Based on the feature value extracted the model tries to predict the disease. Neural network classification model is used with two or three hidden layers using ReLU activation function.

2.3.2 Operational Feasibility:

Operation of a drone depends on the government rules. As in India, drone under 2.4 GHz working frequency are considered as toy drone so that can be used for the implementation of the project. Permission to capture an image of a cotton plant from agricultural institutes or authorized person is required for training and testing of the project.

2.3.3 Implementation Feasibility:

The implemented project will be user-friendly with the help a mobile application. Provided libraries are open source and easy documentation is available. As it also requires large amount of training images to get better result.

2.3.4 Economical Feasibility:

In this project we have use Sublime Text, Jupyter Notebook which is free open source text editor, WAMP server. But cost of camera equipped in drone will be high as focal length of camera increases for improvement in image quality So, the cost of the project will depend on cost of drone rest it is feasible.

2.3.5. Resource Feasibility:

It is also an essential part of a feasibility study. It includes how much time is available to build the new system, when it can be built, whether it interferes with normal business operations, type and amount of resources required, dependencies, and developmental procedures with company revenue prospectus.

2.4 Project Timeline Chart:

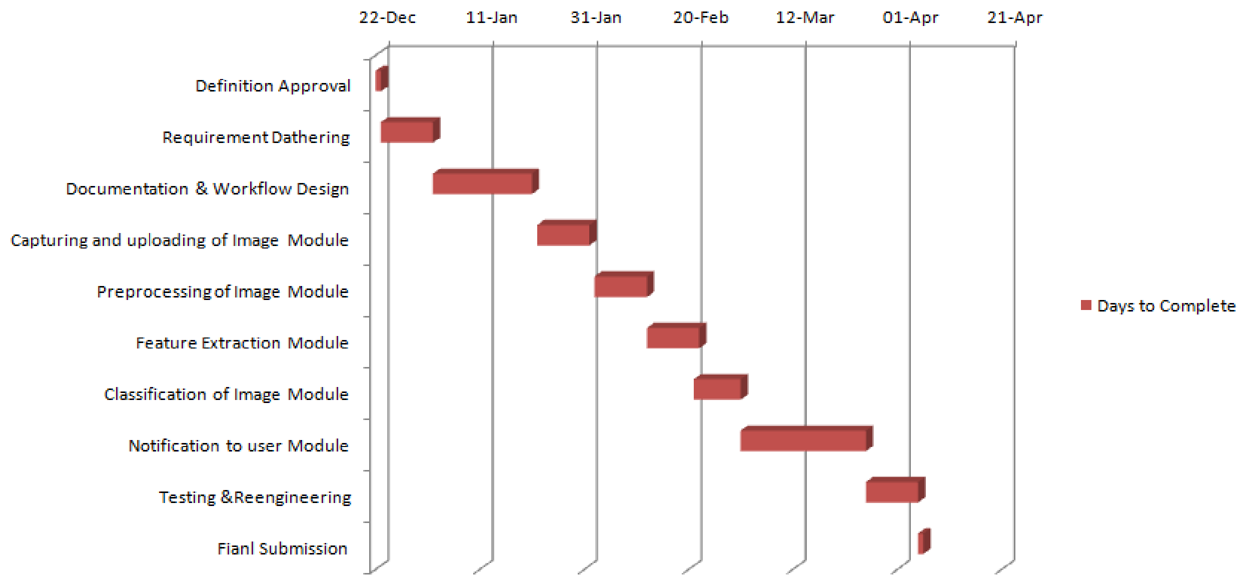

Fig. 2.6 Timeline Chart

2.5 Detailed Module Description:

2.5.1 Capturing and uploading of image:

Capturing of image is performed by using three different ways. First, Camera equipped in drone. Second, mobile camera where real time image of leaf is captured. Last, where plant leaf is grounded and image is captured using mobile camera. During training the model images are taken from internet source and Anand Agricultural University. For testing purpose, all images are captured from AAU's research farms. Then image will be uploaded with the help of mobile application and will be sent to server. If image gets uploaded to server then success is returned otherwise, failure message will be displayed. Failure of image may cause due to poor internet connection or server may fail to start. It is user's responsibility to upload good quality image for getting better result.

