Plant leaf disease detection using image processing and machine learning

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology

in

Electrical and Electronics Engineering and Electronics & Instrumentation Engineering

by

Shrey Gupta(17BEI0077)

Ishan Goel(17BEE0288)

Rahul Gupta(17BEE0358)

Under the guidance of

Dr. G.K Rajini

School of Electrical Engineering,

Vellore Institute of Technology, Vellore

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

June, 2021

DECLARATION

I hereby declare that the thesis entitled "**Plant leaf disease detection using image processing and machine learning**" submitted by me, for the award of the degree of ***Bachelor of Technology in Electronics and Instrumentation/Electrical and Electronics Engineering*** to Vellore Institute of Technology is a record of bonafide work carried out by me under the supervision of Dr G.K Rajini.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place : Vellore

Date : 02 June 2021

Signature of the Candidates

Shrey Gupta 

Ishan Goel 

Rahul Gupta 

CERTIFICATE

This is to certify that the thesis entitled "Plant Leaf disease detection using image processing and machine learning" submitted by Rahul Gupta (17BEE0358), Shrey Gupta (17BEI0077), Ishan Goel (17BEE0288), School of Electrical Engineering, Vellore Institute of Technology, Vellore, for the award of the degree of ***Bachelor of Technology in Electronics and Instrumentation/Electrical and Electronics Engineering***, is a record of bonafide work carried out by him under my supervision, as per the Vellore Institute of Technology code of academic and research ethics.

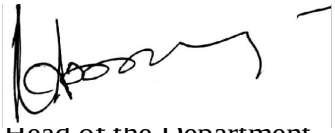
The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place : Vellore

Date : 02.06.2021 **Signature of the Guide**

The thesis is satisfactory

Approved by



Head of the Department

EEE/EIE

DEAN,SELECT

ACKNOWLEDGEMENTS

We are highly grateful to **DR. G. VISWANATHAN** Sir (Chancellor) and **DR. RAMBABU KODALI** Sir (Vice Chancellor) for giving us the opportunity to work on this project.

We are highly grateful to **Dr. S. SIVABALAN (DEAN SELECT School), Dr. N. AMUTHA PRABHA (HOD Dept. of Electronics and Instrumentation Engineering),** and **Dr. I. JACOB RAGLEND (HOD Dept. of Electrical and Electronics Engineering)** for providing us valuable guidance, suggestions, Ideas during whole project work.

We express our sincere thanks to **Dr. G.K. RAJINI,** our guide who guided us with valuable suggestions and guidance for completion of our project. She helped us understand the intricate issues involved in project- making besides effectively presenting it. These intricacies would have been lost otherwise. Our project has been completed successfully only because of her guidance.

We are also thankful to **School of Electrical Engineering** for providing technical support to carry out the project work.

Students Name

Shrey Gupta

Rahul Gupta

Ishan Goel

Executive Summary

Early stage disease detection of plants is very important to curb the infection to avoid the spread and loss. Our project helps to detect plant disease with the help of leaf images of the plant. Basically it is a classification problem where extraction of image features are done based upon the implementation of Image Processing Algorithms and supervised classification machine learning models to classify the plant as healthy or diseased.

Here two segmentation algorithms are deployed which are subroutines for image processing and three machine learning models and comparing the result with every combination on the basis of accuracy, precision, sensitivity, specificity and F1-score.
Here python has been implemented as our language of choice basically because of its extensive collection for machine learning libraries.

CONTENTS

.

List of Figures

List of Tables

List of Abbreviations

RGB	Red Green Blue
BGR	Blue Green Red
SLIC	Simple Linear Iterative Clustering
HSV	Hue Saturation Value
HSL	Hue Saturation Lightness
IDM	Inverse Difference Moment
GLCM	Gray level Co-occurrence Matrix
GUI	Graphic User Interface
GDP	Gross Domestic Product
SGDM	Study Group on Data Management
HSI	Hue, Saturation and Intensity

1. INTRODUCTION

Agriculture is an important sector of India. It contributes about 19.8% of India's GDP in the year 2020-21. Agriculture also provides employment to 60% of India's population. So protecting crops is very essential part to safeguard India's people and economy. So keeping this in mind, machine learning classifiers are implemented on a dataset obtained from the images of the plant leaves, hence achieving early detection of disease if any in the plant. Therefore, safeguarding other neighboring plants from getting damage, therefore reducing losses of farmers which also results in production of good quality of crops.

Pests are responsible for 20 to 40 percentage of crop damage every year. This includes high cost productivity and less yield of crops like wheat, rice, maize, soybean and potato. With the rapid increment in population size, pathogens and insects majorly decrease the food production.

1.1 OBJECTIVE

To do the performance analysis of the implemented classifier(classical Machine Learning models) on a given set of extracted features fed as a training data to the model and pre-processing performed with a given segmentation technique.

To achieve better accuracy in terms of classifying plants as diseased or normal.

1.2 MOTIVATION

Within the framework of disease detection, Plant Disease Detection is a crucial topic. The attention for the disease detecting programs and the promotion of various technologies are increasing tremendously.

In the longer term diseased plants affect human life too. In order to deal with this, the plant diseases can be identified at the initial stage itself and infection control tools can be used to solve pest problems while minimizing risks to people and the environment.

1.3 BACKGROUND

Demands for plant disease detection has been rising day by day globally. It is crucial to detect disease at an initial stage. There is a need for an initial stage prediction system to effectively determine the disease of the plant using a variety of features/algorithms. Even though there are websites that offer this service, their prediction method may not be the best.

1.3.1 Feasibility Study on Plant Chili Disease Detection Using Image Processing Techniques

by Md Shakaff and Rohani Binti S Mohamed Farook Abdul Hallis Bin Abdul Aziz, Zulkifli Bin Husin, Ali Yeon Bin published in January 2012.

MATLAB and LabView software used to detect chilli plant disease using Image Processing. By the help of this algorithm disease can be detected at an early stage and can determine the chilli plant healthiness. Harmful chemicals can also be reduced by utilizing this technique which in turn effects the cost production and quality of chilli[8].

1.3.2 Leaf Disease Detection and Prevention Using Image processing using MATLAB

by Priyanka Pawar, Mira Padwal and Priti Nagane Mira Nagane, Priyanka Bhosale, Prajakta Mitkal published in Feb 2016.

In this, grayscale has been transformed from RGB (Red Green Blue) by pre-processing which in turns removes the unwanted data from the image. Green pixels of solid space are contaminated. For extraction of sickness area specifically 3 computations are Linear SVM (Support Vector Machine), Non-linear SVM and Multiclass SVM [7].

1.3.3 A Framework for Detection and Classification of Plant Leaf and Stem Diseases

by Dheeb Al Bashish, Malik Braik and Sulieman Bani-Ahmad published in October 2010.

Detection of leaf and stem disease is incorporated through Image Processing. Texture of diseased leaf and stem can be analyzed through Color Co-occurrence Method. To classify the plant as diseased, back propagation algorithm has been implemented in neural network [2].

1.3.4 Remote Area Plant Disease Detection Using Image Processing by Sabah Bashir and Navdeep Sharma published in Sept-Oct 2012.

The threshold values are analyzed for every individual pixels value. Object pixel are the pixels if the threshold value is smaller than the pixel value. For future grouping K and Naïve Bayes are to be utilized [1].

1.3.5 Plant Leaf Disease Detection and Classification Using Image Processing Techniques by Prakash M. Mainkar, Shreekant Ghorpade and Mayur Adawadkar published in Sept 2015.

In this RGB is converted into space structure to create color transformation structure of RGB leaf in the first phase. Calculation of the segmented infected object texture and infected cluster's masked cells are to be removed. RGB is to be transformed into HIS (Hue, Saturation and Intensity) and H and S are created from SGDM (Study Group on Data Management) grid [5].

2. PROJECT DESCRIPTION AND GOALS

2.1 Project Description

The early stage disease detection is important to curb an infectious disease so as to not destroy the whole plantation. This project aims to classify the image given of a plant leaf as either healthy or diseased. This is achieved by implementing image processing and supervised machine learning models for classification.

Through image processing useful information/features have been extracted from the image which then is used to train the classification model.

The segmentation algorithms are:

1) Green Masking Technique
2) Super Linear Iterative Clustering(SLIC) Algorithm

The machine learning models are:

1) Gaussian Naïve Bayes
2) Support Vector Machine(SVM)
3) Random Forest

Methodology:

1) Data Collection:

The Data set used for this project is acquired from GitHub. It has 1711 images out of which 905 are healthy leaf images and 806 diseased images.

2) Preprocessing data:

The data is in form of images thus there's need to do some pre-processing before running the segmentation algorithm.

The image is converted to RGB color space from BGR (Blue Green Red) color space. Then it is converted to HSV color space from RGB color space. The image is converted to HSV color space as it results in more precise representation of extracted features of interest in the image. The images are resized into 500pixels*500pixels.

3) Extracting Feature from the image through image processing:

The feature extracted are Hu Moments (shape features), haralic features (textural features), color histogram using various python libraries like OpenCV, mahotas which have inbuilt functions like Hu Moments, calcHist, stddeviation etc to achieve extraction of various features, after extracting features, hey are rescaled that is upper and lower limits are set for them so as to have equal weightage of all the features in the classification.

4) Parting the informational index into preparing and testing:

The dataset obtained after performing above steps are splitted into training and testing dataset using train_test_split function and according to the parameter passed in the function which takes the value of percentage of test data.

5) Training the model:

The various models or the classifiers are trained from the training dataset obtained after performing above step.

6) Extracting the parameter of models using testing part:

After the model is trained, the classifier/model performance is then evaluated i.e. calculating various performance parameters like accuracy, precision, specificity, sensitivity, F1-scores using testing dataset obtained from the fourth step.

2.2 Goals

- To design a system to classify the images of leaves as healthy or diseased
- Implement green masking & SLIC algorithm as segmentation and Gaussian Naïve Bayes, Support Vector Machine & Random forest to achieve the same.

