LEAF RECOGNITION AND MATCHING WITH MATLAB

A Thesis

Presented to the

Faculty of

San Diego State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Computer Science

---

by

Vijaya Bylaiah

Spring 2014

SAN DIEGO STATE UNIVERSITY

The Undersigned Faculty Committee Approves the

Thesis of Vijaya Bylaiah:

Leaf Recognition and Matching with MATLAB

Carl Eckberg

Carl Eckberg, Chair
Department of Computer Science

William Root
Department of Computer Science

MSarkar

Mahasweta Sarkar
Department of Electrical and Computer Engineering

April 17, 2014

Approval Date

DEDICATION

Dedicated to my family and friends.

In the end, the number of prayers we say may contribute to our happiness, but the number of prayers we answer may be of greater importance.

– Dieter F.

ABSTRACT OF THE THESIS

Leaf Recognition and Matching with MATLAB
by
Vijaya Bylaiah
Master of Science in Computer Science
San Diego State University, 2014

Imagine someone hiking in the Swiss mountains, where he finds a weird leaf or flower. This person has always been bad in biology but would like to know more about that plant. What’s its name? Its main features? Is it rare? Is it protected? Etc. By simply taking a picture of the leaf with a Digital Camera, he or she could feed it to the database in his computer and then get all the information regarding the leaf image through an automatic leaf recognition application.

In recent decades, digital image processing, image analysis and machine vision have been sharply developed, and they have become a very important part of artificial intelligence and the interface between human and machine grounded theory and applied technology. These technologies have been applied widely in industries, medicine and agriculture. Finger print recognition is well developed and face recognition is rapidly improving.

As part of this project, the elaboration of such an application has been attempted. The recognition of leaves from photographs implies several steps, starting with image preprocessing, feature extraction, plant identification, matching and testing and finally obtaining the results implemented in MATLAB.

While a botanist could be content with a system for recording a plant species discovered in its natural habitat, to be identified and logged later, this application aims at providing a detailed identification to hikers, campers, etc.

TABLE OF CONTENTS

PAGE

LIST OF FIGURES

PAGE

ACKNOWLEDGEMENTS

It is a pleasure to thank everyone who made this thesis possible. This work would not have been possible without the support from my thesis chair Prof. Dr. Carl Eckberg who gave me the chance to work on this project and gratefully acknowledge the supervision provided by him which has been abundantly helpful and he has assisted me in numerous ways. The discussions I had with him were invaluable. Special thanks to my thesis committee members Prof. William Root and Dr. Mahasweta Sarkar for their support and invaluable advice. My final words go to my family. I want to thank my parents and husband for their love and guidance which helped me in achieving my goals.

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

The human visual system has no problem interpreting the subtle variations in translucency and shading in this Figure 1.1 photograph and correctly segmenting the object from its background.

Figure 1.1. Lotus flower seen as to the naked eye.

Let's imagine a person taking a field trip, and seeing a bush or a plant on the ground, he or she would like to know whether it's a weed or any other plant but have no idea about what kind of plant it could be. With a good digital camera and a recognition program, one could get some useful information. Plants play an important role in our environment. Without plants there will be no existence of the earth's ecology. But in recent days, many types of plants are at the risk of extinction. To protect plants and to catalogue various types of flora diversities, a plant database is an important step towards conservation of earth's biosphere. There are a huge number of plant species worldwide. To handle such volumes of information, development of a quick and efficient classification method has become an area of active research. In addition to the conservation aspect, recognition of plants is also necessary to utilize their medicinal properties and using them as sources of alternative energy sources like bio-fuel. There are several ways to recognize a plant, like flower, root, leaf, fruit etc.

1.2 BACKGROUND

Since recent decades, digital image processing, image analysis and machine vision have been sharply developed, and they have become a very important part of artificial intelligence and the interface between human and machine grounded theory and applied technology. These technologies have been applied widely in industry and medicine, but rarely in realm related to agriculture or natural habitats.

Despite the importance of the subject of identifying plant diseases using digital image processing, and although this has been studied for at least 30 years, the advances achieved seem to be a little timid. Some facts lead to this conclusion:

Methods are too specific. The ideal method would be able to identify any kind of plant. Evidently, this is unfeasible given the current technological level. However, many of the methods that are being proposed not only are able to deal with only one species of plant, but those plants need to be at a certain growth stage in order to the algorithm to be effective. That is acceptable if the plant is in that specific stage, but it is very limiting otherwise. Many of the researchers do not state this kind of information explicitly, but if their training and test sets include only images of a certain growth stage, which is often the case, the validity of the results cannot be extended to other stages.

Operation conditions are too strict. Many images used to develop new methods are collected under very strict conditions of lighting, angle of capture, distance between object and capture device, among others. This is a common practice and is perfectly acceptable in the early stages of research. However, in most real world applications, those conditions are almost impossible to be enforced, especially if the analysis is expected to be carried out in a non-destructive way. Thus, it is a problem that many studies never get to the point of testing and upgrading the method to deal with more realistic conditions, because this limits their scope greatly. Lack of technical knowledge about more sophisticated technical tools. The simplest solution for a problem is usually the preferable one. In the case of image processing, some problems can be solved by using only morphological mathematical operations, which are easy to implement and understand. However, more complex problems often demand more sophisticated approaches. Techniques like neural networks, genetic algorithms and support vector machines can be very powerful if properly applied. Unfortunately, that is often not the case. In many cases, it seems that the use of those techniques is in more demand in the scientific community than in their technical appropriateness with respect to the problem at hand. As a result, problems like over fitting, overtraining, undersized sample sets, sample sets with low representativeness, bias, among others, seem to be a widespread plague. Those problems, although easily identifiable by a knowledgeable individual on the topic, seem to go widely overlooked by the authors, probably due to the lack of knowledge about the tools they are employing. The result is a whole group of technically flawed solutions.

In recent times, computer vision methodologies and pattern recognition techniques have been applied towards automated procedures of plant recognition. Digital image processing is the use of the algorithms and procedures for operations such as image enhancement, image compression, image analysis, mapping, geo-referencing, etc. The influence and impact of digital images on modern society is tremendous and is considered as a critical component in variety of application areas including pattern recognition, computer vision, industrial automation and healthcare industries.

One of the most common methods in leaf feature extraction is based on morphological features of leaf. Some simple geometrical features are aspect ratio, rectangularity, convexity, sphericity, form factor etc.

One can easily transfer the leaf image to a computer and a computer can extract features automatically in image processing techniques. Some systems employ descriptions used by botanists. But it is not easy to extract and transfer those features to a computer automatically.

The aim of the project is to develop a Leaf recognition program based on specific characteristics extracted from photography. Hence this presents an approach where the plant is identified based on its leaf features such as area, histogram equalization and edge detection and classification. The main purpose of this program is to use MATLAB resources.

Indeed, there are several advantages of combining MATLAB with the leaf recognition program. The result proves this method to be a simple and an efficient attempt. Future sections will discuss more on image preprocessing and acquisition which includes the image preprocessing and enhancement, histogram equalization, edge detection. Further on sections introduces texture analysis and high frequency feature extraction of a leaf images to classify leaf images i.e. parametric calculations and then followed by results.

1.3 Related work

Many researchers have made an attempt for plant identification. Some approaches identify the plants based on plant image color histogram, edge features and its texture information. They also classify the plants as trees, shrubs and herbs using complication classifier algorithms. But this proposed thesis work as seen in the Figure 1.2 makes a simple approach by just considering leaf details using simple Support Vector Machine Classifier (SVM) for image classification without many complications. Lots of researchers have proposed many methods for finding out the area of the leaf in an image. Out of these my work uses a simple and a robust area calculation by using another object as reference. Out of many edge detection techniques, this proposed work uses Sobel edge detection algorithm which extracts the boundary pattern successfully.

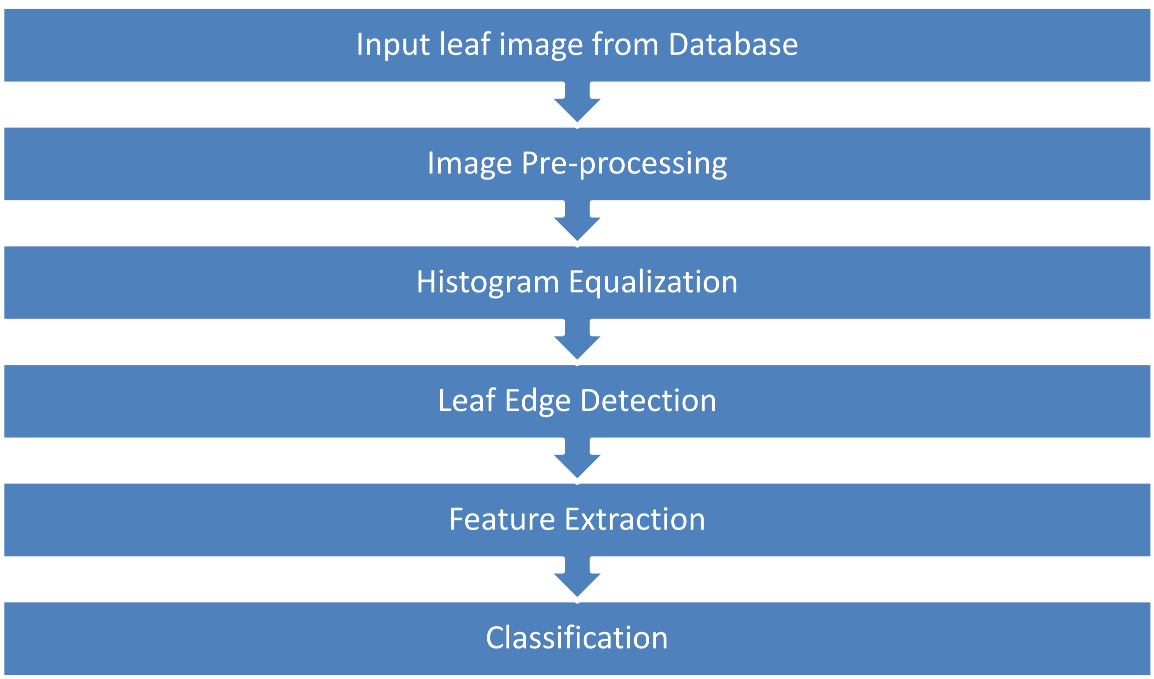

Figure 1.2. Flowchart of our proposed algorithm.

