Bachelor of Science in Computer Science & Engineering

Developing an Android Application for Detecting Mango Anthracnose Using Digital Image Processing

by

Nizia Nahyan

ID: 1504043

Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)

Chattogram-4349, Bangladesh.

May, 2021

Developing an Android Application for Detecting Mango Anthracnose Using Digital Image Processing

Submitted in partial fulfilment of the requirements for
Degree of Bachelor of Science
in Computer Science & Engineering

by

Nizia Nahyan
ID: 1504043

Supervised by

Dr. Mohammad Ibrahim Khan
Professor
Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)
Chattogram-4349, Bangladesh.

The thesis titled '**Developing an Android Application for Detecting Mango Anthracnose Using Digital Image Processing** ' submitted by ID: 1504043, Session 2019-2020 has been accepted as satisfactory in fulfilment of the requirement for the degree of Bachelor of Science in Computer Science & Engineering to be awarded by the Chittagong University of Engineering & Technology (CUET).

Board of Examiners

Chairman

Dr. Mohammad Ibrahim Khan
Professor
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Member (Ex-Officio)

Dr. Md. Mokammel Haque
Professor & Head
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Member (External)

Dr. Kaushik Deb
Professor
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Declaration of Originality

This is to certify that I am the sole author of this thesis and that neither any part of this thesis nor the whole of the thesis has been submitted for a degree to any other institution.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. I am also aware that if any infringement of anyone's copyright is found, whether intentional or otherwise, I may be subject to legal and disciplinary action determined by Dept. of CSE, CUET.

I hereby assign every rights in the copyright of this thesis work to Dept. of CSE, CUET, who shall be the owner of the copyright of this work and any reproduction or use in any form or by any means whatsoever is prohibited without the consent of Dept. of CSE, CUET.

Signature of the candidate

Date:

Acknowledgements

Firstly,this thesis would not be completed without the grace of Almighty Allah. I am very much grateful to Allah for completing my thesis.

Secondly, my gratitude goes to my supervisor Prof. Dr. Ibrahim Khan, Professor of Computer Science and Engineering Department at Chittagong University of Engineering and Technology. Without his proper help and guidance i could not achieve this. His continuous help and assistance enabled me to rectify my errors and start over in certain cases. Because of his critical questions and guidance i progressed through my whole thesis journey.

Thirdly, i thank my parents for helping me when i was struggling and motivating me all the time.

Abstract

Mango Anthracnose is a fungal infection of mango and is recognized as the most important field and post-harvest disease of mango all over the world. To increase save mangoes from becoming rotten and increase production , we introduce an android application system where we can detect anthracnose by the image of the leaves of mango plant using digital image processing. Firstly image is captured on phone camera and saved or image is imported from any folder. Then we upload the image on server database using the app and the image gets collected by the pc where image processing will be performed. Image is segmented by k-means clustering ,features are extracted and classified by Support Vector Machine. After getting the results whether the leaf has anthracnose or not, a resultant image is created point to the result and uploaded to the database and through the application the uploaded image is retrieved and displayed on the app.

Keywords: K-means, SVM(Support Vector Machine), Contour, MySQL, Database, Android Picasso library, OTSU method

Chapter 1

Introduction

1.1 Introduction

Anthracnose is a lethal disease for mango production and causes severe damage to our economic progress. It causes direct loss of fruit if it is left untreated. Post-harvest loss of mango is 17-36 percent due to anthracnose.The most widely used method for this disease detection is simply naked eye observation by experts through which identification and detection of plant diseases is done. For doing so, a large team of experts as well as continuous monitoring of experts is required, which costs very high when farms are large. At the same time, in some countries, farmers don't have proper facilities or even idea that they can contact to experts. Due to which consulting experts even cost high as well as time consuming too. The symptoms of anthracnose are basically seen on leaves, petioles, flower clusters, twigs etc. We are going to work with the images of leaves. On leaves lesions start as small black and brown spots that can extend to large areas [1]. We have to cluster and separate the diseased portion from the background and then extract the features from the portion.

In such condition the suggested technique proves to be beneficial in monitoring large fields of crops and automatic detection of the anthracnose by just seeing the symptoms on the plant leaves makes it easier as well as cheap. The android application that we are building detects the anthracnose from the image of the leaf given to it. Image is uploaded to the local database and by means of the app it is uploaded to the computer where image processing will take place. After the image is analysed and we get the result, another image having the result is generated. Then the image is again inserted to the database and we can access the resultant image from the app and display it. By using this application we can save the mango plants at early stages so that they don't get rotten.

1.2 Framework/Design Overview

The framework of our android system is simple. There are three major steps : 1.Acquirement of the image from phone to the application.

2.Getting the image from the phone to pc and perform image processing.

3.Sending the image with result back to the application. The figure is given below:

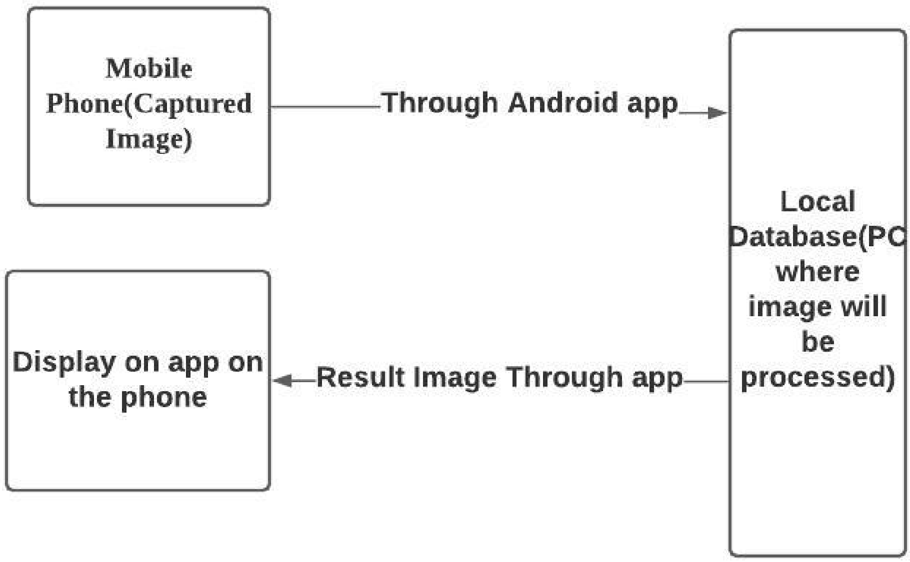


Figure 1.1: System architecture of the proposed framework.