3. TECHNICAL SPECIFICATION

3.1 Hardware Specification:

- 2.3 GHz dual-core Intel Core i5
- Turbo Boost up to 3.6GHz
- 8GB of 2133MHz LPDDR4 onboard memory
- Nvidia Geforce 940mx graphic card

3.2 Software Specification:

- Python 3.7
- ScikitLearn
- Matplotlib
- Pandas and Numpy
- Anaconda and Jupyter Notebook
- Browser to run Jupyter Notebook(Preferably Google Chrome)
- OpenCv
- Mahotas

3.3 Module Description:

- **Pandas:** It is a vast open source library written in Python that enables you to perform data manipulation. It provides a simple way to create, control and change the data.

- **Numpy:** It is the primary library of Python to provide scientific computing, which has a very powerful n-dimensional array object. It also provides tools for accommodating C, C++ etc. It is also helpful in linear algebra, capability of random numbers, etc.
- **Matplotlib:** It is a library in Python specifically meant to plot data and its numerical maths extension NumPy. It gives us an object-oriented Application Program Interface (API) to embed plots into various applications using general Graphic User Interface toolkits.
- **Scikit-Learn:** It is the fundamental library of Python specifically meant for Machine Learning. It consists of various effective and efficient tools to perform Machine Learning and other statistical analysis like regression, classification, feature extraction, clustering, etc.
- **OpenCV:** OpenCV is a library in python which is used for computer vision and image processing.It is an open source library which was originally developed by Intel.
- **Mahotas**: This is an computer vision and image processing library

4. DESIGN APPROACH AND DETAILS:

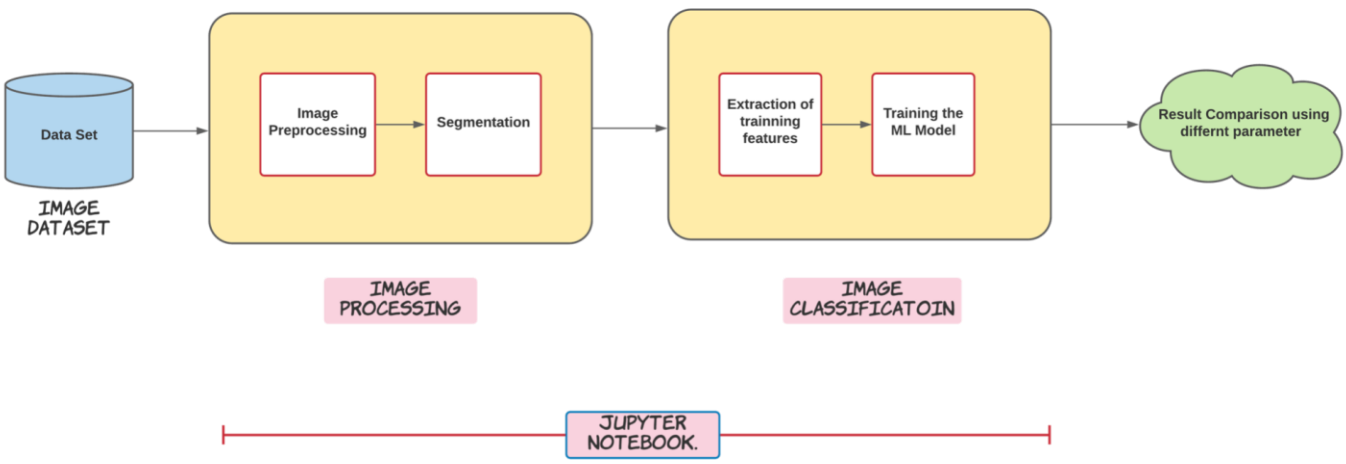


Figure 1: System Architecture

Figure. 1 represents the system architecture which includes image dataset block which contains the leaf images, Image processing block in which fetched dataset is preprocessed and images are segmented using python jupyter notebook, followed up by the extraction of various features from

the images in dataset and then training the classifiers using various libraries of python in Image Classification block.

Note: Python installed on open source computational webtool called jupyter notebook.

4.1 Image Processing:

Image Processing is a domain which represents processing of digital images with the help of a computer in order to extract useful features from an image or to enhance the quality of an image.

The steps involved in Image Processing are:

1. Image acquisition
2. Image preprocessing
3. Image segmentation
4. Extraction of Features

4.1.1 Image acquisition & Image preprocessing

First the data set loaded into a jupyter notebook using the imread command which is available in the OpenCV library. Then preprocessing is carried out to the entire data set using commands available in the OpenCV library the commands are BGR2RGB, HSV2RGB.

4.1.2 Image Segmentation:

In advanced picture preparation and PC vision, image segmentation is implemented to divide a digital image into numerous segments (sets of pixels, otherwise called picture objects). The objective of the segmentation is to streamline and additionally change the portrayal of a picture which is more significant and simpler to investigate. Image segmentation is normally used to find items and limits (lines, bends, and so forth) in pictures.

There are two Segmentation algorithms namely:

1) Green Masking Technique:

This is a threshold based segmentation technique. A range of pixel values are chosen which are called as mask. Each pixel is compared in the image with the chosen mask. IF the value of pixel is not in range it is set to 0 i.e. turned black. ELSE the value is not changed. During implementation, a green mask resulted in pure leaf image and no background color is observed. [8, 9].

2) Super Linear Iterative Clustering (SLIC) Algorithm:

SLIC is a type of clustering based segmentation algorithm. This algorithm creates the super-pixel by dividing image of n pixels into k clusters of equal size as the input parameter. Then the center of these clusters are adjusted according to the distance metric over different iterations. Finally an image is obtained on dividing into k clusters after iterations are over [4, 12].

ALGORITHM:

1) Initialize the clusters centers as the 5-D vector as (L_k, A_k, B_k, x, y) where L_k, A_k, B_k are L,A,B color space values respectively and x, y are the space coordinates of the cluster centers .These centers are initialized over a grid size of $\sqrt{k}/\sqrt{n}$.

2) Now, the image cluster centers are initialized at the point with lowest pixel gradient.

3) Now, each pixel in image is assigned to the closest cluster center in neighborhood of 2S*2S.The closest cluster center is calculated using following distance metric formulae:

$$d_{lab} = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2} \quad \text{..... (4.1)}$$

$$d_{xy} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \quad \text{..... (4.2)}$$

$$D_s = d_{lab} + \frac{m}{S} d_{xy} \quad \text{..... (4.3)}$$

$$S = \sqrt{N/K} \quad \text{..... (4.4)}$$

Where m defines the proximity considered.

4) Cluster centers are changed according to the average of the vectors associated with the concerned cluster.

5) New centers are calculated and residual error E till E<=threshold is obtained.

4.1.3 Features Extraction:

Using the segmented image, extracted features are used to train the machine learning model. Total of 550 features have been extracted.

Hu Moments:

In PC vision and image processing, picture moments are regularly used to portray the state of an item in a picture. These moments catch fundamental data like the space of the article, the centroid (for example the middle (x, y)- directions of the item), the direction, and other attractive properties. Hu proposed 7 moments that can be utilized to describe the state of an article in a picture. Moment-based estimations are regularly computation of the centroid (since all further second estimates are based on the underlying centroid).

Calculation of Hu Moments:

From a stringently factual perspective, "moments" are simply measurable assumptions for an irregular variable.

The customary snapshot of a shape in a double picture is characterized by:

$$M_{ij} = \sum_x \sum_y x^i y^j I(x, y) \quad \text{..... (4.5)}$$

Where I (x, y) is the pixel power esteem at the (x ,y)- organize.

To get interpretation invariance, shape estimations have been considered and compared with the centroid of the shape. The centroid is basically the center(x ,y)- directions of the shape, which has been characterize as x–and y–separately.

$$\bar{x} = M_{10}/M_{00} \quad \text{..... (4.6)}$$

$$\bar{y} = M_{01}/M_{00} \quad \text{..... (4.7)}$$

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q \; I(x, y) \quad \text{..... (4.8)}$$

In any case, these general moments don't have a great deal of discriminative capacity to address shapes, nor do they have any invariant properties. That is why Hu came up with these 7 moments.

$$M_1 = (\mu_{20} + \mu_{02}) \quad \text{..... (4.9)}$$

$$M_2 = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \quad \text{..... (4.10)}$$

$$M_3 = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{30})^2 \quad \text{..... (4.11)}$$

$$M_4 = (\mu_{30} - \mu_{12})^2 + (\mu_{21} - \mu_{03})^2 \quad \text{..... (4.12)}$$

$$M_5 = (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})((\mu_{30} + \mu_{12})^2 - 3(\mu_{21} - \mu_{03})^2) + (3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03})(3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2)$$

..... (4.13)

$$M_6 = (\mu_{20} - \mu_{02})((\mu_{30} + \mu_{12})^2 - (\mu_{21} - \mu_{03})^2) + 4\mu_{11}(\mu_{30} + 3\mu_{12})(\mu_{21} + \mu_{03})$$

..... (4.14)

$$M_7 = (3\mu_{21} - \mu_{30})(\mu_{30} + \mu_{12})((\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2) - (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03})(3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2)$$

..... (4.15)

This thus creates a feature vector of size 7.

$$V = [M_1, M_2, \ldots\ldots, M_7]$$ ..... (4.16)

Haralic Features:

Texture is a description of spatial arrangement of colors or intensities of an image.

Haralick texture feature is statistical based which is calculated from a Gray Level Co-occurrence Matrix, (GLCM), a matrix that counts the co-occurrence of neighboring gray levels in the image. The GLCM is a square lattice that has the element of the quantity of dim levels N in the district of interest (ROI) [3].

Gray Level Co-occurrence Matrix:

- This matrix is the heart of Haralic features.
- This matrix is calculated on gray scale images.
- This is a type of square matrix of order n*n where n is the order of quantization or number of different gray level (i.e. the number of pixel considered for calculating GLCM) in an image.
- There is a direction component involved in this matrix which needs to be chosen before the start of calculation of GLCM
- If red pixel is of interest then these are the degree of direction which should be provided before the start of calculation

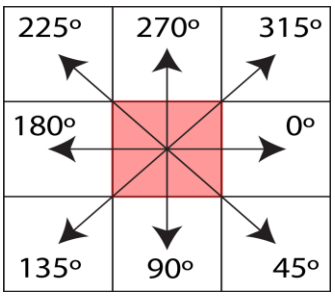


Figure 2: Direction in GLCM

Figure. 2. Shows the various directions which are considered to calculate the GLCM i.e. the occurrences of pixel i from pixel j in various directions like 0 degree to pixel j, 315 degree to pixel j etc. as mentioned above.