2.5.2 Pre-processing of image:

The main objective of pre-processing any image it to improve quality of image. Pre-processing of image also helps to better extract feature from image and reduce the noise present in it. Here for de-noising image we applied Gaussian Blur algorithm which internally convert image from RGB to Gray scale. Gaussian blur returns a matrix which represents the image. Normalization of values is performed so that any outlier present in dataset can be analysed. Then to improve contrast in image Histogram Equalization technique [9] is applied.

2.5.3 Feature extraction from image:

After pre-processing image is feed to GLCM matrix where texture based features are extracted. The graycomatrix function creates a gray level co-occurrence matrix by calculating how frequently a pixel with the particular intensity value i occurs in a specified spatial relationship to a pixel with the value j. By default this spatial relationship is the pixel of interest and its immediate right pixel. However we can specify some other spatial relationship between twos. To create multiple GLCMs, specify an array of offsets to the graycomatrix function. These offsets define pixel relationships of varying direction and distance. Directions can be horizontal, vertical, along two diagonals. Calculating statistics from GLCM matrix also known as Spatial Gray-level Dependence Matrices (SGDM) [10]. The features are as follows:

Contrast: Returns a measure of the intensity contrast between a pixel and its neighbour over the whole image. Range = [0 (size (SGDM, 1)-1)^2]. Contrast is 0 for a constant image. *C(i,j)* is pixel value of location (i,j).

$$Contrast = \sum_{i,j=0}^{n-1} c(i,j)(i,j)^2$$

Energy: Returns the sum of squared elements in the SGDM. Range = [0 1]. Energy is 1 for a constant image.

$$Energy = \sum_{i,j=0}^{n-1} c(i,j)^2$$

Homogeneity: Returns a value that measures the closeness of the distribution of elements in the SGDM to the SGDM diagonal. Range = [0 1] Homogeneity is 1 for a diagonal SGDM.

$$Homogeneity = \frac{\sum_{i,j=0}^{n-1} c(i,j)}{(1+(i-j)^2)}$$

Correlation: Returns a measure of how correlated a pixel is to its neighbour over the whole image. Range = [-1 1] Correlation is 1 or -1 for a perfectly positively or negatively correlated image.

$$Correlation = \sum_{i,j=0}^{n-1} \frac{\{i \cdot j\} \mathrm{X}\, c(i,j) - \{\mu x - \mu y\}}{\sigma x \,\mathrm{X}\, \sigma y}$$

Entropy: Return the randomness of pixels as compared to its neighbour. It is also inversely proportional to energy.

$$Energy = \sum_{i,j=0}^{n-1} -\log_2(c(i,j)) \cdot c(i,j)$$

Mean: Returns an estimate of the intensity of all pixels in the relationships that contributed to the GLCM.

$$Mean = \sum_{i,j=0}^{n-1} i \cdot c(i,j)$$

Standard deviation: The deviation from mean of the intensities of all reference pixels.

$$\text{Standard deviation } = \sqrt[2]{\sum_{i,j=0}^{n-1} c(i,j) \cdot (i-\mu)^2}$$

Now features are extracted from image and based on which decision is made by NN to classify image. Storage of all features values are in CSV file.

2.5.4 Classification of diseases present in image:

Dataset is divided into training and testing dataset in some fixed ratio (Here 80% and 20% respectively). Images present in training dataset are feed to NN. Features extracted from each image are feed to each neuron present in Input layer. Output of input layer is input for next layer. Here ReLU activation function is used which work as follow, if feature value is between certain ranges for that particular neuron then it output 1 otherwise 0. Adam optimizer is used for weight initialisation for each edge and it also adaptively, changes the value of weight iteratively in each Epoch. Model is trained for certain number of Epochs so to get optimum accuracy. Then whole trained model is stored in certain format. To store weights of neurons, number of neurons, number of layers, activation function and optimizer used is stored in H5 format. Then for testing application an API is created using Flask framework. When new image is passed as an argument in testing API then model return its class label. The label is our required result.