Acquirement of the image : Image is captured through camera and saved to gallery. The said image is accessed through gallery or any other folder.

Uploading the image to database and to the pc : We created a localhost database and from the app we upload the image to the app and image is received in pc.In pc the image undergoes all the image processing techniques and resultant image is created.

Sending the image back to the application : After generating the result image we update the database with the result image and from the database the image is displayed on the application.

1.3 Difficulties

While building this system i faced various kinds of difficulties. Overcoming them and rectifying my mistakes was not an easy task. There were many errors in the codes that i wrote. But the major difficulties were :

1. Thresholding : All the images are different. Their background is different and so to get foreground and area of interest , an average threshold is set that can work in most images. After much attempt the average threshold was set using OTSU thresholding.

2. Creating and updating Local Database : Connecting to localhost was difficult. And updating it was not working earlier.

3. Dataset : There was no database of the anthracnose of mango. I had to process each and every image as there was a problem of thresholding. I also captured lots of pictures through my phone camera and used them.

1.4 Applications

There are many applications of this android application. It gives great service to the mango farmers out there. Other applications are ;

1. Android application saves so much time and money rather than hiring an inspection team.

2. The application can be upgraded and can be used other diseases of plants.

3. This crates an opportunity for image processing to be used in other areas.

4. This application can be used by many agricultural companies whether governmental or non-governmental.

1.5 Motivation

Main motivation is to decrease the production loss of mango. Other motivation is to protect the mangoes. The main targets are :

1. To decrease production loss of the farmers.

2. To improve accessibility of the farmers to anthracnose detection as fast as possible.

3. To reduce the cost that is spent on curing the diseases of plants everyday.

4. To implement this system in the real environment and evaluate the system.

1.6 Contribution of the thesis

This proposed system is the integration of android and digital image processing combined. Main objective of this protect was to develop an android app for anthracnose detection. The main contributions are :

1. The images that are trained and tested maximum of them give proper results and the a lot of images among them are captured by me in real-life environment.

2. The dataset is created properly. Each of the images were processed and used. This is why image processsing results were so good.

3. The model gives good amount of accuracy and precision.

4. A developed user interface of the application.

1.7 Thesis Organization

The rest of the part of the report goes by :

1. Chapter 2 describes all the existing projects and works that have already implemented similar systems

2. Chapter 3 gives us the details of the methodology of the proposed system.

3. Chapter 4 describes about the dataset that has been used and gives us details about the analysis of the performance measure of the model.

4. Chapter 5 gives us the overall summary and gives a sneak peek about future works.

1.8 Conclusion

In this section we got to know about the basics of this proposed system and the difficulties that we faced while doing the project. There were motivations that have to be followed by myself. Introduction basically gave a short brief about the project and on the next chapter backgrounfd and related works will be presented.

Chapter 2

Literature Review

2.1 Introduction

From the past three to five years there has been several articles similar to the work that i have done developing this system. This section of literature review deals with some related works that are mentioned below :

2.2 Related Literature Review

First of all in first paper,they developed a system for groundnut leaf disease detection. For the input RGB image a color renovation constitute is formed. This RGB iamge is converted into HSV image. Plane Separation is the next step and the separation is done on color features. Then using back propagation algorithm the groundnut leaf disease is detected [2].

In the next work, they developed a plant disease recognition model using leaf image classification and used deep convolutional networks. The model was able to detect 13 types of plant disease. First all the images are collected and saved on the database. They performed CNN deep level classification with the help of Caffe, a deep learning framework developed by Berkley Vision and Learning Centre. the average precision was 96.3 percent [3].

In this paper an image processing technique is implemented as a smart phone application. It is incorporated with portable molecular analysis equipment. parts of the affected lesions of fruits and leaves are segmented and separated. Analysis is based on number of spots, their area and color features. the the features are compared with the predetermined values of data [4].

In the next paper, they propose a system to detect leaf/stem disease of plants. They collect all the images and perform image processing techniques. They images are segmented using k-means segmentation and then images are passed through pre-trained neural network. The neural network is based on statistical classifier and had precision about 93 percent [5].

The next paper proposes a smart phone application and detect diseases of mostly citrus plants. The application does not have to be connected to remote server. Features like diseased area, color, number of spots are taken into account. These feature data along with weather metadata form disease signatures that can easily be defined by end users. The extracted features of the test photograph are compared with the disease signature to get the result. The accuracy was recorded around 70 to 80 percent [6].

In this paper diseases of rose plants are detected using digital image processing techniques. The techniques include image acquisition, image segmentation, feature extraction, statistical analysis and classification [7].

This project focuses on disease detection of soybean plants using image processing. It provides input to the DSS(Decision Support System) which provides help to the farmers. For the segmentation of area of interest multi thresholding was used. And for classification of features SVM classifier was used [8].

This paper describes the feasibility of CNN in disease detection for soybean leaf images taken in natural environment. The model is designed on the basis of LeNet architecture to perform soybean disease classification [9].

This paper proposes an algorithm for segmentation that will be helpful in disease detection as well as classification of the disease types. The algorithm proposed here is generic algorithm which is used for detection [10].

This paper focuses on detecting the bacterial fungal disease of alphonso mango. First iamge is captured through camera and goes for processing. The image is pre-processed by histogram streching method. Watershed algorithm is used for segmentation. Then the image is classified and result is achieved by template matching technique [11].

This paper proposes the survey of image processing techniques used for disease detection. The method used for detection is mostly remote sensing techniques used in images and detection of area of interest [12].

This paper focuses on the detection of plant disease using image processing with MATLAB functions. The classifier used is SVM. [13]

2.3 Conclusion

In these chapter we discussed all the related works similar to my work. There were disease detection systems but there were a few works which incorporated android application. So there is a chance future works and improvement of our proposed system.

Chapter 3

Methodology

3.1 Introduction

The methodology of our proposed system can be divided into two parts. 1. Building the android application and incorporating with database 2. Image processing of the fetched image and generate a resultant image. In this chapter we will discuss about our methodology that incorporates our proposed system.

3.2 Diagram/Overview of Framework

The overview of the two methodologies are given below :

3.2.1 Building Connection between android application and PC :

App connects to database and PC. The PC sends resultant image to database and App.