Research on the utilization of moments for object characterization in both invariant and non-invariant tasks has received considerable attention in recent years. A substantial amount of work has been done on leaf shape based plant classification and recognition. Wu et al. [1], extracted 12 commonly used digital morphological features which were orthogonalized into 5 principal variables using PCA. They used 1800 leaves to classify 32 kinds of plants using a probabilistic neural network system. Wang et al. [2], employed centroid contour distance (CCD) curve, eccentricity and angle code histogram (ACH). Fu et al. [3] also used centroid-contour distance curve to represent leaf shapes in which an integrated approach for an ontology-based leaf classification system is proposed. For the leaf contour classification, a scaled Recognition of plants by Leaf Image using Moment Invariant and Texture Analysis CCD code system is proposed to categorize the basic shape and margin type of a leaf by using the similar taxonomy principle adopted by the botanists. Then a trained neural network is employed to recognize the detailed tooth patterns.

The CCD system takes a plant image as input and finds the matching plant from a plant image database and is intended to provide users a simple method to locate information about their plants. With a larger database, the system might be used by biologists, as an easy way to access plant databases. Max-flow min-cut technique is used as the image segmentation method to separate the plant from the background of the image, so as to extract the general structure of the plant. Various color, texture and shape features extracted from the segmented plant region are used in matching images to the database. Color and texture analysis are based on commonly used features, namely color histograms in different color spaces, color co-occurrence matrices and texture maps. As for shape, some new descriptors are introduced to capture the outer contour characteristics of a plant. While color is very useful in many CBIR (content based image retrieval) problems, in this particular problem, it introduces some challenges as well, since many plants just differ in the particular hue of the green color. Results show that for 54% of the queries, the correct plant image is retrieved among the top-15 results, using a database of plants from different plant types. Moreover, the tests are also performed on a clean database in which all the plant images have smooth shape descriptors and are among the images. The test results obtained using this clean database increased the top-15 retrieval probability to 68%. The image enhancement processing can make objects in the source image clear.

Due to the different shapes and sizes of image blocks of leaves, they could be separated and extracted from sources. Then, by using image analysis tools from Matlab, these characters such as sizes, radius, perimeters, solidity, and eccentricity could be calculated. Then, using them as input data, create a radial basis function Divide the input data into two parts. Select one part to train the network and the other to check the validity of the model.

Finally, input data from other image frames under the same condition could be used to check the model. The world of plants is very vast than the worlds of animals or birds or bugs.

Each of these fields traditionally used field guide books, readily purchased, to enable vacationers, scouts or hikers to identify encountered species. Compendious online guides have been enabled by computers and databases. But the goal of this thesis is leaf identification using Digital image processing of a fairly small number of very common occurrences of brush or cacti encountered in the southwest region of the United States taken by a Digital camera. The assumption is that the user will take a picture of a fairly large brushy plant, and feed the snap to the database of this desktop application which uses greater image processing tools provided by MATLAB to identify leaves in less responsive time and accuracy, thus limiting the search to a relatively small set of such plants. The hope is that this will fill a small but useful niche in dynamic identification.

- Chapter 2 discusses the state of the art of this project.
- Chapter 3 gives a brief picture on Biology point of view of this topic and its importance.
- Chapter 4 gives the system overview of this project.
- Chapter 5 explains in detail on the Histogram equalization, which is the image enhancement technique used in this image processing.
- Chapter 6 explains on the SOBEL Edge detection technique and filters applied in this project.
- Chapter 7 explains briefly on the Image processing tools provide by MATLAB that has been implemented in this project.

CHAPTER 2

USAGE

The state-of-the-art of leaf/plant/tree recognition is nowadays used by botanist. However, the program developed in that case is specific for one task and is of no use in more general applications. Actually, these simple techniques focus only on a few features (like color), and are not efficient in a more general purpose.

Consequently more general image classification methods are used, because it is a widespread topic, and there are a lot of well-known features (such as color histogram, SIFT (Scale invariant feature transform), HOG (Histogram of oriented gradient), Shape descriptors, OCR (Optical Character Recognition). In order to focus on the main structure of the program, the MATLAB implementation, the database retrieving and specific feature creation, we will take benefit of the built functions available in MATLAB for Digital Image Processing. The final program as seen in Figure 2.1 provides a segmentation algorithm. In addition, like in most of the image recognition programs, a database of plant or leaf picture has to be done, as well as a learning method to extract the features for the database, and a matching method to retrieve the best match from the database. Several additional small programs have been implemented to gather information for results.

A unique set of features are extracted from the leaves by slicing across the major axis and parallel to the minor axis. Then the feature points are normalized by taking the ratio of the slice lengths and leaf lengths (major axis). These features are used as inputs to the SVM. The SVM Classifier was trained with few simple leaves from a different plant species.

> Input data preparation: Once the feature extraction was complete, two files were obtained. They were: (1) Training texture feature data and (2) Test texture feature data
>
> Classification using Support Vector Machine based on Hyper plane classifier : A software routine was written in MATLAB that would take in .mat files representing the training and test data, train the classifier using the train files and then use the test file to perform the classification task on the test data. Consequently, a MATLAB routine would load all the data files (training and test data files) and make modifications to the data according to the proposed model chosen.

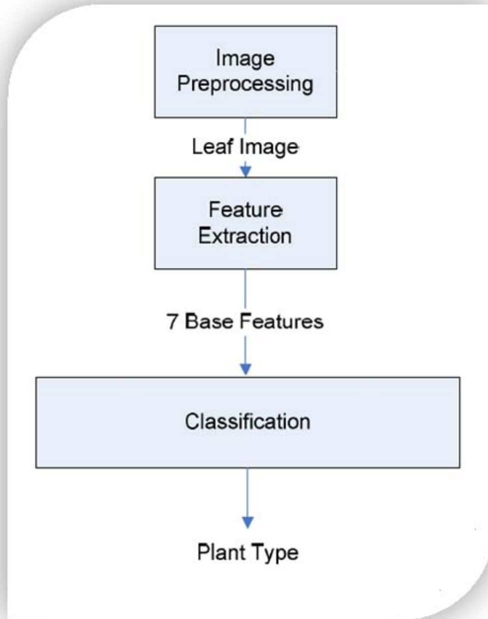

Figure 2.1. Main stages of the system.

CHAPTER 3

IMPORTANCE, PAPERS & BOOKS

3.1 BIOLOGY POINT OF VIEW

In leaves recognition research, a lot has been done about general features extraction or recognition between different classes of objects. In case of specific domain recognition, taking into account the unique characteristics that belong to this category, improves the performance of the system. Despite the high technical aspect of this project, dealing with leaves gives a biological connotation. A very basic knowledge on leaves has to be learned and knowing the perspective of how biologists themselves recognizing a leaf is and add on.

Biologists also emphasize the importance of leaves; indeed their size, their shape, their disposition can vary very much and be a good mean for differentiating similar blooms. The disposition of the leaves on the stem can be alternate, opposed or whorled as illustrated in Figure 3.1. The nervation of the leaf can be of different types; there are leaves with dichotomic, parallel, palmate, pinnate nerves. These features are well explained below.

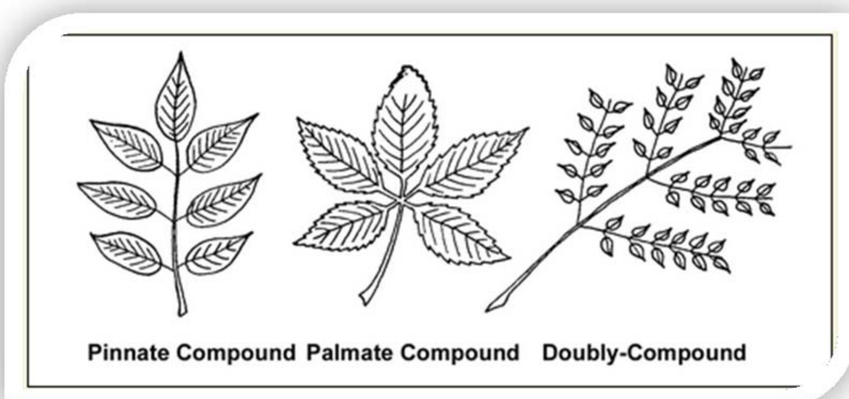

Figure 3.1. Compound leaves.

3.2 IMPORTANCE

In object recognition research a lot have been done about general features extraction or recognition between different classes of objects. In case of a species domain recognition, taking into account the unique characteristics that belong to this category, improves the performance of the system.

Despite the high technical aspect of this project, dealing with leaves, gives it a biological connotation. Some basic knowledge about leaves have to be learned and concepts about how the biologists themselves recognize leaves has to be studied. The next two paragraphs are devoted to these experiences.

Precision Botany (PB) refers to the application of new technologies in plant identification. Computer vision can be used in PB to distinguish plants from its species level, so that an identification can be applied on the size and number of plants detected for the classification purpose. This is focused on the application of computer vision for identification purposes of species in Stemonoporus genus. Surveys reveal that there are 3711 flowering plant species in Sri Lanka [4]. Out of these, 926 are endemic [5, 6]. Since some of these have minute variations, identification of these species has become difficult. Accurate and speedy identification of plants has become a time consuming and a fuzzy work due to non-availability of a computerized scientific plant identification system. Design and implementation of image-based plant classification system is a long felt.

Biologists **receive a large number of requests to identify plants for people,** many species of plants look very similar on their leaves, and botanists will turn to identifying the species based on their structure or other morphologies.

There are three main parts to a leaf:

1. The **base** which is the point at which the leaf is joined to the stem.
2. The **stalk** or **petiole** is the thin section joining the base to the lamina - it is generally cylindrical or semicircular in form.
3. The **lamina** or **leaf blade** is the wide part of the leaf

Leaves can be of many different shapes:

Primarily, leaves are divided into **simple** - a single leaf blade with a bud at the base of the leafstem; or **compound** - a leaf with more than one blade. All blades are attached to a single leafstem. Where the leafstem attaches to the twig there is a bud.

