Leaf Recognition and Classification System using Shape and Colour Features with Random-Forest Classifiers

A Report Submitted

in Partial Fulfillment of the Requirements

for the Degree of

Bachelor of Technology

in

Computer Science & Engineering

Submitted By:

Name	***Registration no.***
Adarsh Goswami	*20154166*
Kumar Sitesh	*20154165*
Kohinoor Meshram	*20154112*
Kumar Shubham	*20154129*
Akhi Halder	*20154082*

under the guidance of:

Prof. Dharmender Singh Kushwaha

to the

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

MOTILAL NEHRU NATIONAL INSTITUTE OF TECHNOLOGY

ALLAHABAD, Uttar Pradesh [India]

April, 2018

UNDERTAKING

*I declare that the work presented in this report titled "**Leaf Recognition and Classification System using Shape and Colour Features with Random-Forest Classifiers**", submitted to the Computer Science and Engineering Department, Motilal Nehru National Institute of Technology, Allahabad [Uttar Pradesh - India], for the award of the **Bachelor of Technology** degree in **Computer Science & Engineering**, is my original work. I have not plagiarized or submitted the same work for the award of any other degree. In case this undertaking is found incorrect, I accept that my degree may be unconditionally withdrawn.*

April, 2018
Allahabad[UP - India]

Adarsh Goswami 20154166

Kumar Sitesh 20154165

Kohinoor Meshram 20154112

Kumar Shubham 20154129

Akhi Halder 20154082

CERTIFICATE

*Certified that the work contained in the report titled "**Leaf Recognition and Classification System using Shape and Colour Features with Random-Forest Classifiers**", by Adarsh Goswami, Kumar Sitesh, Kohinoor Meshram, Kumar Shubham and Akhi Halder has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.*

(Prof. Dharmender Singh Kushwaha)
Computer Science and Engineering Dept.
M.N.N.I.T, Allahabad [UP - India]

April, 2018

Preface

Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or characteristics/features associated with that image. Nowadays, image processing is among rapidly growing technologies. It forms core research area within engineering and computer science disciplines too.

In this project we wanted to do something which could benefit the botanists, laymen etc if implemented as a product.We are recognizing plant species based on leaf features. In support of this report we are providing the results and conclusion of experiment.Any constructive feedback is cordially invited.

Acknowledgement

We would like to express our sincere gratitude to our mentor Prof. Dharmender Singh Kushwaha for providing his invaluable guidance,comments and suggestions throughout the course of the project. His encouraging nature and motivational words always boosted our moral.

The completion of this project required a lot of effort and guidance at every step of development.

We as a team have highly benefited and gained a lot of knowledge about image processing techniques and machine learning algorithms after completion of this project. Finally we would also like to thank our professors, seniors, colleagues for supporting us and enabling us to opt a different approach to our project.

Contents

List of Figures

Chapter 1

Introduction

1.1 Problem Statement

This problem is based on leaf recognition and classification using image processing and machine learning. The dataset used in the project comprises of leaves of 32 different species and different machine learning techniques were used and compared to get the maximum accuracy in classification.

1.2 Motivation

Plants of different species need to be classified by Botanists and Agricultural Researchers in their day-to-day work. But sometimes they come across some rare species of plants or when they can't classify a leaf as medicinal or poisonous. If there is a global database of most of well-known species of plants, it'll ease the work for the professionals working in the related fields. Today's GPUs and even handheld devices can use the trained model for the classification. The mobile application of this project with some improvement can be used in near future for mobile leaf classification.

1.3 Machine Learning Techniques used

1.3.1 kNN Classifier

kNN is a simple classification technique which works when there is little or no prior knowledge about the distribution of the data.
k in the algorithm refers to the number of neighbors the algorithm considers for classification. The classification works in the fashion that a circle is drawn and k neighbors are inscribed in it.Then based on the majority voting the classification is done.(Note : k must always be an odd number to avoid a draw in the votes).Usually euclidean distance is used for kNN,but other distances such as hamming distance, manhattan distance, etc can also be used. Advantages of this algorithm are that it can learn complex models fairly easily and it is robust in nature.On the other hand disadvantages of this algorithm is that we need to determine the value of 'K'.And in case of high dimensional data low computational efficiency and false intuition can be there.

1.3.2 Decision Tree Classifier

A decision tree is a flowchart like structure where each internal node denotes a test on an attribute, each branch represents an outcome of a test and each leaf or terminal node is a class label.The topmost node in a tree is a root node.It uses a tree like model of decisions.So the advantage this approach holds is that it is easy to understand, interpret and visualize. It can handle both numerical as well as categorical data.The main disadvantage this approach holds is that small variation in data can lead to a completely different tree ,this is called Variance. Also over-complex trees can be made that do not generalize the data well and can lead to Overfitting.This splitting can stop when a user defined criteria is met. The splitting can result in fully grown trees until a stopping criteria is met,but fully grown trees are likely to overfit data and therefore Pruning can be done.
Growing a tree involves these following steps:

- *Knowing which features to choose.*
- *What are the conditions for splitting.*
- *Knowing when to stop.*
- *Pruning.*

1.3.3 Random Forest Classifier

Random forest algorithm is one of the most popular and powerful machine learning algorithm that is capable of performing both regression and classification tasks. As the name suggests the algorithm operates by creating a multitude of decision trees at training time and outputting the class that is the mode of the classes(in case of classification) or mean prediction(in case of regression) of the individual trees.In general more the number of trees in the model more robust is the prediction and therefore higher is the accuracy.
Advantage of this method is that this method can handle large dataset and won't overfit the data when there are more number of decision trees.Disadvantage of this method are that this method is good for classification problem but not so good for regression problems and we have a little control on what the algorithm does. Growing of a random forest occurs in the following fashion :

- *Firstly out of the N samples or images.Then a sample of these N samples is taken but with replacement.*
- *If there are M input variables or features , a number m¡M is specified such that at each node m variables are selected at random.The best split out of these m is used to split the node.The value of m is held constant while we grow the forest.*
- *Each tree is allowed to grow to the largest extent possible without any pruning.*
- *Prediction is done by aggregating the result of the n trees.*

Chapter 2

Related Work

Research works in the field of Leaf Detection and Classification have been numerous.We have studied various previous works done in this field.
Some of the works are :

Jyotismita Chaki and Ranjan Parekh work[5] on leaf recognition using Gabor Filter. Plant leaf images of three plant types are analyzed using Gabor Filter by varying the filter parameters. Leaf images are convolved with Gabor filters followed by a separation of the real and imaginary portions of the signal. Absolute difference between the real and imaginary signals form the scalar feature value used for discrimination. Associated parameters like filter size, standard deviation, phase shift and orientation are varied to investigate which combination provides the best recognition accuracies. Classification is done by subtracting the test samples from the mean of the training set. Accuracy obtained is comparable to the other works. He takes into account features like pattern,texture, and shape.The overall accuracy obtained was 100 percent on the three kind of leaves taken in the experiment.Such a classification can work effectively in case of less number of leaves and can run with limited hardware as well.This paper aims to achieve further prospects such as invariance to rotation and combining other shape and descriptive features such as fourier descriptors and Hough transform.

Another work[4] in the field of leaf recognition and classification based on Shape based features and Neural Network Classifiers is done by Jyotismita Chaki and Ranjan Parekh.This paper proposes an automated system for recognizing plant species based on leaf images. Plant leaf images corresponding to three plant types, are analyzed using two different shape modeling techniques, the first based on the Moments-Invariant (M-I) model and the second on the CentroidRadii (C-R) model. For the M-I model the first four normalized central moments have been considered and studied in various combinations viz. individually, in joint 2-D and 3-D feature spaces

for producing optimum results. For the C-R model an edge detector has been used to identify the boundary of the leaf shape and 36 radii at 10 degree angular separation have been used to build the feature vector. To further improve the accuracy, a hybrid set of features involving both the M-I and C-R models has been generated and explored to find whether the combination feature vector can lead to better performance. Neural networks are used as classifiers for discrimination. The data set consists of 180 images divided into three classes with 60 images each. Accuracies ranging from 90%-100% are obtained which are comparable to the best figures reported in extant literature.