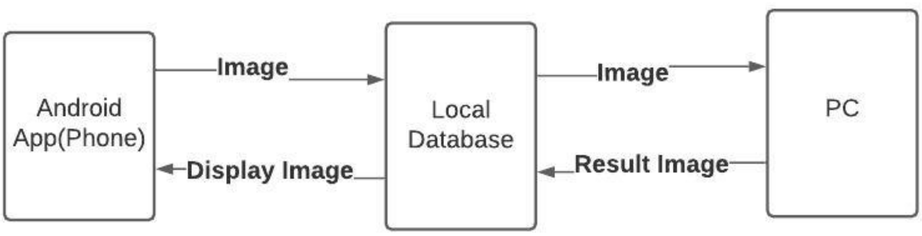


Figure 3.1: Overview of the Android and PC connection.

3.2.2 Image Processing :

Image processing techniques are given below :

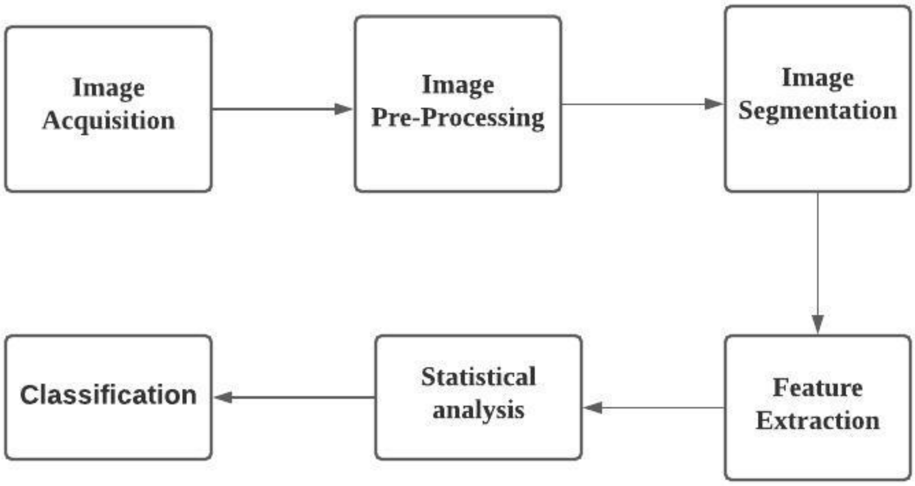


Figure 3.2: Overview of the Image Processing steps of the image.

3.3 Detailed Explanation

Just like methodology we are also going to separate the explanation in two parts. 1.Connecting App and PC 2.Image processing steps in PC

Connecting App and PC :

1. We have to have a local database where we can save the images with their imagepath that we are calling the 'url'.

My database looks like this ;

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	id	int(22)			No	None		AUTO_INCREMENT	Change	Drop	More
2	url	text	utf8mb4_general_ci		No	None			Change	Drop	More
3	name	varchar(100)	utf8mb4_general_ci		No	None			Change	Drop	More

Figure 3.3: Creating the database.

2. Launch the application on phone : After i run the app

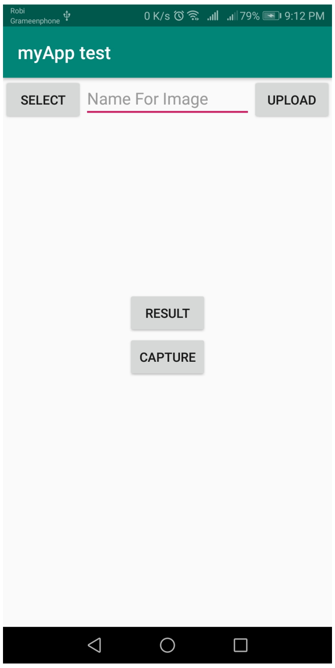


Figure 3.4: After launching the app.

3. After pushing the ’Capture’ button i can capture the image of the leaf.

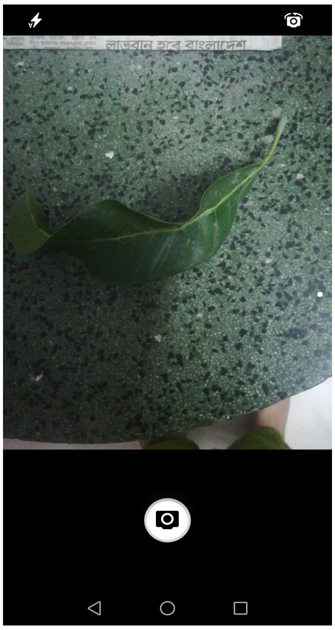


Figure 3.5: Opening the camera.

4.Pushing the ’SELECT’ button : i can also import any image from gallery or select captured image

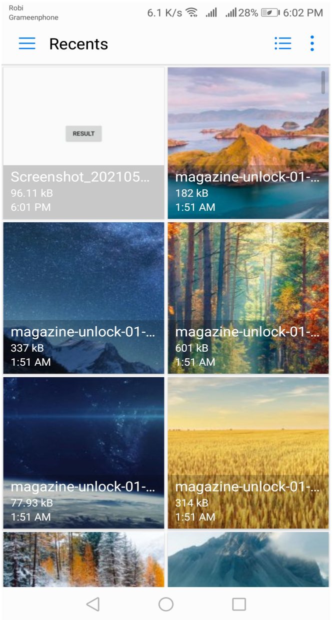


Figure 3.6: Selecting the 'SELECT' button.

5. Then we can search for any folder of the image

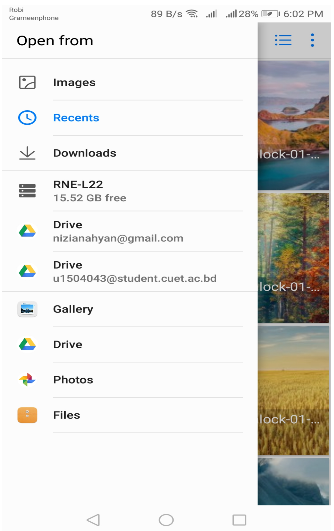


Figure 3.7: Searching through folders.

6. After selecting the folder we can fetch any image

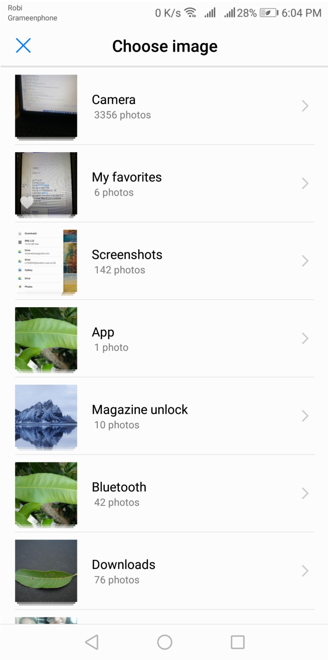


Figure 3.8: Choosing 'Gallery' folder.