Calculation of GLCM:

1. Provide the gray scale image of quantization 'n' and direction 'd'
2. A square matrix mat[n][n] created
3. $\forall\, i,j \ \in n|\ mat[i][j] = \ count\ of\ pair\ such\ that\ j\ is\ in\ direction\ d\ from\ i$
4. Transpose of this matrix is added to itself to form symmetric matrix
5. Then each element is divided with the sum of all elements in the matrix to normalize it.
6. This matrix is the GLCM.

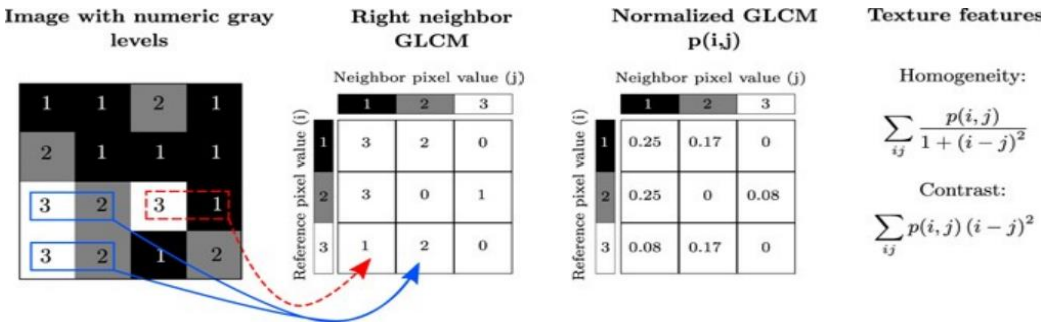


Figure 3: Calculation of GLCM

Figure 3. represents the occurrence of neighboring pixel i from the referenced pixel j in 0deg direction, and the third matrix shows the normalized matrix by dividing the matrix with the sum

of all the values in the matrix, fourth figure indicates the features mathematical representation calculated form the normalized matrix.

Color Feature extraction using color histogram:

Color Histogram is the most advanced tool for separating the shading highlight of a picture. It addresses the picture from an alternate point of view. It addresses the recurrence dispersion of shading containers in a picture. It checks similar pixels and stores them. [10, 11]

Color feature extraction using color co-occurrence matrix:

- The color co-occurrence matrix is another method to find out color features of an image.
- Some of the features extracted are: Hue Energy, Saturation (HSL) entropy, Saturation (HSV) entropy, a(L*a*b) entropy, Saturation (HSL) IDM, Value IDM, Red SumMean, Green SumMean, Grey SumMean, Light SumMean etc [11].

4.2 Image Classification:

Plant disease detection is a classification problem where leaf is classified as either diseased or healthy. Hence supervise classification learning models are used.

Classification Algorithms used are:

1) Gaussian Naïve Bayes

2) Support Vector Machine (SVM)

3) Random Forest Algorithm

4.2.1 Gaussian Naïve Bayes

Naïve Bayes is a probabilistic Artificial Intelligence calculation dependent on the Bayes Theorem, utilized in a various ways of arrangements. Bayes' Theorem is a basic numerical equation utilized for figuring contingent probabilities. There are some assumptions made by this algorithm which impacts on the accuracy of this algorithm.

1) It assumes all features are independent

2) It gives equal importance to all the features i.e. it assumes that all the features have equal weight in classification.

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)} \quad \text{..... (4.17)}$$

Algorithm:

1. From Training set find the prior probability

P(diseased)=(diseased)/(diseased + healthy)

P (healthy)=(healthy)/(diseased + healthy).

2. Find the mean, variance for all the 550 features given the target is plant is healthy or diseased separately from the test data, i.e. if plant is healthy calculate the mean and variance of the given feature.

3. From the test data, for all the leaves it can be predicted whether the leaf belongs to which class i.e. healthy or diseased.

Mathematical Formulation of:

$$P\left(y|x_{1\,,\ldots\ldots,}\ x_n\right) = \frac{P(x_1|y)P(x_2|y)\ldots P(x_n|y)P(y)}{P(x_1)P(x_2)\ldots P(x_n)} \quad \text{..... (4.18)}$$

$$P\left(y|x_{1\,,\ldots\ldots,}\ x_n\right) \propto P(y)\prod_{i=1}^{n} P(x_i|y) \quad \text{..... (4.19)}$$

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}}\ \exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right) \quad \text{..... (4.20)}$$

Where:

- P(y)=prior probability
- μ_y = mean of feature x given class Y or in this case diseased plant or healthy one (from training data)
- X_i= i^{th} feature value for the test data
- σ_y^2 = variance of feature x given class Y or in this case diseased plant or healthy one (from training data)

4.2.2 Support Vector Machine (SVM):

This is a supervised machine learning algorithm which can be used for both classification and regression type of problems. Two class SVM model is used in this project (diseased class and healthy class). It works well with a larger amount of features and data as well. The objective of SVM is to plot a hyperplane that divides the labelled data into two halves.

These points are called support vectors. There are multiple such hyperplanes but there is one more constraint i.e. it needs to maximize the margin distance. Margin distance is calculated as distance between two hyper planes [5].

4.2.3 Random Forest:

This is a classification algorithm which includes subroutines called as decision tree. In this algorithm the data set is splitted into multiple parts and different decision trees are formed.The final value is calculated by taking an average of the output of different decision trees and the algorithm returns the result. Bagging method is used in Random Forest i.e. multiple models are fitted into one algorithm.

A subset of information is made by haphazardly choosing x number of highlights (sections) and y number of models (columns) from the first dataset of n highlights and m models[5].

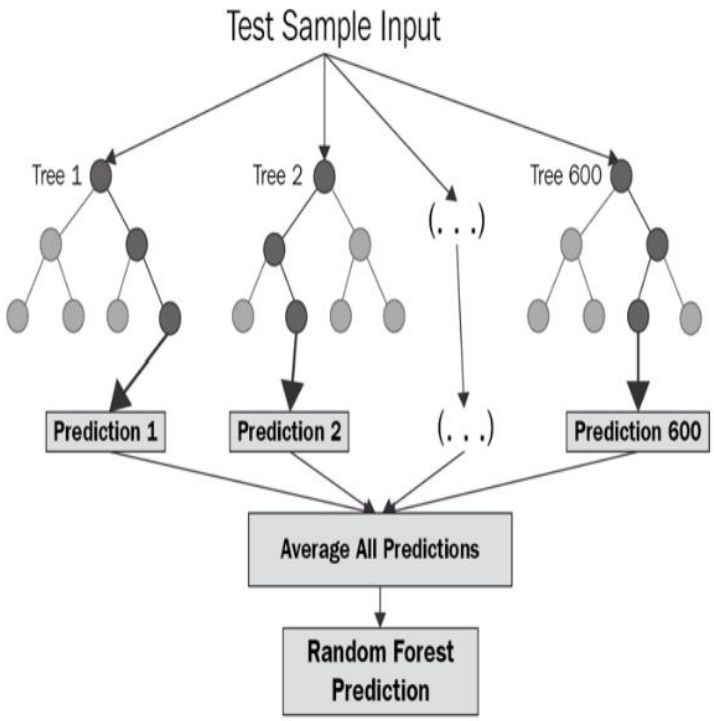


Figure 4: Random Forest

Figure 4. represents the various decision trees built by the random forest algorithm to take the decision based on the average of all the results obtained from them.

Information Gain

The information gain is the amount by which the Entropy of the system reduces due to the split that is implemented. The algorithm has created the tree using observations. It is splitted with respect to feature X when Gain(S,A) is maximum among the remaining features.

$$E = -\sum_i^C p_i \, log_2 p_i \qquad \text{..... (4.21)}$$

The entropy of the framework is determined by the above equation where p(x) is the likelihood of getting class x from those 14 individuals (expecting). It has two classes here one is Yes and the other is No in the objective segment.

E(S)= -[p(Yes)*log(p(Yes))+ p(No)*log(p(No))];

Here yes means diseased otherwise healthy

Basic Algorithm:

1. A← the "best" choice characteristic for a hub N.
2. Assign A as a choice property for the hub N.
3. For each worth A (attribute/characteristic), make another relative of hub N.
4. Sort preparing guides to leaf hubs.
5. On the off chance that preparation models are completely characterized, stop else emphasize over new leaf hubs.

5. SCHEDULE, TASKS, MILESTONE

Task	Description	Timeline	Duration
Problem Statement	The goals and objectives of the Plant Disease Detection was defined using different algorithms to solve the problem and compare them.	09.02.2021 - 11.02.2021	2 days
Literature Survey	Analysis of existing models used to solve the Plant Disease Detection Problem Comparison of standard Image Segmentation and machine learning algorithms.	12.02.2021 - 15.02.2021 & 20.02.2021 - 25.02.2021	10 days
Dataset Collection	Various dataset providers like UCI repository, Kaggle and GitHub were searched for a good dataset relevant to the problem.	26.02.2021 - 28.02.2021	3 days

Data Set Pre-Processing	The dataset was pre-processed according to our needs.	01.03.2021 - 05.03.2021	5 days
Image Segmentation (Green Masking Technique)	Attempted to apply Green Masking Technique for Image Segmentation.	06.03.2021 - 10.03.2021	5 days
Image Segmentation (SLIC Algorithm)	Attempted to apply SLIC Algorithm for Image Segmentation.	11.03.2021 - 15.03.2021	5 days
Learnt about Classification Algorithms	Researched about different classification algorithms which would suit our problem statement.	20.03.2021 - 31.03.2021	12 days
Implementation of Naïve Bayes Algorithm	Attempted to apply Naïve Bayes Algorithm for Image Classification.	01.04.2021 - 03.04.2021	3 days
Implementation of SVM	Attempted to apply SVM Algorithm for Image Classification.	04.04.2021 - 08.04.2021	5 days
Implementation of Random Forest	Attempted to apply Random Forest Algorithm for Image Classification.	09.04.2021 - 15.04.2021	7 days
Options to improve accuracy	Modifications like normalizing the data or using Standard or MinMax Scalar to convert the highly varying data to a simpler form in a hope to increase the accuracy.	16.04.2021 - 17.04.2021	2 days

Comparison of Result	Compared the performance of classification algorithms on the basis of different parameters.	18.04.2021 - 19.04.2021	2 days
Theoretical Comparison	Did theoretical comparison of classification algorithms.	20.04.2021 - 21.04.2021	2 days
Thesis	The final step was to compile all my work review and prepare my project thesis.	10.05.2021 - 23.05.2021	14 days

Table 1: Schedule, Task, Milestone

Table 1 represents the various processes, methodologies implemented in the project, their description and timeline.