Leaves may be arranged on the stem either in an alternate arrangement – leaves that are not places directly across from each other on the twig; or in an opposite arrangement – 2 or 3 leaves that are directly across from each other on the same twig.

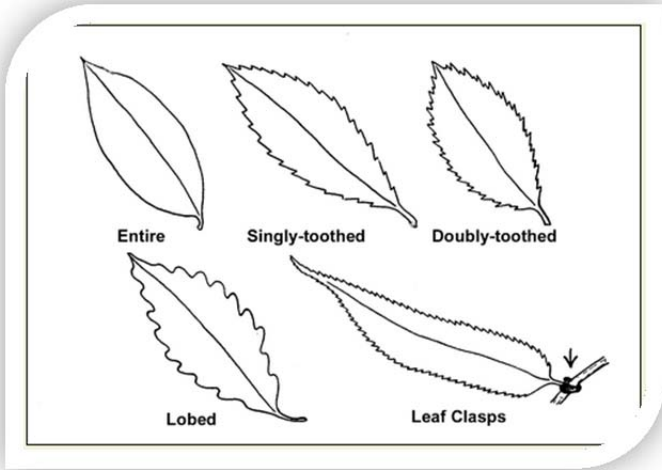

Figure 3.2. Simple leaves – Margin structure.

The margin (the edge of a leaf) as seen in Figure 3.2 may be entire, singly-toothed, doubly-toothed, or lobed.

Compound leaves may be palmate – having the leaflets arranged round a single point like fingers

On the palm of a hand; or pinnate – when the leaves are joined on the two sides of the stalk, like the vanes of a feather as seen in the Figure 3.1.

Leaf arrangements are pretty straightforward to figure out. Need to look for the nodes and then determine how many leaves are coming off each node. If there's only one leaf per node, then need only to determine whether the arrangement is alternate or spiral, and it's usually pretty obvious.

So that's it for basic leaf terms , In conclusion this basic learning has been of great use as it helped understand how professionals approach the sensitive task of recognizing leaves. Many factors come into play, whether the color, the shape, the symmetry of the leaves or some more subtle as simple or compound leaves. The accent has been put on the important role of the leaves; with their alternate disposition or their palmate nerves they are many clues for recognition. After all, every concept introduced here became a significant feature for the project!

CHAPTER 4

SYSTEM OVERVIEW

To identify an item is to recognize the item and associate it with its appropriate name. Such as, the automobile in front of any house is a Honda Accord. Or, a large woody plant in the park is a tree, more specifically a Doug-fir. Identifying a landscape or garden plant requires recognizing the plant by one or more characteristics, such as size, form, leaf shape, flower color, odor, etc., and linking that recognition with a name, either a common or so-called scientific name. Accurate identification of a cultivated plant can be very helpful in knowing how it grows (e.g., size shape, texture, etc.) as well as how to care and protect it from pests and diseases.

First let's look at some **common characteristics of plants that are useful in identifying them**. Now if the same was in a **botany class dealing with plant systematics**, the field of study concerned with identification, naming, classification, and evolution of plants, we would spend a good deal of time on the **reproductive parts of plants**, i.e., mostly the various parts of the flowers, i.e., ovary, stigma, etc. Structural similarity of reproductive parts is an important means by which plants are categorized, grouped, named, and hence identified. However, with many horticultural plants, especially **woody plants, one may have to make an identity without regard to flowers**, for often flowers are not present or are very small, and other characteristics may be more obvious. Some plants characteristics are so obvious or unique that we can recognize them without a detailed examination of the plant.

Pattern recognition is a very important field within computer vision, and the aim of pattern recognition/classification is to classify or recognize the patterns based on extracted features from them. The pattern recognition involves three steps (1) Pre-processing (2) Feature Extraction (3) Classification. In Pre-processing one usually process the image data so it should be in suitable form e.g. one gets an isolated objects after this step. In second step measure the properties of object of interest and in third step, determine the class of object based on features. A brief explanation on the pattern recognition is given in the Figure 4.1.

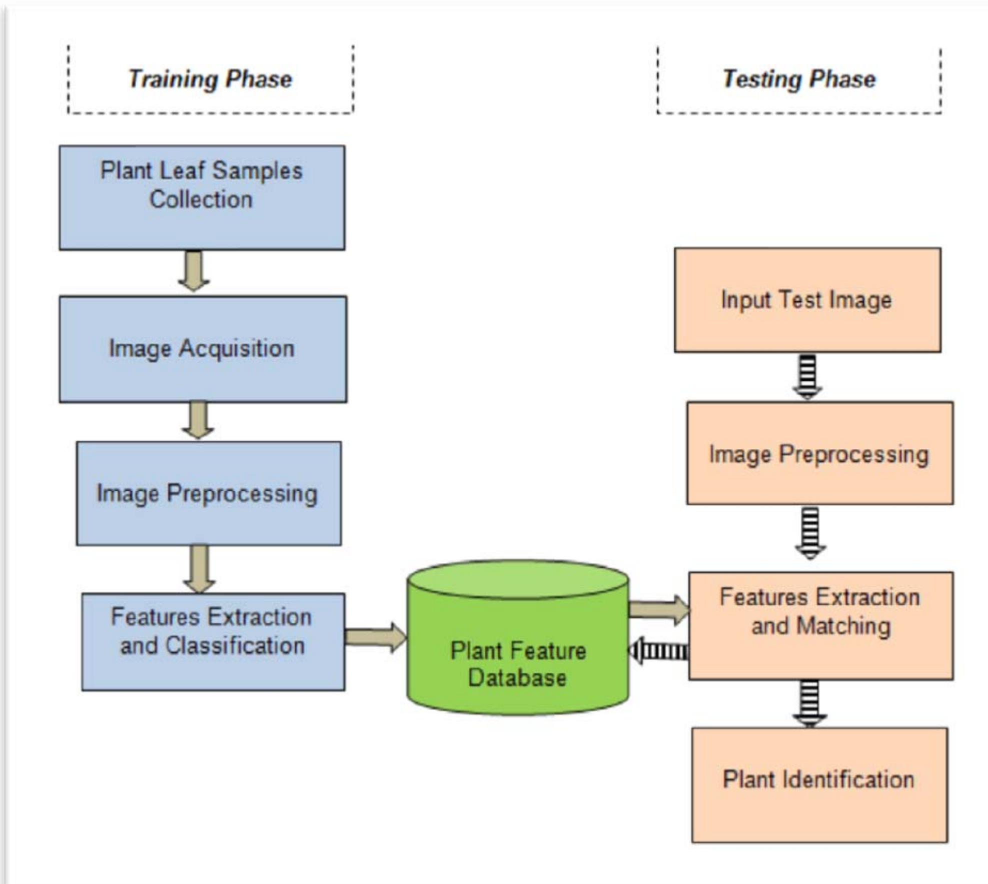

Figure 4.1. Main pattern recognition steps.

4.1 IMAGE PREPROCESSING

Before the operations, some of the leaf images are rotated manually for helping the program to arrange leaf apex direction to the right side. Afterwards, automatic preprocessing techniques are applied to all of the leaf images. These preprocessing steps are illustrated on an image as seen in Figure 4.2, while ignoring the color information. As a result, only Gray component for each pixel is computed from the color image by

$$\text{Gray} = 0.299 * R + 0.578 * G + 0.114 * B \tag{4.1}$$

Where R, G and B correspond to the color of the pixel [7, 8], respectively.

The rectangle of interest (ROI) of the leaf image should include all the pixels their gray values are smaller than a specific threshold [9], and then the binary image of the leaf is retrieved. In this approach the threshold is automatically gotten according to the histogram of the leaf gray image. Then the contour of leaf can be extracted.

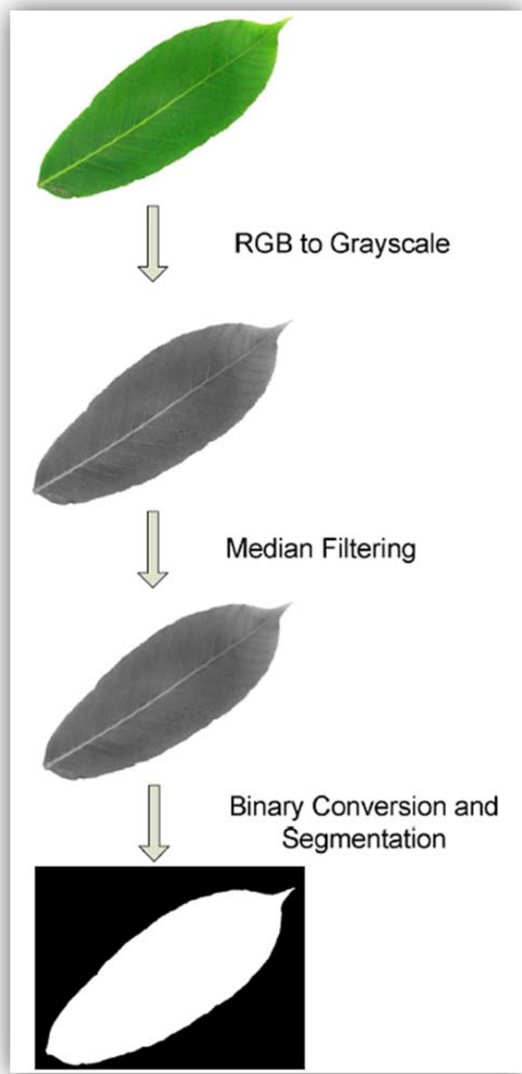

Figure 4.2. Image preprocessing steps.

4.2 Feature Extraction

After pre-processing, in pattern recognition, the important and essential task is to measure the properties of an object because objects have to be detected based on these computed properties. In the feature extraction step, the task is to describe the regions based on chosen representation, e.g. a region may be represented by its boundary and its boundary is described by its properties (features) such as color, texture, etc.

There are two types of representations, an external representation and internal representation. An external representation is chosen when the primary focus is on shape characteristics. An internal representation is selected when the primary focus is on regional properties such as color and texture. Sometimes the data is used directly to obtain the descriptors such as in determining the texture of a region, the aim of description is to quantify a representation of an object. This implies, one can compute results based on their properties such length, width, area and so on.