7. After choosing 'Gallery' we can fetch any gallery image

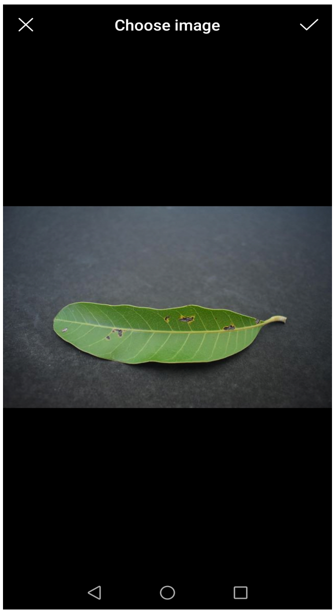


Figure 3.9: Choosing image.

8. After choosing the image it is displayed on the app and we can push the 'UPLOAD' button

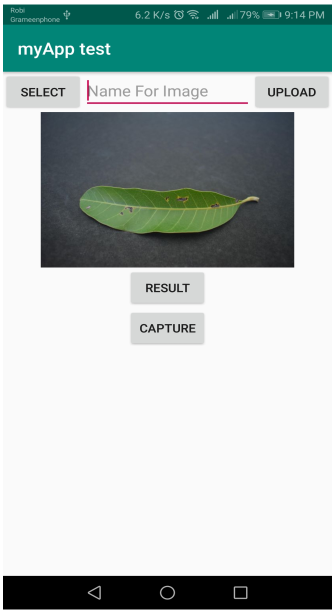


Figure 3.10: Fetched image.

9. After uploading the image the image is received at the PC end. This is the picture of the folder before uploading :

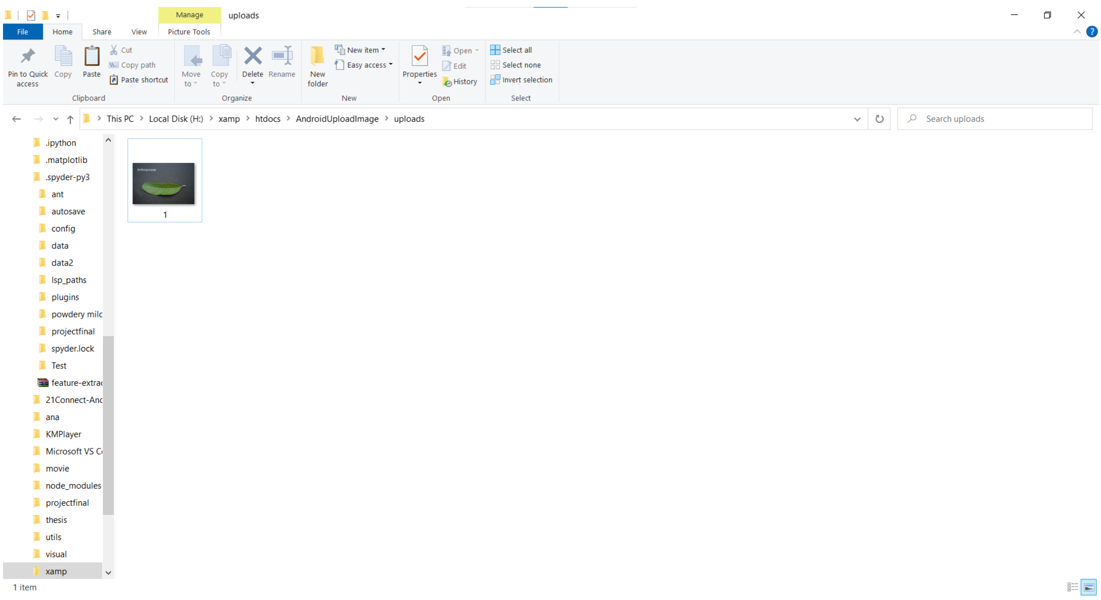


Figure 3.11: Before uploading.

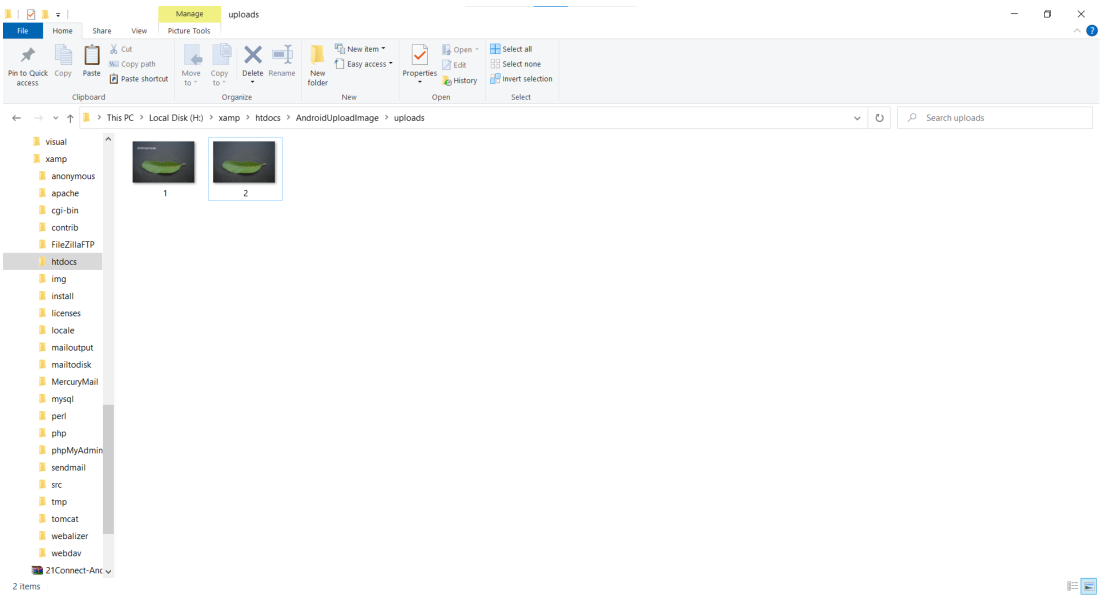


Figure 3.12: After uploading.