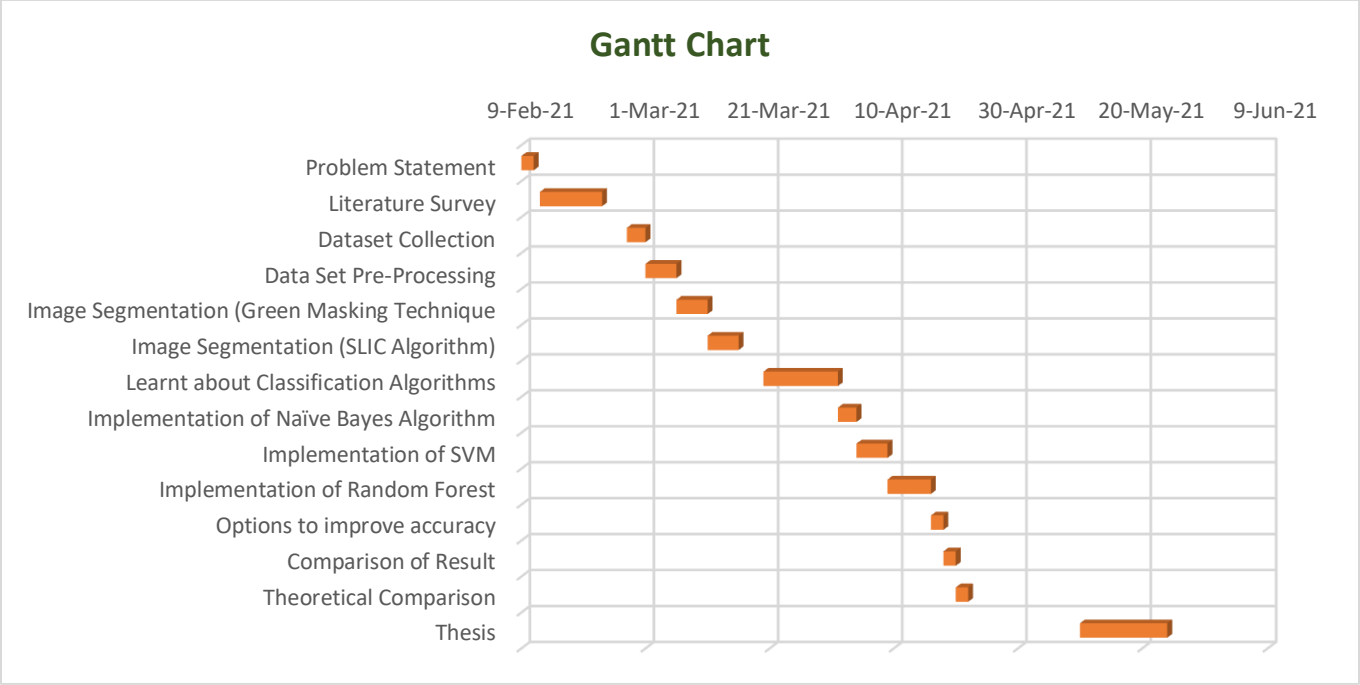


Figure 5: Gantt chart

Figure 5. represents the chunk of time taken by various processes carried out in this project.

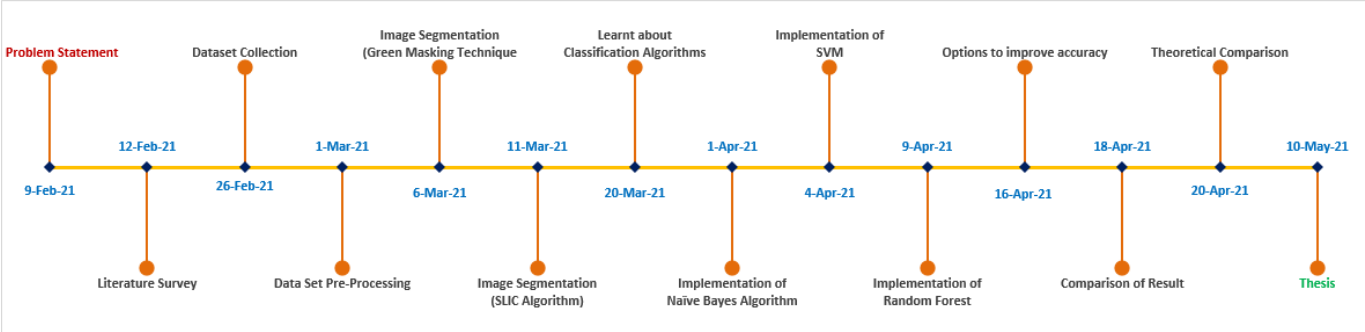


Figure 6: Milestone chart

Figure 6. displays the time at which a particular process or methodology is implemented successfully.

6. PROJECT OUTPUTS:

6.1 Green Masking Segmentation output

Output description: first image is the original image, second image is image in LAB color space, third image is a binary image i.e. it has only two color pixels value (leaf part is separated from the rest) ,fourth and fifth images are obtained by blackening the pixels of leaf surrounding leaving the spots and surrounding of leaf respectively, last image is obtained after masking (fourth image and original image bitwise and) is applied on original leaf image which results into the testing image used in extracting features.

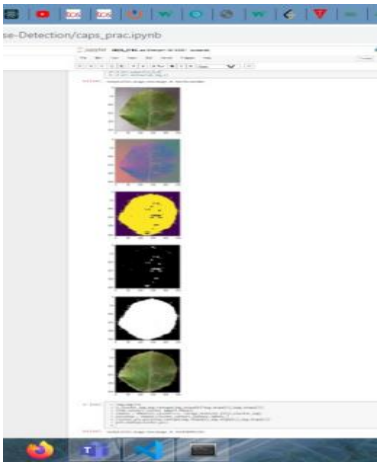


Figure 7: Green Masking Segmentation output

Figure 7 shows the various stages or images involved to finally create a mask to be implemented on original leaf image.

6.2 SLIC Segmentation output:

The first image depicts the clustering of image in 4 parts which is based on the pixel similarity and pixel proximity, second image is the result of turning some of those clusters which are not useful for extracting features into black pixels and other clusters original leaf.

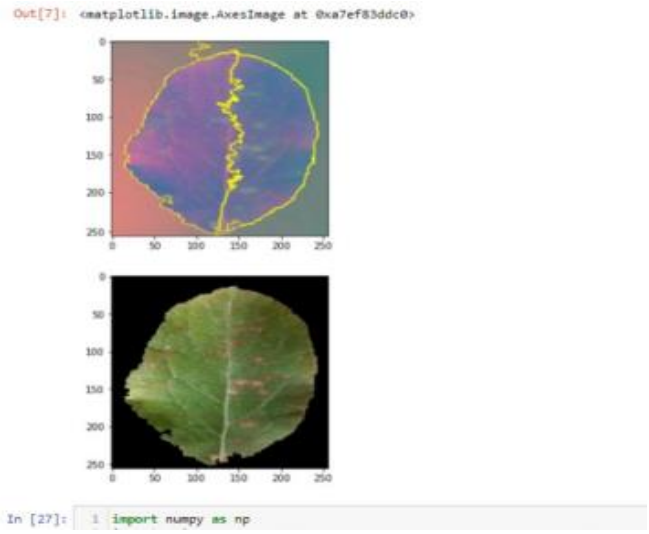


Figure 8: SLIC Segmentation output

Figure 8. represents the output after performing clustering using SLIC segmentation technique and using those clusters to create a segmented image.

6.3 Gaussian Naïve Bayes output:

a) Using Green Masking:

The following output is obtained, as this algorithm uses Bayes theorem to label certain regions into different classes, therefore accuracy is slightly less on a lower side as it is less adaptable to image intensity and many other parameters.

Accuracy 86.8421052631579%

[[117 40] [5 180]]

[STATUS] splitted train and test data... Train data: (1367, 549) Test data: (342, 549)

Precision=TP/TP+FP: 95.90163934426229 %

Sensitivity=TP/TP+FN: 74.52229299363057 %

Specificity=TN/TN+FP: 97.29729729729729 %

F1-Score: 83.87096774193547 %

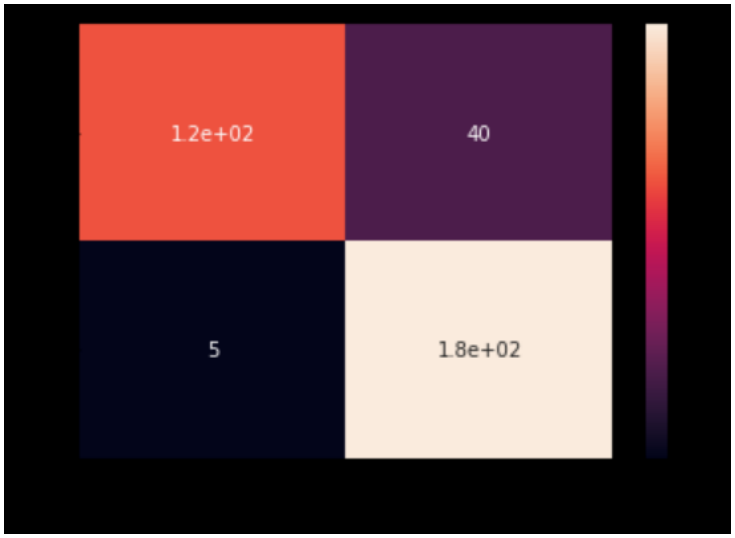


Figure 9: Confusion Matrix of Gaussian naïve Bayes using Green Masking Technique

Figure 9. is the matrix which shows the true positives (120, actual diseased and predicted diseased), true negatives (180, actually healthy and predicted healthy), false positives (5, actually healthy but detected diseased) and false negatives 40, predicted healthy but actually diseased.

b) Using SLIC Segmentation:

The following results are obtained using SLIC segmentation and Gaussian Classifier where accuracy is less as compared to masking due to high percentage error in segmentation part, which subsequently has an effect over other performance parameters also.