2.5.5 Notification to user:

After model predicts the class label, server sends the notification message to the user. It can be viewed from notification section in mobile application. Notification message contains disease by which plant is affected and control measures to reduce loss in production. Notification is sent by callback API from server application. In server, trained model is stored.

Chapter-3

System Design

3.1Use Case Diagram:

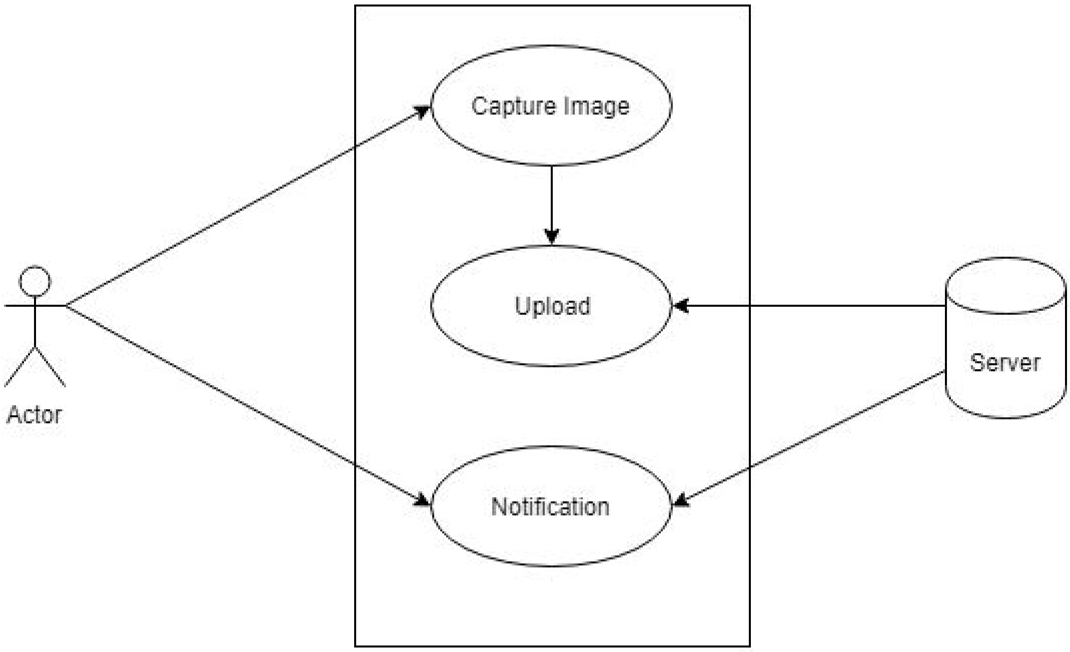

Fig. 3.1 Use Case Diagram

3.2 Data Flow Diagram:

Level 0:

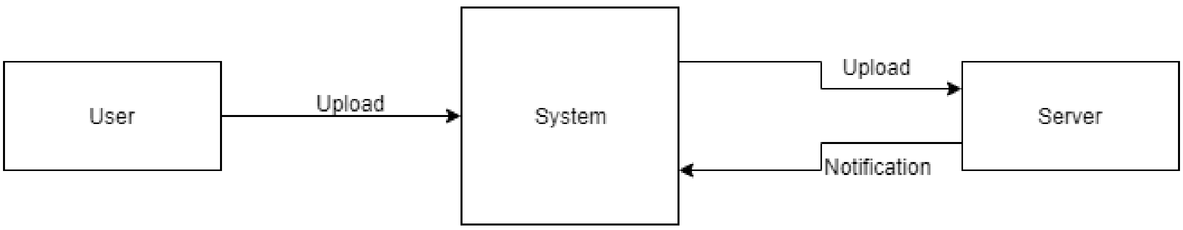

Fig. 3.2 DFD Level 0

3.3 Sequence Diagram:

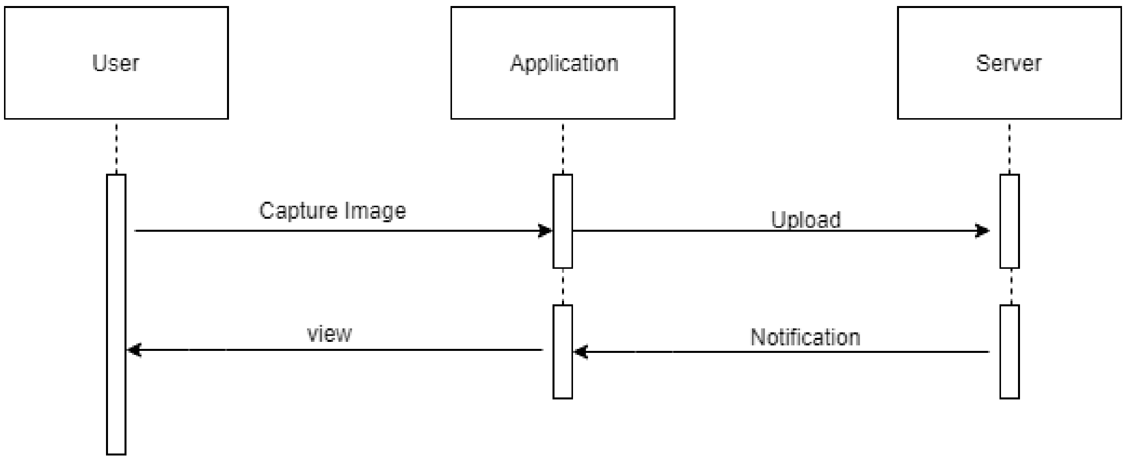

Fig. 3.3 Sequence Diagram

3.4State Diagram:

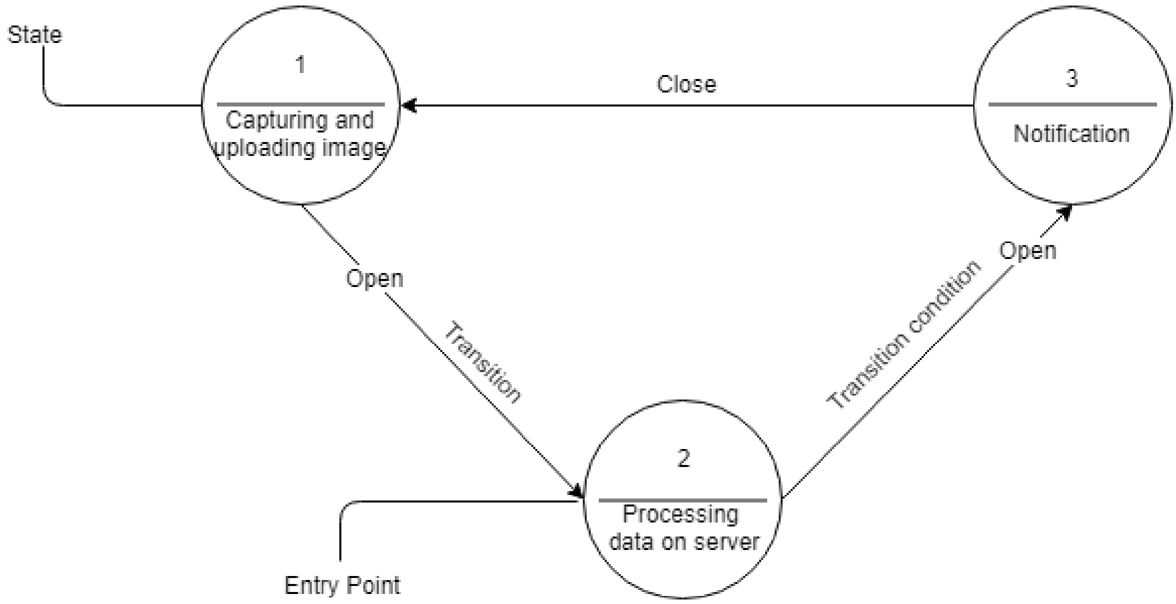

Fig. 3.4 State Diagram

3.5Context Diagram:

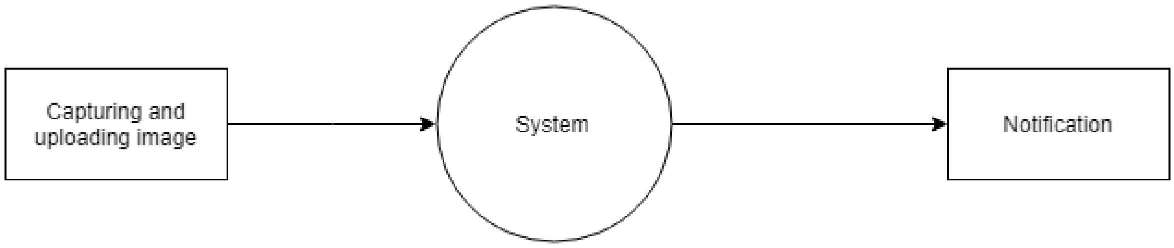

Fig. 3.5 Context Diagram

Chapter – 4

Implementation and Testing

4.1 User Interface and Snapshots

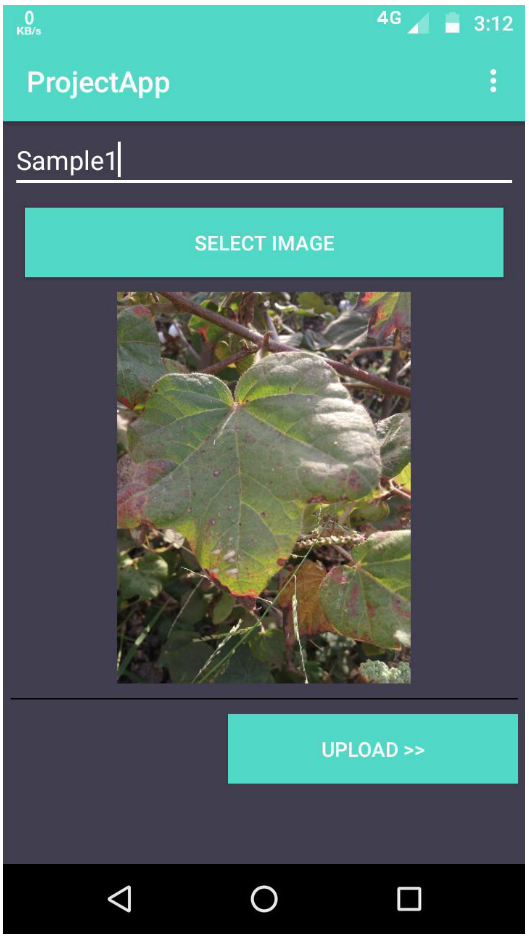

Fig. 4.1 Upload Image to server

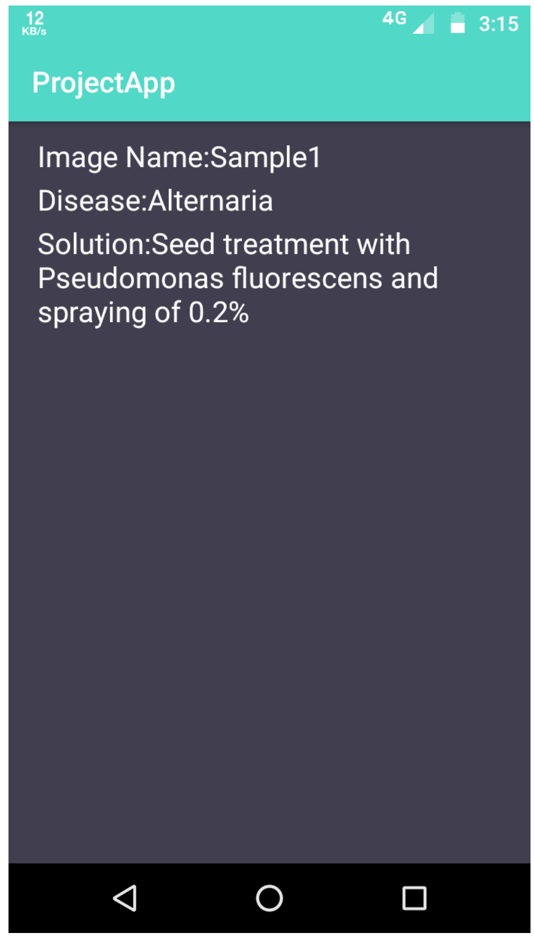

Fig. 4.2 Notification

4.2 Test cases and Result:

Test Case No.	Observed Disease	Disease predicted by image captured by drone	Disease predicted by image captured by drone from surface	Disease predicted by image captured by mobile	Classification
1	Alternaria	Alternaria	Alternaria	Alternaria	True
2	Healthy	Alternaria	Alternaria	Alternaria	False
3	Alternaria	Alternaria	Alternaria	Alternaria	True
4	Alternaria	Alternaria	Alternaria	Alternaria	True
5	Healthy	Healthy	Healthy	Alternaria	False
6	Bacterial Blight	Bacterial Blight	Bacterial Blight	Bacterial Blight	True
7	Cercospora	Cercospora	Cercospora	Cercospora	True
8	Cercospora	Healthy	Healthy	Cercospora	False

9	Bacterial Blight	Bacterial Blight	Bacterial Blight	Bacterial Blight	True
10	Cercospora	Cercospora	Cercospora	Cercospora	True

Table 4.1 Classification results for different images

4.3 Result Analysis:

The statistics of dataset used to develop this system is shown as below:

1. Total No. of images in the dataset - 275
2. No. of images for Alternaria disease - 40
3. No. of images for Anthracnose disease - 72
4. No. of images for Bacterial Blight disease - 72
5. No. of images for Cercospora disease - 34
6. No. of images for Healthy cotton - 57
7. No. of images used for training a model - 220
8. No. of images used for testing a model - 55

No.	Disease	Classification Accuracy
1	Alternaria	85%
2	Bacterial Blight	79%
3	Cercospora	70%
4	Anthracnose	31%

Table 4.2 Classification Accuracy for each disease

Conclusion and Future work

