Mobile Application For Determining Input Level Of Fertilizer And Detecting Diseases In Crops

By

Tasneem Tazeen Rashid
Thuza
Md. Sazzad Hossain

A thesis submitted in partial fulfillment of the requirements for the Bachelors In Science Degree

Department of Computer Science and Engineering
BRAC University, September 2014

Abstract

Economic development and food security in Bangladesh significantly depend on agriculture production. Despite progresses made in agricultural techniques, farmers in Bangladesh still face difficulties in determining required levels of fertilizer and detecting different diseases during cultivation period. Using adequate fertilizer during specific periods of production and early detection of diseases ensure effective and efficient use of fertilizer and good harvest of crops. On the other hand, mobile phone usage has increased exponentially among the population of Bangladesh.. People from all walks of life are using mobile phones and different associated applications for gaining economic and social benefits. However, very few mobile phone applications benefit agricultural production and specifically aim farmers.

This thesis is aimed at developing an android application to generate an automated mobile-based system to serve two purposes: a) determine the amount of input level of fertilizer for paddy, and b) detect diseases of potato through image processing of leaves. The purpose of such development was to assemble an application with a user friendly interface and implement some effective algorithms considering the problems.

In order to develop the application, different algorithms, approaches and languages were examined that can ensure a user friendly interface. Determining the input level of fertilizer for paddy was executed in android using “Principal Component Analysis” algorithm. On the other hand, two specific potato leaf diseases— “Early Blight” and “Late Blight”— were detected employing “Mathematical morphology”, “Leaf Vein Detection” and “Blob detection” algorithms in MATLab. The result showed that color of paddy leaf matched precisely with the LCC to determine the input level of Urea while the application successfully recognized affected parts in the potato leaf. Evaluating with the real time data, this thesis can be a very promising application to help the farmers with their harvesting procedure.

Acknowledgement

We would like to thank Md. Zahangir Alom for agreeing to supervise this thesis. His patience and confidence has been a great source of encouragement and without his guidance this paper would not have come to existence. His dedication to this paper deserves to be reciprocated with immense gratitude.

We would also like to thank Rubel Biswas for supporting as co-supervisor in this thesis.

A special thanks to the thesis committee for taking the time to review and evaluate this paper.

Last but not the least, a cordial thanks to Dr. Anil Kumar Das, Director of ICT, Department of Agricultural Extension, Ministry of Agriculture for motivating and providing useful data to conduct the thesis.

Table of Contents

List of figures:

List of tables:

Chapter 1: Introduction

1.1 Background of the problem

Major items of food production in Bangladesh include rice, potato, wheat, pulses, oilseeds, sugarcane and vegetables. These food items account for almost 80 percent of the total calorie and protein intake for the population of the country [39]. Among them, rice and potato forms a significant portion of staple food.

Rice:

Over the years, rice (*Oryza sativa*) production has shown significant growth due to adoption of high-yielding facilities, for example fertilizer and irrigation. Fertilizer, as a key input into modern agriculture, accounts for 15 per cent of the total cost of rice production [40]. Urea is usually used as fertilizer. Farmers use Urea in the highest quantity among all other fertilizers. Often untimely and irrational usages of urea not only impact on the production but also cause significant wastage of resources that could have been utilized for other purposes. The International Rice Research Institute (IRRI) developed a Leaf Color Chart (LCC) to help in calculating required Urea level for varieties of "BORO" and "AMON" paddy and reduce wastage. According to the Bangladesh Economic News, LLC could have saved 1.5 Lac tons of Urea during the production year of 2010 [41].

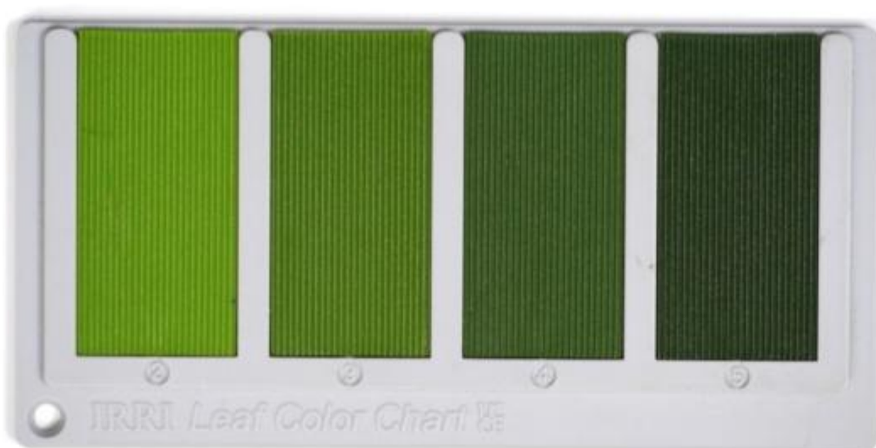


Figure 1: Leaf Color Chart

How LCC works:

International Rice Research Institute also suggested how to use the LCC [42].

- Color of the paddy leaf are matched with LCC; during "BORO" season starting from 21 days after transplantation and during "AMON" season15 days after transplantation as well as periodically after every 10 days till paddy become matured.
- For direct sowing of paddy grown from seeds in a muddy field, color of the leaf needs to be matched after 25 days in "BORO" season and after 15 days for "AMON" season.
- In each occasion 10 healthy leaves have to be matched.
- The healthy leaves are the ones that are youngand completely grown. Color of their middle part need to be compared with LCC.
- For best result LCC should be used between 9.00 to 11.00 AM and 2.00 to 4.00 PM.
- Standardized LCC consists of four color shades from yellowish green to dark green with corresponding numbers 2, 3, 4, and 5 which correspond to different LCC value.
- If color any leaf match between two options, then the value is considered as an average between the two. For example if the color of the leaf matches options between and 4, the LCC value will be 3.5.
- For transplanted paddy, if LCC value is 3.5 or less than 3.5; and for sowed paddy if LCC value is 3 or less than 3 for six or more leaves, 7.5 Kg of Urea will be needed for "BORO" and 9.0 Kg of Urea will be needed for "AMON" for each acre of land.

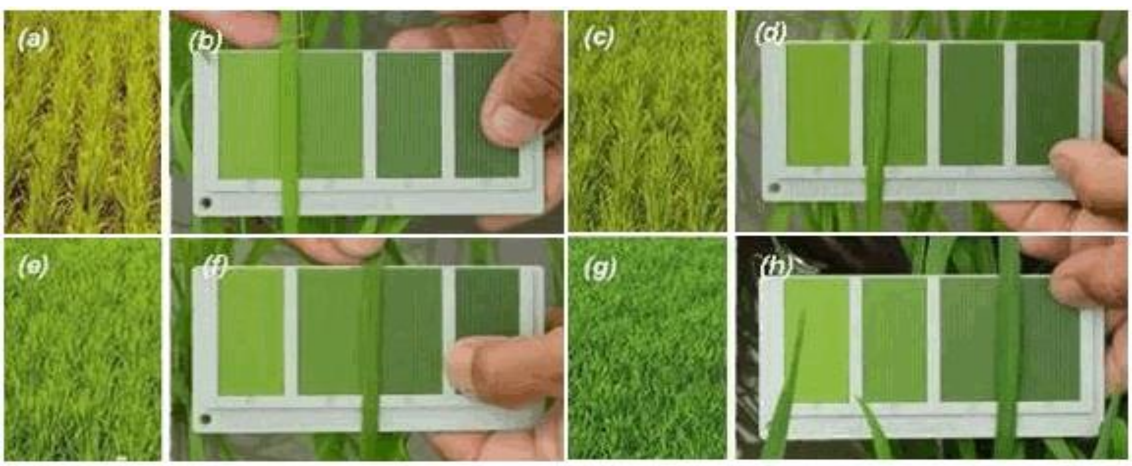


Figure 2 : Matching different paddy leaves with LCC

Potato:

Potato (*Solanum tuberosum*) is another staple food in developed countries which accounts for 37 per cent of the total potato production in the world [43]. In Bangladesh, farmers produce around 82 lac tons of potato each year against an annual consumption of 60 to 65 lac tons [49]. Generally, 30 to 50 per cent of the production is lost due to different diseases [5]. “Early Blight” (caused by the fungus *Alternariasolani)* and “Late Blight” (caused by the water mold *Phytophthorainfestans)* are the two common yet deadly diseases of potato [44] [45]. These diseases cause a high amount of loss in the total potato production in the country. Some scientists have come up with a potato variety resistant to Late Blight diseases [47]. Unlike paddy, there are no easily available methods to detect different diseases in potato.

Symptoms of “Early Blight”:

Different symptoms indicate whether the crop is being affected by the disease or not [1]. These include:

- Small, irregular to circular dark brown spots on the lower leaves appears as foliar symptoms. These spots may be sized from a pinpoint to 1/8 inch in diameter.
- As the spots enlarge, they become restricted by leaf veins and take on an angular shape.
- Early in the growing season, those lesions may become larger – up to 1/2 inch in diameter and may seem like a series of dark concentric rings alternating with bands of light tan tissue
- A band narrow in shape of chlorotic tissue sometimes surrounding each lesion, and extensive chlorosis of infected foliage develops over time.
- At the closing stage of the growing season, the upper leaves of infected potato plants may be sprinkled with numerous small early blight lesions and gradually, lesions may join together to cover a large area of the leaf.
- Heavily infected leaves eventually wither and die but usually remain attached to the plant.

Figure 3 : Leaf affected by the disease "Early Blight"

Symptoms of "Late Blight":

Similarly, different symptoms indicate whether the crop is being affected by "Late Blight" or not [2]. These include:

- Small, circular to irregular in shape lesions, which show up 3 to 5 days after infection in the leaves with purple ranged color.
- The lower leaves are primarily affected, where the micro climate is more humid.
- Nevertheless, if weather conditions are favorable and the pathogen has been carried into the field by air currents they may occur on upper leaves.
- Generally development of the lesions begins to on the compound leaf near the point of attachment to the petiole (which is often cupped) or at the leaf edges, where dew is retained longest.
- In time of cool, moist weather, lesions expand rapidly into large, brown or black dark spots, often surrounded by a pale green to yellow border.
- Lesions are not bounded by leaf veins and they can cause young expanding leaves can be misshaped if formed at leaf tips or edges.
- Entire leaves may become blighted and killed within a few days as new infections occur and existing lesions coalesce.

- The lesions continue developing along the length of the stem and can remain active even in hot, dry weather.

Figure 4 : Leaf affected by the disease “Late Blight”

1.2 Motivation

Realizing the significance of the problems associated with paddy and potato productions with an intention to contribute to agricultural development in Bangladesh motivated this work to develop a user friendly application for the persons associated with agriculture development in Bangladesh. Our supervisor Md. Zahangir Alom was a great inspiration to proceed. A special inspiration came by Dr. Anil Kumar Das, Director of ICT, Department of Agricultural Extension, Ministry of Agriculture, who inspired and provided some information related to this thesis.

1.3 Prevision works on the problem

Some authors have worked on the problems addressed in this thesis; Table 1 summarizes some of those publications.