Stephen Gang Wu,Forrest Sheng Bao, Eric You Xu, Yu-Xuan Wang, Yi-Fan Chang and Qiao-Liang Xiang work[8] on leaf detection and classification using Probabilistic Neural Networks is also worth noting.

The approach uses 12 features extracted using PCA to be provided as an input to PNN.The Probabilistic Neural Networks(PNN) was trained over 1800 leaves using 32 species and an accuracy of more than 90% was obtained.. Compared with other approaches, this algorithm is an accurate artificial intelligence approach which is fast in execution and easy in implementation.

Chuanlei Zhang, Shanwen Zhang, and Weidong Fang[7] have also worked in this field, detecting and classifing leaves on the basis of Local Discriminative Tangent Space Alignment.It is often hard for machine learning algorithms to correctly classify a leaf when it has an irregular structure.This method based on Local Discriminative Tangent Space Alignment encapsulates the geometric and discriminative information into a local patch.

Boran Sekeroglu and Yücel Inan work in this field using neural networks.

This work has been carried out in the goal of introduction of leaves identification or classification using ANNs. The neural networks have proved their ability to give high efficiency in different applications.The entire work is divided into three groups which resulted in different accuracies all around 90%. The results obtained in this work proved that the use of ANN for classification of plants based on the images of their leave is a promising idea. It is proving the ability to use neural networks for leave recognition tasks and for machine vision use in the classification process.

Milan Sulc and Jirı Matas work[6] in this field involving the use of histograms.Two kind of histograms were used one from the border and one from the interior.. The histogrammed local features are an improved version of a recently proposed rotation and scale invariant descriptor based on local binary patterns (LBPs). Describing the leaf with multi-scale histograms of rotationally invariant features derived from sign- and magnitude-LBP provides a desirable level of invariance. The representation does not use colour.

V. Satti[10] recognition and classification model that uses colour and shape features to train an Artificial Neural Network(ANN) *model with accuracy of 93.3% and a* k Nearest Neighbour *model with accuracy of 85.9%. Flavia leaf dataset was used to extract shape and texture features.*

A recognition model for medicinal plants was developed by Arun[1] and colleagues, which uses Greytone Spatial Dependency Matrices*(GTSDM), grey textures and* Local Binary Pattern*(LBP) operator features as input to the model. Without preprocessing of images, an accuracy of about 94% was achieved using dataset comprising 250 different images divided into 5 different classes.* Locally Modified Linear Discriminant Embedding Algorithm*(MLLDE) was used by Zhang[7] and colleagues on ICL plant leaf dataset. The dataset comprised of 750 different leaf images belonging to one of 50 diferent species. A modification to the same was proposed by Kadir*

Chapter 3

Software Requirements Specification

3.1 Introduction

3.1.1 Purpose

The purpose of this document is to provide a detailed description of leaf recognition and classification system based on the shape and color features. It will give the step by step description of method used to classify plant species on the basis of their shape,colour,intensity features. This project if developed as an application can be used in botany,farming,medicines,etc.Also it will be a great help to the new people working in any field related to plants as even a small mistake in terms of mis-classification can lead to disaster.

3.1.2 Scope

This system can serve as a base model for plant species detection using image of leaf. This system is based on shape, color and intensity features.After training the model an application developed using the model will be light weight and helpful for people in this field.

3.1.3 Glossary

- ***Dataset :*** *Collection of images for leaf recognition and classification system.*
- ***Training data :*** *Part of the dataset used for training the algorithm.*
- ***Testing data :*** *Part of dataset used for testing the algorithm.*
- ***Accuracy :*** *Ratio of actual output to expected output is termed as accuracy.*
- ***Client :*** *Entity providing the leaf image for recognition.*
- ***System :*** *Entity providing the desired output on recognition.*

3.1.4 References

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

3.1.5 Overview of Document

The next section, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next section.
The third section, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.
Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

3.2 Overall Description

3.2.1 System Environment

The complete system is composed of three parts.The client,who gives an input leaf image; the feature generator which pre-processes the image and extract features

from the input image and thirdly the classifier which classifies the image in a class among the 32 classes present.

3.3 Requirement Specification

3.3.1 Functional Requirements

The system uses 32 different species of leaves and recognits the output after processing and analysing the features using machine learning algorithms.
The system if developed as a product would help in detection of plant species and would be useful to botanists and every other person present in the field of farming, medicines, etc.
The user needs to give an image to the system and the system would provide an output as the recognited species.

3.3.2 Non-Functional Requirements

We can build the system on any python version but we have preferred using the latest version Python 3.6 because of its compatibility with previous versions and its easy availability. We have used various python libraries for this system.OpenCV lirary is used for image processing and finding features from the processes images, os library is used for traversing in the directories of the database, scikit-learn(sklearn) library is used for recognition classifiers.
Also, this system will need to process a lot of data during execution. Hence, a fast processor is required for quick analysis.

3.4 Data Models

3.4.1 Flow Chart

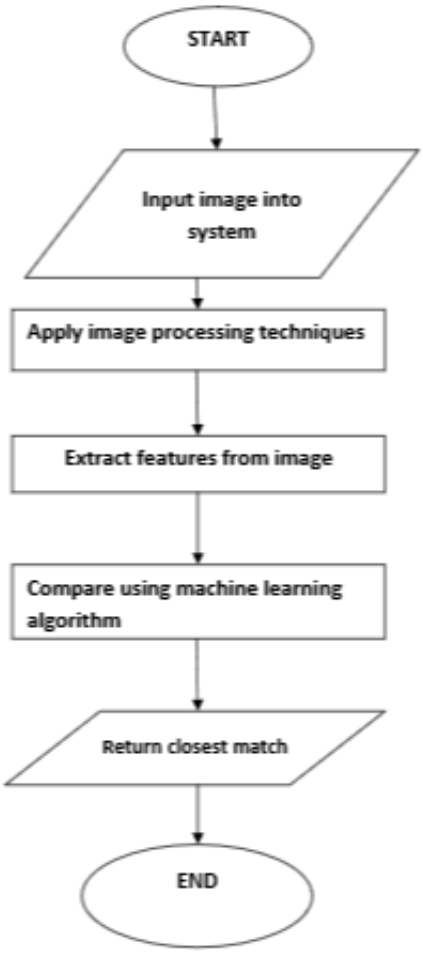


Figure 1: Flow Chart of the Leaf Recognition System

Chapter 4

Proposed Work

4.1 Dataset

*Folio Leaf Dataset[2] from UCI is used.It contains 32 different species, with each containing 20 images.RGB images with dimensions 2322*4128 pix are used.*

Beaumier du perou	*Ficus*	*Pomme Jacquot*	*Carricature plant*
Eggplant	*Duranta gold*	*Star Apple*	*Coffee*
Fruitcitere	*Ashanti blood*	*Barbados Cherry*	*Ketembilla*
Guava	*Bitter Orange*	*Sweet Olive*	*Chinese guava*
Hibiscus	*Coeur Demoiselle*	*Croton*	*Lychee*
Betel	*Jackfruit*	*Thevetia*	*Geranium*
Rose	*Mulberry Leaf*	*Vieux Garcon*	*Sweet potato*
Chrysanthemum	*Pimento*	*Chocolate tree*	*Papaya*

4.2 Preprocessing of dataset

Various image processing techniques are used to make the input images compatible for feature extraction.

4.2.1 Resizing

*Resizing of input image is necessary so as to balance the tradeoff between the speed and detailed features on the train and test data.The images are resized to 512*512 pix images. The background algorithm used by the resize function is Bilinear Interpolation using OpenCV library.*
[3]

Figure 2: (512*512) Resized Leaf Image

4.2.2 GrayScaling

It is done to reduce the number of color channels to work on. So, the colored image(RGB) is converted to black and white(GRAYSCALE). It is also easier to work on grayscale image for edge detection and extraction. It uses the Luminosity formula-¿ Y=0.299R + 0.587G + 0.114B.