- *Area:* Area represents number of pixels in the leaf region. Binary form of our leaf image has black background and white leaf. In this image, number of white pixels represents the area of the leaf.
- *Major Axis:* Major axis is denoted as a line, which lies between apex and base of the leaf.
- *Minor Axis:* Minor axis of the ellipse that has the same normalized second central moments as the leaf region.
- *Perimeter:* Perimeter is the distance around the boundary of leaf region.
- *Convex Hull:* Convex hull represents the smallest convex polygon that encapsulates the leaf region.
- *Minor Axis Length Ratio of Major Axis Length:* This feature is denoted as ratio of minor axis length to major axis length. It is reverse of the aspect ratio that is used in the literature.

4.3 CLASSIFICATION (RECOGNITION)

Once the features have been extracted, then these features are to be used to classify and identify an object using SVM classifier to classify plants based on shape-related features of leaf such as aspect ratio, rectangularity, area ratio of convex hull, perimeter ratio of convex hull, sphericity, circularity, eccentricity, form factor and invariant moments.

In general pattern recognition systems, there are two steps in building a classifier: training and testing (or recognition). These steps can be further broken down into sub-steps.

Training:

1. Pre-processing: Process the data so it is in a suitable form.
2. Feature extraction: Reduce the amount of data by extracting relevant information, usually results in a vector of scalar values.
3. Model Estimation: From the finite set of feature vectors, need to estimate a model (usually statistical) for each class of the training data.

Recognition:

1. Pre-processing:
2. Feature extraction: (both steps are same as above)

3. Classification: Compare feature vectors to the various models and find the closest match. One can match the feature vectors obtained in training set.

The algorithm has three main parts: Training, Classification, Segmentation and distance measurement.

CHAPTER 5

HISTOGRAMS OF ORIENTED GRADIENTS FOR LEAF DETECTION

5.1 HISTOGRAM STRETCHING IS USED TO ENHANCE THE CONTRAST

Contrast is the difference between maximum and minimum pixel intensity. An important class of point operations is based upon the manipulation of an image histogram or a region histogram. The most important examples are described below.

Frequently, an image is scanned in such a way that the resulting brightness values do not make full use of the available dynamic range. This can be easily observed in the histogram of the brightness values shown in Figure 5.1. By stretching the histogram over the available dynamic range we attempt to correct this situation. If the image is intended to go from brightness 0 to brightness 2^B-1, then one generally maps the 0% value (or *minimum* as defined) to the value 0 and the 100% value (or *maximum*) to the value 2^B-1. The appropriate transformation is given by:

$$b[m,n] = \left(2^B - 1\right) \bullet \frac{a[m,n] - \text{minimum}}{\text{maximum} - \text{minimum}} \tag{5.1}$$

This formula, however, can be somewhat sensitive to outliers and a less sensitive and more general version is given by:

$$b[m,n] = \begin{cases} 0 & a[m,n] \le p_{\text{low}}\% \\ \left(2^B - 1\right) \bullet \dfrac{a[m,n] - p_{\text{low}}\%}{p_{\text{high}}\% - p_{\text{low}}\%} & p_{\text{low}}\% < a[m,n] < p_{\text{high}}\% \\ \left(2^B - 1\right) & a[m,n] \ge p_{\text{high}}\% \end{cases} \tag{5.2}$$

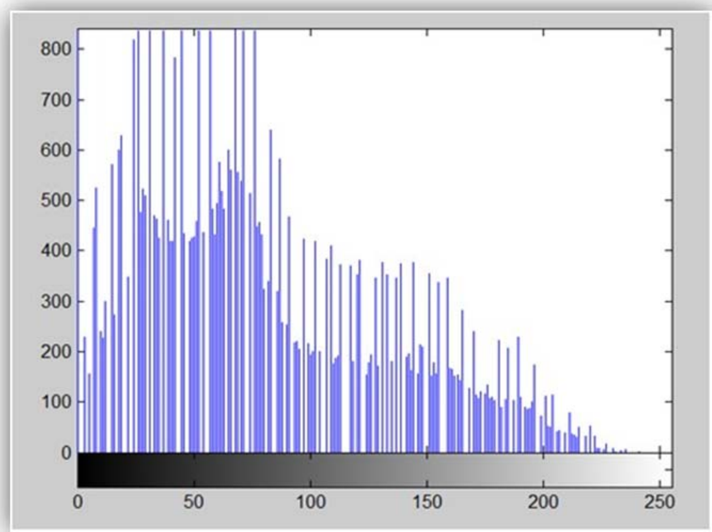

Figure 5.1. The stretched histogram of the image.

In this second version one might choose the 1% and 99% values for $p_{low}\%$ and $p_{high}\%$, respectively, instead of the 0% and 100% values represented by eq. . It is also possible to apply the contrast-stretching operation on a regional basis using the histogram from a region to determine the appropriate limits for the algorithm. Note that in eqs. and it is possible to suppress the term 2^B-1 and simply normalize the brightness range to $0 <= b[m,n] <= 1$. This means representing the final pixel brightnesses as reals instead of integers but modern computer speeds and RAM capacities make this quite feasible.

5.2 Contrast Stretching

Consider Figure 5.2. The histogram of the image (Figure 5.2) is shown in Figure 5.3. Now we calculate contrast from this image.

Contrast = 225.

Now we will increase the contrast of the image. Increasing the contrast of the image: The formula for stretching the histogram of the image to increase the contrast is

$$g(x,y) = \frac{f(x,y) - fmin}{fmax - fmin} * 2^{bpp} \tag{5.3}$$

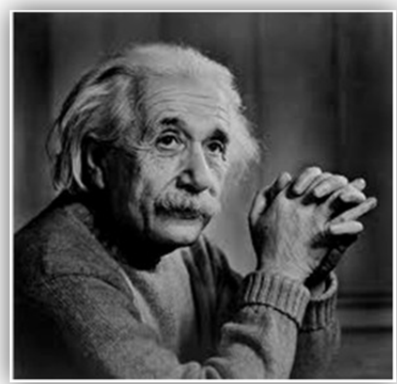

Figure 5.2. Image used for contrast stretching.

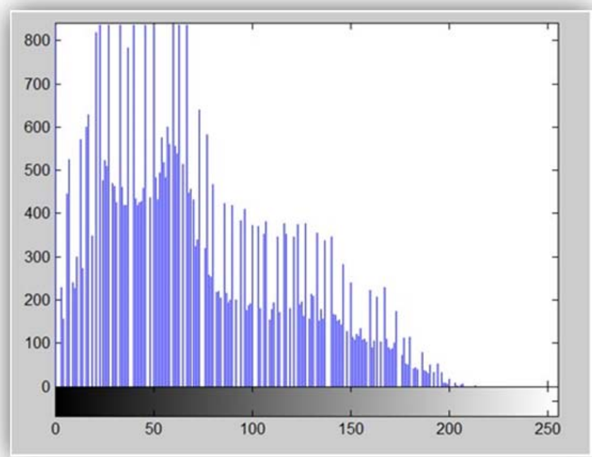

Figure 5.3. Histogram of the image.

The formula requires finding the minimum and maximum pixel intensity multiply by levels of gray. In our case the image is 8bpp, so levels of gray are 256. The minimum value is 0 and the maximum value is 225. So the formula in our case is

$$g(x,y) = \frac{f(x,y) - 0}{225 - 0} * 255 \tag{5.4}$$

Where f(x,y) denotes the value of each pixel intensity. For each f(x,y) in an image, we will calculate this formula. After doing this, we will be able to enhance our contrast. The image in Figure 5.4 will appear after applying histogram stretching.

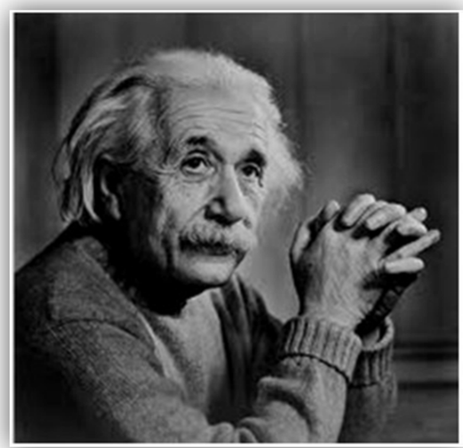

Figure 5.4. Image after applying histogram stretching.

The stretched histogram of this image has been shown in Figure 5.1. Note the shape and symmetry of histogram. The histogram is now stretched or in other means expands. Have a look at it.

In this case the contrast of the image can be calculated as Contrast = 240

Hence we can say that the contrast of the image is increased. Note: this method of increasing contrast does not work always, but it fails on some cases.

5.3 Failing of Histogram Stretching

As we have discussed, in the Figure 5.5 the algorithm fails on some cases. Those cases include images when there is pixel intensity 0 and 255 are present in the image. Because when pixel intensities 0 and 255 are present in an image, then in that case they become the minimum and maximum pixel intensity which ruins the formula like this.

Putting fail case values in Equation 5.3:

$$g(x,y) = \frac{f(x,y) - 0}{255 - 0} * 255 \tag{5.5}$$

Simplify that expression gives

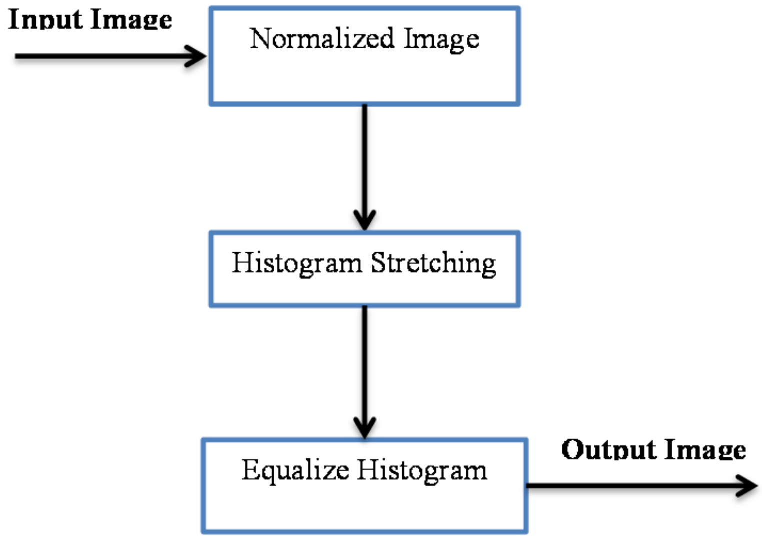

Figure 5.5. Block diagram of the histogram stretching approach.

$$g(x,y) = \frac{f(x,y)}{255} * 255$$

$$g(x,y) = f(x,y) \tag{5.6}$$

That means the output image is equal to the processed image. That means there is no effect of histogram stretching has been done at this image.