Table 1: List of related publications

Discussed topic	Author(s) andpublication year	Name of the publication	Algorithms used
Input level of fertilizer	Joshua, Van Zesar, Philip John, Ricardo Stoles, Stephen R Tatel, Andy V 2012	Android-based image processing application for rice nitrogen management.	Leaf Color Chart Color Histogram analysis Pixel bitwise operation Z-test one proportion
	Ruprah,Taranpreet Singh 2012	Face recognition based on PCA algorithm.	Neutral networking Back propagation Principal component analysis Euclidean distance
Disease detection	Naikwadi, Smita Amoda,Niket 2013	Advanced image processing for detection of plant disease.	Color co-occurrence method Neutral network
	Anthonys, G. Wickramarachchi,N. 2009	An image recognition system for crop disease identification of paddy fields in Sri Lanka	Mathematics Morphology CIE *a*b* color space Color texture Membership function

There are many face recognition proposals employing PCA. Ruprah (2012) approached to apply PCA algorithm in two systems. One was to compare the Euclidean distance and the other one was Histogram Equalization Technique. The result of the system made an emphasized Euclidean Distance classifier rather than the histogram technique.

Joshuaet. al. (2012) proposed to implement Leaf Color Chart (LCC) on Android OS using Histogram Analysis and pixel bitwise operation with a greater accuracy. It is to be noted that it was one of the very few approaches to apply image processing for implementing LCC on an Android-Based device. The outcome was the project provided particular upgrades necessary for the improvement of the LCC in future application.

Naikwadiand Amoda (2013) proposed and experimentally evaluated a solution based on software for automatic disease detection and classification of plant leaf. In order to achieve the goal, they took two steps successively after the segmentation phase. In the first step, they identified the mostly green colored pixels. The next step of masking pixels are based on specific threshold values that are computed using Otsu's method. Mostly the green pixels are masked.

They used histogram technique and edge detection technique in order to find the shape. As their result illustrate they were able to generate an automated output with a satisfying accuracy.

Earlier,Anthonysand Wickramarachch (2009)approached to create an automated image processing for detecting paddy diseases. They started their image processing with some digital images of leaf infected with paddy disease. They applied a method of mathematics morphology to segment these images. Texture, shape and color features of color image of disease spot on leaf were extracted after that. A classification method of membership function was used to discriminate between the three types of diseases. Their main goal was to recognize some of the paddy diseases in Sri Lanka using image recognition techniques. The results obtained on real data were comparatively satisfactory (more than 70 per cent recognized). The rate of the recognition time was less than 2 seconds.

Bryan S. Morse, Brigham Young University, 1998–2000, introduced some problems regarding threshold. There was also some discussion about find threshold values in some methods. But the results were not always promising.

1.4 Methodology

The aim of this thesis was to help field level workers in the agricultural sector by calculating input level of fertilizers of paddy and detecting diseases of potato through an android application.

In order to develop the application, different algorithms, approaches and languages were examined that can ensure a user friendly interface. To proceed with the detection steps of input level of fertilizer, the main approach was to apply the Principal Component Analysis also known as PCA algorithm with a view to recognizing the color level of any requested image by the user with the Leaf Color Chart (LCC) provided by IRRI. It was aimed to implement this PCA algorithm in android operating system creating a new library in JAVA for applying the algorithm.

On the other hand, for the detection of two specific potato diseases - “Early Blight” and “Late Blight” - “Blob detection”, “Mathematical morphology”, “Shape Detection and Color

Extraction" , "Leaf Vein Detection" and "Detect diameter of the Blob" algorithms were employed in MATLab.

Thus the framework of the application suggests different application steps to serve the two purposes:

Application steps of detecting input level of fertilizer:

Step 1. Select option.

Step 2. Select image of the chosen leaf from the camera, crop to fix height and width.

Step 3. Repeat step 2 for ten times in total.

Step 4. Press send button after finishing step 3.

Step 5. Application will send information to the backend code of PCA algorithm.

Step 6. A message will generate after completing the compilation of the code.

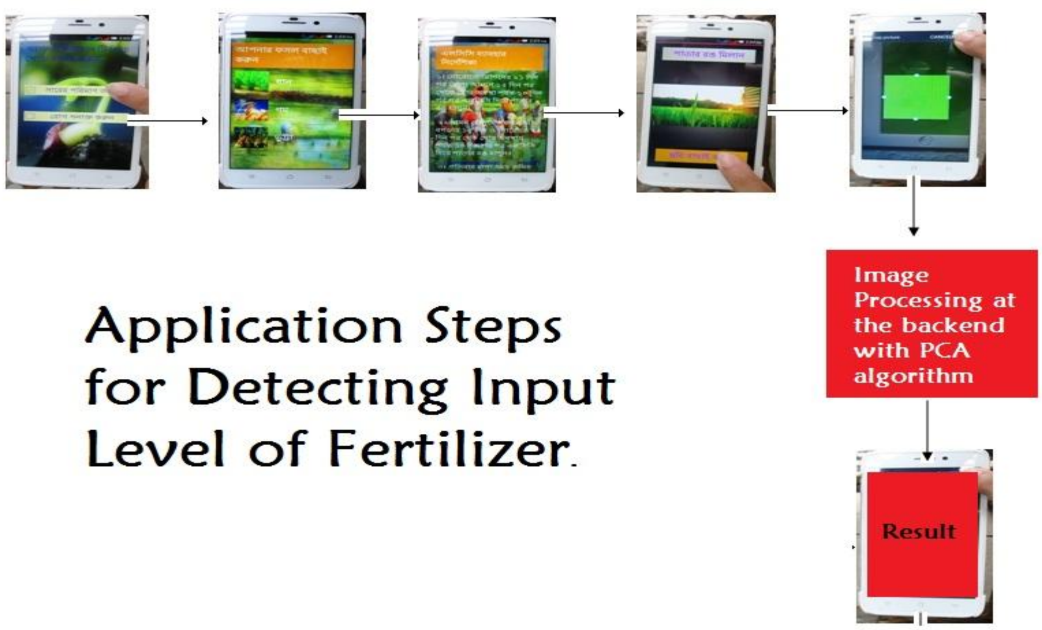


Figure 5: Steps of detecting input level of fertilizer

Application steps of detecting leaf diseases in potato:

Step 1. Connect the phone with internet.

Step 2. Select option.

Step 3. Select image of the affected leaf from the camera.

Step 4. Press send button after finishing step 2.

Step 5. Application will connect to a server to save the image and run an "index.php" file.

Step 6. The PHP file will run an exe file of the MATLab code through shell of the server.

Step 7. Inside the MATLab code features will be extracted through image processing.

Step 8. MATLab code will return a text message and load it to another PHP file.

Step 9. With JSON parsing the text message will be retrieved and delivered to the user.

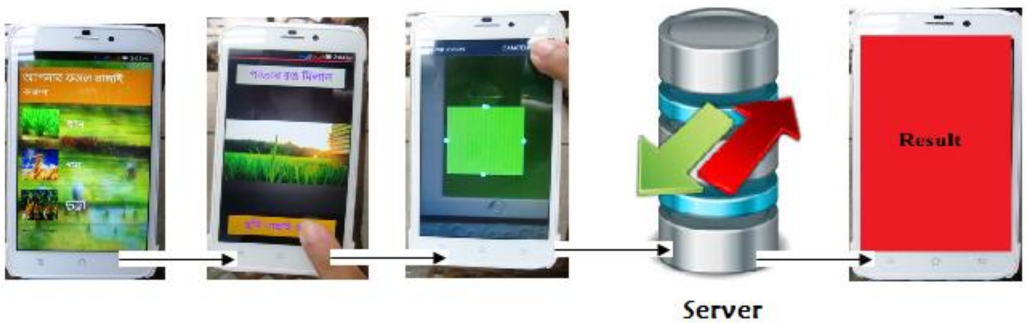


Figure 6 : Steps of Detecting leaf diseases in potato

1.5 Outline of the thesis

The thesis is organized in 9 chapters. After the introduction in first chapter, chapter 2 compiles a literature review that helped in conceptualizing the application to address the problem and different concepts associated with the application. Chapter 3 describes the proposed system of the application while chapter 4 describes some details of some experimental approaches. Chapter 5 discusses the results. The concluding chapter 6 summarizes the findings and future work possibilities are indicated in chapter 7.

Chapter 2: Literature Review

2.1 Determining input level of fertilizer for paddy

"Understanding Principal Component Analysis Using a Visual Analytics Tool"

by Dong Hyun Jeong, Caroline Ziemkiewicz, William Ribarsky and Remco Chang

[6]This paper emphasized on Principle Component Analysis (PCA) as a mathematical procedure mostly used in data analysis and also signal processing. The results and procedures are difficult to understand and so it is often judged as a black box operation whose. The paper mostly focused to provide an elaborate explanation of PCA algorithm, based on a designed visual analytics tool which visualizes the results of principal component analysis. It also supports a set of interactions to help the user in better understanding and utilizing PCA. It had the relationship between PCA and single vector decomposition (SVD), the method used in our visual analytics tool as introduction. After that, a detailed description of the interactive visual analytics tool, including advantages and limitations, is provided. [7] tells that Principal Component Analysis is central to the study of multivariate data.

"Practical Approaches to Principal Component Analysis in the Presence of Missing Values"

by Alexander Ilin, Tapani Raiko

[8]This paper introduced Principal component analysis (PCA) as a data analysis technique (classical) that figured out linear transformation of data that retain the maximal variance that occur. They studied a case where some of the values of the given data were missing. It also showed many properties of the issue which were mostly relevant to nonlinear models. For instance, over fitting and incomplete locally optimal solutions were two of them. A probabilistic formulation of PCA provides a good foundation for handling missing values. They proposed formulas for achieving such goal. In case of high dimensional and greatly sparse data, a severe

problem occurs as over fitting. However, traditional algorithms for PCA are very slow. So, they introduced an algorithm which they claimed to be novel fast and extended that to variant Bayesian learning. Distinguished versions of PCA are compared in experiments which are artificial, demonstrating the regularization effects and posterior variance was modeled. By applying it to the by applying it to the Netflix problem the scalability of the proposed algorithm was demonstrated.

2.2 <u>Disease Detection for potato</u>

“Blob Detection”

by Anne Kaspers

[12]The author in this paper claimed that detecting low-level objects in an image was a primary. Such low level shapes were named as blobs. They can both be 2D or 3D. Blobs can be found in different ways. It relies on their scale and can be detected using local operations if the image can be represented as multi-scale. The paper showed several blob detection methods. After that, there was an effort to make a fair comparison without performing any kind practical experiments. It showed that blobs can be defined and localized in different ways. While approaching that, several methods were used and each of them had its own ability and potential.

“A Robust Blob Recognition and Tracking Method in Vision-based Multitouch Technique”

by Feng Wang, Xiangshi Ren, Zhen Liu

[17]The basic proposal of vision-based multi-touch approach was image processing with high performance ability. The paper introduced two different algorithms were presented to resolve the problems of blob detection as well as tracking their position. One of the algorithms was image contour transformation algorithm. It was adopted to calculate the existence of finger touch area. Another algorithm was Minimum Distance First (MDF) algorithm which was modeled to

identify and track the corresponding blobs in two consecutive images. The result of the paper was regarding whether the efficiency of those two algorithms fully met the requirement of any real time and practical multi-touch system.