10. After getting the image we do many image processing techniques and get the resultant image. after getting the image we update the database on our localhost.

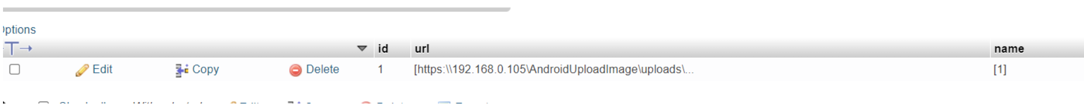


Figure 3.13: Updated database.

11. There is an issue here when we uploaded the image to app and sent it to PC. Then after getting the image the image is moved to other folder because we have to keep the xampp folder empty. when we get the result as another image we save the image in the xampp folder and update the database. In this case we have to restart the app and re-run as we inserted into the database. After restarting we do not have to upload any image. We just have to push the result button and we will get the result image.

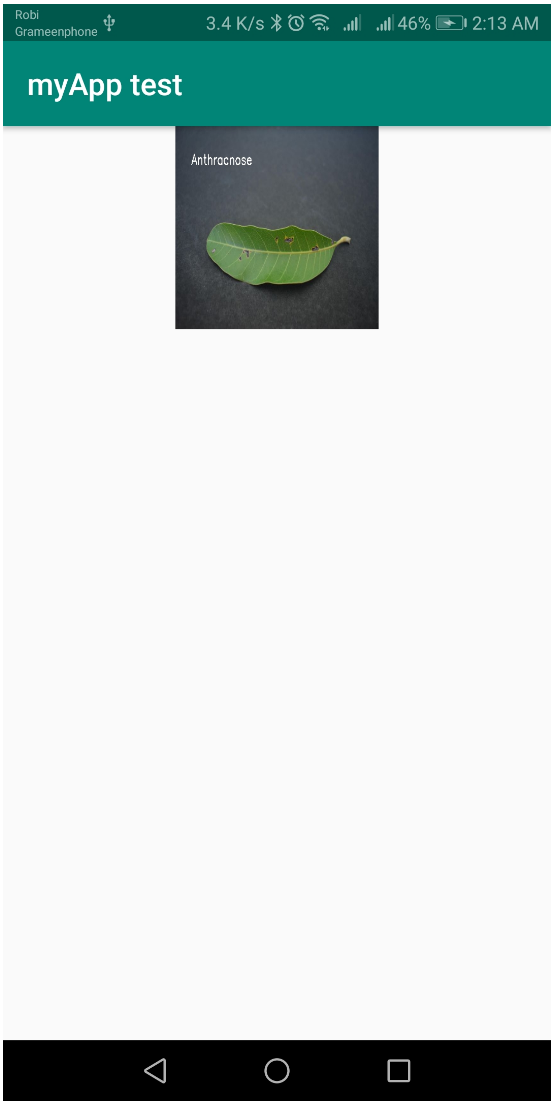


Figure 3.14: Result corporated into the fetched image.

3.3.1 Image Processing Techniques :

Image Acquisition :

This is already understood as i discussed the fetching of the image in previous subsection. image is captured through the camera and saved on the phone and then we fetch the image by application.

Image pre-processing :

i think it is the most important part of my work as there were many different types of leaf images so it was very difficult to apply same techniques to them as their background was different, noise was different and area of interest was different. The main points of pre-processing were :

Background Removal : Background Removal was difficult as the images werwe different. For removing background otsu thresholding was used.

Otsu Thresholding : Otsu Thresholding iterates through all the possible values and calculating a spread for all the possible pixels values falling each side of the threshold values [14]. As coding respective i am using Python OpenCV for image processing. There is a built in function for Otsu thrsholding. It is 'cv2.THRESH_OTSU()' [15]. It is better than Local and Global thresholding.

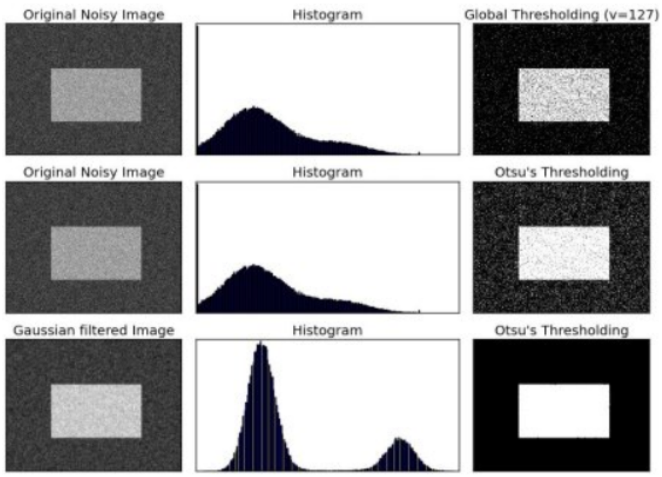


Figure 3.15: Outputs of Otsu thresholding.

In my work otsu thresholding is used for background removal. Figure is given below : The fetched image is given below :

Figure 3.16: Original Image

But my background of the image was not removed properly even after Otsu thresholding. There were many holes in the result image.

Figure 3.17: Otsu Thresholding of image.

To fill the holes and properly remove background i used Morphological Transformations.

Morphological Transformation : Morphological operations are simple operations that are performed over binary images by the use of another input called Structuring Element or Kernel. There are many morphological operations like erosion and Dilation. Dilation fixes the holes in the images and erosion removes excessive boundaries of image [16].

Opening : It is erosion followed by dilation. in OpenCV there is a function for opening , that is 'cv2.morphologyEx(img, cv.MORPH_OPEN, kernel)'

Closing : It is dilation followed by erosion. The function is 'cv2.morphologyEx(img, cv.MORPH_CLOSE, kernel)'.

In my case Opening was used in the image after it was thresholded. After opening ackground was removed and we got the leaf foreground.

Figure 3.18: After doing Opening operation on the image.

Image segmentation :

Image Segmentation is a process where a digital image is partitioned into many sub groups called image objects. For getting area of interest, segmentation is a very useful approach. By segmenting we can draw lines, specify boundaries and separate the area of interest [17].

In my case K-means Clustering was used for segmentation.

K-means Clustering : K-means is an unsupervised machine learning algorithm. It segments the dataset into k groups exactly.