Figure 3: Grayscaled Leaf Image

4.2.3 Thresholding

It is used to convert the grayscaled images to binary images. OTSU Thresholding is applied over grayscaled images. As binary images are easier to work upon when processing involves edge and contour detection.

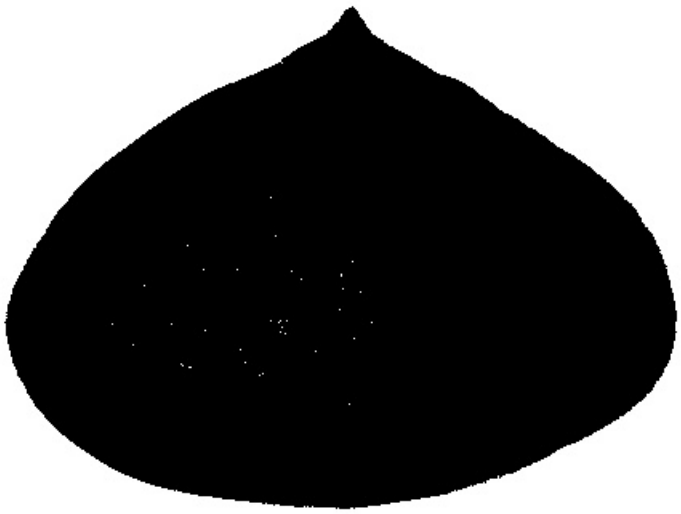


Figure 4: Leaf Image after OTSU Thresholding

4.2.4 Opening and Inverse Thresholding

Erosion followed by Dilation(Opening) is applied to minimize the noise in the background on the output of OTSU thresholding images. The output of opening operation is provided as an input to inverse Threshold function. It is used to create the inverse of the binary image.

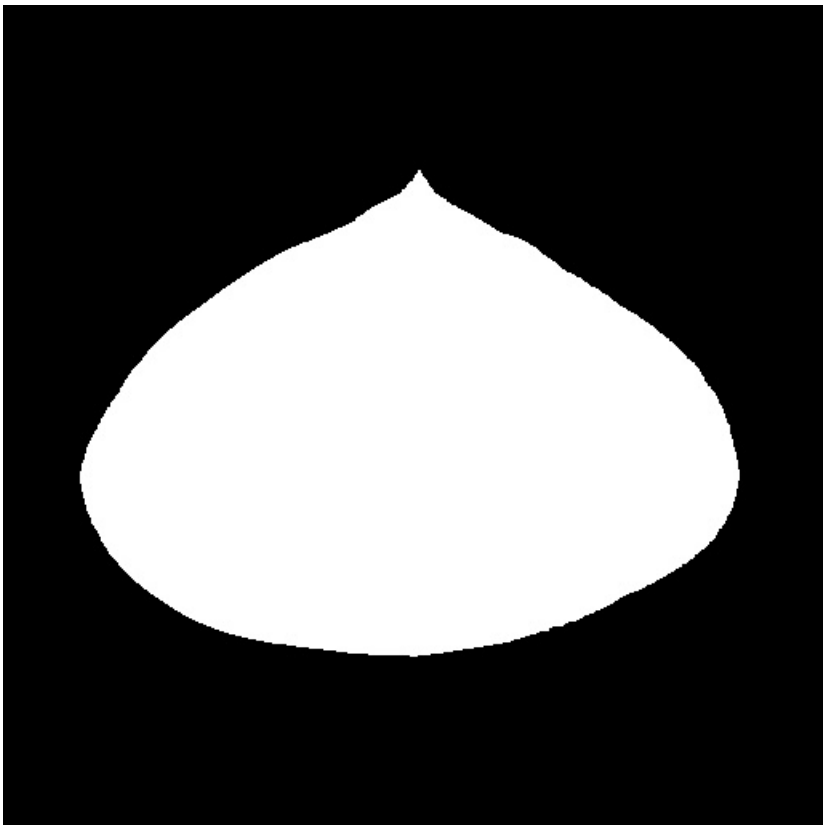


Figure 5: Leaf Image after Opening and Inverse Thresholding operations

4.3 Feature Extraction

Several shape, color, intensity and gradient based features are computed for training the machine learning algorithm. Total of 810 features are extracted on the basis of leaf shape, colour and gradient change.[9]

4.3.1 Extracting Contours

Contours can be explained simply as a curve joining all the continuous points (along the boundary), having the same intensity. Contours find their use in object detection,shape analysis and recognition.Before finding contours image need to be normalized to CV_8UC1 type.

Figure 6: Leaf Image with extracted Contours(in red)

4.3.2 Area and Perimeter

Using contours,Moments is calculated and using which area of the leaf is calculated. Perimeter of leaf is calculated using arclength() of OpenCV library.
[3]

4.3.3 Convex Hull

Convex hull is the smallest area bounding the entire leaf. In mathematics it represents the convex closure of a set X of points in euclidean plane. Using extracted contours Convex hull surrounding the leaf is estimated.
Hull Area and Perimeter are calculated using OpenCV functions.
[3]

Figure 7: Convex Hull around the Leaf

4.3.4 Aspect Ratio

Bounding rectangle is generated using OpenCV library, which is used to calculate the width and height of bounding rectangle. Aspect ratio is calculated as the ratio of width to height.

4.3.5 Other Shape features

White area ratio(WR), Perimter to area ratio(PtoA), Area of hull to Area of leaf ratio(ahtal), Perimeter of hull to perimeter of leaf(PHtP) are also calculated.

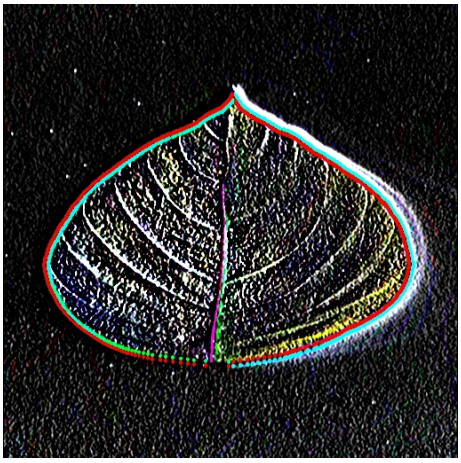
Aspect Ratio	***White Area Ratio***	***Perimeter to Area***	***Perimeter to Hull***	***Hull Area Ratio***
Width/Length	*Area of leaf/(Length * Width)*	*Perimeter of leaf/Area of leaf*	*Perimeter of hull/Perimeter of leaf*	*Area of leaf/Area of hull*

4.3.6 Color Histogram

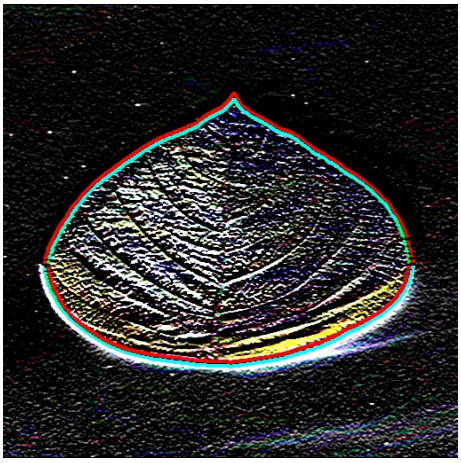
Histograms are used in image processing to give us an intuition about contrast, brightness and intensity distribution in an image. It is simply a plot with pixels values from 0-255 serving as the X-axis in the corresponding number of pixels in Y-axis.
*We are extracting 256*3 features using color histogram in which the image is spliteed into 3-planes of R,G,B respectively. 512*512 RGB images are used as input.*

4.3.7 Histogram of Oriented Gradient(HOG)

This technique counts occurrences of gradient orientation in localized portion of an image. In HOG feature descriptor the distribution of direction of gradients are used as features.
Gradients of an image are useful because the magnitude of gradients is large around the edges and corners and we know that a major portion of information is packed in edges and corners, and the flat regions contain lesser information.