"Morphological Grayscale Reconstruction in Image Analysis: Applications and E client Algorithms"

by Luc Vincent

[22]In this paper Morphological reconstruction is described as a part of a set of image operators. It was sometimes referred to as geodesic. When the case is binary,, reconstruction the connected components of a binary image I (the mask) which are marked by a (binary) image J contained in I are simply extracted. It was proposed that the transformation was possible to be extended to the grayscale case. The proposed it because they thought that it might turn out to be very useful for analysis tasks of several images. At the very beginning of the paper two formal definitions of grayscale reconstruction was provided. After that, use of grayscale reconstruction in various image processing applications was introduced. It aimed at demonstrating the usefulness of this transformation for image filtering and segmentation tasks. In the final segment, the paper focused on implementation issues such as the standard parallel and sequential approaches to reconstruction, their common drawback etc. With a view to developing this scenario, a new algorithm was propose, which was based on the preference of regional maxima creating use of breadth-first image scanning. It was implemented with a queue of pixels. Its merging with the previous technique resulted in a hybrid grayscale reconstruction algorithm.

"An Implementation of Leaf Recognition System using Leaf Vein and Shape"

byKue-Bum Lee and Kwang-Seok Hong

[23]The paper proposed and implemented a leaf recognition system using the leaf vein and shape. Fast Fourier Transform (hereinafter, FFT) methods with distance between contour and centroid on the detected leaf image was used by the proposed approach. Total 21 leaf features were extracted for the leaf recognition. They included the distance feature between centroid and all points on the leaf contour and frequency domain data by FFT that was performed using the

distances. In short, 10 features of all the 21 leaf features were extracted using distance, FFT magnitude, and phase, the other 10 features were extracted using the digital morphological features using four basic geometric features and five vein features, and the last one feature was extracted using the convexhull. For the verification of the approach, images of 1907 leaves were applied to classify 32 kinds of plants. Practical results showed that, the proposed leaf recognition system showed an average recognition rate of 97.19 per cent. They confirmed that the recognition rate of the proposed leaf recognition system was better than that of the existed leaf recognition method which was very strong claim.

2.3 Different concepts associated with the application

2.3.1 Principal Component Analysis:

Principal Component Analysis is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components.

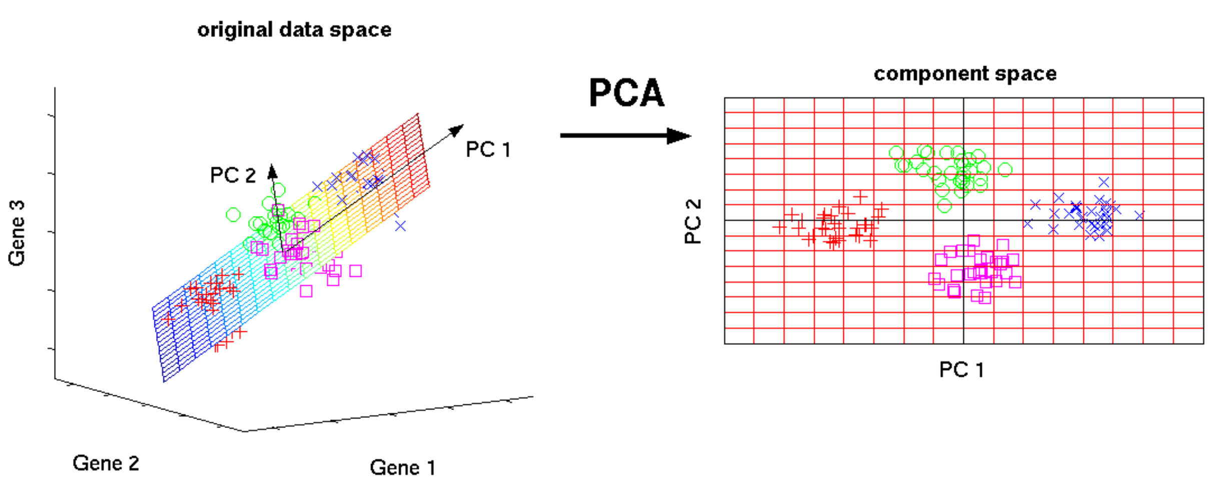


Figure 7 : Principal Component Analysis (PCA)

Basically, PCA projects a high dimensional data into lower dimensional subspace. PCA steps [12]: Transform a N X d matrix X into an N X m matrix Y

- Subtract the mean and calculate the covariance matrix C:

  Steps of calculating Covariance matrix (d X d matrix):

  - ✓ $C = \frac{1}{N-1} X^T X$
  - ✓ $C_{i,j} = \frac{1}{N-1} \sum_{q=1}^{N} X_{q,i} . X_{q,j}$
  - ✓ $C_{i,i}$ (diagonal) is the variance of the variable i.
  - ✓ $C_{i,j}$ (off-diagonal) is the covariance between variables i and j.
- Calculate the Eigenvectors of the covariance matrix (Orthonormal)
  - ✓ If A is a square matrix, a non-zero vector v is an eigenvector of A if there is a scalar λ (eigenvalue) such that

    $$Av = \lambda v$$
- Select m eigenvectors that correspond to the largest m eigenvalues to be the new basis.

[41]In MATLab this PCA algorithm can be used for face recognition system. This paper has followed and implemented steps of face recognition system[3] only with a modified version in JAVA which will be discussed later in this write up.

2.3.2 Blob detection

In the field of computer vision, blob detection refers to mathematical methods that are aimed at detecting regions in a digital image that differ in properties, such as brightness or color, compared to areas surrounding those regions. Informally, a blob is a region of a digital image in which some properties are constant or vary within a prescribed range of values; all the points in a blob can be considered in some sense to be similar to each other [49]. [42]One of the algorithms to detect blob is Linderberg's watershed-based grey-level blob detection algorithm.

For the purpose of detecting grey-level blobs(local extreme with extent) from a watershed analogy, Lindeberg developed an algorithm based on pre-sorting the pixels, alternatively connected regions having the same intensity, in decreasing order of the intensity values. Then, comparisons were made between nearest neighbors of either pixels or connected regions.

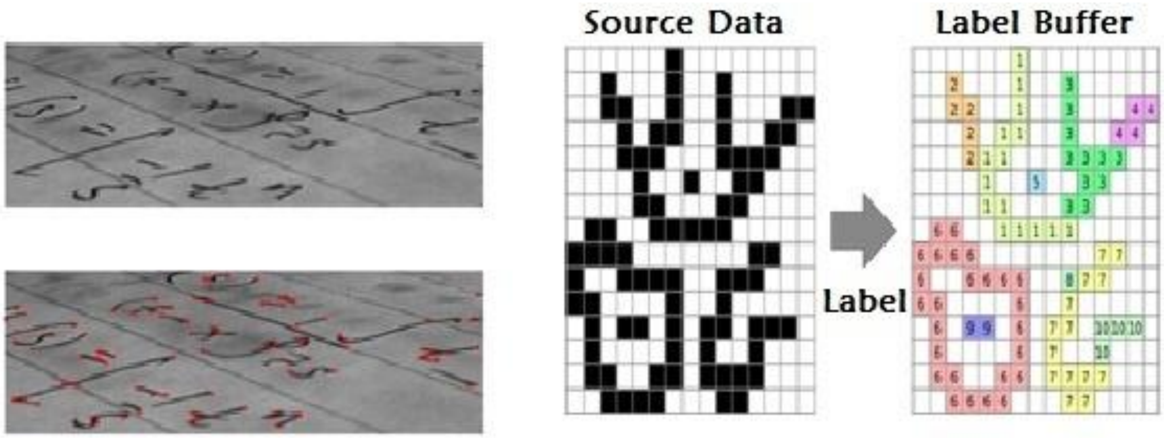


Figure 8 : Blob Detection with connected regions

2.3.3 Mathematical morphology

Mathematical morphology is a theory and technique for the analysis and processing of geometrical structures, [31] based on set theory, lattice theory, topology and random functions [49]. In binary image morphology, an image is viewed as a subset of a Euclidean space $\mathbb{R}^d$ or the integer grid $\mathbb{Z}^d$, for some dimension d.

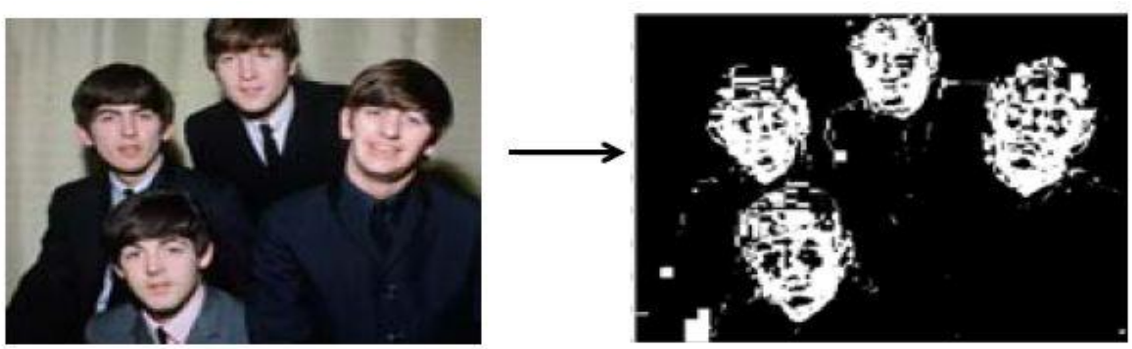


Figure 9 : RGB to binary conversion of an image for mathematical morphology

Structuring element:

The basic idea in binary morphology is to probe an image with a simple, pre-defined shape, drawing conclusions on how this shape fits or misses the shapes in the image. [37]This simple “probe” is called structuring element and is itself a binary image. Some examples of structuring elements denoted by B:

- Let $E=\mathbb{R}^2$; *B* is an open disk of radius *r*, centered at the origin.
- Let $E=\mathbb{Z}^2$; *B* is a 3x3 square, that is,*B*={(-1,-1), (-1,0), (-1,1), (0,-1), (0,0), (0,1), (1,-1), (1,0), (1,1)}.
- Let $E=\mathbb{Z}^2$; *B* is the "cross" given by:*B*={(-1,0), (0,-1), (0,0), (0,1), (1,0)}.

Basic operators:

There are four basic operators:

- **<u>Erosion:</u>**

  Erosion of the binary image A by the structuring element B is defined by:

  $A \ominus B = \{z \in E | B_z \subseteq A\}$ where B_z is the translation of B by the vectors, i.e., $B_z = \{b + z | b \in B\}$, $\forall z \in E$ .When the structuring element B has a center (e.g., B is a disk or a square), and this center is located on the origin of E, then the erosion of A by B can be understood as the locus of points reached by the center of B when Bmoves inside A.