The main idea is to define k centres. The centres should be chosen in an intelligent way. Then distance is measured between the other dataset and the centres. The points are assigned to the groups that the centres are in based on how close they are with their centres. After the first iteration the centres are changed by mean average and is iterated again like first time [18]

$$J(V) = \sum_{i=1}^{c} \sum_{j=1}^{c_i} \left(\| x_i - v_j \| \right)^2$$

The euclidian distance is measured by this formula .

The pseudocode for k-means is given :

Algorithm 1 K-Means Clustering

```
while True do
    for i = 1 to m ... do
        c^(i) := index (1 to k) of cluster centroid closest to x^(i)
    end for
    for k = 1 to K ... do
        μ_(k) := average (mean) of points assigned to cluster k.
    end for
end while
```

Figure 3.19: Pseudo code of k-means clustering

Image Segmentation with Color Spaces :

Color image segmentation is based on color feature of image pixels. It assumes that homogeneous color of the image corresponds to seperate clusters [19]

After Image segmentation we get :

Figure 3.20: After k-means Segmentation showing the curling part and diseased part of iamge

We can see from the comparison to original all the diseased area are discovered. Even the curling of the leaf that is also symptom of anthracnose.

After cropping from the original, the image looks like :

Figure 3.21: After final segment.

Feature Extractions :

Features are the characteristics of the segmented image that is calculated as they are input to the classification model of the system.

Our features are :

1. Width and Height :

To find the area and other features we have to find the contours. Contours mean curve joining all the points of the pixels of the same intensity. Through 'cv2.findContour()' method we get the areas of the diseased part of the image. And fro the area get the width and height by bounding box.

2. Mean or Color index :

The area is achieved then we crop those areas and get the mean value of the intensity values. It is close to the intensity of brown as anthracnose diseased parts are brown in color.

3. Standard Deviation : Standard Deviation means how much the other intensities differ from the mean.

The extracted features of the image are :

```
48 22
         mean  width  height  std deviation
0  114.381266     48      22      45.107391

In [13]:
```

Figure 3.22: Features.

Image Classification :

The classification model used here is SVM. SVM creates a hyperplane boundary between the two classes that are fed into the classification model. and when the test data is given it puts it on the two of the classes.

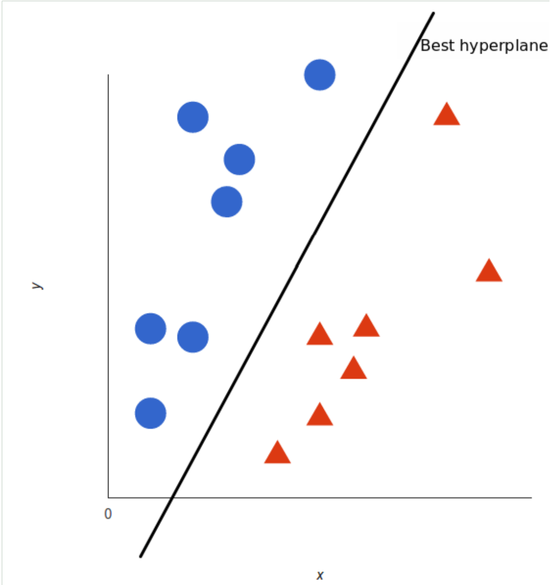


Figure 3.23: SVM.

for prediction we split the train and test data to 70 and 30 percent and used the features that we extracted.

We implemented a ’tkinter’ User interface to easily fetch the image from the app and read the results. We pushing each buttons and call each function in the python code that we implemented.

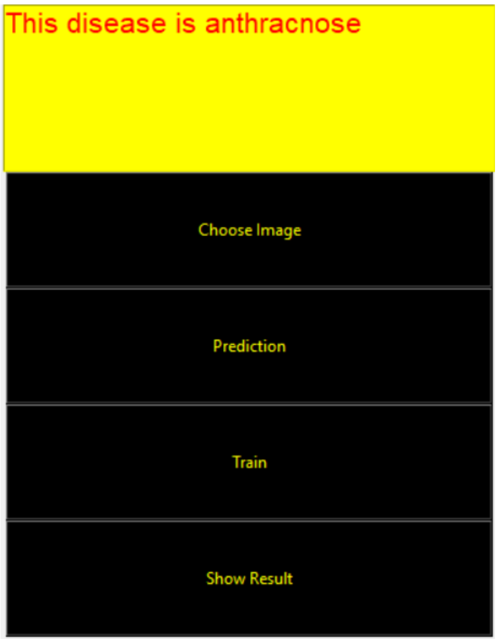


Figure 3.24: Result of the classification.

3.4 Implementation

Launching app : After launching we fetch the image

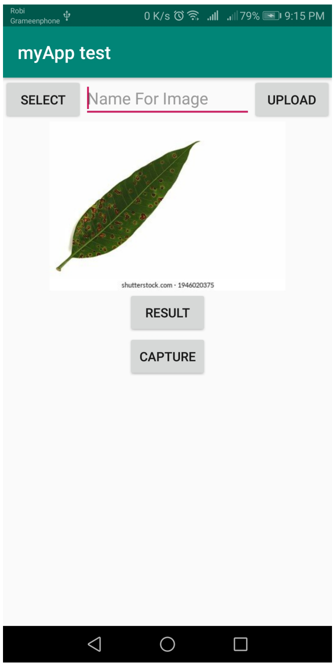


Figure 3.25: Fetching Image.

The image :

Figure 3.26: The Image.

After Image processing cropping the final diseased area :

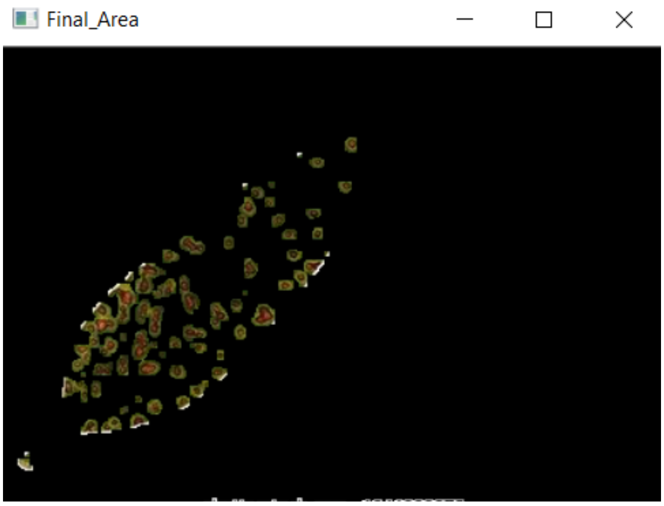


Figure 3.27: Diseased parts.