Accuracy 85.38011695906432%

[[114 43] [7 178]]

[STATUS] splitted train and test data... Train data: (1367, 549) Test data: (342, 549)

Precision=TP/TP+FP: 94.21487603305785 %

Sensitivity=TP/TP+FN: 72.61146496815287 %

Specificity=TN/TN+FP: 96.21621621621621 %

F1-Score: 82.01438848920864 %

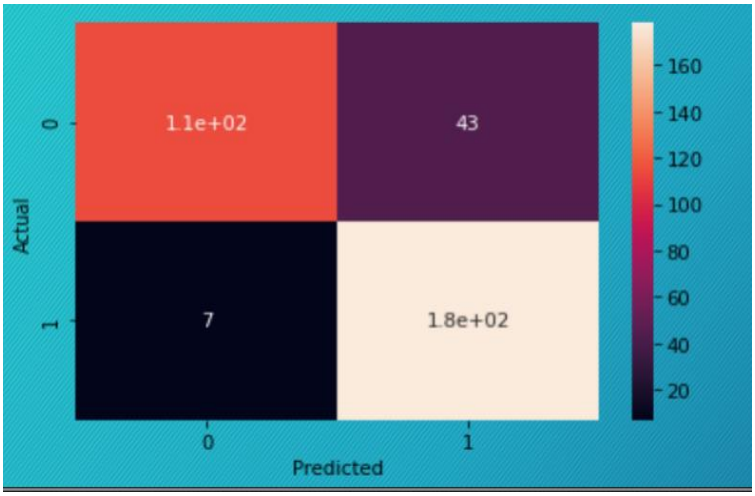


Figure 10: Confusion Matrix of Gaussian naïve Bayes using SLIC Technique

Figure 10. is the matrix which shows the true positives (110, actual diseased and predicted diseased), true negatives (180, actually healthy and predicted healthy), false positives (7, actually healthy but detected diseased) and false negatives (43, predicted healthy but actually diseased.

6.4 Support Vector Machine:

a) Using Masking Segmentation:

The performance parameters values obtained from the support vector machine are given below ,and in this case accuracy is quite high(93.85%) comparatively higher than Gaussian Naïve Bayes as well as other parameters making it more suitable for plant disease detection.

Accuracy: 93.85964912280701%

[[149 8] [13 172]]

Precision=TP/TP+FP: 91.9753086419753 %

Sensitivity=TP/TP+FN: 94.90445859872611 %

Specificity=TN/TN+FP: 92.97297297297297 %

F1-Score: 93.41692789968651 %

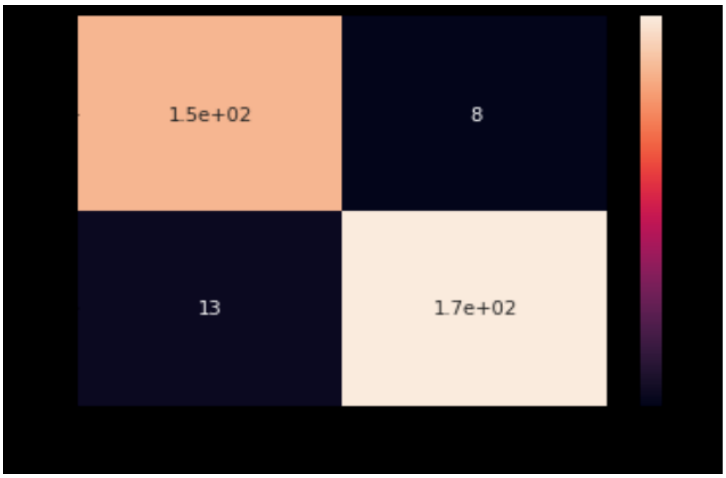


Figure 11: Confusion Matrix of SVM using Green Masking Technique

Figure 11. is the matrix which shows the true positives (150, actual diseased and predicted diseased), true negatives (170, actually healthy and predicted healthy), false positives (13, actually healthy but detected diseased) and false negatives (8, predicted healthy but actually diseased.

b) Using SLIC Segmentation:

In this case SVM performs better than masking and there is a significant increase in accuracy i.e. (95.61%), precision, and specificity and in sensitivity as well.

Accuracy: 95.6140350877193%

[[154 3] [12 173]]

Precision=TP/TP+FP: 92.7710843373494 %

Sensitivity=TP/TP+FN: 98.08917197452229 %

Specificity=TN/TN+FP: 93.51351351351352 %

F1-Score: 95.35603715170278 %

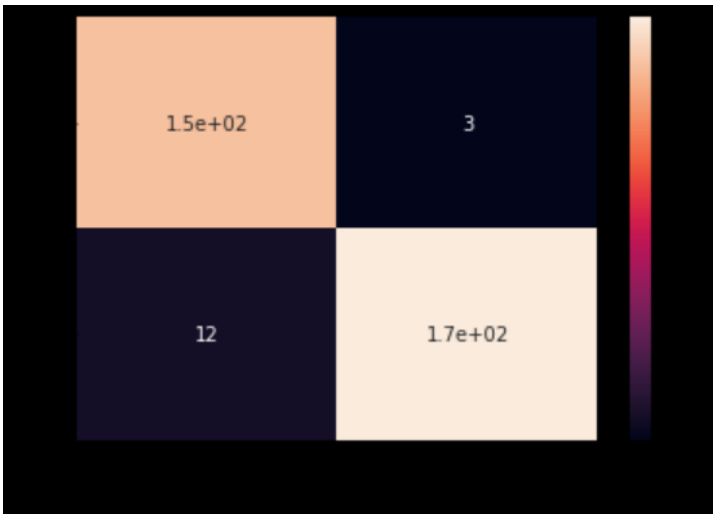


Figure 12: Confusion Matrix of SVM using SLIC Technique

Figure 12. is the matrix which shows the true positives (150, actual diseased and predicted diseased), true negatives (170, actually healthy and predicted healthy), false positives (12, actually healthy but detected diseased) and false negatives (3, predicted healthy but actually diseased.

6.5 Random Forest

As it can be seen from the below Random forest achieves the best accuracy, precision, specificity and sensitivity results as compared to Gaussian naïve Bayes and Support vector machine.

a) Using Masking Segmentation:

In this case Accuracy of 97.36%, precision of 96.83%, Sensitivity of 97.45% and F1 score of 97.14% is obtained.

Accuracy: 97.36842105263158%

[[153 4] [5 180]]

Precision=TP/TP+FP: 96.83544303797468 %

Sensitivity=TP/TP+FN: 97.45222929936305%

Specificity=TN/TN+FP: 97.29729729729729 %

F1-Score: 97.14285714285715 %

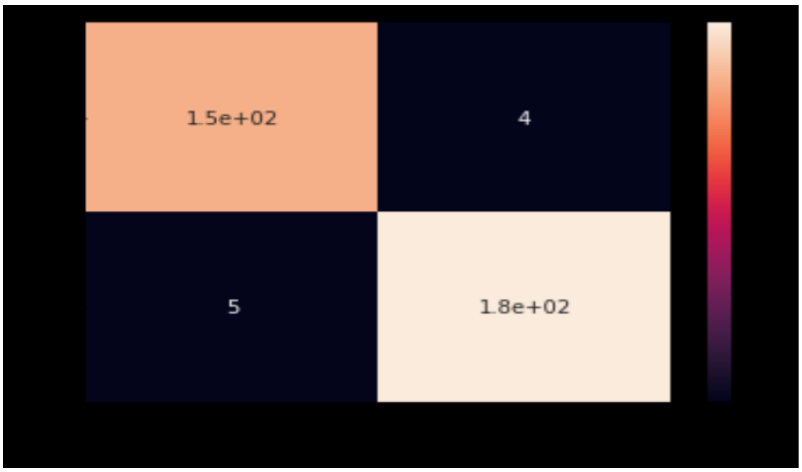


Figure13: Confusion Matrix of Random Forest using Green Masking Technique

Figure 13. is the matrix which shows the true positives (150, actual diseased and predicted diseased), true negatives (180, actually healthy and predicted healthy), false positives (5, actually healthy but detected diseased) and false negatives (4, predicted healthy but actually diseased.

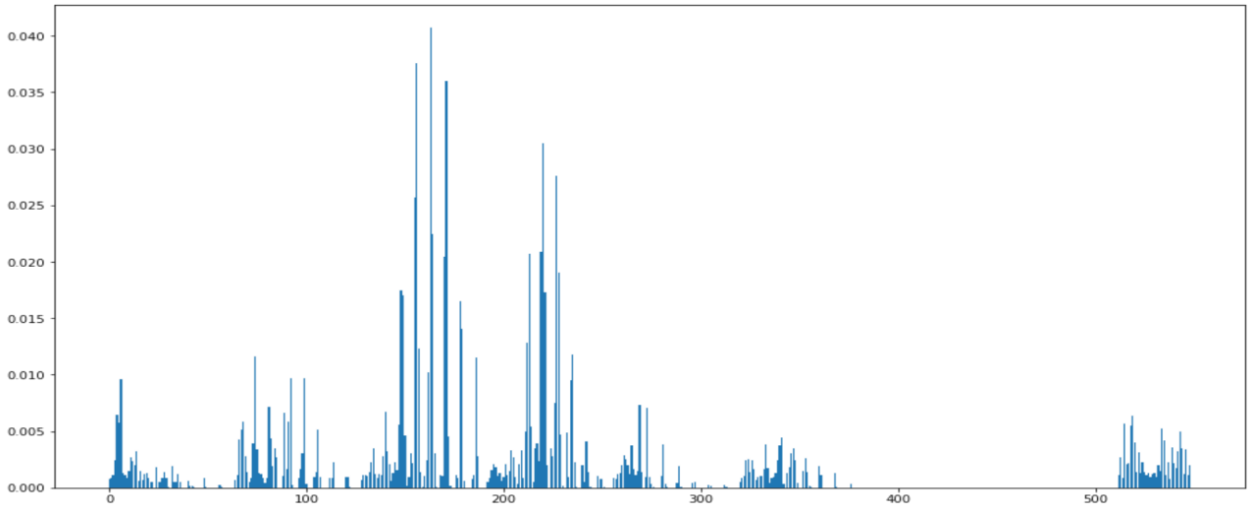


Figure 14: Weight of each feature of Random Forest using Green Masking Technique

Figure 14. represents the contribution of each feature (in percentage) in classification of plant as a diseased or a healthy.

b) Using SLIC Segmentation:

In this case the performance of the classifier remains the same irrespective of the segmentation technique followed. It achieves the accuracy of 97.36%, precision of 96.835%, Sensitivity of 97.45%, specificity of 97.29%.