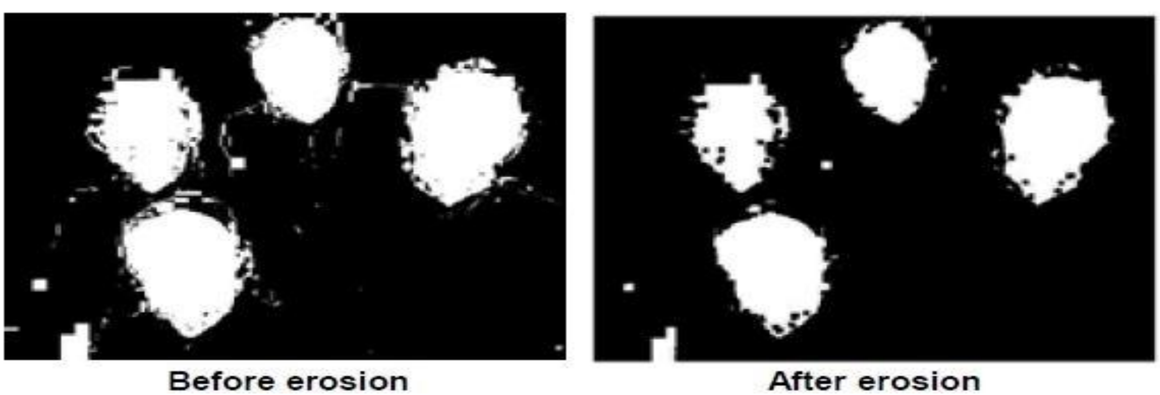


Figure 10 : Example of erosion.

- **<u>Dilation:</u>**

  Dilation of A by the structuring element B is defined by:

  $$A \oplus B = \bigcup_{b\in B} A_b .$$

The dilation is commutative, also given by:

$$A \oplus B = B \oplus A = \bigcup_{a \in A} B_a.$$

If B has a center on the origin, as before, then the dilation of A by B can be understood as the locus of the points covered by B when the center of B moves inside A.

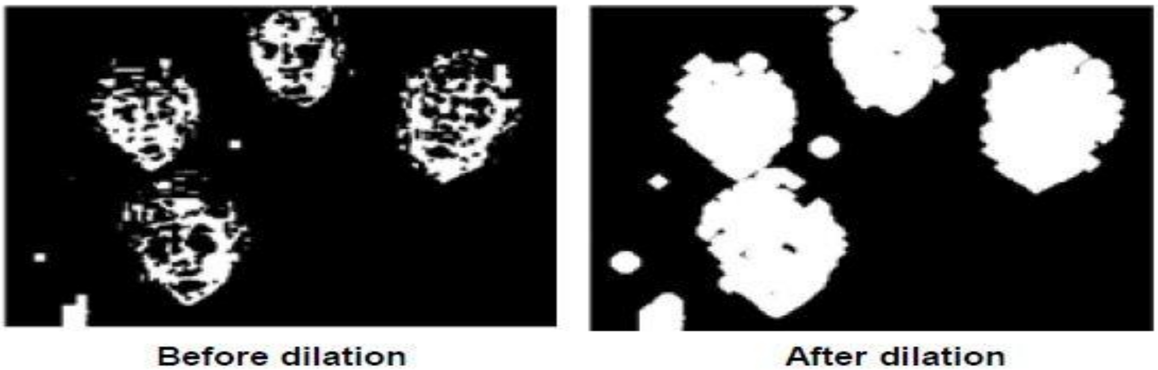


Figure 11 : Example of dilation.

- **<u>Opening:</u>**

  Opening of A by B is obtained by the erosion of A by B, followed by dilation of the resulting image by B:

  $$A \circ B = (A \ominus B) \oplus B.$$

  The opening is also given by

  $$A \circ B = \bigcup_{B_x \subseteq A} B_x$$

  ,which means that it is the locus of translations of the structuring element B inside the image A.

- **<u>Closing:</u>**

  Closing A by B is obtained by the dilation of A by B, followed by erosion of the resulting structure by B:

  $$A \bullet B = (A \oplus B) \ominus B.$$

  The closing can also be obtained by $A \bullet B = (A^c \circ B^s)^c$, where X^c denotes the component of X relative to E (that is, $X^c = \{x \in E | x \notin X\}$).

2.3.4 Color extraction

Color extraction is a process where value of each pixel of the image is the key component to retrieve the color of any object. Generally, an image can be in different forms: RGB form (3 dimensional), Grayscale form (2 dimensional), Binary form (1 dimensional). Grayscale and binary form of an image is basically a restrained transformation of RGB form. So depending on various structures, color extraction is different. For RGB form, values of Red, Green and Blue, for Grayscale form gray shades ranging white to black and for Binary image only Black and White are extracted.

2.3.5 Leaf Vein detection

Most of the time, physiological and morphological conditions of any plant is reflected to its leaf. To be precise in this thesis, veins are considered as one of the parameters to distinguish two different types of blight in the potato leaf.

2.3.6 Shell execution in PHP

In computing, a shell is a user interface for access to an operating system's services. In general, operating system shells use either a command-line interface (CLI) or graphical user interface (GUI), depending on a computer's role and particular operation [49]. Shell of a computer can be open and run a program by a PHP instruction. In this thesis as computer will be used as a server, this operation was implemented.

2.3.7 Android camera setting

Working of android camera smoothly is very essential in the implemented Android application since it would get the input of images from the user. Android has a camera API or camera Intent available which captures images easily.

Chapter 3: Proposed system implementation

3.1 Determining fertilizer level of paddy

3.1.1 Data acquisition:

For image processing, accumulating database is the most important part. To proceed with the approach, at the beginning the LCC chart was collected from the Department of Agricultural Extension located in Khamarari, Dhaka. As LCC has four different color components, picture of each component was taken at different environment numbers of time. Environment was ensured to be different by two means: taking images during different time of the day and with different devices (mobile phone). As per instruction by IRRI to get the best result LCC should be used between 9.00 to 11.00 AM and 2.00 to 4.00 PM, data acquisition followed this suggestion [4].

3.1.2 Camera setting

For detecting the input level of fertilizer camera of the application was modified in a special way. The user had to take the picture inside a fixed crop tool. This crop tool was programmed in android manually. The crop tool was fixed with 200x200 pixels. Inside the crop tool, user had to take the picture of the paddy leaf. This was set so that no environmental noise may hamper the classification of the process. Adding to that, the camera settings in android was set such that, the program would turn on the camera for ten times and take pictures.

3.1.3 Data representation:

Images were stored in a matrix of rows and columns with format. When the image was taken it was in RGB 3D representation form. It was vectorized in 2D grayscale form for further calculation. After running the eigenfacecore algorithm three files named “A.txt”, “m.txt” and “eigenfaces.txt” were generated. They were saved in the resource directory of android. From there, while compiling the code, values were imported. Once the PCA algorithm was finished compiling, the result was stored in a text file consisting of an array for ten different images in the resource directory of android. Later, comparing the value the final result was shown to the user.

3.1.4 Principal Component Analysis:

PCA algorithm in this paper was subjected through three processes.

Creating train database

At the very beginning a set of images are taken to train the system for further recognition of any unknown image. As discussed previously images are stored in 3D matrices from the camera. For deploying the algorithm the images need to be vectorized and stored in a matrix as database. In this step, RGB to Grayscale conversion is obtained. There is a formula to convert the RGB values into a Grayscale range which is [3]:

$$X = 0.299 \times R + 0.5870 \times G + 0.1140 \times B$$

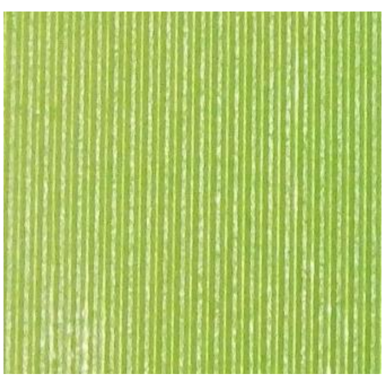
Here, X = Grayscale value,

R = Red value of a certain pixel

G = Green value of a certain pixel

B = Blue value of a certain pixel

Each image of the train database will be stored in one column which will be the factor indicating the image number for recognition.



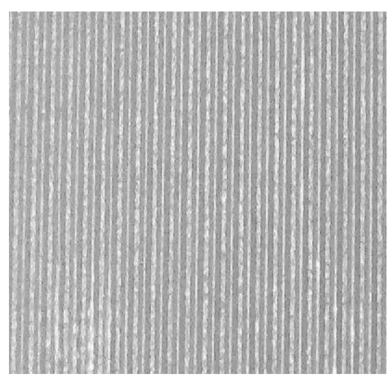


Figure 12: RGB image to Grayscale image

Calculating EigenfaceCore

This is the most important part of the process. This function gets the 2D matrix containing all the training image vectors. And return three 3 outputs [4]. They are:

- Mean matrix: It takes the same pixel's value from all the images in train database and calculates the mean of them to store in a new matrix called "mean".

- A matrix: this matrix is the centered image vectors. Calculation of this matrix shows at what distance from the mean matrix each train images are standing.

"Eigenfaces" matrix: PCA is done by the decomposition of covariance matrix. Principal components are calculated by covariance matrix, by:

$$C = X.X^T$$

Reference to this paper X is basically the each vectorized image. However, this results a large matrix. Thus, to overcome this, a principle called "Snapshot Eigenface method" is followed. To reduce the matrix dimension a surrogate of the covariance is calculated:

$$L = X^T.X$$

Eigen values (D) of L are calculated by and the Eigen vector (V) is calculated with the correspondent formula:

$$X.V = V.D$$

Eigen values of are calculated and less than a certain threshold value they are sorted. Thus number of nonzero vectors maybe less than the number of training images.

At the end, all the eigenvectors of covariance matrix is the Eigen faces. With the below formula:

$$\text{Eigenfaces} = A * L\_eig$$

In this thesis, only the calculation of Eigen values and Eigen vector was taken from MATLab, as for fertilizer recognition only the three text files are need. It did not even need to save the entire train database as images at the backend. All the collected data were vectorized and sent to generate with the further formulas.

For recognition with the test database part of a test image is taken from the user and vectorized it.

Then all the centered images are projected into a face space by multiplying in Eigenface basis.

$$ProjectedImage = Eigenfaces * A(one\ column\ at\ time)$$

Again, following the same procedure as calculating train database difference between test image and mean of the entire train images should be subtracted. Along with that test image is projected in the face space.

At the end of the PCA steps Euclidian distance between projected test image and projection of all the images are calculated and with the minimum distance with the test image, train image is recognized as the expected result.

In this thesis paper LCC value is set for a range of train images. If the test image is near any of the test image the correspondence value of LCC is stored as result.

Most importantly, recognition process is continued up to ten times to follow the instruction of LCC measurement.

3.1.5 Template matching

Another approach was implemented to experiment the fertilizer recognition system. This approach was defined by few steps:

- Images were taken as train data
- Each image was converted to grayscale
- Each pixel was added and an average was calculated.
- This calculation is called for each shade of LCC.
- The in between mean was also calculated.
- While testing an image given by the user was converted in the same way and compared.