(a) Gradient of the Leaf Image along X-Axis(SobelX Operator)



(b) Gradient of the Leaf Image along Y-Axis(SobelY Operator)

Figure 8: Sobel Operations on Leaf

4.4 Rectifying Features Values

Analyzing the feature data visually, a threshold is decided for the values of a particular feature. If the data was found less than the threshold it was assumed to

be because of some noise in the background and to rectify it mean value of rest of the values of that particular feature is calculated and assigned to the faulty feature values.

4.5 Training on Dataset

The data set was divide into the training set and testing set with a split ratio of 30 percent.

4.5.1 K-Nearest Neighbours(kNN)

The principle behind nearest neighbour methods is to find a predefined number of training samples closest in distance to the new point, and recognit the label from these. We tried to achieve accuracy using kNN on our data set by tuning the values of K(n neighbors) and giving more weightage to the closer points(weight). The accuracy in the range 70-77 % was achieved at k=3.

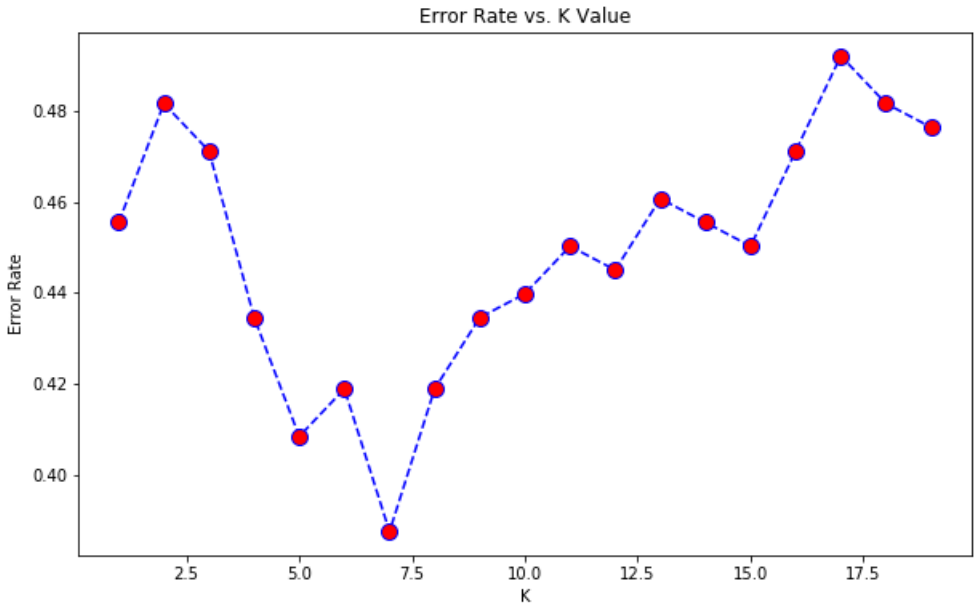


Figure 9: Graph plot for [Error Rate vs k]

4.5.2 Decision Tree

Decision Tree are a non-parametric supervised learning used for classification and regression.The model created using this machine learning algorithm recognits the target value by learning simple decision rules from the training data set. It divide the working area into subareas by identifying various features. It divides the tree into branches until purity condition is reached or information gains becomes zero. Accuracy around 68-72% was achieved.

4.5.3 Random Forest Classifier

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and use averaging to improve the recognitive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement.Initially, the model was trained on the training set and then a graph was plotted by tuning the parameters. The major changes in the accuracy were observed on tuning the maximum depth of the decision tree classifiers. The maximum accuracy obtained using this model is 85-90% taking number of decision tree classifiers(n estimators)=400. A graph was also constructed by varying the number of decision tree classifiers.

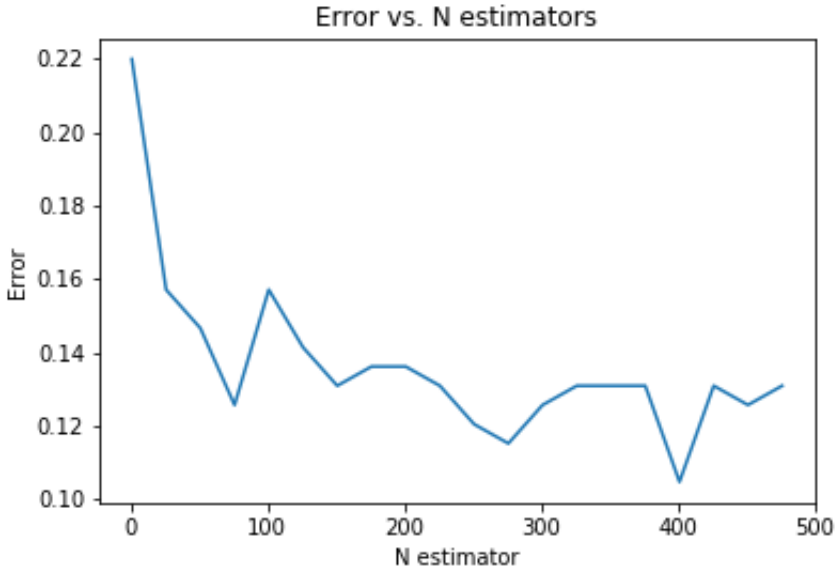


Figure 10: Graph plot for [Error Rate vs n$_e$*stimators*]