Generating the result image :

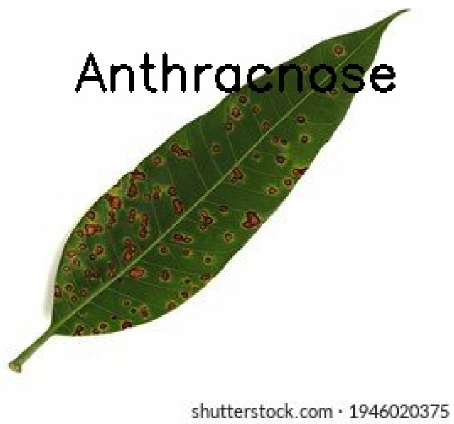


Figure 3.28: The Image.

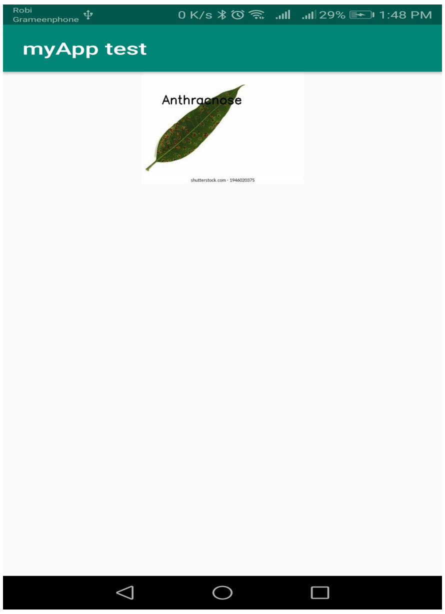


Figure 3.29: Result.

This is after getting the result on the app.

3.5 Conclusion

In this chapter step-by-step explanation is given about the methodology. The result analysis is discussed in next chapter.

Chapter 4

Results and Discussions

4.1 Introduction

In the previous chapter we have discussed all the steps in the methodology. Here we will talk about the technology used in thi system building.

For android design , Android Studio Version '3.5.5 was used and for Android device HUAWEI NOVA 2i (android version 8.0.0) was used.

For database localhost and XAMPP was used.

For image processing , PC (Windows 10 pro, CORE i5 processor) was used. Python3 (Jupyter Notebook and Spyder) was used and libraries like Ski-kit learn and OpenCV was used for image processing.

4.2 Dataset Description

Anthracnose detection of mango is not common and we donot see that much often. So i created my own dataset. There are 150 images that i collected and 30 images that i personally captured using my phone.

Figure 4.1: Dataset.

After feature extraction all the features were saved in a dataset CSV file from which data was fed into machine learning model.

	mean	width	height	std deviati	type
0	105.9139	8	13	30.75294	1
1	116.5873	38	25	43.19102	1
2	98.94223	62	42	28.64134	1
3	85.87148	13	10	48.62231	1
4	90.95167	18	12	37.39424	1
5	98.04774	17	29	26.61731	1
6	121.2642	32	25	47.61703	1
7	102.953	7	12	19.0599	1
8	113.1261	18	26	37.22362	1
9	94.20317	5	7	36.91006	1
10	91.34866	33	11	27.96746	1
11	95.00208	6	8	19.97	1
12	105.7351	17	15	14.46956	1
13	135.3199	26	48	44.02689	1
14	92.43898	18	6	44.45591	1
15	76.47225	14	19	33.68342	1
16	110.7153	7	8	10.46835	1
17	89.47891	61	46	35.63911	1
18	97.47738	7	8	32.33278	1
19	121.9242	7	20	25.35984	1
20	86.91423	274	53	39.82198	1
21	81.74088	56	24	31.16419	1
22	71.33872	18	15	27.80247	1
23	80.82247	20	16	36.92946	1
24	81.70864	7	9	26.42095	1
25	74.81827	28	16	23.26698	1
26	104.26	13	23	37.05448	1
27	73.55137	18	10	26.44658	1
28	97.51994	12	17	35.93986	1

Figure 4.2: Feature Dataset.

4.3 Impact Analysis

The system we built can have a huge impact on socially and economically . Thousands of farmers will be hugely benefitted by using this system.

4.3.1 Social and Environmental Impact :

Every year lots of mango fields are destroyed because of anthracnose disease. By using the app their time and money will be saved and they will earn a lot.

4.3.2 Economic Impact :

There will be increase in production of mango if the disease is detected as early stages. Because of this rotten mango will not affect the mangoes if the disease is detected and early.

4.4 Evaluation of Framework

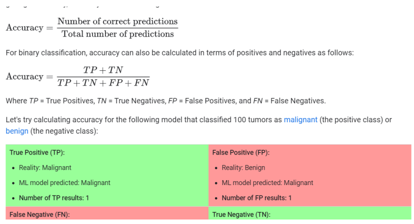
The framework that we developed is implemented properly and works fine. There is no network issues, error regarding the framework.

4.5 Evaluation of Performance

The android app is built properly and works perfect. So there is no way to measure how proper this works but we can calculate the decision making of the model that we developed by measuring the accuracy and precision.

Accuracy :

Accuracy is the fraction of predictions our models got right [20].



$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

For binary classification, accuracy can also be calculated in terms of positives and negatives as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where *TP* = True Positives, *TN* = True Negatives, *FP* = False Positives, and *FN* = False Negatives.

Let's try calculating accuracy for the following model that classified 100 tumors as malignant (the positive class) or benign (the negative class):

True Positive (TP): • Reality: Malignant • ML model predicted: Malignant • **Number of TP results: 1**	**False Positive (FP):** • Reality: Benign • ML model predicted: Malignant • **Number of FP results: 1**
False Negative (FN):	**True Negative (TN):**

Figure 4.3: Accuracy equation.

Evaluating the decision model : Using both SVM and Logistic Regression

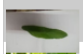
	Image	Accuracy	Precision
	Image_0012	94.07%	93.89%
	Image_0045	93.65%	93.62%
	Image_0054	85.34%	82.96%
	Image_0017	89.84%	86.90%
	Image_0069	95.67%	94.97%
	Image_0110	93.81%	92.72%
	Image_0034	79.56%	80.62%
	Image_0078	89.51%	90.70%

Figure 4.4: Accuracy comparison by SVM.