3.2 Disease detection for potato

3.2.1 Data acquisition:

In the case of disease detection for potato Dr. Mahbuba Begum, Senior Officer, Plant pathology, Tuber Crops Research centre, Bangladesh Agricultural Research Institute was contacted as the primary source of collecting database. This division of the paper does not require any stored data. All the data found from all the sources were used to determine the characteristics of each disease. After that feature extraction helped to gain more knowledge to determine “Early blight” or “Late blight” of potato.

3.2.2 Camera setting

Like in the part of determining level of fertilizer, this part also requires special camera settings. Intention of this thesis was to deploy a camera which will crop and capture an image of the effected region to remove some noise of the image at the base level. Though crop and capture option is still under development in this thesis, capture and crop feature of the camera was obtained perfectly.

3.2.3 Data representation

Images are read in different ways on the basis of requirements. For color extraction RGB color image was in need, for blob detection image was transferred to binary image also grayscale image was used to detect the leaf veins.

3.2.4 Equation generation for detecting diseases

The symptoms of each disease were categorized into five different categories and through some logical reasoning an equation was generated to determine whether the leaf is affected by “Early Blight” or “Late Blight”. The logical components were set at some initial conditions. They are:

1. Roundness, R $= \begin{cases} \text{True , if blob is round} \\ \text{False , if blob is not round} \end{cases}$

2. Color, Cl $= \begin{cases} \text{True , if blob is Brown} \\ \text{False , if blob is Purple} \end{cases}$

3. Centric Ring, Cr $= \begin{cases} \text{True , if blob is centric ring is found} \\ \text{False , if blob is not centric ring is found} \end{cases}$

4. Leaf Vein, L = $\begin{cases} \text{True , if blob is limited by the vein} \\ \text{False , if blob is not limited by the vein} \end{cases}$

5. Diameter , D = $\begin{cases} \text{True , if diameter of the blob is 1/8 inch} \\ \text{1/8 inch} \end{cases}$

From the above logics a K-map was constructed and an equation was generated. Adding to that the equation was reduced as below:

$$Y = Cr + R'Cr'V + RCr'Cl'V + Cr'Cl'V'(R \oplus D)'$$

3.2.5 Blob detection

Blob detection is done for two different images, one for RGB image and one for binary image.

RGB Binary blob detection:

RGB blob detection is specifically implemented to get the effected regions out of the color leaf image. This is because; one of the primary distinguishing feature between "Early blight" and "Late blight" is color components of the affected region. Generally, early blight's affected regions vary in between the brown range.

To do this blob detection the green pixels of the image is masked with respect to a thresholding value. This thesis paper came up with a unique thresholding formula with does not need any method to calculate threshold. Pixels are masked only with a simple condition:

(image(i,j,1)<image(i,j,2)) && (image(i,j,2) > image(i,j,3))

This means, let image is taken by the user, i and j are pixel values of row and column. By 1, means value of red color of that pixel, 2 means green and 3 means blue color of the picture. Generally for green pixel, green value is always higher than red value and blue value.

After finishing the thesholding all the regions are extracted with binary blob detection. Projecting that binary blob detection into the RGB image colored regions are extracted. From there an average value of R, G and B is calculated and matched with the database created for different disease containing rages of colors for different diseases.

Figure 13: RGB Blob detection

Binary blob detection:

In this section, same threshold condition is taken and turned the green pixels into black and affected region to white, causing a binary image transformation and morphological analysis for blob detection is done.

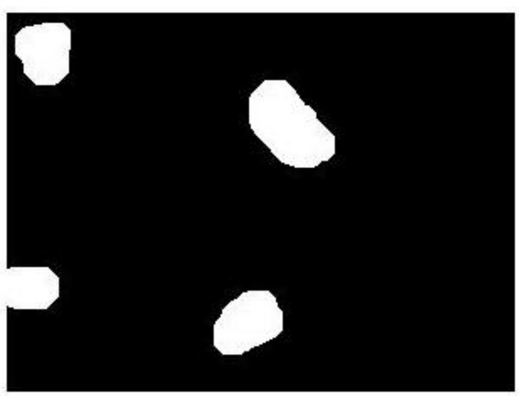


Figure 14: Binary Blob Detection

3.2.6 Morphological analysis for blob

Through binary blob detection, the threshold binary image is obtained. Then filling method was applied to fill up any wholes inside the white affected region. Morphological analysis for blob detection followed by erosion, dilation and labeling connected region.

Erosion:

As disease detection part is employed in MATLab, default erosion method is used. It was observed that more or less affected regions were oval to disk shaped. So the image was probed with structuring element SE of 'disk' value with 10 times. This value was determined by experimenting all the database of images. After that images were eroded with the specific structuring element. At this point all the noises are demolished.

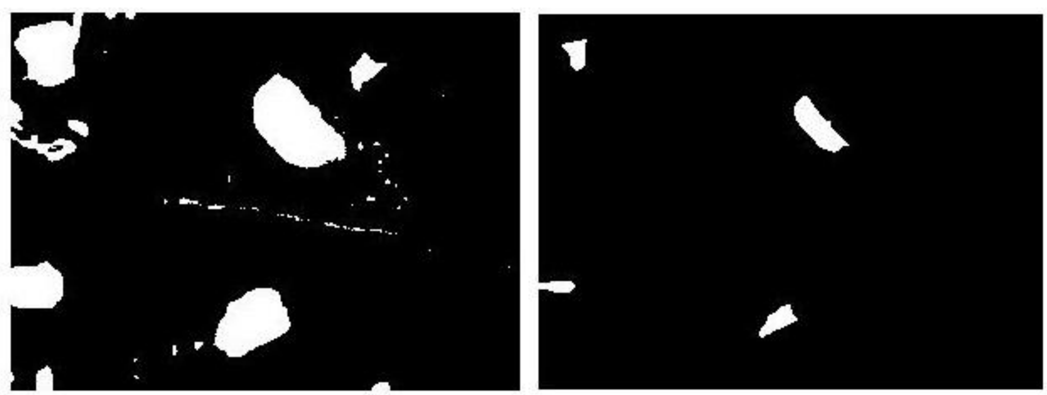


Figure 15: Erosion in Binary image

Dilation:

Dilation means expanding the shapes and probing with the structuring element. The same structuring element used to erode was used to dilate. Basically, at first image is reducing the noise with 'disk' probing through erosion. On the other hand, to get back the actual shape of the non-noise parts dilation is done.

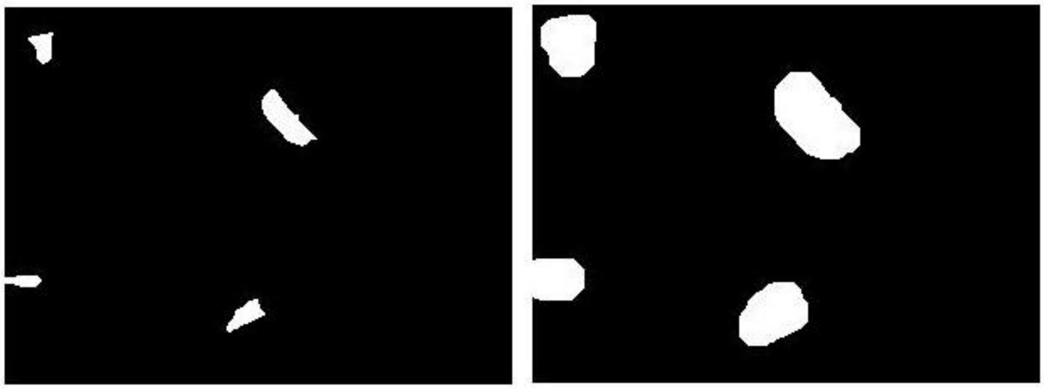


Figure 16: Dilation from image gone through erosion

Labeling the connected regions:

Labeling is done in binary image. This basically finds the connected components of a binary image. All the pixels in connected components are given by a label. The searching of the connected components is done in the column-wise fashion, in other words, in top-bottom scan order. To continue with the process, a number should be given as a parameter either 4 or 8. In MATLab 8 is the default value which defines which pixels are considered neighbors. The method is given below:

```
bwlabel(binary_image,8)
```

This method returns the connected regions and number of the connected regions.

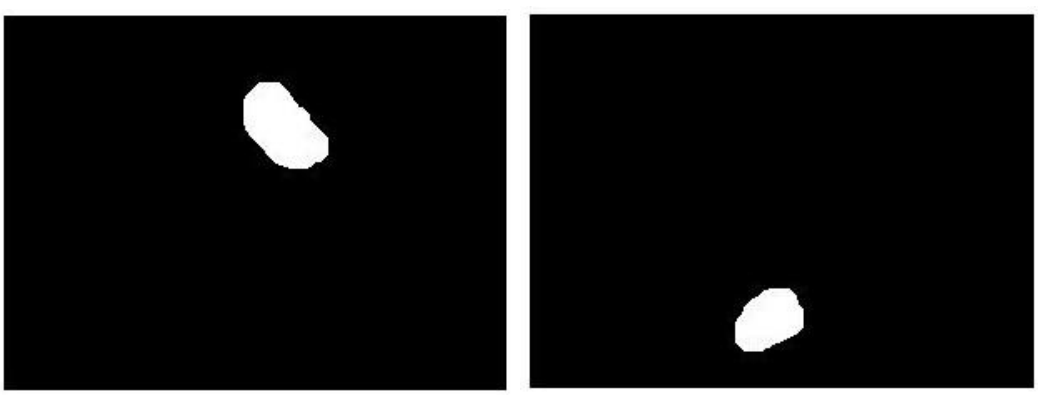


Figure 17: Retrieved blobs from labeling

3.2.7 Shape detection

All ready through blob detection algorithm, blobs are detected in the affected leaf's image. Now getting the region properties from a method called "regionprop", shape of the blob can be detected primarily. Two conditions were checked to resolve this factor.

1. Check the ratio of "maximum length of the axis" and "minimum length of the axis" :

    a. Find maximum length of the axis of the region of the blob by:

    "regionprops(binary_image, 'MajorAxisLength')"

    b. Find maximum length of the axis of the region of the blob by:

    "regionprops(binary_image, 'MinorAxisLength')"

    c. Calculate the ratio between them and compare with the experimented difference range. If the distance is within the range then first condition to be the blob a circular region, is full filled.

2. Get the "Area" and "Perimeter" of the region, and calculate the roundness with the formula [5]:

$$\text{Circularity, R} = \frac{Perimeter^{\ 2}}{4*\pi*\text{Area}}$$

If R is around 1 to 1.3 or less than that, the region is circular or it is irregular in shape.

By this two circumstances shape of the region can be detected primarily on the blob of the affected region.