4.6 Trained Dataset Overview

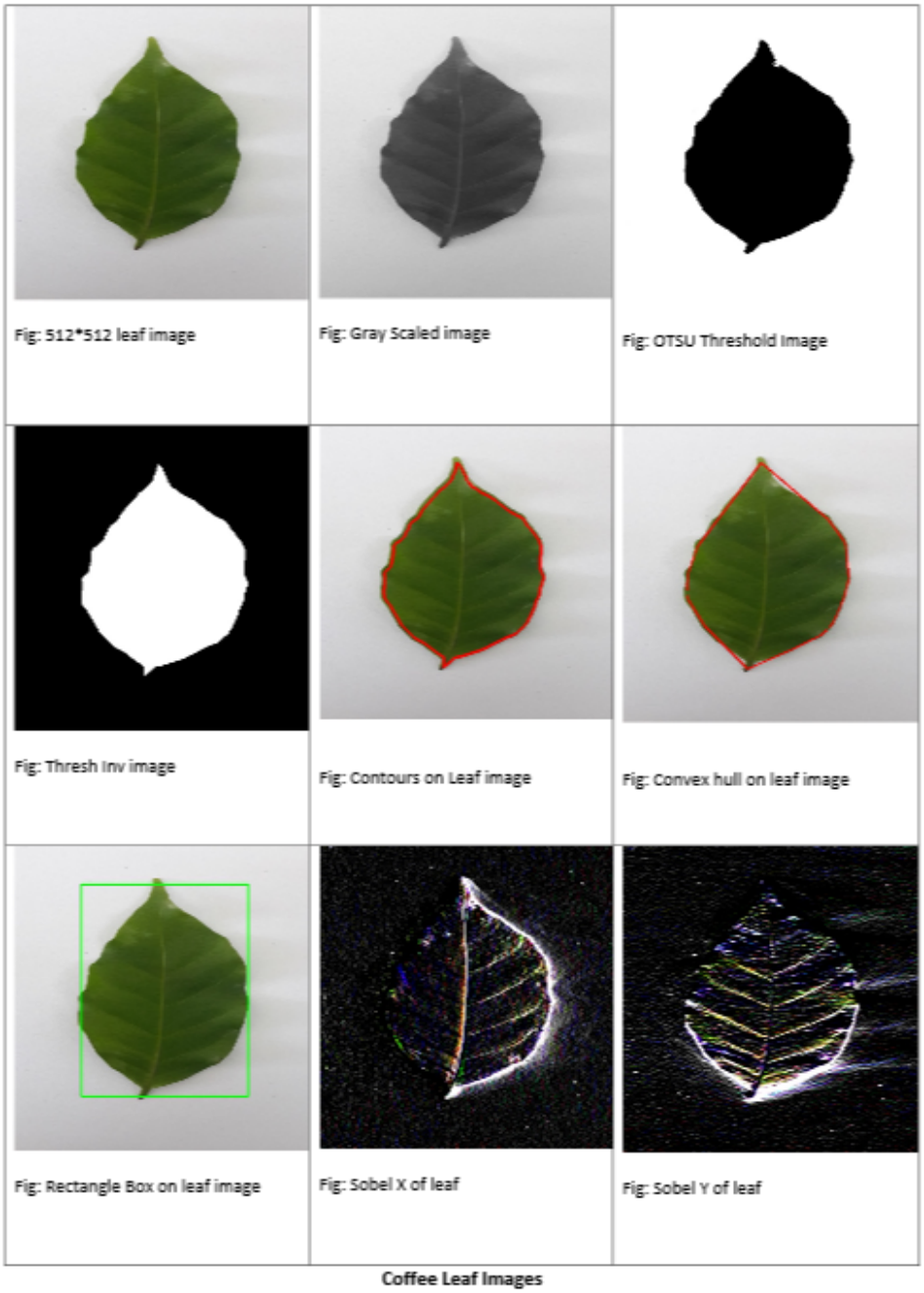


Figure 11: Processed and Pre-Processed Leaves used in Training Dataset : ***Coffee***

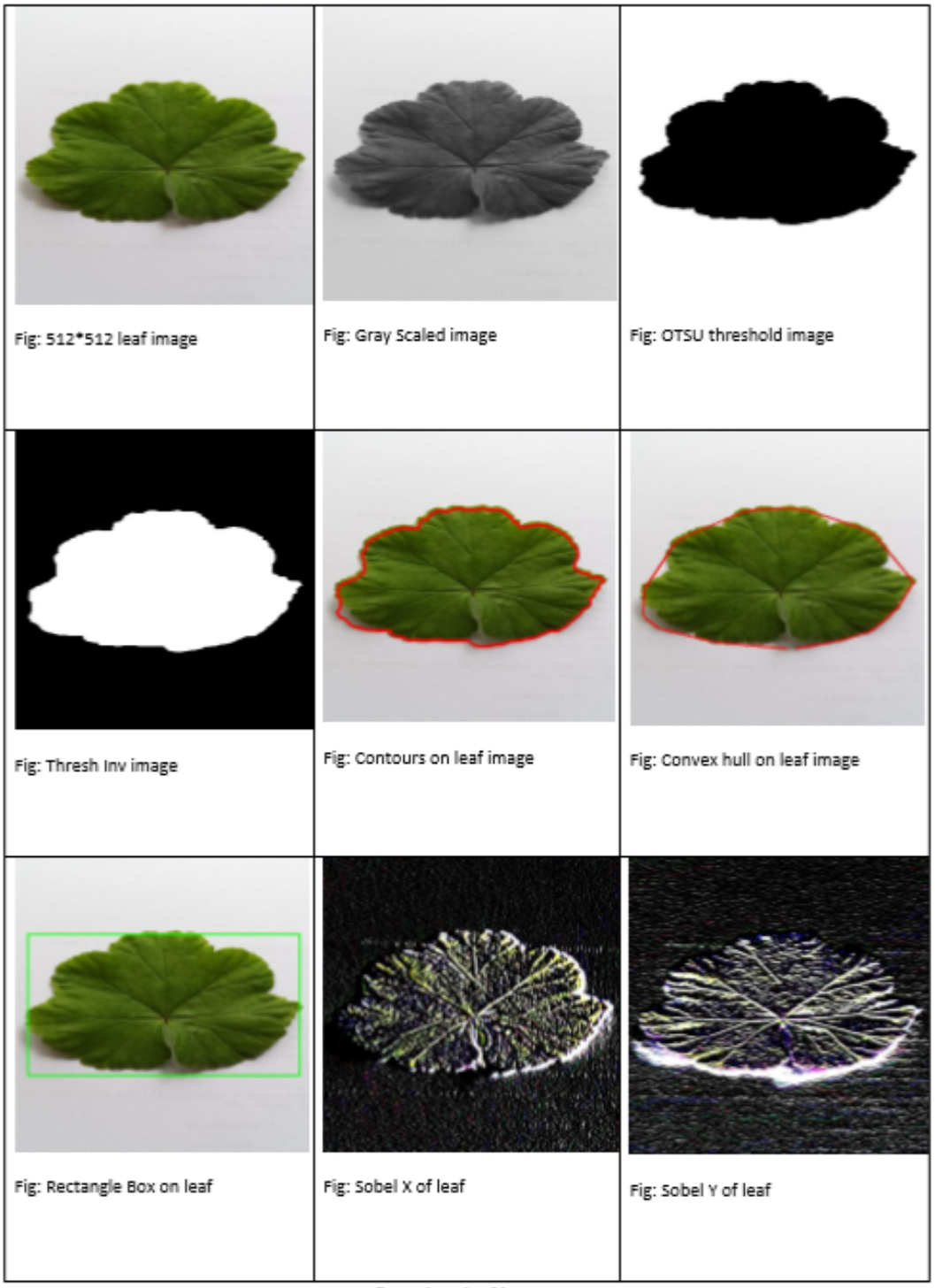


Figure 12: Processed and Pre-Processed Leaves used in Training Dataset : ***Geranium***