	Image	Accuracy	Precision
	Image_0012	95.6%	94.3%
	Image_0045	92.89%	90.1%
	Image_0054	89.76%	90.043%
	Image_0017	96.7%	96.83%
	Image_0069	86.45%	87.1%
	Image_0110	93.29%	93.6%
	Image_0034	79.56%	80.62%
	Image_0078	94.73%	92.85%

Figure 4.5: Accuracy comparison by Logistic Regression.

4.6 Discussion :

The difficulty in image processing is that each image is different. Sometimes the features are not extracted properly and this results the error in decision. Sometimes there is so much noise in the image that image processing does not work in this case.

4.7 Comparison :

In comparison with the paper [3] the accuracy was 95.2 % whereas it was just image processing but my system is incorporated with application. So in my opinion the use of our system mght be more useful.

In comparison with paper [4] the accuracy with good threshold was 90-98 % but with lower than threshold the accuracy falls to 70 % . Other than that although it was also an application it was not android. It was only for windows phone ,notebook .

4.8 Conclusion

Results and findings was added in this chapter.

Chapter 5

Conclusion

5.1 Conclusion

For this paper an android application is implemented using image processing techniques for detecting anthracnose disease of mango in real life environment.

5.2 Future Work

This system focuses on detection of anthracnose disease but in future there might be chnce to include several disease detection not only for mango but also for rice, jute etc. which are the natural crops of Bangladesh.

In android application more options can be added to send the result image to message or any other app.

References

[1] *Management of anthracnose in mango - the hindu*, https://www.thehindu.com/sci-tech/agriculture/Management-of-anthracnose-in-mango/article16300329.ece, (Accessed on 05/20/2021) (cit. on p. 1).

[2] R. M. and S. A. N. A., 'Groundnut leaf disease detection and classification by using back probagation algorithm,' in *2015 International Conference on Communications and Signal Processing (ICCSP)*, 2015, pp. 0964–0968. DOI: 10.1109/ICCSP.2015.7322641 (cit. on p. 6).

[3] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk and D. Stefanovic, 'Deep neural networks based recognition of plant diseases by leaf image classification,' *Computational Intelligence and Neuroscience*, vol. 2016, pp. 1–11, 2016. DOI: 10.1155/2016/3289801. [Online]. Available: https://doi.org/10.1155/2016/3289801 (cit. on pp. 6, 30).

[4] N. Petrellis, 'A smart phone image processing application for plant disease diagnosis,' in *2017 6th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, 2017, pp. 1–4. DOI: 10.1109/MOCAST.2017.7937683 (cit. on pp. 6, 31).

[5] D. Al Bashish, M. Braik and S. Bani-Ahmad, 'A framework for detection and classification of plant leaf and stem diseases,' in *2010 International Conference on Signal and Image Processing*, 2010, pp. 113–118. DOI: 10.1109/ICSIP.2010.5697452 (cit. on p. 7).

[6] N. Petrellis, 'A smart phone image processing application for plant disease diagnosis,' *2017 6th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, pp. 1–4, 2017 (cit. on p. 7).

[7] V. Sawarkar and S. Kawathekar, 'A review: Rose plant disease detection using image processing,' 2018 (cit. on p. 7).

[8] Y. Dandawate and R. Kokare, 'An automated approach for classification of plant diseases towards development of futuristic decision support system in indian perspective,' in *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2015, pp. 794–799. DOI: 10.1109/ICACCI.2015.7275707 (cit. on p. 7).

[9] S. Wallelign, M. Polceanu and C. Buche, 'Soybean plant disease identification using convolutional neural network,' in *FLAIRS Conference*, 2018 (cit. on p. 7).

[10] V. Singh, Varsha and A. Misra, 'Detection of unhealthy region of plant leaves using image processing and genetic algorithm,' *2015 International Conference on Advances in Computer Engineering and Applications*, pp. 1028–1032, 2015 (cit. on p. 7).

[11] A. Nadarajan and A. Prof.Thamizharasi, ‘Detection of bacterial canker disease in mango using image processing,’ *IOSR Journal of Computer Engineering*, vol. 19, pp. 01–08, 2017 (cit. on p. 7).

[12] J. G. A. Barbedo, ‘Digital image processing techniques for detecting, quantifying and classifying plant diseases,’ *SpringerPlus*, vol. 2, no. 1, Dec. 2013. DOI: `10.1186/2193-1801-2-660`. [Online]. Available: `https://doi.org/10.1186/2193-1801-2-660` (cit. on p. 8).

[13] P. S. Veling, ‘Mango disease detection by using image processing,’ *International Journal for Research in Applied Science and Engineering Technology*, vol. 7, no. 4, pp. 3717–3726, Apr. 2019. DOI: `10.22214/ijraset.2019.4624`. [Online]. Available: `https://doi.org/10.22214/ijraset.2019.4624` (cit. on p. 8).

[14] *Otsu thresholding - the lab book pages*, `http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html`, (Accessed on 05/22/2021) (cit. on p. 17).

[15] *Opencv: Image thresholding*, `https://docs.opencv.org/master/d7/d4d/tutorial_py_thresholding.html`, (Accessed on 05/22/2021) (cit. on p. 17).

[16] *Opencv: Morphological transformations*, `https://docs.opencv.org/master/d9/d61/tutorial_py_morphological_ops.html`, (Accessed on 05/22/2021) (cit. on p. 19).

[17] *What is image segmentation or segmentation in image processing?* `https://www.analytixlabs.co.in/blog/what-is-image-segmentation/`, (Accessed on 05/22/2021) (cit. on p. 19).

[18] *Data clustering algorithms - k-means clustering algorithm*, `https://sites.google.com/site/dataclusteringalgorithms/k-means-clustering-algorithm`, (Accessed on 05/22/2021) (cit. on p. 20).

[19] D. Khattab, H. M. Ebied, A. S. Hussein and M. F. Tolba, ‘Color image segmentation based on different color space models using automatic GrabCut,’ *The Scientific World Journal*, vol. 2014, pp. 1–10, 2014. DOI: `10.1155/2014/126025`. [Online]. Available: `https://doi.org/10.1155/2014/126025` (cit. on p. 20).

[20] *Classification: Accuracy | machine learning crash course*, `https://developers.google.com/machine-learning/crash-course/classification/accuracy`, (Accessed on 05/22/2021) (cit. on p. 29).