CHAPTER 6

SOBEL EDGE DETECTION

Edge detection is more popular for identifying discontinuities in gray level than detecting isolated points and thin lines .The edge is the boundary between two regions with relatively distinct gray level properties. It is assumed here that the transition between two regions can be properties. The transition between two regions can be determined based on the gray level discontinuities .The Sobel operator performs a 2-D spatial gradient measurement on an image and so emphasizes regions of high spatial frequency that correspond to edges. In the input grayscale image, approximate gradient magnitude is also identified at each point by the edge detector. The operator consists of a pair of 3x3 convolution kernels which is rotated by 90 degree [10]. The convolution masks of the Sobel detector are given in Figure 6.1.

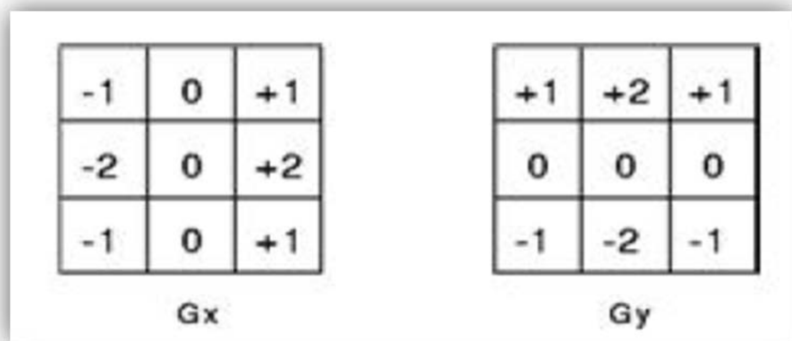

Figure 6.1. Sobel mask.

Input: A Sample Image.

Output: Detected Edges.

Step 1: Accept the input image.

Step 2: Apply mask Gx,Gy to the input image.

Step 3: Apply Sobel edge detection algorithm and the gradient.

Step 4: Masks manipulation of Gx,Gy separately on the input image.

Step 5: Results combined to find the absolute magnitude of the gradient.

Step 6: The absolute magnitude is the output edges.

CHAPTER 7

SUPPORT VECTOR MACHINE (SVM) CLASSIFIER

SVMs (Support Vector Machines) are a useful technique for data classification. Classification task usually involves separating data into training and testing sets. Each instance in the training set contains one \target value" (i.e. the class labels) and several attributes" (i.e. the features or observed variables). The goal of SVM is to produce a model (based on the training data) which predicts the target values of the test data given only the test data attributes.

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (*supervised learning*), the algorithm outputs an optimal hyperplane which categorizes new examples. Let's consider the following simple problem:

We are given a set of n points (vectors) : x1, x2, x3, ….. xn such that xi is a vector of length m, and each belong to one of two classes we label them by "+1" and "-1". So our training set is $(x_1, y_1), (x_2, y_2), \dots (x_n, y_n)$

$$\forall i \quad x_i \in R^m , \ y_i \in \{+1, -1\} \tag{7.1}$$

We want to find a separating hyperplane that separates these points into the two classes. "The positives" (class "+1") and "The negatives" (class "-1"). Let's introduce the notation used to define formally a hyperplane:

$$f(x) = \beta_0 + \beta^T x, \tag{7.2}$$

where β is known as the *weight vector* and β_0 as the *bias*. For a linearly separable set of 2D-points which belong to one of two classes, find a separating straight line.

In Figure 7.1 you can see that there exist multiple lines that offer a solution to the problem.

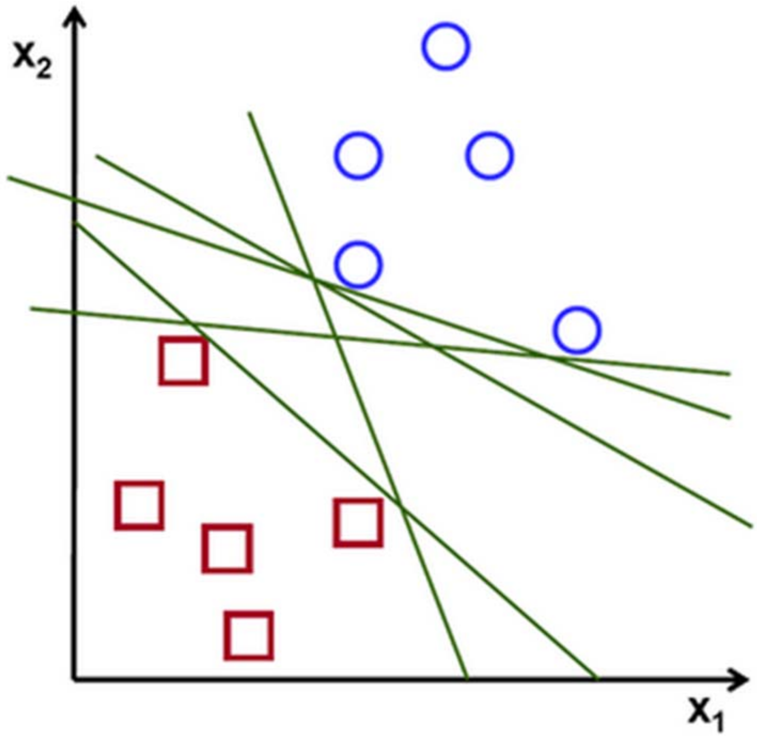

Figure 7.1. Green color hyperplane separating two classes of red squares and blue circles.

A line is bad if it passes too close to the points because it will be noise sensitive and it will not generalize correctly. Therefore, our goal should be to find the line passing as far as possible from all points.

Then, the operation of the SVM algorithm is based on finding the hyperplane that gives the largest minimum distance to the training examples. Twice, this distance receives the important name of **margin** within SVM's theory. Therefore, the optimal separating hyperplane *maximizes* the margin of the training data which is depicted well in the Figure 7.2.

The optimal hyperplane can be represented in an infinite number of different ways by scaling of β and β_0. As a matter of convention, among all the possible representations of the hyperplane, the one chosen is

$$|\beta_0 + \beta^T x| = 1 \tag{7.3}$$

Where x symbolizes the training examples closest to the hyperplane. In general, the training examples that are closest to the hyperplane are called **support vectors**. This representation is known as the **canonical hyperplane**.

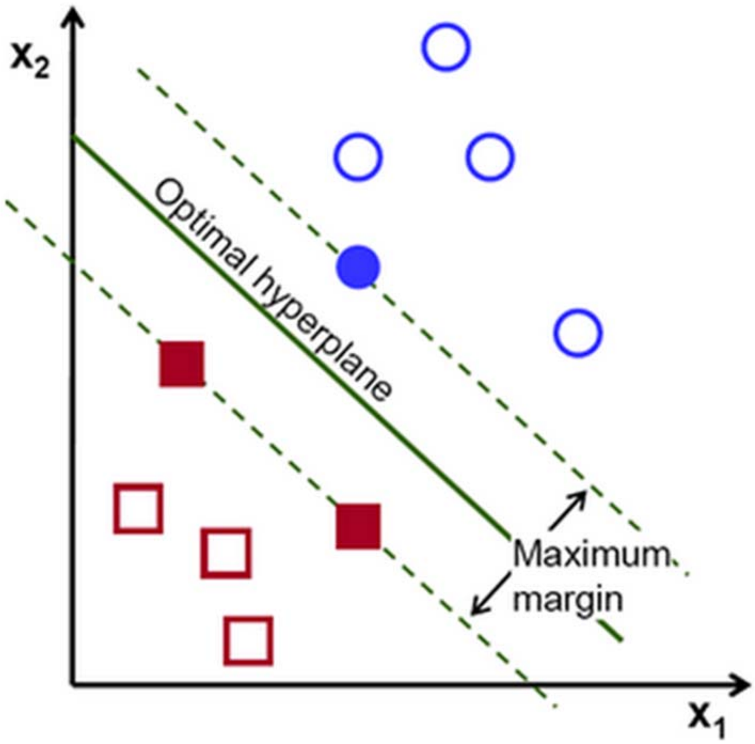

Figure 7.2. Finding an optimal hyperplane.

The data usually contain noises, which result in overlapping samples in pattern space [11], and there may produce some outliers in the training data set. So we need to remove these outliers from the training data set so that a better decision boundary can be easily formed. We here apply smoothening method to remove those points that do not agree with the majority of their k nearest neighbors [11-16].

In particular, by comparing with the 1-NN and k-NN classifiers [11-16], it can be found that the SVM classifier can not only save the storage space but also reduce the classification time under the case of no sacrificing the classification accuracy.

CHAPTER 8

MATCHING DATABASE

8.1 Available Databases

Caltech University has been working on leaf recognition and has therefore created a dataset consisting of 102 leaf categories and more than 8000 images [17]. The leaves mainly come from USA. Downloads or more information can be found. This database is very attractive since at least 40 images of the same category are present, which is essential for a good recognition at a large scale.

8.2 Resulting Database

Finally by mixing a bit the three sources exposed in the previous paragraph, the final database contains 10 images. The repartition is roughly the following 60% from the Caltech dataset, 30% from the other website [18]. The leaves belong to different category.

8.3 Results

In order to test the efficiency one can collect additional pictures of flowers present in the database and see if the system recognizes them. But to have significant results another set of suitable test images would have to be found. So a ground truth evaluation of the database has been conducted.

It consists of going through all the images in the database and search the second best match (the first one obviously being the same image). If the leaf image indicated is part of the same category as the leaf under test then it's a successful recognition. By doing this for the whole database, the performance of the system can be evaluated by establishing the recognition rate. Figure 8.1 best describes the flow chart of the system achieved.