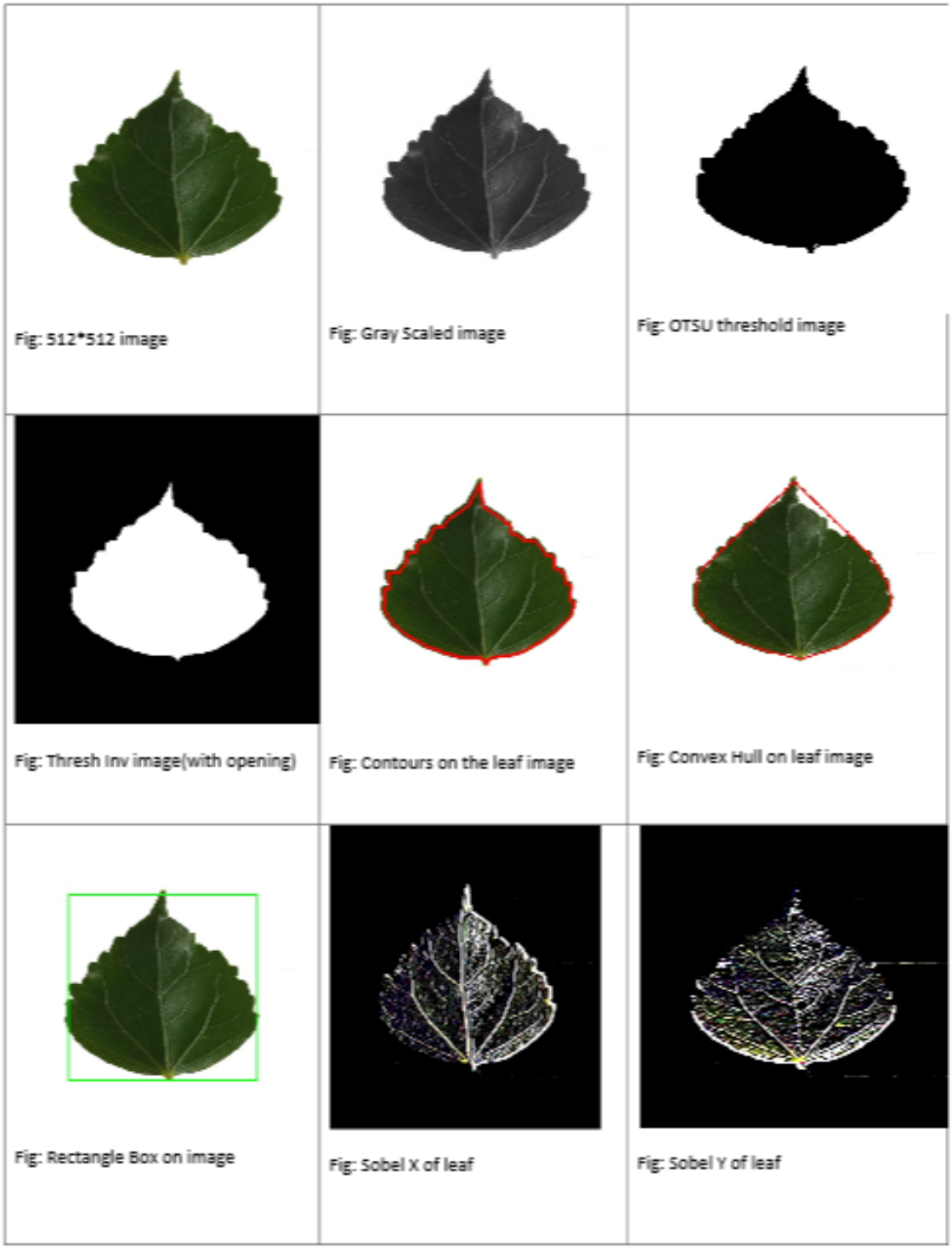


Figure 13: Processed and Pre-Processed Leaves used in Training Dataset : ***Hibiscus***

Chapter 5

Experiment and Results Analysis

Various machine learning algorithm were applied on the given data set and we strived for achieving better accuracy.

5.1 Feature Dataset Description

The final dataset for the model training is a 636 x 810 matrix of features. Total of 636 leaves corresponding to one of the 32 species are used. The dataset columns correspond to the features of the leafs:

- *Columns* 0-767*: Histogram [R,G,B]*
- *Columns* 768*: Aspect Ratio*
- *Columns* 769*: White Area Ratio*
- *Columns* 770*: Perimeter to Area Ratio*
- *Columns* 771*: Perimeter of Hull to Perimeter of Leaf Ratio*
- *Columns* 772*: Area of Hull to Area of Leaf Ratio*
- *Columns* 773-809*: Histogram of Oriented Gradients [HOG]*

5.2 Terms related to Result Analysis

1. ***True Positive (TP)****: These refer to the positive tuples that were correctly labeled by the classifier.*

2. ***True Negative (TN)****: These are the negative tuples that were correctly labeled by the classifier.*

3. ***False Positive (FP)****: These are the negative tuples that were incorrectly labeled as positive.*

4. ***False Negative (FN)****:These are the positive tuples that were mislabeled as negative.*

5. ***Confusion Matrix****: A confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known.*

6. ***Recall (Sensitivity)****:Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes.*

$$Recall = \frac{TP}{TP + FN}$$

7. ***Precision****: Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.*

$$Precision = \frac{TP}{TP + FP}$$

8. ***Accuracy****: Accuracy is a ratio of correctly predicted observation to the total observations.*

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

9. ***F1-score****:F1 Score is the weighted average of Precision and Recall.*

$$Precision = \frac{2 * (Recall * Precision)}{(Recall + Precision)}$$

5.3 Results

5.3.1 k Nearest Neighbour

Accuracy- 68-72%

Class	Leaf	Samples	Correct	Incorrect	Precision	Recall	F1-Score
0	Ashanti blood	6	6	0	0.75	1.00	0.86
1	Barbados Cherry	4	2	2	0.40	0.50	0.44
2	Beaumier du perou	8	4	4	0.80	0.50	0.62
3	Betel	5	3	2	0.60	0.60	0.60
4	Bitter Orange	2	2	0	0.33	1.00	0.50
5	Carricature plant	3	2	1	0.33	0.67	0.44
6	Chinese guava	6	5	1	0.62	0.83	0.71
7	Chocolate tree	5	4	1	0.80	0.80	0.80
8	Chrysanthemum	5	3	2	1.00	0.60	0.75
9	Coeur Demoiselle	3	3	0	0.60	1.00	0.75
10	Coffee	2	1	1	0.10	0.50	0.17
11	Croton	4	3	1	0.75	0.75	0.75
12	Duranta gold	4	3	1	0.43	0.75	0.55
13	Eggplant	4	3	1	0.75	0.75	0.75
14	Ficus	5	5	0	0.62	1.00	0.77
15	Fruitcitere	3	2	1	0.50	0.67	0.57
16	Geranium	7	4	3	1.00	0.57	0.73
17	Guava	2	1	1	1.00	0.50	0.67
18	Hibiscus	5	2	3	1.00	0.40	0.57
19	Jackfruit	6	4	2	1.00	0.67	0.80
20	Ketembilla	6	4	2	0.44	0.67	0.53
21	Lychee	7	5	2	0.83	0.71	0.77
22	Mulberry Leaf	6	1	5	0.33	0.17	0.22
23	Papaya	6	4	2	0.67	0.67	0.67
24	Pimento	9	9	0	0.82	1.00	0.90
25	Pomme Jacquot	4	1	3	0.33	0.25	0.29
26	Rose	4	2	2	1.00	0.50	0.67
27	Star Apple	6	1	5	1.00	0.17	0.29
28	Sweet Olive	7	2	5	1.00	0.29	0.44
29	Sweet potato	4	3	1	1.00	0.75	0.86
30	Thevetia	5	5	0	0.83	1.00	0.91
31	Vieux Garcon	6	1	5	0.33	0.17	0.22
	TOTAL	159	100	59	0.72	0.63	0.63

Figure 14: Classification report for kNN model (with 30% training data and k=6)

5.3.2 Decision tree

Accuracy- 68-72%

Class	Leaf	Samples	Correct	Incorrect	Precision	Recall	F1-Score
0	Ashanti blood	7	7	0	1.00	1.00	1.00
1	Barbados Cherry	4	3	1	0.75	0.75	0.75
2	Beaumier du perou	3	2	1	1.00	0.67	0.80
3	Betel	6	2	4	0.40	0.33	0.36
4	Bitter Orange	7	3	4	0.75	0.43	0.55
5	Carricature plant	2	1	1	0.50	0.50	0.50
6	Chinese guava	2	2	0	0.67	1.00	0.80
7	Chocolate tree	8	8	0	0.89	1.00	0.94
8	Chrysanthemum	4	4	0	0.44	1.00	0.62
9	Coeur Demoiselle	3	1	2	0.50	0.33	0.40
10	Coffee	9	5	4	1.00	0.56	0.71
11	Croton	6	5	1	0.62	0.83	0.71
12	Duranta gold	4	2	2	0.50	0.50	0.50
13	Eggplant	2	1	1	0.20	0.50	0.29
14	Ficus	5	4	1	0.80	0.80	0.80
15	Fruitcitere	6	4	2	0.67	0.67	0.67
16	Geranium	3	3	0	1.00	1.00	1.00
17	Guava	5	4	1	0.67	0.80	0.73
18	Hibiscus	5	4	1	0.57	0.80	0.67
19	Jackfruit	10	6	4	1.00	0.60	0.75
20	Ketembilla	3	3	0	0.60	1.00	0.75
21	Lychee	3	1	2	0.20	0.33	0.25
22	Mulberry Leaf	6	4	2	0.57	0.67	0.62
23	Papaya	9	4	5	1.00	0.44	0.62
24	Pimento	4	3	1	0.50	0.75	0.60
25	Pomme Jacquot	3	3	0	1.00	1.00	1.00
26	Rose	5	4	1	0.80	0.80	0.80
27	Star Apple	3	2	1	0.67	0.67	0.67
28	Sweet Olive	4	2	2	1.00	0.50	0.67
29	Sweet potato	7	4	3	0.80	0.57	0.67
30	Thevetia	7	7	0	1.00	1.00	1.00
31	Vieux Garcon	4	4	0	0.80	1.00	0.89
	TOTAL	159	112	47	0.76	0.70	0.71