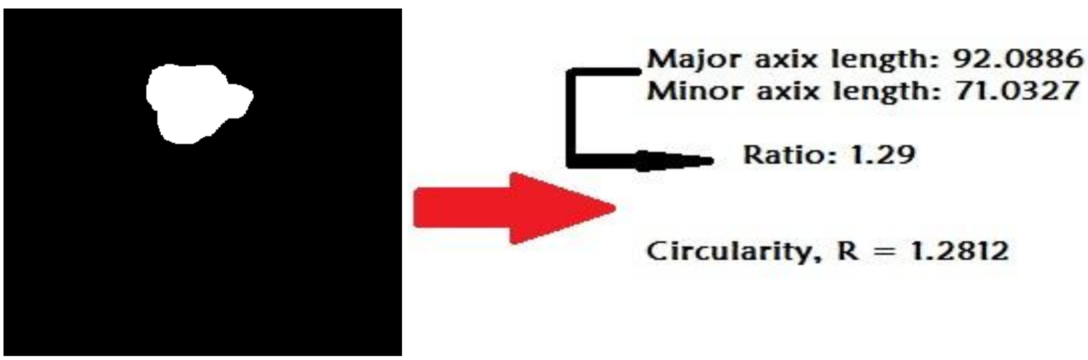


Figure 18: Shape Detection

3.2.8 Color extraction

Color extraction was done with getting the value from an image histogram. After collecting the RGB blobs in the leaf the centroid of the regions were calculated and the region was cropped with a fixed positions. Then the image was converted to HSV color scale. After that valuess plotted in a histogram. From the comparative analysis, it was determined that for "Brown" regions are with 0.2 and "Purple" regions are with in 0.8. The threshold was calculated by the below step:

$$\text{Threshold} = \frac{0.2+0.8}{2}, \text{Which is } 0.5$$

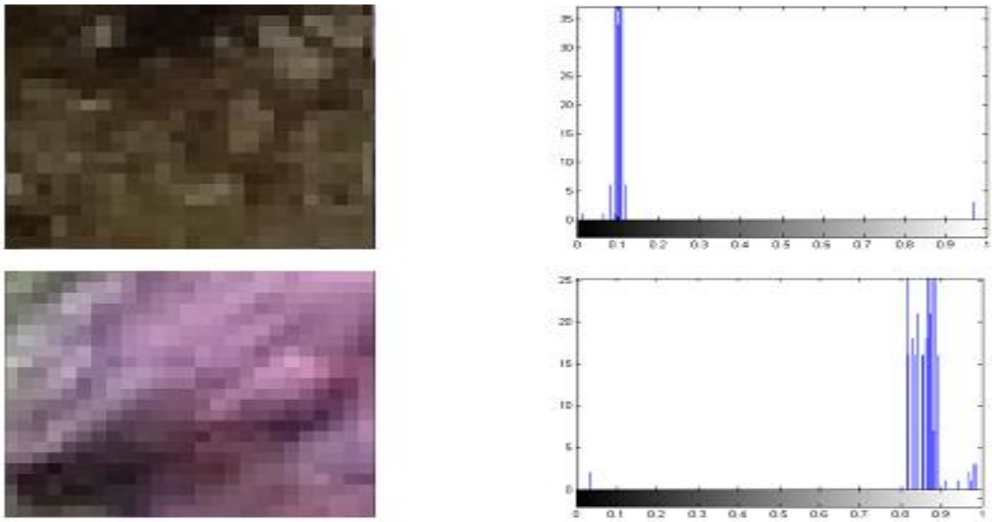


Figure 19: Color extraction.

Then when a new image will come, histogram will be calculated in the same process and then is the value of the region is less than 0.4, the region color is brown otherwise color is purple.

3.2.9 Concentric ring detection

One of the most significant differences between “Early Blight” and “Late Blight” is appearance of concentric rings in early blight. To determine whether the affected blob is affected by any concentric ring a very easy yet accurate algorithm can be used. At the beginning two images are generated a binary image and blob extracted RGB image. Then the RGB blob image is converted to binary with the traditional formula in MATLab. Then area of each images are calculated.

Figure 20 : Area ratio for determining concentric ring in Late Blight

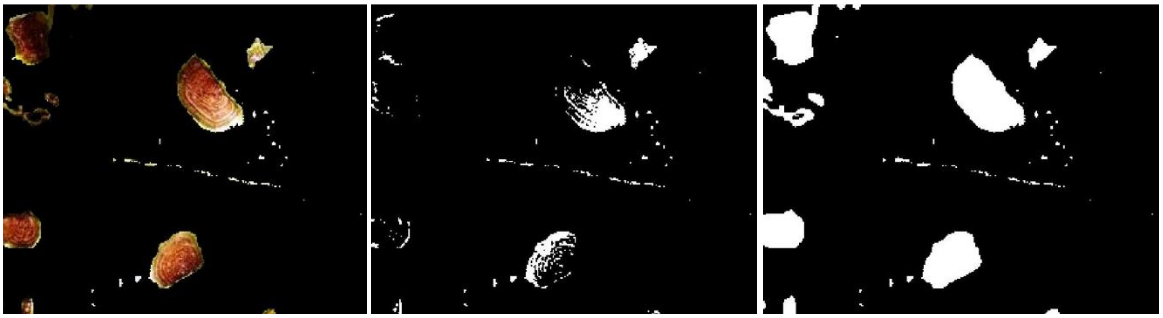


Figure 21 : Area ratio for determining concentric ring in Early Blight

From the above two images it can be seen early blight's affected regions are less white due to concentric ring. Thus a ration range was determined (above 60%) to figure the appearance of concentric ring.

3.2.10 Leaf vein detection

Leaf vein detection is a very challenging task. There are a lot of approaches to detect vein in a healthy leaf. Yet, detecting vein in a disease affected leaf is more challenging. This is because; the leaf has some noise for the disease. Which is why, sometimes vein detection algorithm may over do the effected region under vein category. Thus veins cannot be detected perfectly. This thesis was conducted following two approaches for determining the leaf vein. Between these two approaches, the best one was implemented with respect to the situation of leaf having disease parts as noise.

Here, a new method of RGB to grayscale conversion was established [3]:

$$Y = \left(\frac{(H+90)\%360}{360}\right) + 1 - V)$$

To employ this method in this thesis, blobs were masked as black pixels.

Figure 22: Masked affected regions

Then the formula was generated in this image to proceed further to detect vein in the leaf.

Figure 23 : Grayscale image with the established formula.

After that, grayscale morphological image processing was implemented in this picture. For this operation two methods of morphological image processing was implemented simultaneously.

1. Bot-Hat-Transformation
2. Top-Hat-Transformation

Bot-Hat-Transformation:

$$f \bullet b - f$$

This can extract the little darker regions in the image or get the gray differences between these regions and their backgrounds.

Top-Hat-Transformation:

$$f - f \circ b$$

This can extract the little brighter regions in the image or get the gray differences between these regions and their backgrounds.

To handle some uneven illumination occurred in the image, difference between the bot-hat image and the top-hat image is taken.

$$g = (f \bullet b - f) - (f - f \circ b)$$

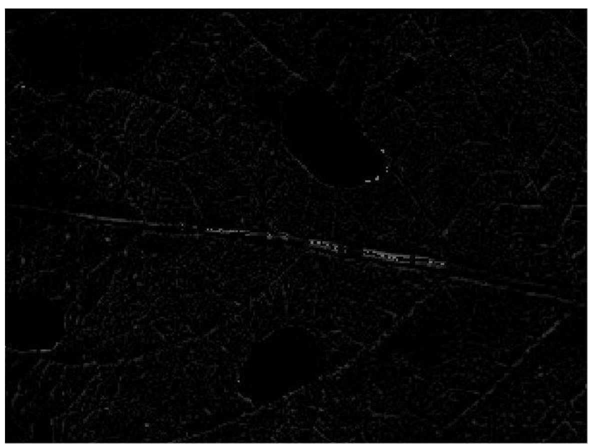


Figure 24 : Applying difference of Top-Hat-Transformation and Bot-Hat-Transformation

Applying a gray threshold value, masking is done and finally the image below is generated:

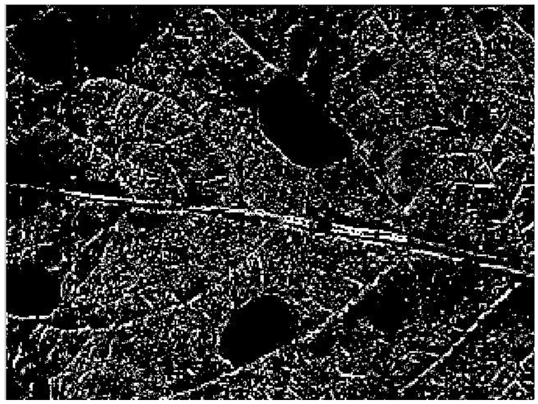


Figure 25 : Detected leaf vein.

Still it is not a perfect detection; this part of the thesis is aimed to develop in future as detecting veins are very important to differentiate between “Early blight” and “Late blight”.

3.2.11 Diameter detection:

This part of the thesis is a proposed approach. More experiment should be taken to improve it.

- Create a database indicating a technique to determine per pixel information taking picture from a particular distance
- Then the pixel value and the maximum axis length should be co-related so that an approximate diameter can be determined.

3.3 Client and Server Interaction

This part of the thesis is going to execute at a server. The users will the take the picture and when they will press the send button the image will be send to the server allocated for the code execution.

3.3.1 Shell Execution

As currently the computer is acting as a server application will connect through internet. Beforehand, an “exe” file of the MATLab code was generated and saved to a local server. In this thesis “XAMPP” was used to fulfill the purpose. On the other hand, the picture will be send a saved in the server and execute a PHP file to run the “exe” file of the MATLab through shell execution.

3.3.2 Parsing the result

Once the MATLab file is executed through shell, a result will be saved in a PHP file, and the application will fetch the information from the PHP file and show it to the user.

Chapter 4: Details on some experimental approaches

4.1 Hierarchical approach for vein detection from affected leaf

As mentioned earlier, vein detection in a disease affected leaf is quite difficult, another approach was taken to check the vein detection algorithm's working percentage.

In this part, the RGB image of the leaf excluding the affected part was converted to grayscale with the traditional formula [51]

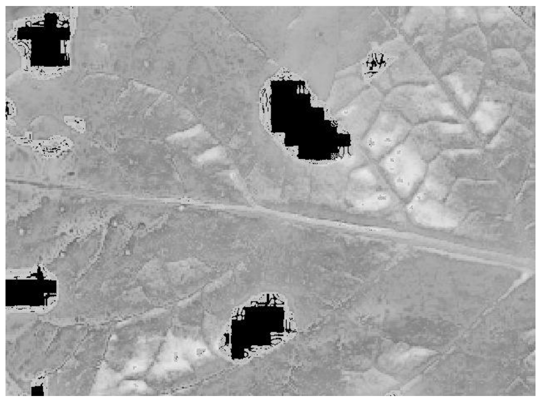


Figure 26 : RGB to Grayscale conversion

After that, an image opening method was implied.

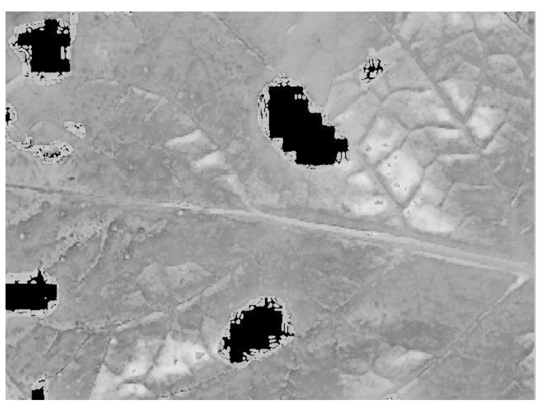


Figure 27 : Apply the opening image processing

At the end, difference between the grayscale image and opened image is calculated to the get the veins in the leaf.