Accuracy: 97.36842105263158%

[[153 4] [5 180]]

Precision=TP/TP+FP: 96.83544303797468 %

Sensitivity=TP/TP+FN: 97.45222929936305 %

Specificity=TN/TN+FP: 97.29729729729729 %

F1-Score: 97.14285714285715 %

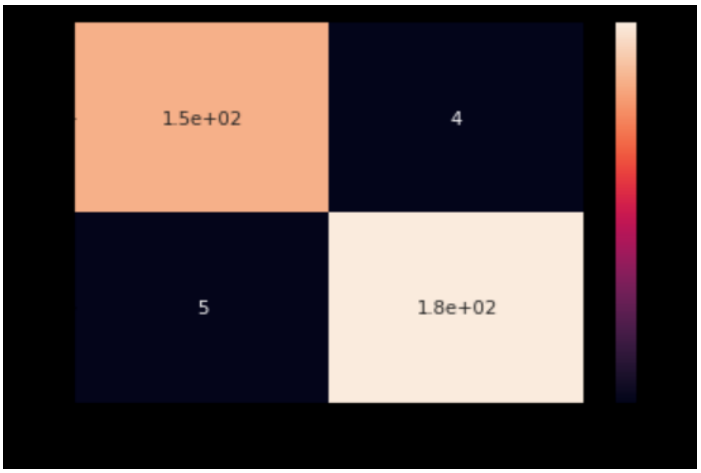


Figure 15: Confusion Matrix of Random Forest using SLIC Technique

Figure 15. represents the matrix which shows the true positives (150, actual diseased and predicted diseased), true negatives (180, actually healthy and predicted healthy), false positives (5, actually healthy but detected diseased) and false negatives (4, predicted healthy but actually diseased.

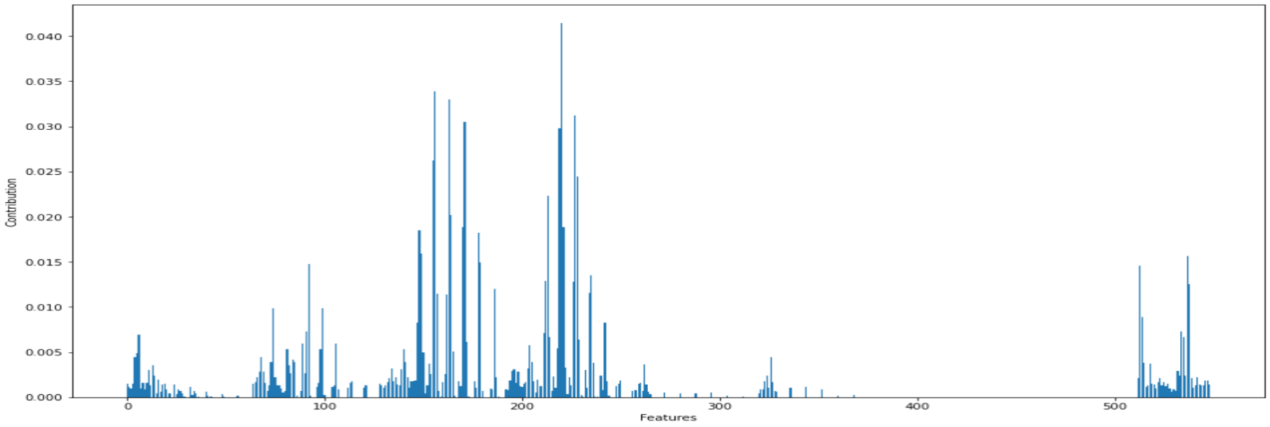


Figure 16: Weight of each feature of Random Forest using SLIC Technique

Figure 16. represents the contribution of each feature (in percentage) in classification of plant as a diseased or a healthy.

7. Results and Discussion:

The below results shows the values for the performance parameters: Accuracy, Precision, Sensitivity, Specificity, F1-score for each of the classifiers for Masking technique in the form of Bar Charts and table for comparing the performance of the various classifiers .

7.1 Bar Graph Comparison using Masking segmentation:

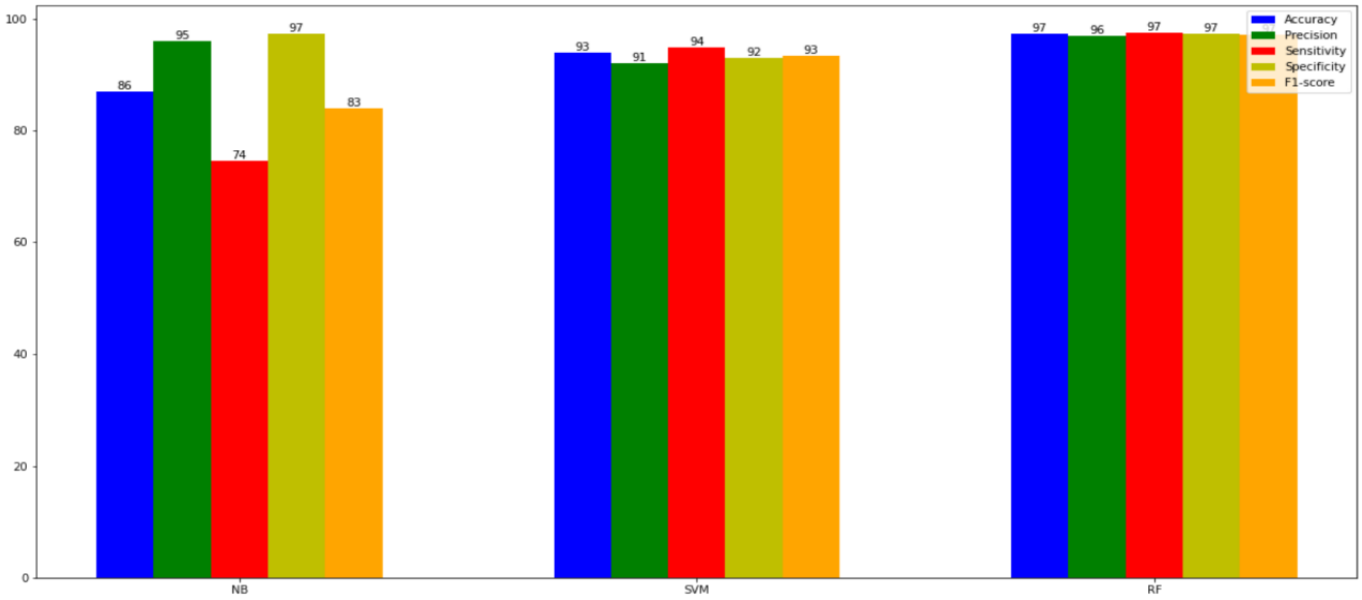


Figure 17: Bar Graph comparison of classification algorithm using Green Masking Technique

Figure 17 represents bar graph which shows x axis as various classifiers and each of their performance parameters, y-axis shows the value in percentage for each of the entities mentioned on x-axis.

Algo used	Accuracy(%)	Precision(%)	Sensitivity (%)	Specificity(%)	F1_score(%)
Gaussian Naive Bayes	86.842105	95.901639	74.522293	97.297297	83.870968
Support Vector Machine	93.859649	91.975309	94.904459	92.972973	93.416928
Random Forest	97.368421	96.835443	97.452229	97.297297	97.142857

Table 2: Tabular Comparison of classification algorithm using Green Masking Technique

Table 2. displays the values of performance parameters in (%) for the different classifier used.

7.2 Bar Graph and Tabular Comparison using SLIC segmentation

The below results shows the values of performance parameters: Accuracy, Precision, Sensitivity, Specificity, F1-score for each of the classifiers for SLIC Segmentation technique in the form of Bar Charts and table for comparing the performance.

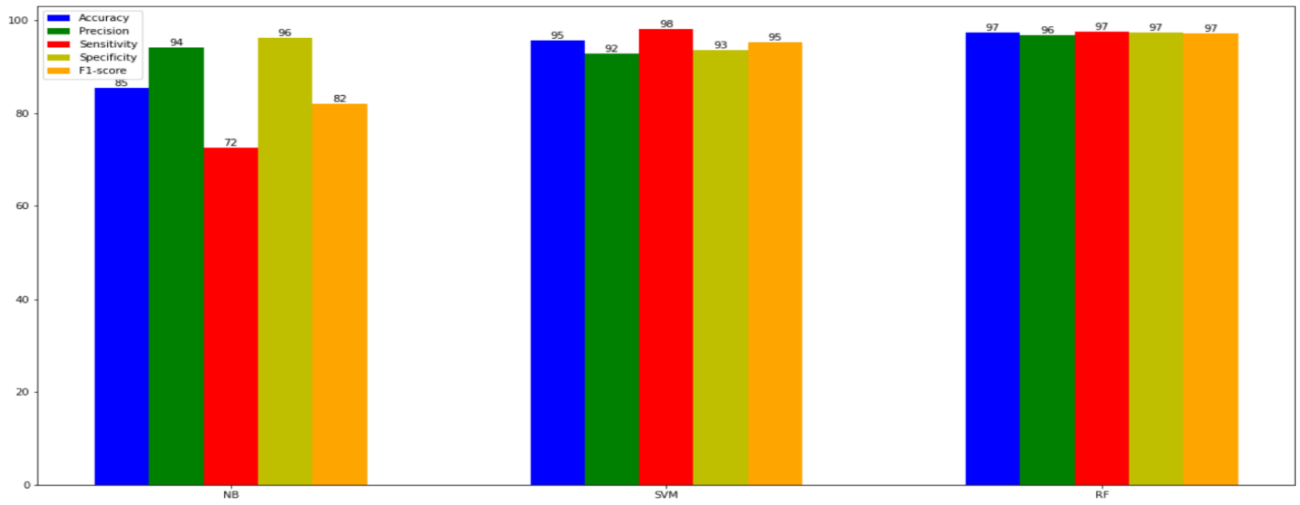


Figure 18: Bar Graph comparison of classification algorithm using SLIC Technique

Figure 18 represents bar graph in which x-axis shows the various classifiers and each of their performance parameters, y-axis shows the value in percentage for each of the entities mentioned on x-axis.