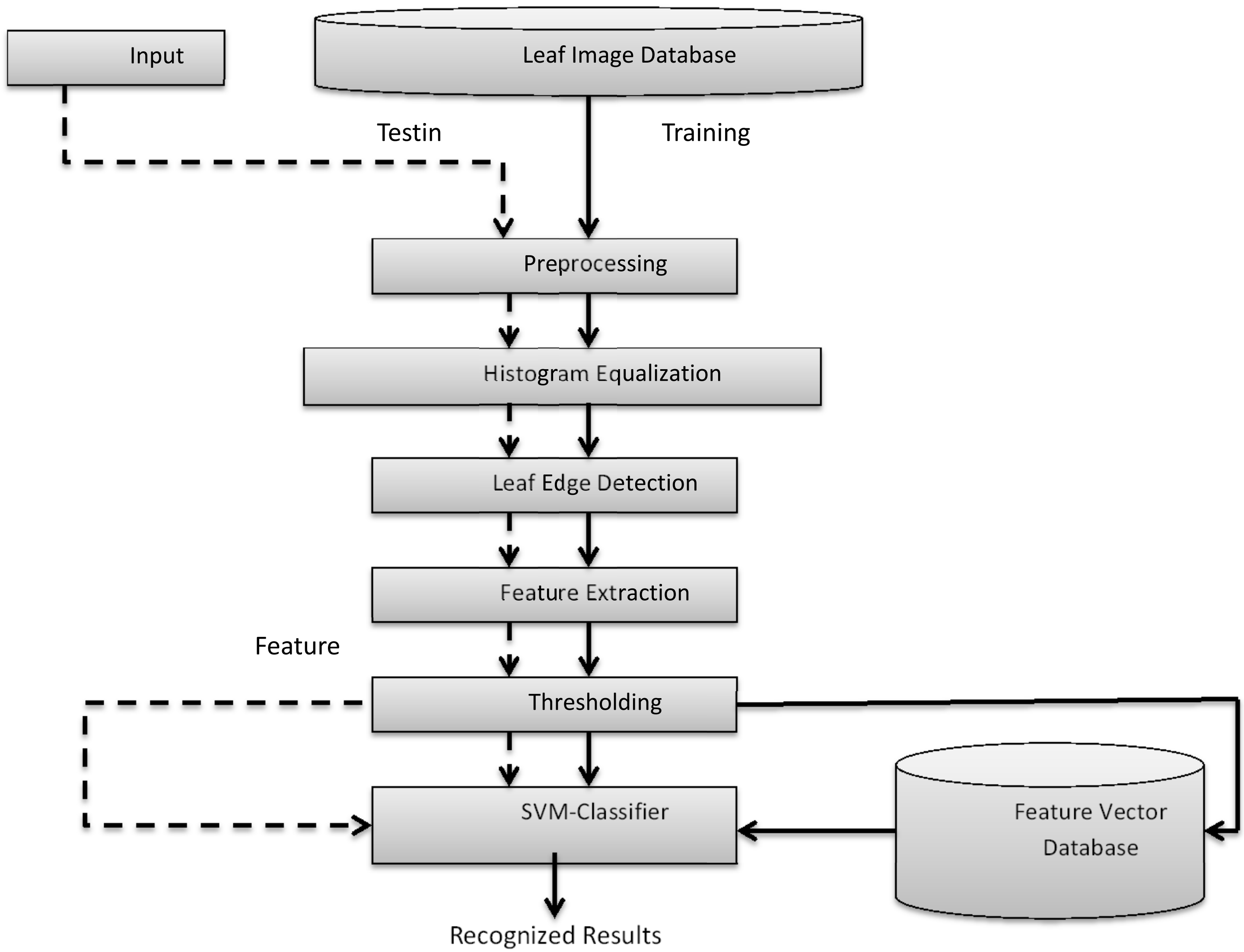

Figure 8.1. Flow chart of the system achieved.

CHAPTER 9

SCREEN SHOTS

The below list of snapshots Figure 9.1 to Figure 9.13 describe the flow of the project.

Figure 9.1. Snapshot of the menu window.

9.1 MENU

The Menu window displays all the image processing steps that has to be carried out on a leaf image.

9.2 GETTING THE DATABASE LEAF

Here we obtain all the leaf images converted to grayscale from RGB which is stored in the database to display.

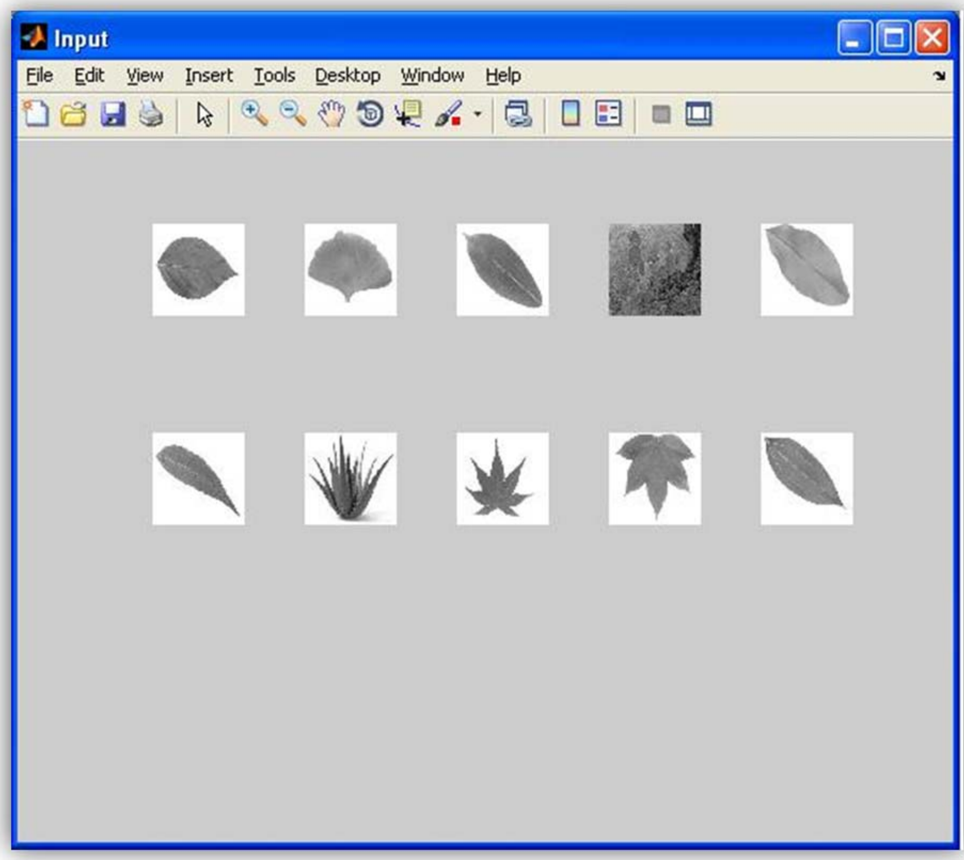

Figure 9.2. Snapshot of the getting images from the database.

Figure 9.3. Snapshot of the leaf detection.

9.3 LEAF PROCESS

Leaf process uses the image and performs smoothing on the leaf image.

9.4 HISTOGRAM EQUALIZATION FOR LEAF

The histogram equalization is performed on the image.

Figure 9.4. Snapshot of the leaf pre-processing.

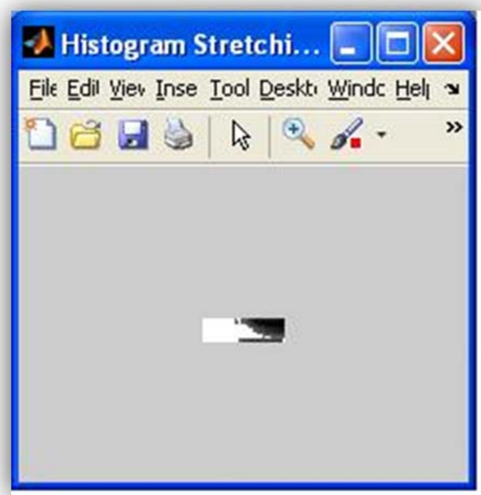

Figure 9.5. Snapshot of the histogram equalization.

9.5 Feature Extraction

The Feature points are extracted from the leaf image.

9.6 Testing

Selection of input leaf image upon which RGB2gray and then smoothing is performed.

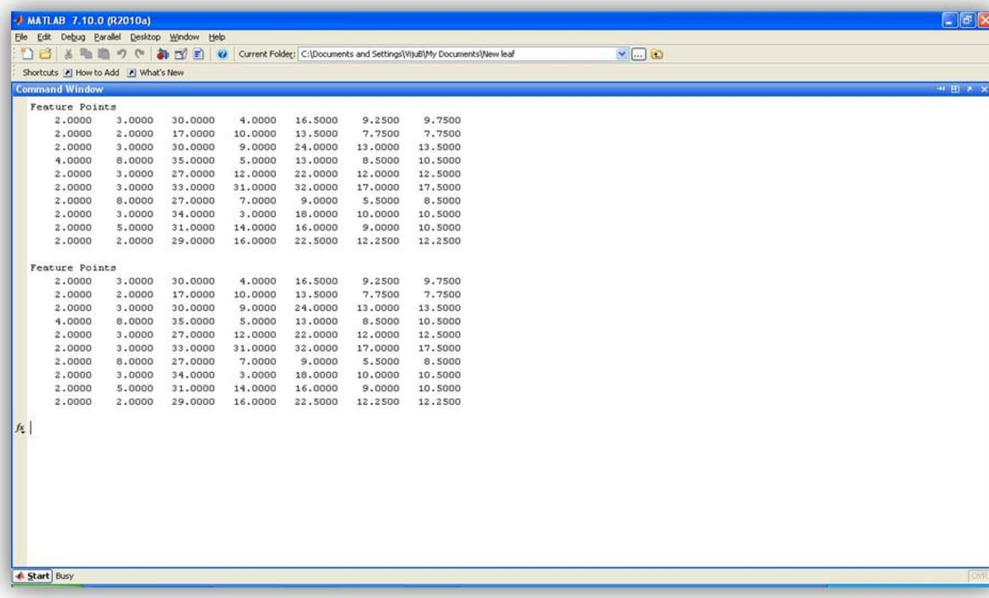

Figure 9.6. Snapshot of feature points retrieved from feature extraction.