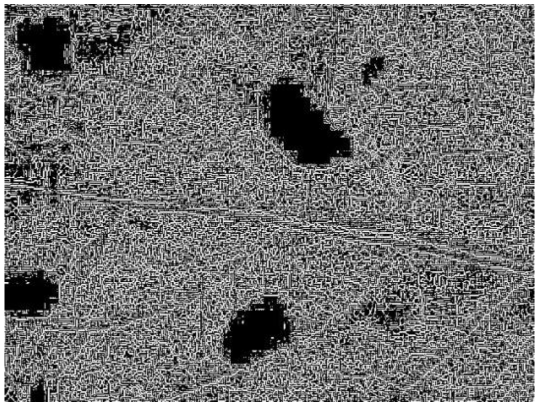


Figure 28 : Leaf vein extraction

Again, this works perfectly only with healthy leaves.

4.2 Android camera setting:

User has to enter the image for detecting input level of fertilizer, expecting the image to be cropped properly with only the green area. To make this working process more user friendly, an attempt was taken to insert a cropping tool when the camera is opened for inputting image. For this purpose, an idea was adopted to draw a canvas on the camera with the fixed width and height and crop only the image inside it.

It is still in developing session but it will serve the purpose in future.

Chapter 5: Experimental results and discussions

5.1 Environmental setup

1. Android
2. MATLab, 2010 version
3. JAVA
4. Notepad++
5. XAMPP server
6. Windows 7

5.2 Database

Collecting database was the most challenging part of this application. Field level data were required for more accuracy as this application was dependent on agricultural crops. However by the time the development of the application was on full working process, it was not easy to gather data on both paddy and potato. As seen in the crop calendar in Figure 28, during fall session (September-December), the development of the application was at the primary stage of accumulating knowledge on the topic. The work on approaching to different algorithm was possible during the spring session (January-April) while the actual implementation of the algorithms was possible during summer session (May-August). As the chart [48] shows by that time harvest of both paddy and potato was over thus was not possible to collect data from the field. Thus, the available database was not elaborate. However, the algorithms worked well with the current database that works for other environments too.

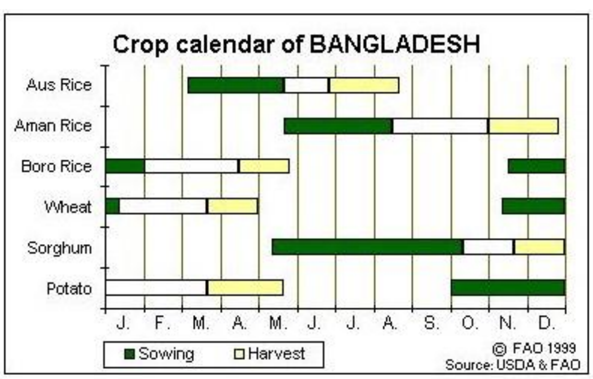


Figure 29 : Crop calendar Of Bangladesh

5.3 Challenges

The development of the application faced several challenges.

Android application crashed frequently causing difficulties while implementing the algorithms in a time bound period. Also the application needed implementing unusual camera settings, for example opening the camera view ten times in a row or cropping the image while capturing the image. Thus, handling the camera often created complexity in the application.

Availability and collecting database for the application was another challenge. Although the Leaf Color Chart helped in developing the application, but gaining access to such data was both time consuming and difficult since they can be accessed from very few authentic sources.

5.4 Performance analysis

5.4.1 Fertilizer recommendation system (PCA)

We took 22 images as database. And calculated the below table:

TP = 16	TN =1
FP = 4	FN = 1

Formula for precision calculation: $\frac{tp}{tp+fp} = \frac{16}{16+4}$ = .80

Formula for recall calculation: $\frac{tp}{tp+fn} = \frac{16}{16+1}$ == .94

Formula for accuracy calculation: $\frac{tp}{tp+tn+fp+fn} = \frac{16}{16+1+4+1}$ x 100 % = 72%

5.4.2 Fertilizer recommendation system (Template)

We took 38 images as database. And calculated the below table:

TP = 31	TN = 3
FP = 2	FN = 2

Formula for precision calculation: $\frac{tp}{tp+fp} = \frac{31}{31+2} = .93$

Formula for recall calculation: $\frac{tp}{tp+fn} = \frac{31}{31+2} = .93$

Formula for accuracy calculation: $\frac{tp}{tp+tn+fp+fn} = \frac{31}{31+3+2+2}$ x 100 % = 81.5%

5.4.3 Disease segmentation

We took 34 images as database. And calculated the below table:

TP = 30	TN = 2
FP = 1	FN = 1

Formula for precision calculation: $\frac{tp}{tp+fp} = \frac{30}{30+1}$ =.967

Formula for recall calculation: $\frac{tp}{tp+fn} = \frac{30}{30+1}$ =.967

Formula for accuracy calculation: $\frac{tp}{tp+tn+fp+fn} = \frac{30}{30+2+1+1}$ x 100 % = 94.1%

5.5 Results:

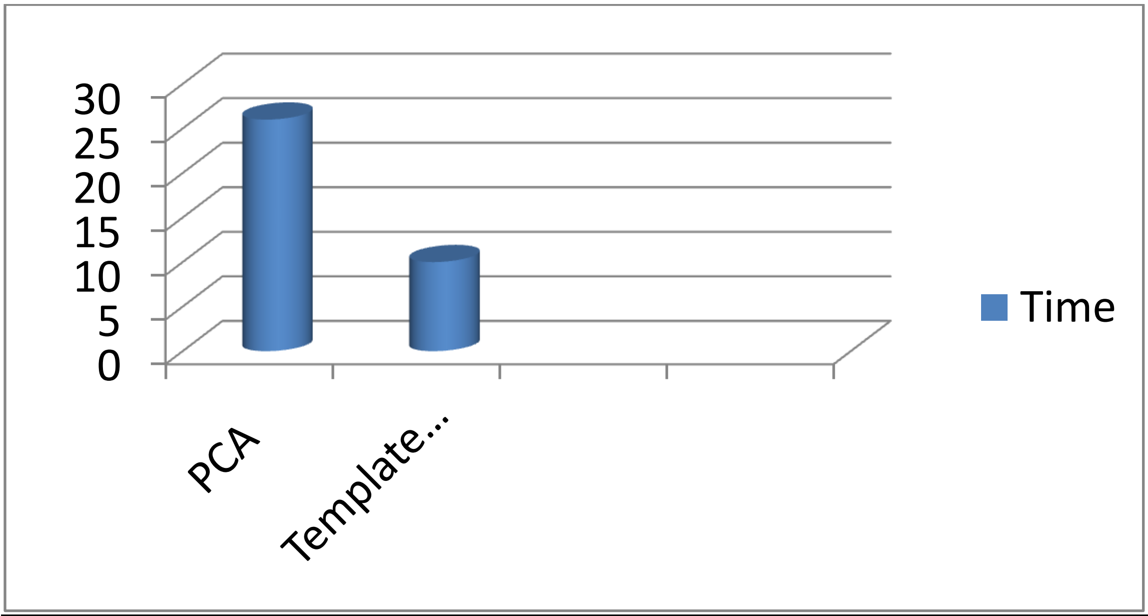


Figure 30 : Comparison between PCA and Template matching with time constrain

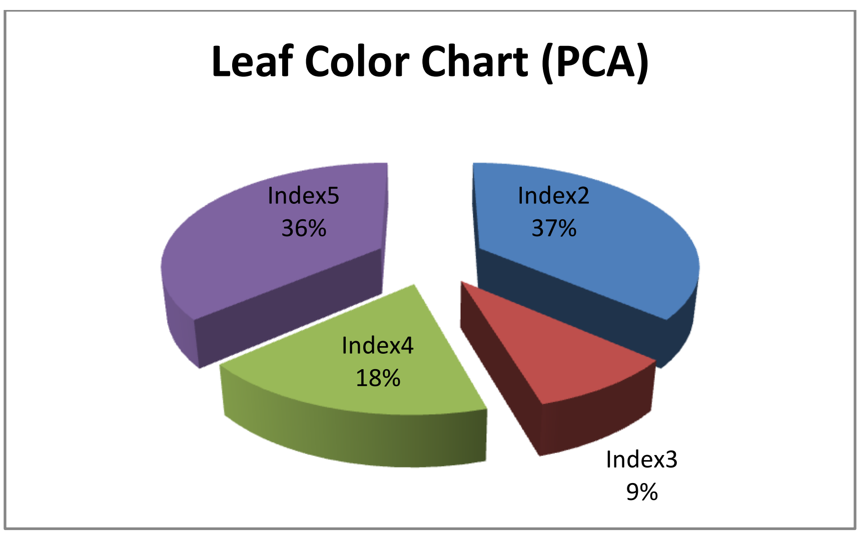


Figure 31 : Comparison in pie chart

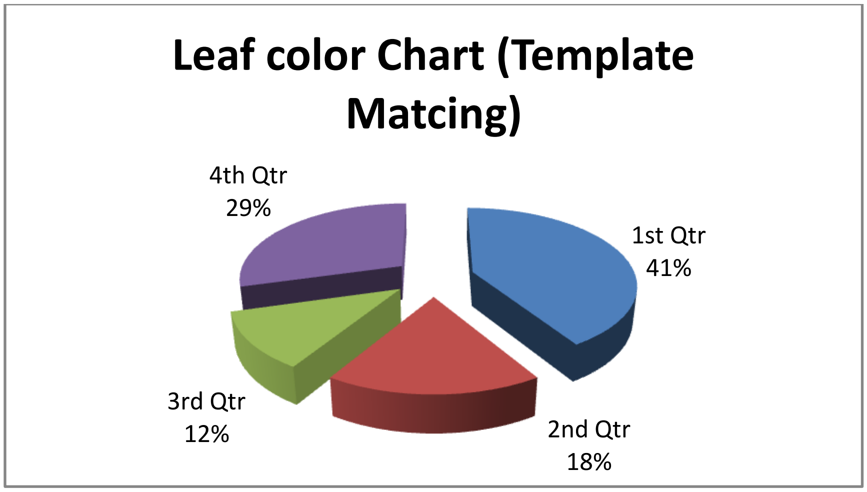


Figure 32: Comparison of template matching algorithm in pie chart

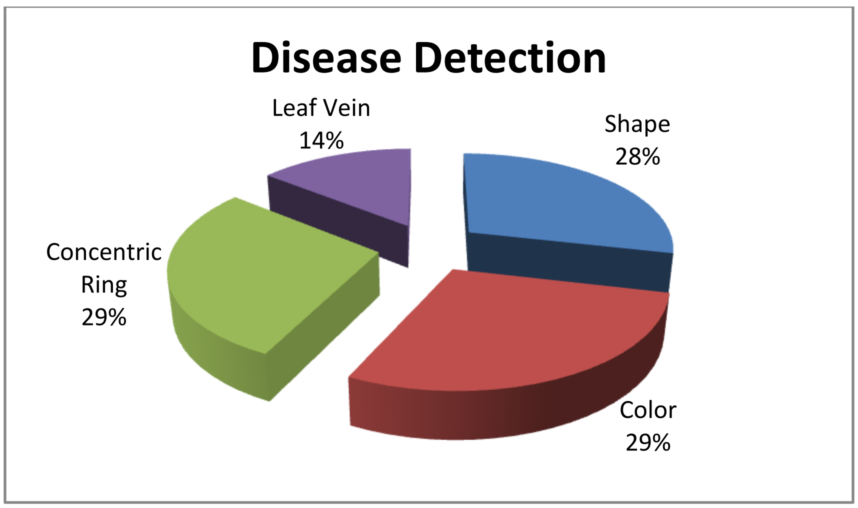


Figure 33 : Efficiency of disease detection process in pie chart

From all the results and comparison it can be said that, with a powerful source of data an almost perfect result for both features of this thesis is not impossible.