Algorithm used	Accuracy (%)	Precision (%)	Sensitivity (%)	Specificity (%)	F1_score (%)
Gaussian Naive Bayes	85.380117	94.214876	72.611465	96.216216	82.014388
Support Vector Machine	95.614035	92.771084	98.089172	93.513514	95.356037
Random Forest	97.368421	96.835443	97.452229	97.297297	97.142857

Table 3: Tabular Comparison of classification algorithm using SLIC Technique

Table 3. displays the values of performance parameters in (%) for the different classifier used.

7.4 Overall Analysis:

The Overall Analysis provides different conclusions that can be drawn from the performance of the various classifiers i.e. which classifier is the best and which is not suitable for the detection of disease in plants.

1) **Gaussian Naïve Bayes performs the least beneficial among the three:**

As it is a parametric model, thus numerical condition works under explicit conditions so that its accuracy is less compared to others as it is not flexible and assumes parameters are independent which might not be the case.

In simple words,

P(Z| X, Y) = P(Z|X) * P(Z|Y), which naive Bayes classification assumes.

2) **SVM works better than Naïve Bayes**

As it chooses the most optimal model of all the models it generates i.e. hyperplane with the maximum margin. Kernel can be used to increase the dimensionality of the feature set, therefore making separation of data into classes easier than in the lower dimensionality. But works less efficiently than random forest.

3) **Random forest works best**

As at each level of splitting the node it considers the best feature to separate into classes in the form of maximum information gain. Since it's a rule based algorithm, decision making is better than other algorithms.

8. COST ANALYSIS:

No paid software has been used in this project. Also the data set was freely available on GitHub. Thus the overall cost of the project is null exclusive System cost as well.

System (Laptop) cost – INR 55,000/-

9. SUMMARY:

The solution for the problem (Plant disease detection using Image Processing and Machine Learning) is carried out using the different combinations of segmentation and classification algorithms. Before performing image segmentation the images in the data set are preprocessed for uniform results.Both algorithms (SLIC and Green Masking Techniques) were implemented in code with the help of the OpenCV library in python.

In image classification, three different algorithms (Gaussian Naïve Bayes, SVM and Random Forest) which results in a good amount of variation. Through the data, the least performing algorithm was Gaussian naive Bayes which had accuracy of around 85%. Next was SVM whose accuracy was around 95% and the best performing algorithm was Random Forest with an accuracy of around 98%. Random Forest performs the best because at each level it chooses the best possible split hence resulting in high accuracy. The implementation of these algorithms were done with the help of ScikitLearn library in python.

10. REFERENCES:

[1] Sabah Bashir and Navdeep Sharma "Remote Area Plant Disease Detection Using Image Processing" by IOSR Journal of Electronics and Communication Engineering Sept-Oct 2012.

[2] Dheeb Al Bashish, Malik Braik and Sulieman Bani-Ahmad "A Framework for Detection and Classification of Plant Leaf and Stem Diseases" 2010 IEEE International Conference on Signal and Image Processing.

[3] D. Gadkari. 2004. "Image Quality Analysis Using GLCM," M.E. thesis, University of Central Florida, Orlando, Florida.

[4] Cong, Jinyu & Wei, Benzheng & Yin, Yilong & Xi, Xiaoming & Zheng, Yuanjie. (2014). Performance evaluation of SLIC algorithm on medical image processing. 24. 3231-8. 10.3233/BME-141145.

[5] Prakash M. Mainkar, Shreekant Ghorpade and Mayur Adawadkar "Classification and Detection of Plant Leaf Disease Using IP Techniques" International Journal of Innovative and Emerging Research in Engineering Volume 2, Issue 4, 2015, e-ISSN: 2394 – 3343, p-ISSN: 2394 – 5494.

[6] Prakash M. Mainkar, Shreekant Ghorpade and Mayur Adawadkar "Plant Leaf Disease Detection and Classification Using Image Processing Techniques**"** International Journal of Innovative and Emerging Research in Engineering Plant Leaf Disease Detection and Classification Using Image Processing Techniques January 2015.

[7] Priyanka Pawar, Mira Padwal and Priti Nagane Mira Nagane, Priyanka Bhosale, Prajakta Mitkal "Leaf Disease Detection and Prevention Using Image processing using MATLAB" International Journal of Recent Trends in Engineering and Research 2016.

[8] Zulkifli Bin Husin, Abdul Hallis Bin Abdul Aziz, Ali Yeon Bin Md Shakaff and Rohani Binti S Mohamed Farook "Feasibility Study on Plant Chili Disease Detection Using Image Processing Techniques" 2012 IEEE Third International Conference on Intelligent Systems Modelling and Simulation, pp. 978-0-7695-4668-1/12.

[9] Sachin D. Khirade, A. B. Patil, "Detection of Plant Disease and classifying them Using Image Processing" 2015 IEEE International Conference on Computing Communication Control and Automation, pp. 978-1-4799-6892-3/15.

[10] Q. Zhao, J. Yang, and H. Liu, “Retrieval of Stone Images using Color Histogram”, in IEEE International Conference on Image Analysis and Signal Processing, 2009, pp. 157-161.

[11] C. H. Lin., R. T. Chen, and Y. K. Chan, “Image retrieval system of a smart Content based on color and texture feature”, Image and Vision Computing, Vol. 27, No. 6, 2009, pp. 658–665.

[12] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Su¨ sstrunk, “State-of-the-Art Superpixel Methods in comparison to SLIC Superpixels method” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 34, No. 11, November 2012

[13] https://www.semanticscholar.org/paper/Leaf-Disease-Detection-and-Prevention-Using-Image-Mitkal-Pawar/d5ff6592e2b2a5ab946f05a653e04010ffe681c2

[14] https://towardsdatascience.com/

[15] https://victorzhou.com/

[16] https://haralick.org/journals/TexturalFeatures.pdf

[17] https://en.wikipedia.org/wiki/Image_moment

Appendix A

```
In [3]: # Converting each image to RGB from BGR format

        def rgb_bgr(image):
            rgb_img = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
            return rgb_img
```

Code to convert from BGR to RGB color space.

The above code gets the image in HSV color space from BGR color space.

```
# image segmentation using Green Masking Technique
def img_segmentationMasking(img):
    rgb_img=cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
    lab_img=cv2.cvtColor(rgb_img,cv2.COLOR_RGB2LAB)
    g_m=cv2.inRange(lab_img,np.array([0,130,0]),np.array([250,255,255]))
    g_m=~g_m
    d_m=cv2.inRange(lab_img,np.array([0,130,135]),np.array([255,255,235]))
    d_m=d_m
    f_m=d_m+g_m
    seg_img=cv2.bitwise_and(rgb_img,rgb_img,mask=f_m)
######end working code
    return seg_img
```

Implementation of Green Masking Technique

This is a code for implementing masking technique where first RGB image is converted to LAB space, then using range function it separate the pixels falling in certain range which are categorized as useful and non-useful part which is converted to black region.

```
# Image Segmentation using SLIC Algorithm
def img_segmetationSLIC(img):
    from skimage import segmentation
    import numpy as np
    import cv2
    from skimage import morphology
    from matplotlib import pyplot as plt
    from skimage.segmentation import mark_boundaries
    print(mask.shape)
    segments = segmentation.slic(image, compactness=1, n_segments=3)
    seg_img=np.zeros((256,256,3))
    region2=segments==2
    region0=segments==0
    region1=segments==1
    region3=segments==3
    print(region1)


    for i in range(0,256):
        for j in range (0,256):
            if region0[i][j]==True:
                image[i][j][0]=0
                image[i][j][1]=0
                image[i][j][2]=0
    for i in range(0,256):
        for j in range (0,256):
            if region1[i][j]==True:
                image[i][j][0]=0
                image[i][j][1]=0
                image[i][j][2]=0
    seg_img=image
    seg_img=cv2.cvtColor(seg_img, cv2.COLOR_LAB2RGB)
    for i in range(0,256):
         for j in range (0,256):
                if seg_img[i][j][0]==0 and seg_img[i][j][1]==64 and seg_img[i][j][2]==194:
                    seg_img[i][j][0]=0
                    seg_img[i][j][1]=0
                    seg_img[i][j][2]=0
    return seg_img
```

Implementation of SLIC Technique

The above code helps to implement slic segmentation technique for which skimage library is used which results in image clustering i.e image is divided in clusters.

```
def fd_hu_moments(image):
    image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    feature = cv2.HuMoments(cv2.moments(image)).flatten()
    return feature
```

Extraction of Hu Moment

The above code is a function definition which helps in calculating hu moments i.e shape features using CV2 library function Hu-Moments when this function is called passing image as argument.

```
def fd_haralick(image):
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    haralick = mahotas.features.haralick(gray).mean(axis=0)
    return haralick
```

Extraction of Haralick Feature

The above code is a function definition which helps in calculating haralic i.e textural features using mahotas library function features. haralick when this function is called passing image as argument.

```
def fd_histogram(image, mask=None):
        image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
        hist  = cv2.calcHist([image], [0, 1, 2], None, [bins, bins, bins], [0, 256, 0, 256, 0, 256])
        cv2.normalize(hist, hist)
        return hist.flatten()
```

Extraction of Color Feature using Histogram

The above code is a function definition code which extracts the color histogram bins of h, s, v component using cv2 **library function calchist.**

```python
def fd_color(image):
    import mahotas
    import cv2
    import numpy as np
    from skimage.feature.texture import greycomatrix
    test_hsv=cv2.cvtColor(image,cv2.COLOR_BGR2HSV)

    h,s,v=test_hsv[:,:,0],test_hsv[:,:,1],test_hsv[:,:,2]

##h,s mean and std_dev
    h_mean,h_std=cv2.meanStdDev(h)
    s_mean,s_std=cv2.meanStdDev(s)
    color_fea=np.array([h_mean,h_std,s_mean,s_std])
#     print(h_mean,h_std,s_mean,s_std)
    haralick_fea = mahotas.features.haralick(v).mean(axis=0)
    color_fea=np.append(color_fea,haralick_fea)
    return color_fea
```