Figure 15: Classification report for Decision Tree model (with 30% training data)

5.3.3 Random Forest Classifier

Accuracy- 85-90%

Class	Leaf	Samples	Correct	Incorrect	Precision	Recall	F1-Score
0	Ashanti blood	7	7	0	0.88	1.00	0.93
1	Barbados Cherry	4	3	1	0.75	0.75	0.75
2	Beaumier du perou	3	2	1	1.00	0.67	0.80
3	Betel	6	4	2	1.00	0.67	0.80
4	Bitter Orange	7	3	4	0.75	0.43	0.55
5	Carricature plant	2	2	0	1.00	1.00	1.00
6	Chinese guava	2	2	0	0.50	1.00	0.67
7	Chocolate tree	8	8	0	1.00	1.00	1.00
8	Chrysanthemum	4	4	0	1.00	1.00	1.00
9	Coeur Demoiselle	3	3	0	1.00	1.00	1.00
10	Coffee	9	7	2	1.00	0.78	0.88
11	Croton	6	6	0	1.00	1.00	1.00
12	Duranta gold	4	4	0	1.00	1.00	1.00
13	Eggplant	2	2	0	0.67	1.00	0.80
14	Ficus	5	5	0	1.00	1.00	1.00
15	Fruitcitere	6	5	1	0.83	0.83	0.83
16	Geranium	3	3	0	0.75	1.00	0.86
17	Guava	5	5	0	0.83	1.00	0.91
18	Hibiscus	5	3	2	0.50	0.60	0.55
19	Jackfruit	10	6	4	1.00	0.60	0.75
20	Ketembilla	3	3	0	1.00	1.00	1.00
21	Lychee	3	3	0	0.75	1.00	0.86
22	Mulberry Leaf	6	5	1	1.00	0.83	0.91
23	Papaya	9	9	0	1.00	1.00	1.00
24	Pimento	4	4	0	1.00	1.00	1.00
25	Pomme Jacquot	3	3	0	0.75	1.00	0.86
26	Rose	5	4	1	1.00	0.80	0.89
27	Star Apple	3	2	1	0.40	0.67	0.50
28	Sweet Olive	4	3	1	0.60	0.75	0.67
29	Sweet potato	7	6	1	1.00	0.86	0.92
30	Thevetia	7	7	0	0.78	1.00	0.88
31	Vieux Garcon	4	4	0	0.80	1.00	0.89
	TOTAL	159	137	22	0.89	0.86	0.86

Figure 16: Classification report for Random Forest Classifier model (with 30% training data and n-estimators)

Accuracy improved by almost 15-20% in moving from kNN to Random Forest Classifier.

5.4 Experimental Testing

Testing on the trained Random Forest Classifier(RFC) model was carried out with 16 set of random leaves. All of the features were extracted from the test leaves as a dataframe:

```
In [108]: print(X_test)
```

	c0	c1	c2	c3	c4	c5	c6	c7	c8	c9	...	hog27	hog28	\
16	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	271983	244525	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	127454	116143	
22	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	74386	76233	
28	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	87269	71166	
29	0.0	0.0	1.0	2.0	3.0	2.0	6.0	5.0	5.0	4.0	...	69222	65066	
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	61772	59094	
16	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	205351	163435	
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	90677	84070	
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	89852	76220	
26	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0	0	
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	78352	54634	
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0	0	
18	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0	0	
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	66923	62667	
22	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	49839	43807	
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	114255	75605	

	hog29	hog30	hog31	hog32	hog33	hog34	hog35	hog36
16	197633	147620	127991	100514	75988	73400	51498	22691
3	94178	75008	65563	49324	47057	38241	29101	13619
22	67336	56592	54004	48760	48528	52794	51002	18510
28	70911	63327	56766	62647	63662	51824	46849	16405
29	60263	68165	66665	54562	49882	62914	66419	29787
5	47493	36027	37452	32752	26224	22195	23040	9070
16	161603	136910	106571	81683	54809	49037	41456	14909
1	59086	44611	40509	35774	29189	32159	24512	8032
5	64467	60907	51948	44279	32316	30117	25585	7733
26	0	0	0	0	0	0	0	0
10	58407	62030	49413	58844	53262	58926	58370	24142
6	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0
1	61635	56679	52055	44839	38581	42524	35441	12494
22	41512	41226	40892	44071	38123	39764	35501	18441
10	70177	63049	48036	40651	33480	35278	30547	19478

```
[16 rows x 810 columns]
```

Figure 17: Test Data (Features obtained from 16 leaves)

The test dataframe comprises of 16 rows each corresponding to the feature vector of individual leaf, and 810 columns each corresponding to a feature in the feature vector.
The original dataframe consists of 636 rows and 810 columns corresponding to all the leaves in the Folio Leaf dataset[2].

	c0	c1	c2	c3	c4	c5	c6	c7	c8	c9	...	hog27	hog28	hog29	hog30	hog31	hog32	hog33	hog34	hog35	hog36
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	89804	71722	65279	55502	50232	40066	39386	37418	34413	14259
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	81322	74840	74895	75336	72746	63477	62765	56901	55041	21602
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	74556	64864	73908	64143	61186	58857	52659	40105	37659	15805
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	19913	16036	17360	18164	19399	19422	14562	13273	13846	4593
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	36619	34532	37540	32073	35572	33101	31760	36835	27877	9044
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	50603	48804	47686	44363	49460	45543	44291	36349	33400	15296
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	44061	39867	44096	49373	44671	41745	41071	37829	26463	14503
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	66949	56970	55886	49032	46113	46225	39350	38077	35658	13295
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	63959	55998	50582	49385	42044	45254	39617	36474	35752	14408
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	57461	41386	48837	48667	40133	36625	34922	38585	41534	18098
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	107561	80884	72358	66651	58275	48614	39526	38431	31125	14676
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	67628	52774	51589	44465	51831	46629	43702	38682	31440	12768
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	61218	53012	52976	49973	55603	56085	49035	56028	49617	18045
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	60612	58209	60245	56391	52971	56978	47290	39668	34744	12597
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	68336	59339	46602	45371	38825	38412	38106	33292	27212	12085
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	56771	48367	48481	56270	58055	53354	50769	43413	31154	14746
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	112284	97545	75913	67094	59297	42590	38864	35592	30044	11965
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	71005	54204	59754	45662	46899	35776	25339	22597	31386	22285
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	58024	34181	33947	27186	21448	20219	16659	13927	21692	8620
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	85732	75667	70312	62360	58542	55414	44154	36934	33644	11553
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	104720	106505	85594	78075	74019	63868	51210	51749	42385	19035
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	79335	67941	62397	55098	51423	54964	50076	44807	40544	13854
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	73137	61751	57711	59200	50002	50837	47273	44351	37079	16311
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	51241	43837	45703	47749	44473	41754	40395	34196	26376	12104
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	90677	84070	59086	44611	40509	35774	29189	32159	24512	8032
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	39588	29099	36971	34081	33544	33251	27262	26319	24795	11429
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	61291	60597	49024	43314	50858	41498	39500	27437	35655	18606
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	29316	19755	23715	19725	19162	13060	11508	8948	14284	10707
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	56387	55689	44195	46581	48405	46311	58117	59977	51937	25373
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	76644	55608	68733	65616	55473	50517	46929	40114	48997	18932
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
30	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	130217	110875	94503	74168	66832	65495	67287	59414	50253	19020
30	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	11.0	...	118278	86852	78654	74561	67347	64861	53197	51469	42746	18464
30	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	90833	72086	78199	69264	64640	63371	47530	45811	41918	23866
30	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	117164	102341	101402	96336	84843	76763	67043	57627	55819	22463
30	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	90781	83029	87530	81245	59434	70429	72011	69119	60261	28485
30	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	22.0	...	125477	103358	83814	71900	62763	54988	47372	43426	55470	20007
30	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	113820	108041	87201	80550	63802	55723	50020	41370	36649	15236
30	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	36.0	...	111348	97072	72096	72478	64247	65702	50287	51370	50440	25890
30	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	108383	96757	75663	67896	60651	58716	44591	36031	48964	25256
30	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	4.0	...	82625	73712	69238	69209	57704	56332	54876	66215	57372	26027
31	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	101270	88934	82654	77534	71961	73691	80335	70039	70950	32021
31	0.0	0.0	0.0	1.0	0.0	0.0	2.0	0.0	12.0	9.0	...	146816	116894	103196	85565	95160	69086	54430	46718	36197	24991
31	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	101258	79939	69088	54612	44355	41509	35849	29123	35741	13349
31	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	2.0	...	131549	154094	142838	153787	145816	161451	156588	145164	132070	57144
31	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	106087	102626	92036	88307	76425	62088	67391	58656	44980	21669
31	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	122859	124833	100192	73359	74836	58061	52859	41973	35201	17363
31	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	157039	94217	94952	95520	75849	81333	68139	54078	46320	15026
31	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	102371	97451	86874	70685	55535	51214	41691	38166	50888	18941
31	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	178832	139915	113573	112528	87280	70471	62579	51207	48238	22053
31	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	89520	88826	79784	59887	67120	64514	48565	47551	50182	23698
31	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	79047	74446	78334	79878	91212	91847	81048	79207	70650	28997
31	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	126953	107887	86427	76849	64098	43666	36928	44215	43685	18664
31	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	90056	73399	65499	56555	61026	49737	31549	39838	45050	15886
31	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	116448	83787	61646	46694	48376	33119	32013	35633	29973	13891
31	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	82681	77938	82165	94530	84541	77644	72121	71340	84450	45311
31	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	100154	76877	56103	49331	44199	47614	48065	45359	41944	15540
31	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	3.0	...	92001	73078	70547	60958	55826	48288	44589	36131	27162	10792
31	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	84993	74243	80164	71063	80269	85672	74702	71327	73276	36056
31	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	90282	83326	87090	82821	61980	51126	44476	41782	36545	13634
31	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	71147	54726	40635	43826	38346	35369	33746	34493	48356	26412