This system is developed for the farmers and researchers to early detection of disease in cotton plant by examining the leave. Through this system, we have tried to highlight the problems associated with the cultivation of cotton and causes of low yield loss in the developing countries like India. But there is still lot to do such as follows:

1. This system will detect only disease caused by fungi and bacteria. More disease can be included that occurred on the cotton ball.
2. By collecting more images for each disease we can improve accuracy.

So there is no limit to for further development of this system.

References

[1] Prakash M. Mainkar, Shreekant Ghorpade, Mayur Adawadkar, "Plant Leaf Disease Detection and Classification using Image Processing Technique" International Journal of Innovative and Emerging Research in Engineering, Volume 2, 2015

[2] Premalatha V., Valarmathy. S., Sumithra M.G., "Disease Identification in Cotton Plant Using Spatial FCM & PNN Classifier" International Journal Innovative Research in Computer and Communication Engineering, Volume 3, April 2015

[3] Anuradha Badage, "Corp Disease Detection using Machine Learning: Indian Agriculture" International Research Journal of Engineering and Technology, Volume 5, September 2018

[4] Reena Tijare, Pawan Khade, Rashmi Jain, "The Survey of Disease Identification of Cotton Leaf" International Journal Innovative Research in Computer and Communication Engineering, Volume 3, November 2015

[5] Sachin D. Khirade, A. B. Patil, "Plant Disease Detection Using Image Processing" International Conference on Computing Communication Control and Automation, 2015

[6] https://en.wikipedia.org/wiki/Gaussian_blur

[7] https://dzone.com/articles/introduction-to-perceptron-neural-network

[8] https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

[9] https://en.wikipedia.org/wiki/Histogram_equalization

[10] http://matlab.izmiran.ru/help/toolbox/images/enhanc15.html