Extraction of Color Feature using CCM Matrix

The above code is a function which helps to extract color features of 'v' component from color co-occurrence matrix of 'v' and also calculating standard deviation and mean of h and s component. When it is called and image is passed as an argument.

```python

ptr=1
# data=pd.dataframe()
for training_name in train_labels:
    # join the training data path and each species training folder
    dir = os.path.join(train_path, training_name)


    current_label = training_name
    if(ptr==1):
        images_per_class=805
    else:
        images_per_class=904

    for x in range(1,images_per_class+1):

        file = dir + "/" + str(x) + ".jpg"


        image = cv2.imread(file)
        image = cv2.resize(image, fixed_size)




        RGB_BGR       = rgb_bgr(image)
        BGR_HSV       = bgr_hsv(RGB_BGR)
#         IMG_SEGMENT   = img_segmentation(RGB_BGR,BGR_HSV)
        IMG_SEGMENT   = img_segmentation(image)



        fv_hu_moments = fd_hu_moments(IMG_SEGMENT)
        fv_haralick   = fd_haralick(IMG_SEGMENT)
        fv_histogram  = fd_histogram(IMG_SEGMENT)
        fv_color_fea=   fd_color(IMG_SEGMENT)

#       fv_gabor      = fd_gabor(IMG_SEGMENT)

        global_feature = np.hstack([fv_histogram, fv_haralick, fv_hu_moments,fv_color_fea])



        # update the list of labels and feature vectors
        labels.append(current_label)
        global_features.append(global_feature)

    print("[STATUS] processed folder: {}".format(current_label))
    ptr+=1
#     print(fv_color_fea)
#     print(len(fv_histogram) )
print("[STATUS] completed Global Feature Extraction...")
```

Extracting features from all images

The above code is extracting features from all the images in the dataset using for loop and collecting those features in array and storing label of images in array therefore creating training dataset for classification.

Implementation:

```
#Gaussian Naiive Bayes |
from sklearn.naive_bayes import GaussianNB
clf_gauss=GaussianNB()
clf_gauss.fit(trainDataGlobal,trainLabelsGlobal)
NB_pr=clf_gauss.predict(testDataGlobal)
acc_NB = clf_gauss.score(testDataGlobal,testLabelsGlobal)
print('Accuracy',str(acc_NB*100)+"%")
cm=confusion_matrix(testLabelsGlobal,NB_pr)
print(cm)
Precision_NB=(cm[0][0])*100/(cm[0][0]+cm[1][0])
Sensitivity_NB=(cm[0][0])*100/(cm[0][0]+cm[0][1])
Specificity_NB=(cm[1][1])*100/(cm[1][1]+cm[1][0])
import seaborn as sm
sm.heatmap(cm,annot=True)
plt.xlabel('Predicted')
plt.ylabel('Actual')
print("[STATUS] splitted train and test data...")
print("Train data  : {}".format(trainDataGlobal.shape))
print("Test data   : {}".format(testDataGlobal.shape))
print("Precision=TP/TP+FP:",(cm[0][0])*100/(cm[0][0]+cm[1][0]),'%')
print("Sensitivity=TP/TP+FN:",(cm[0][0])*100/(cm[0][0]+cm[0][1]),'%')
print("Specificity=TN/TN+FP:",(cm[1][1])*100/(cm[1][1]+cm[1][0]),'%')
f_NB=2*(Precision_NB*Sensitivity_NB)/(Precision_NB+Sensitivity_NB)
print("F1-Score:",f_NB,'%')
```

Implementation of Gaussian Naïve Bayes

In the above code, Gaussian Naïve Bayes is implemented using Gaussian NB library imported from sklearn package and testing the classifier on certain parameters such as specificity, sensitivity, precision etc.

```
#Support Vector Machine
import pandas as pd
import numpy
from sklearn.svm import SVC
# clf = SVC(c=1,gamma=10)
clf=SVC(random_state=9)
#### now your job is to fit the classifier
clf.fit(trainDataGlobal, trainLabelsGlobal)
#### using the training features/labels, and to
#### make a set of predictions on the test data
pred=clf.predict(testDataGlobal)
#### store your predictions in a list named pred
from sklearn.metrics import accuracy_score,confusion_matrix
acc_svm = accuracy_score(pred, testLabelsGlobal)
print('Accuracy:',str(acc_svm*100)+"%")
cm=confusion_matrix(testLabelsGlobal,pred)

print(cm)
Precision_svm=(cm[0][0])*100/(cm[0][0]+cm[1][0])
Sensitivity_svm=(cm[0][0])*100/(cm[0][0]+cm[0][1])
Specificity_svm=(cm[1][1])*100/(cm[1][1]+cm[1][0])
import seaborn as sm
sm.heatmap(cm,annot=True)
plt.xlabel('Predicted')
plt.ylabel('Actual')
print("Precision=TP/TP+FP:",(cm[0][0])*100/(cm[0][0]+cm[1][0]),'%')
print("Sensitivity=TP/TP+FN:",(cm[0][0])*100/(cm[0][0]+cm[0][1]),'%')
print("Specificity=TN/TN+FP:",(cm[1][1])*100/(cm[1][1]+cm[1][0]),'%')
f_svm=2*(Precision_svm*Sensitivity_svm)/(Precision_svm+Sensitivity_svm)
print("F1-Score:",f_svm,'%')
```

Implementation of SVM

The above code helps in implementing support vector machine classifier using sklearn library and then calculating evaluating metrics such as accuracy, specificity and sensitivity.

```
#Random Forest
from sklearn.ensemble import  RandomForestClassifier
from sklearn.metrics import accuracy_score
import pandas as pd
import numpy
clf=RandomForestClassifier(n_estimators=1000,criterion='entropy',random_state=0)
clf.fit(trainDataGlobal,trainLabelsGlobal)
pred=clf.predict(testDataGlobal)
acc_rf=accuracy_score(pred,testLabelsGlobal)
print("Accuracy:",str(acc_rf*100)+"%");
cm=confusion_matrix(testLabelsGlobal,pred)

print(cm)
Precision_rf=(cm[0][0])*100/(cm[0][0]+cm[1][0])
Sensitivity_rf=(cm[0][0])*100/(cm[0][0]+cm[0][1])
Specificity_rf=(cm[1][1])*100/(cm[1][1]+cm[1][0])
import seaborn as sm
sm.heatmap(cm,annot=True)
plt.xlabel('Predicted')
plt.ylabel('Actual')
print("Precision=TP/TP+FP:",(cm[0][0])*100/(cm[0][0]+cm[1][0]),'%')
print("Sensitivity=TP/TP+FN:",(cm[0][0])*100/(cm[0][0]+cm[0][1]),'%')
print("Specificity=TN/TN+FP:",(cm[1][1])*100/(cm[1][1]+cm[1][0]),'%')
f_rf=2*(Precision_rf*Sensitivity_rf)/(Precision_rf+Sensitivity_rf)
print("F1-Score:",f_rf,'%')
```

Implementation of Random Forest

The above code helps in implementing Random Forest classifier using sklearn library and then calculating evaluating metrics such as accuracy, specificity and sensitivity.

Curriculum vitae

Name	:	Shrey Gupta
Father's name	:	Naresh Gupta
Date of Birth	:	06/05/1999
Nationality	:	Indian
Sex	:	Male
Company placed	:	Accenture
Permanent Address	:	239/1, Purva Deen Dayal, Near DAV College, Roorkee, Uttarakhand.
Phone Number	:	7060369666
Email Id	:	shreyrocks41084@gmail.com

CGPA: 8.66

Placement Details: **Accenture**

Name	:	Rahul Gupta
Father's name	:	Rakesh Gupta
Date of Birth	:	06-08-1999
Nationality	:	Indian
Sex	:	Male
Company placed	:	Tata Consultancy Services
Permanent Address	:	H 403 Shree nand nagar vejalpur Ahmedabad 380051
Phone Number	:	9773473853
Email Id	:	rg681999@gmail.com

CGPA: 8.37

Placement Details: **Tata Consultancy Services**

Name	:	Ishan Goel
Father's name	:	Sanjay Goel
Date of Birth	:	22/08/1998
Nationality	:	Indian
Sex	:	Male
Company placed	:	Tata Consultancy Services
Permanent Address	:	LIG-92, Phase-1, Shivlok Colony, Haridwar, Uttarakhand.
Phone Number	:	8979762897
Email Id	:	ishangoel1502@gmail.com

CGPA: 9.35

Placement Details: **Tata Consultancy Limited**

Project Title	Plant Disease Detection using Image Processing And Machine Learning
Team Members (Names and Reg Nos.)	Shrey Gupta-17BEI0077, Ishan Goel-17BEE0288, Rahul Gupta-17BEE0358
Faculty Guide	Dr. G.K. RAJINI
Semester / Year	**VIII / IV year**
Project Abstract (not more than 200 words)	Agriculture sector is the backbone of Indian economy as well as vital for human survival, therefore it is necessary that the farmers are provided with advance technologies and farming equipment's, so that the quality of crops does not get compromised. In this project a methodology is proposed which can help in detection of plant disease if any in its early stages itself .Therefore helping the farmer to take the appropriate action in order to avoid damage to all the crops .This methodology is based on the Image processing and Machine learning techniques which are used to segment the leaf images and classifying the images in two classes i.e. diseased or healthy respectively.
List **codes** and **standards** that significantly affect your project.	International Standardization of Image Coding WG 1 - digital compression and coding of still pictures (JPEG).
List at least two significant **realistic design constraints** that are applied to your project.	- With the increasing input data, processing of the data becomes much more complex and time consuming. - The images of plant leaves should be very clear, otherwise segmentation and detection of disease will be difficult.

Briefly explain two **significant trade-offs** considered in your design, including options considered and the solution chosen	- With the increasing input data, processing of the data becomes much more complex and time consuming although accuracy improves. **Solution-** To handle large amount of data the appropriate classifier or algorithm is selected and parameters of the algorithm are tuned optimally. - To detect a diseased plant, farmers needs to capture a good quality image of plant. This is a very time consuming process. **Solution –** A model can be created such that it could directly take the input feed from drone and detect the diseased plant.