5.5.1 *Comparison between two vein detection system*

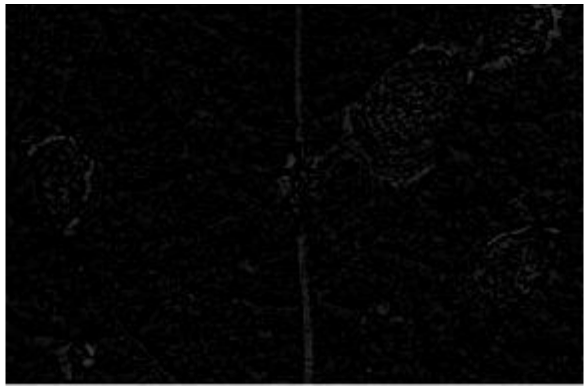


Figure 34: comparatively better detection

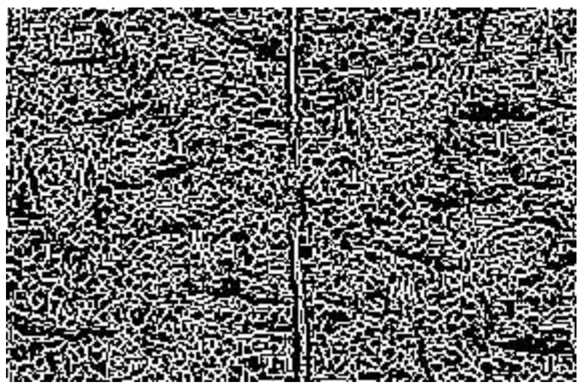


Figure 35: Vein detection with too many noises

From the above two images it can be seen both of the algorithms used was not perfect. The first image approach was a bit better than the second experimental algorithm.

Chapter 6: Conclusion

6.1Conclusion

Bangladesh has a primarily agrarian economy. Agriculture production along with other factors, depend on the efficient and effective use of fertilizer and early detection of diseases. The farmers and the persons involved with helping in the production need assistance to ensure good harvest. Mobile phone has become available at the grass-root level providing different social and economic benefit. The aim of this thesis was to develop a user friendly automated system for the farmers that will help them in determining input level of fertilizer for paddy and detection diseases of potato leaves without bringing an expert to the field.

To develop the application, PCA algorithm was implemented by creating a JAVA library in android for input level of fertilizer. Along with that, an online server-based system for leaf disease detection of potato was implemented to gather data from the user, send it to a server where MATLab generated the desired output. The android parsed the result through PHP command to send it to the user.

The development process of the application also illustrates the challenges of using image processing for Computer Science and Engineering. Yet with enough data and knowledge, development of different applications can meet the demand of diversified user groups.

6.2 Future work

The application developed in this thesis tried to make it as user friendly as possible. Yet it recognizes that many farmers may not be able to use it, especially the illiterate ones. In future the aim will be to make it accessible for those who are not able to read properly.

This application only addressed two crops, while the farmers of Bangladesh produce different varieties of crop. Opportunities remain to develop working algorithms with other crops.

The application also recognized that since PCA algorithm is implemented in JAVA, this application can be implemented in Windows phone as well. This creates possibilities to develop the application to be used in windows phone.

References:

[1] Phillip Wharton and William Kirk : Early Blight

[2] William Kirk, Phillip Wharton, Ray Hammerschmidt, Firas Abu-el Samen and Dave Douches: Late Blight

[3] Xiaodong Zheng,Xiaojie Wang : Leaf Vein Extraction Based on Gray-scale Morphology. I.J. Image, Graphics and Signal Processing, 2010, 2, 25-3. Published Online December 2010 in MECS

[4] Original version by Amir Hossein Omidvarnia, October 2007

[5] A.Hemlata1, Mahesh Motwani2: Face Detection By Finding The Facial Features And The Angle of Inclination of Tilted Face. IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 2, No 1, March 2013. ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784 16

[6] Dong Hyun Jeong, Caroline Ziemkiewicz, William Ribarsky and Remco Chang: Understanding Principal Component Analysis Using a Visual Analytics Tool

[7] I. T. Jollie. Principal Component Analysis. Springer, second edition, 2002.

[8] Alexander Ilin, TapaniRaiko: Practical Approaches to Principal Component Analysis in the Presence of Missing Values. Journal of Machine Learning Research 11 (2010) 1957-2000

[9] Otsu, N., "A Threshold Selection Method from Gray- Level Histograms," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, No. 1, 1979, pp. 62-66.

[10] M. Brand. Fast online svd revisions for light weight recommender systems. In SDM 2003, pages 83{91,2003.

[11] SmitaNaikwadi, NiketAmoda: ADVANCES IN IMAGE PROCESSING FOR DETECTION OF PLANT DISEASES. International Journal of Application or Innovation in Engineering & Management (IJAIEM) Web Site: www.ijaiem.org Email: editor@ijaiem.org, editorijaiem@gmail.com Volume 2, Issue 11, November 2013

[12] Anne Kaspers: Blob Detection

[13] Morphological Image Processing by Steven W. Smith

[14] R. H. Sherrier and G. A. Johnson, “Regionally adaptive histogram equalization of the chest,” IEEE Trans. Med. Imag., vol. MI-6, pp. 1–7, 1987.

[15] M. Kurucz, A. A. Bencz_ur, and K. Csalogany.Methods for large scale SVD with missing values.In KDDCup 2007, 2007.

[16] C. R. González and E.Woods, Digital Image Processing. Englewood Cliffs, NJ: Prentice Hall, 1992.

[17] Feng Wang, XiangshiRen, Zhen Liu: A Robust Blob Recognition and Tracking Method in Vision-based Multitouch Technique

[18] J. Serra and P. Salembier, “Connected operators and pyramids,” presented at the SPIE. ImageAlgebra and Mathematical Morphology, SanDiego, CA, Jul. 1993.

[19] M. Turk and A. Pentland. Face recognition using eigenfaces. In Computer Vision and Pattern Recognition, pages 586{591.IEEE Computer Society, 1991.

[20] M. E. Wall, A. Rechtsteiner, and L. M. Rocha.Singular value decomposition and principal component analysis.in A Practical Approach to Microarray Data Analysis D.P. Berrar, W. Dubitzky,M.Granzow, eds. pages 91-109, Kluwer: Norwell, MA, 2003.

[21] Nicolas Morizet, FrédéricAmiel, InsafDrisHamed, Thomas Ea: A Comparative Implementation of PCA Face Recognition Algorithm

[22] Luc Vincent: Morphological Grayscale Reconstruction in Image Analysis: Applications and Efficient Algorithms

[23] Kue-Bum Lee and Kwang-Seok Hong: An Implementation of Leaf Recognition System using Leaf Vein and Shape. International Journal of Bio-Science and Bio-Technology Vol. 5, No. 2, April, 2013

[24] J. Kasperek, “Real time morphological image contrast enhancement in FPGA,” in LNCS, New York: Springer, 2008.

[25] P. Soille, Morphological Image Analysis: Principle and Applications. New York: Springer-Verlag, 2003.

[26] C. Damerval and S. Meignen. Blob detection with wavelet maxima lines. IEEE Signal Processing Letters, 14(1):39–42, 2007.

[27] P.-E. Forssén and G. Granlund. Robust multi-scale extraction of blob features. In Proceedings of the 13th Scandinavian Conference on Image Analysis, LNCS 2749, 11–18, 2003.

[28] Rein-Lien Hsu, Mohamed Abdel-Mottaleb, and Ani1 K.Jain, “ Face Detection In Color Images”, IEEE 2001.

[29] L. Vincent and E. R. Dougherty, “Morphological segmentation for textures and particles,” in Digital Image Processing Methods, E. R. Dougherty, Ed. New York: Marcel Dekker, 1994, pp. 43–102.

[30] S. Lu, G. Tsechpenakis, D. N. Metaxas, M. L. Jensen, and J. Kruse. Blob Analysis of the Head and Hands: A Method for Deception Detection, Hawaii International Conference on System Science (HICSS'05), 2005.

[31] H. Heijmans, Morphological Image Operators. New York: Academic, 1994.

[32] L. G. Minor and J. Sklansky.Detection and segmentation of blobs in infrared images. IEEE Tranactions on Systems Man and Cyberneteics, SMC-11(3):194–201, March 1981.

[33] J. Short, J. Kittler, and K. Messer, “A comparison of photometric normalization algorithms for face verification,” presented at the IEEE Int. Conf. Automatic Face and Gesture Recognition, 2004.

[34] S. Hinz, Fast and Sub-pixel Precise Blob Detection and Attribution. In: Proceedings ICIP’05, Genua. pp. 457–460, 2005.

[35] S. Malik and J. Laszlo. “Visual touchpad: A two-handed gestural input device”. In Proc. ICMI2004, ACM Press, 2004, pp. 289-296.

[36] K.Sreedhar1 and B.Panlal: Enhancement Of Images Using Morphological Transformations, International Journal of Computer Science & Information Technology (IJCSIT) Vol 4, No 1, Feb 2012.

[37] I. R. Terol-Villalobos, “A multi-scale contrast approach on Morphological connected contrast mappings” Opt. Eng., vol. 43, no. 7, pp. 1577–1595, 2009.

[38] XiaodongZheng: Leaf Vein Extraction Based on Gray-scale Morphology. December 2010 in MECS

[39] http://archive.thedailystar.net/beta2/news/food-security-building-on-a-success-story/

[40] http://www.banglajol.info/index.php/JARD/article/viewFile/1455/1420

[41]http://bangladesheconomy.wordpress.com/2010/02/10/leaf-colour-charts-may-save-1-5-lakh-tonnes-of-urea-this-season/

[42] http://www.knowledgebank-brri.org/lcc.php

[43] http://www.banglajol.info/index.php/BJAR/article/viewFile/6452/4947

[44] http://www.potatodiseases.org/lateblight.html

[45] http://www.potatodiseases.org/pdf/early-blight-bulletin.pdf

[46] http://www.mathworks.com

[47] http://www.thedailystar.net/big-boost-for-potato-yield-24165

[48] http://www.foodsecurityatlas.org/bgd/country/availability/agricultural-production

[49] http://www.wikipedia.org

[50] http://people.cs.pitt.edu/~iyad/PCA.pdf