636 rows × 810 columns

Figure 18: Original Dataframe

A Random Forest Classifier model was trained on the extracted features using the python libraries of Scikit-Learn (sklearn), a machine learning techniques and tools library. This model was trained on 97.5% of original data (620 leaves) and rest 2.5% comprised of the test data (16 leaves). After training, testing was done using the feature vectors of the 16 images.
A precision of 91% and f1-score of 87% was achieved. Following is the Confusion Matrix *of the experiment results. The confusion atrix contains* False Positives *(FP),* True Positives *(TP),* False Negatives *(FP) and* True Negatives *(TN). The values along the leading diagonal (top left to right bottom) shows the number of correct predictions in a class.*

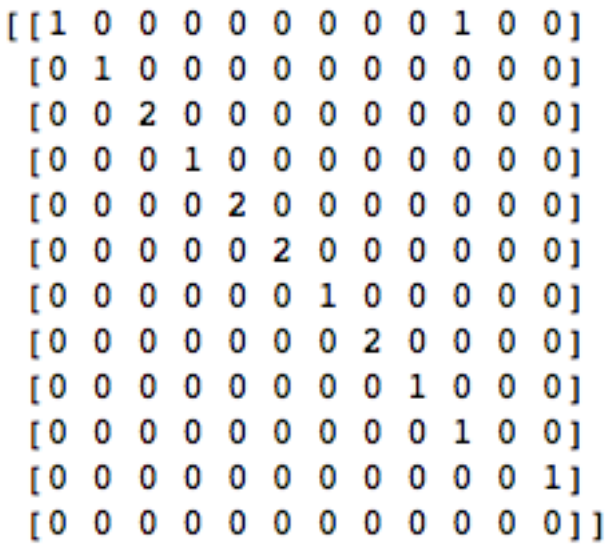


Figure 19: Confusion Matrix of the predictions

The metrics package of sklearn was used to generate a classification report, that compares the test data's original classes to the predicted classes and gives the precision and f1-score for the experimental testing. The precision only takes the TP's and TN's into account while calculating the accuracy of the results. Whereas, f1-score also takes FP's and FN's into account.

	precision	recall	f1-score	support
1	1.00	0.50	0.67	2
3	1.00	1.00	1.00	1
5	1.00	1.00	1.00	2
6	1.00	1.00	1.00	1
10	1.00	1.00	1.00	2
16	1.00	1.00	1.00	2
18	1.00	1.00	1.00	1
22	1.00	1.00	1.00	2
26	1.00	1.00	1.00	1
28	0.50	1.00	0.67	1
29	0.00	0.00	0.00	1
31	0.00	0.00	0.00	0
avg / total	0.91	0.88	0.87	16

Figure 20: Classification report of the predictions

```
y_test = geranium    pred = geranium
y_test = betel    pred = betel
y_test = mulberry leaf    pred = mulberry leaf
y_test = sweet olive    pred = sweet olive
y_test = sweet potato    pred = vieux garcon
y_test = caricature plant    pred = caricature plant
y_test = geranium    pred = geranium
y_test = barbados cherry    pred = barbados cherry
y_test = caricature plant    pred = caricature plant
y_test = rose    pred = rose
y_test = coffee    pred = coffee
y_test = chinese guava    pred = chinese guava
y_test = hibiscus    pred = hibiscus
y_test = barbados cherry    pred = sweet olive
y_test = mulberry leaf    pred = mulberry leaf
y_test = coffee    pred = coffee
```

Figure 21: Actual classes of test leaves vs the Predicted classes

14 out of the 16 test leaves were predicted correctly by our Leaf Recognition and Classification System that is based on Random Forest Classifier, with about 88% accuracy. More randomness of data and size of test dataset also affects the accuracy of the system. In some other cases, the system had an accuracy of 91%.
Following are the test classes and predicted classes of the 16 leaves:

Chapter 6

Conclusion and Future Work

*In this report we demonstrate an approach to recognize 32 different species of plant using their leaf images.The original images 2322*4128 pix were preprocessed and a number of features were extracted from them.After feature extraction we have applied various machine learning techniques for prediction.The best accuracy was found using Random Forest Classifier around 85-88%.*
The blend of learning and knowledge acquired during project work and its implementation is presented in this report.We have strived to make a model which if implemented as an application could help the botanist,farmers and even the layman.Compared with the related works going on our system has a comparable accuracy and is expected to produce result faster.
In the future the system can be made better by applying neural networks and making a user friendly UI so that it would be convenient for common people to use our model.

References

[1] C.H. Arun, W.R. Sam Emmanuel, D. D. *International journal of computer applications. 1–9. Texture feature extraction for identification of medical plants and comparison of different classifiers.*

[2] C.H. Arun, W.R. Sam Emmanuel, D. D. *Folio dataset. UCI Machine Learning Repository.*

[3] doxygen. *Opencv python tutorials.*

[4] Jyotismita Chaki, R. P. *Plant leaf recognition using shape based features and neural network classifiers.*

[5] Jyotismita Chaki, R. P. *Plant leaf recognition using gabor filter.*

[6] Milan Sulc, J. M. *Texture-based leaf identification.*

[7] S. Zhang, Y. L. Elsevier: Neurocomputing, *vol. 74.*

[8] Stephen Gang Wu, Forrest Sheng Bao, E. Y. X. Y.-X. W. Y.-F. C. Q.-L. X. *A leaf recognition algorithm for plant classification using probabilistic neural network.*

[9] Trishen Munisami, Mahess Ransurn, S. K. S. P. *Plant leaf recognition using shape features and colour histogram with k-nearest neighbours classifiers. 740–747.*

[10] V. Satti, A. Satya, S. S. *International journal of engineering science and technology(ijest). 874–879. An automatic leaf recognition system for plant identification using machine vision technology.*

[10] [1] [7] [9] [3] [2]