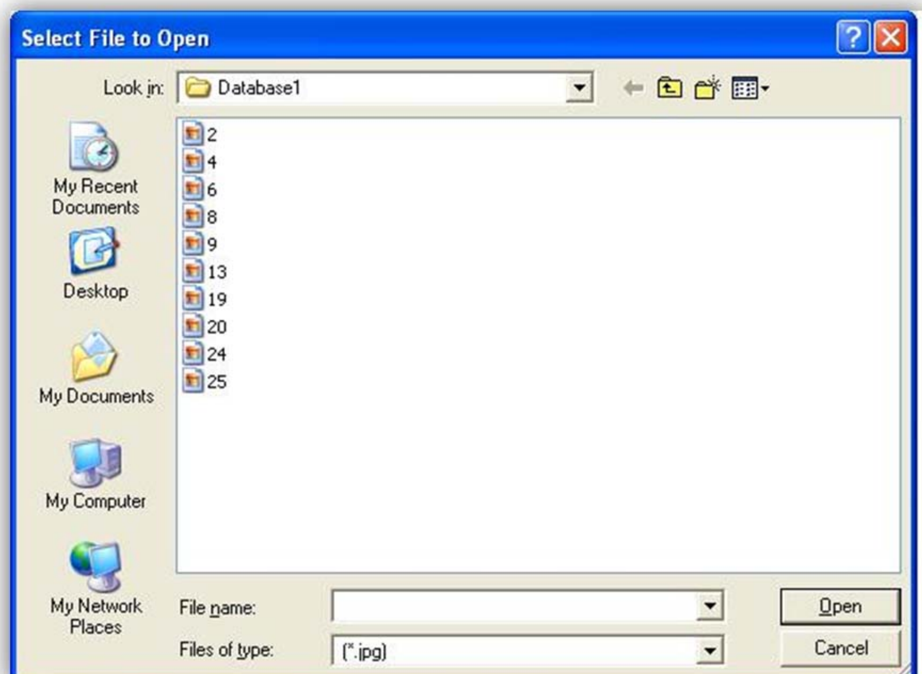

Figure 9.7. Snapshot of the image selection window – To select an image for testing.

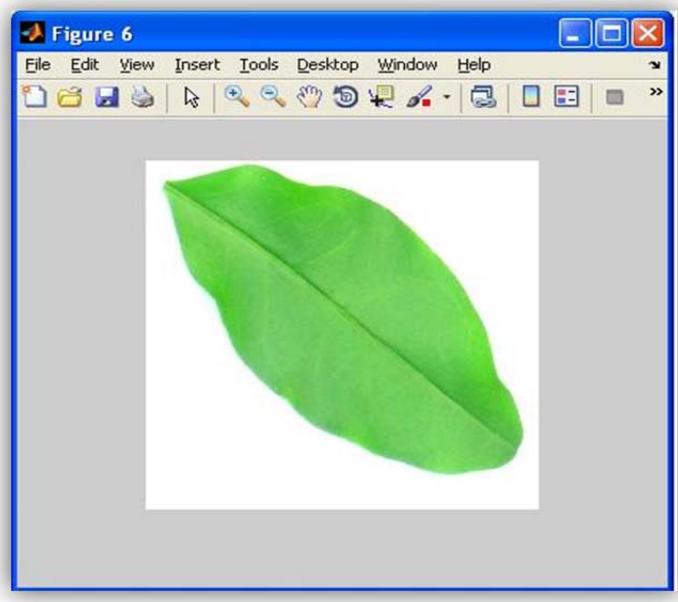

Figure 9.8. Snapshot of the image selected.

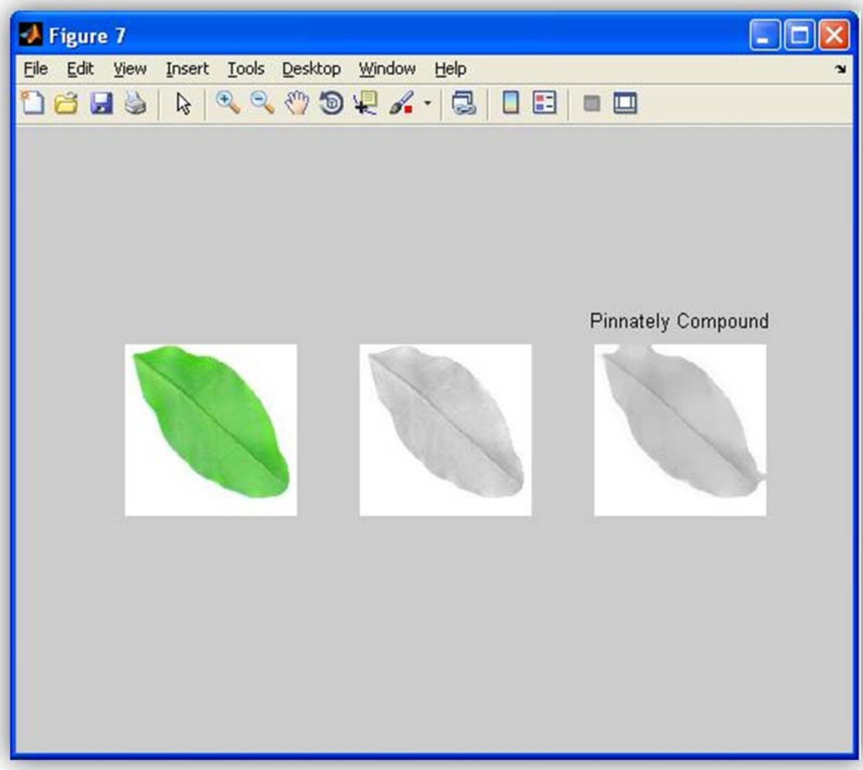

Figure 9.9. Snapshot of the images where the image is converted to RGB2GRAY and then smoothing is performed.

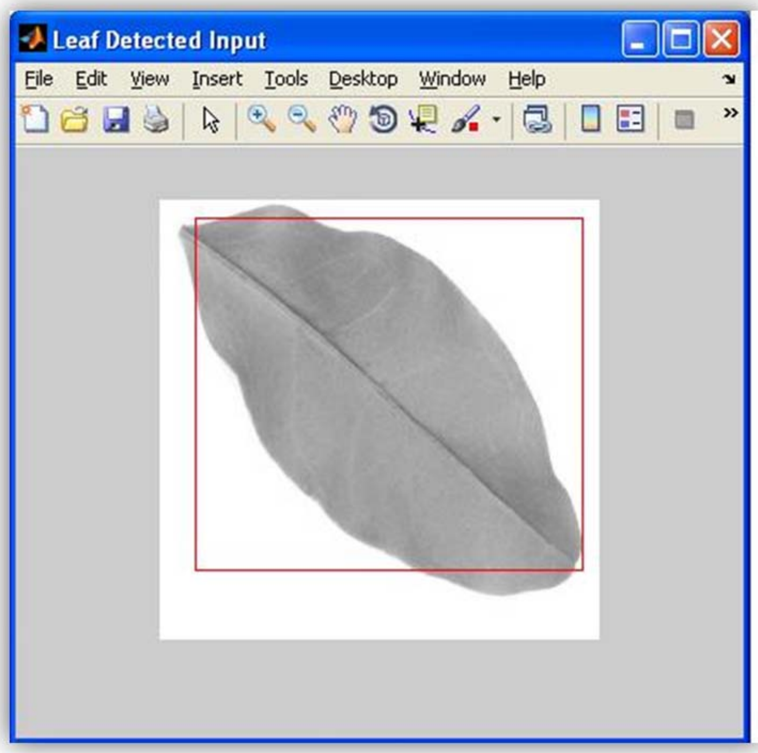

Figure 9.10. Snapshot of the image after applying edge detection.

Figure 9.11. Snapshot of the result window.

9.7 Leaf Input Detected

The input leaf is detected against the leaf image in the database.

9.8 Matching

Matching displays the result.

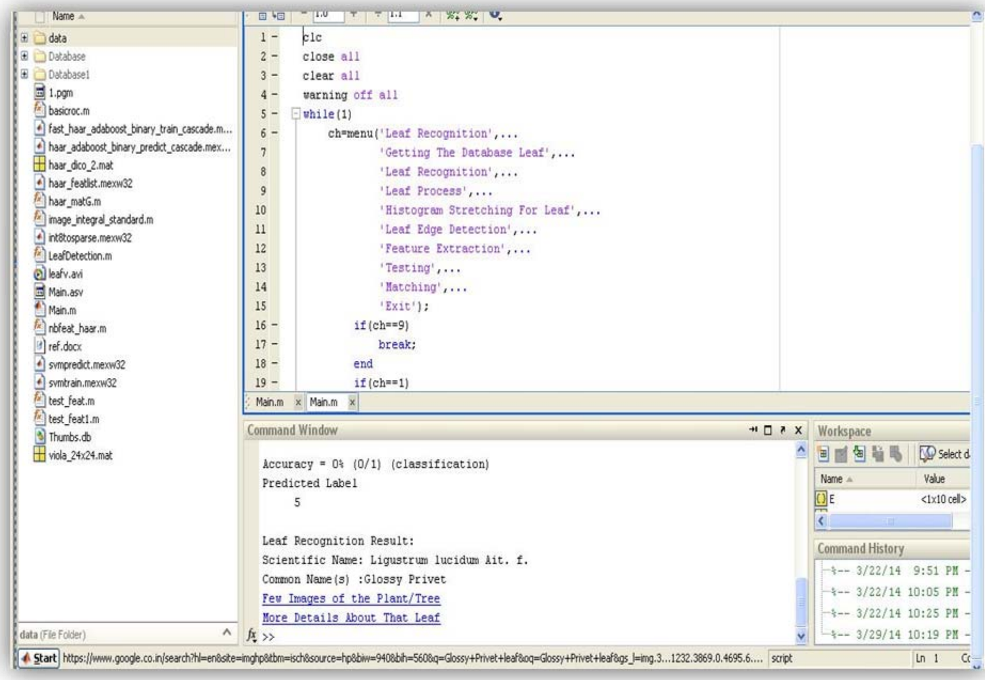

Figure 9.12. Snapshot of the label and Wikipedia link derived from the result.

9.9 LEAF LABEL OUTPUT

The command window displays the family name and a link to Wikipedia to display more details about the leaf detected in web browser.

9.10 WIKIPEDIA LINK

The Wikipedia link opens up in a browser and displays more information for the user about the leaf detected.

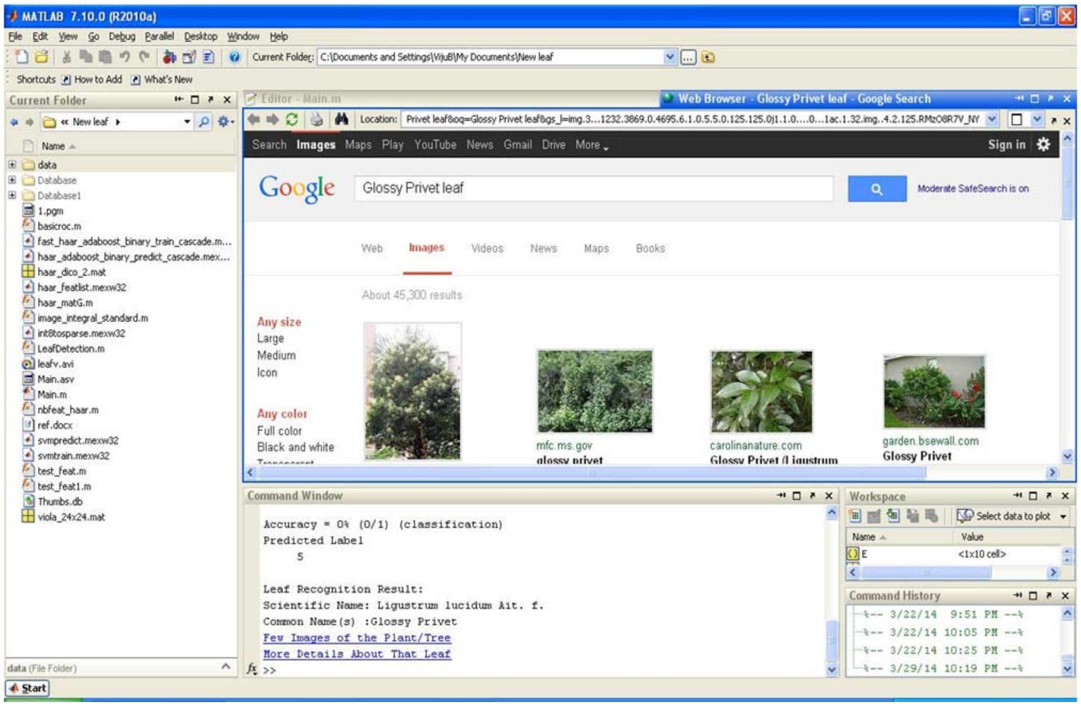

Figure 9.13. Snapshot of the Wikipedia of the image obtained in the leaf.

CHAPTER 10

IMAGE PROCESSING TOOLBOX

Image Processing Toolbox™ provides a comprehensive set of reference-standard algorithms and graphical tools for image processing, analysis, visualization, and algorithm development. You can perform image enhancement, image deploring, feature detection, noise reduction, image segmentation, geometric transformations, and image registration. Many toolbox functions are multithreaded to take advantage of multi core and multiprocessor computers. Image Processing Toolbox supports a diverse set of image types, including high dynamic range, Gig pixel resolution, embedded ICC profile, and topographic. Graphical tools let you explore an image, examine a region of pixels, adjust the contrast, create contours or histograms, and manipulate regions of interest (ROIs). With toolbox algorithms you can restore degraded images, detect and measure features, analyze shapes and textures, and adjust color balance [19].

10.1 MATLAB SUMMARY

Matlab is a commercial "Matrix Laboratory" package which operates as an interactive programming environment. It is a mainstay of the Mathematics Department software lineup and is also available for PC's and Macintoshes and may be found on the CIRCA VAX’s. Mat lab is well adapted to numerical experiments since the underlying algorithms for Mat lab's built-in functions and supplied m-files are based on the standard libraries LINPACK and EISPACK .Mat lab program and script files always have filenames ending with ".m"; the programming language is exceptionally straightforward since almost every data object is assumed to be an array. Graphical output is available to supplement numerical results [19].

10.2 KEY FEATURES OF MATLAB

- High-level language for technical computing
- Development environment for managing code, files, and data
- Interactive tools for iterative exploration, design, and problem solving
- Mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, and numerical integration

- 2-D and 3-D graphics functions for visualizing data
- Tools for building custom graphical user interfaces
- Functions for integrating MATLAB based algorithms with external applications and languages, such as C, C++, Fortran, Java, COM, and Microsoft® Excel®

CHAPTER 11

SUMMARY AND OBSTACLES

This has been a hard project which brought me a lot of knowledge. Now, I would have done things differently, as I am going to explain below in further work. As it was already said, we cannot use the common features as they are really not efficient. A better approach is to implement several specific features. Although using different specific features could be the solution, the results are surprisingly poor. The database used for this project is not exhaustive and the results could be really improved.

This study confirms the importance of leaf length, width, area and perimeter since the results obtained by the feature selection method selected these features as the most discriminant ones and combined them with other morphological features increased the results to 85 %.

Considerable deviations suggest needs for further improvement of the system. However, as the study was based on a limited sample size, reconfirmation of findings is needed with an adequate sample size. As the automated system is a novel method of identification of plants we believe that the performance, accuracy and results obtained are at least promising and have a potential in real plant identification application.

The biggest limitation of our system is, it requires user help in the pre-processing stage. Another limitation is its inability to work with images with complicated background. We would like to overcome these limitations.

Image processing is a parallelizable application, we could parallelize this application and desktop application is much faster and MATLAB image processing tools help in achieving accurate results.

To improve this program, several changes have to be made. The first one refers to the database: we must find a better one, with higher resolution, and better image position.

I haven’t had time to worry about flowers or berries, using leaves as leaves look relatively the same all time the year whereas flowers are seasonal but provide valuable aid in spring or summer.

Ignoring the profile of the entire plant, its best identified by their total profile because of the nature of what we are doing even 50 is plenty. Our goal was specifically chosen so that we could a good job with 13 or 15 species, if we kept things modest, the limitations of the Digital image processing machine wouldn’t be tested.

My initial goal was to achieve digital image processing over a smart-phone but due to few constraints to obtain effective responsive interactions which involve high complexity cost, running these algorithms on a smart-phone are often unattainable.

My future goal is to extend the current image processing desktop application to an android based smart-phone app, one way to effectively achieve is to offload the processing of algorithms on to a high performance server over the network and this would typically work as a client server image processing system. Using the Android based smart-phone; the input image to be identified is captured and sent to a server (eg. PHP Server) via HTTP. A script on the server invokes the server-side application (eg. MATLAB) to process the image which in turn sends the results to the smartphone via http. It would be nice to triple the set of brushy plants that the application can identify.

We can deduce from all this work that this application, even if it works well, can’t be used as a completely reliable tool. It can only help to begin in this domain, for interested people, but we should not trust it if the life of someone is at risk.

WORKS CITED

[1] S. Wu, F. Bao, E. Xu, Y. Wang, Y. Chang, and Q. Xiang. A leaf recognition algorithm for plant classification using probabilistic neural network. In *7th IEEE International Symposium on Signal Processing and Information Technology*, Cairo, Egypt, 2007.

[2] Z. Wang, Z. Chi, and D. Feng. Shape based leaf image retrieval. *IEEE P-Vis Image Sign.* 150:34–43, 2003.

[3] H. Fu, Z. Chi, D. Feng, and J. Song. Machine learning techniques for ontology-based leaf classification. In *8th IEEE International Conference on Control, Automation, Robotics and Vision*, Kunming, China, 2004.

[4] M. D. Dassanayake, editor. *A revised handbook to the flora of Ceylon*, volume 4. CRC Press, Boca Raton, 2003.

[5] M. Ashton, S. Gunathilleke, N. Zoysa, M. D. Dassanayake, N. Gunathilleke, and S. Wijesundera. *A field guide to the common trees and shrubs of Sri Lanka*. Wildlife Heritage Trust Publications, Sri Lanka, 1997.

[6] L. K. Senarathne. *A checklist of the flowering plants of Sri Lanka*. National Science Foundation Publishers, Sri Lanka, 2001.

[7] R. C. Gonzalez and R. E. Woods. *Digital image processing*, 2nd edition. Prentice Hall, Upper Saddle River, NJ, 2004.

[8] M. Sonka, V. Hlavac, and R. Boyle. *Image processing, analysis, and machine vision*, 2nd edition. Cengage Learning, Stamfort, CT, 2003.

[9] Otsu, N. A threshold selection method from gray-level histograms. *IEEE T. Syst. Man. Cyb.* 9:62–66, 1979.

[10] P.-E. Danielsson and O. Seger. Generalized and separable Sobel operators. In H. Freeman, editor, *Machine vision for three–dimensional scenes*, pages 347-380. Academic Press, CA, 1990.

[11] Y.-Y. Wan, J.-X. Du, D. S. Huang, Z. Chi, Y.-M. Cheung, X.-F. Wang, and G.-J. Zhang. *Bark texture feature extraction based on statistical texture analysis*. In Proceedings of the 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, pages 482–485, Hong Kong, 2004.

[12] D. S. Huang. *Systematic theory of neural networks for pattern recognition.* Publishing House of Electronic Industry, Beijing, 1996.

[13] D. S. Huang. The local minima free condition of feedforward neural networks for outer-supervised learning. *IEEE T. Syst. Man. Cy. B.* 28:477–480, 1998.

[14] B. V. Dasarathy. *Nearest neighbor NN norms: NN pattern classification techniques.* IEEE Computer Society Press, Washington, DC, 1991.

[15] T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE. T. Inform. Theory* 13:21–27, 1967.

[16] P. E. Hart. The condensed nearest neighbor rule. *IEEE. T. Inform. Theory* 14:515–516, 1967.

[17] F. F. Lei, M. Andreetto, and M. A. Ranzato. Caltech 101, 2006. http://www.vision.caltech.edu/Image_Datasets/Caltech101/, accessed Jun. 2013.

[18] New York State Department of Economic Development. Leaf Identifier, 2013. http://fallgetaways.iloveny.com/landing_leaf_identifier.html, accessed Aug. 2013.

[19] Math Works Inc. Matlab R2012a documentation, 2014. http://www.mathworks.com/help/index.html, accessed May 2013.