Democratic and Popular Republic of Algeria
Ministry of Higher Education and Scientific Research
Ahmed Draia University - Adrar
Faculty of Science and Technology
Department of Mathematics and Computer Science

جامعة أحمد دراية.أدرار-الجزائر
Université Ahmed Draia.Adrar -Algérie

A Thesis Presented to Fulfill the Master's Degree in Computer Science

Option : Intelligent Systems.

Title

Recognition of Leaf Images Based on generic fourier descriptor Using SVM

Prepared by
Mohamed Aymen ADNANE and Abedelkader ABAIDI

Supervised by
Mr. Mohammed Kohili

In front of a jury composed of

- President Dr OMARI mohammed
- Examiner Mr CHOUGUEUR Eldjilali
- Examiner Mr DEMRI Mohammed

Academic Year 2017/2018

Acknowlegement

Our thanks go first of all to all powerful ALLAH, for guiding our steps towards the path of knowledge.

We would like to thank our major Mr Mohamed kohili for his patience, for his precious advice and for all the conveniences and ease he brought us during our study and realization of this project.

We also express our gratitude to all the teachers who have collaborated in our studies. Without forgetting of course to thank profoundly all the people who participated directly or indirectly in the realization of this work.

Finally, that our dear parents and families find here the expression of our most sincere and profound thanks in recognition of their sacrifices, support and encouragement.

ADNENE Mohamed Aymen

We are grateful to ALLAH ALMIGHTY,

for enabling us to fulfill this work.

I would like to express my gratitude to my research supervisor, Prof. Mr.KOHILI Mohammed for his invaluable advice, guidance and his enormous patience throughout the development of the research.

I would like to thank my great parents, who never stop giving of themselves in countless ways,

My dearest fiancee D.M, who leads me through the valley of darkness with light of hope and support, My beloved family the symbol of love and giving.

I owe profound gratitudefor my teachers for their valuable advices and encouragement and for giving us very useful information that helped us a lot in completing this project.

I also want to pay a bundle of thanks to each and every person who has put indeed effort to make this project possible.

Toall the people in my life who touch my heart, I dedicate this research.

To all my beloved friends Halim, Aymen, Romaissa, Fatima, Aljomet and Ali…

ABAIDI Abedelkader

Abstract

We present in this thesis an experimental study, which aims to do an optimal selection of SVM model for the recognition of leaves. Include the generic Fourier descriptor for features extraction and type of classification (one-against-all) the choice of these parameters has a great influence on the final performance of the classification and the computation time. This work required the realization of a complete system of recognition offline of leaves, the generation of database of 32 classes each class has more than 50 images, which is approximately 1907 images. Preliminary results are very encouraging and promising compared to the general literature survey.

Key words

Recognition, leaves, Support Vector machine SVM, SVM multiclass, Generic Fourier Discriptor GFD.

Résumé

Nous présentons dans cette thèse une étude expérimentale, qui vise à faire une sélection optimale du modèle SVM pour la reconnaissance des feuilles. Inclure le descripteur générique de Fourier pour l'extraction des caractéristiques et le type de classification (un contre tous), le choix de ces paramètres a une grande influence sur la performance finale de la classification et le temps de calcul. Ce travail a nécessité la réalisation d'un système complet de reconnaissance hors ligne des feuilles, la génération d'une base de données de 32 classes, chaque classe a plus de 50 images, soit environ 1907 images. Les résultats préliminaires sont très encourageants et prometteurs par rapport à l'étude de la littérature générale.

Mots clés

Reconaissance, feuille, Support machine vectorielle SVM, SVm multiclasse, Générique Fourier Discriptor GFD.

Table of contenents

CHAPTER 1 : Leaf recognition study

CHAPTER 2 : GFD and SVM

CHAPTER 3 : Experemental results and disscussion

List of Figures

List of tables

General Introduction

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.

Among the techniques used for the recognition of leaves, we find support vector machines (SVM) based on the theory of statistical learning [33]. The SVM introduced at the beginning of the 90s, they realize a great success in statistical learning theory. Today, we can say without exaggerating that these machines have supplanted neural networks and other techniques of learning. Indeed, they are widely used in statistical learning and have had a great deal of success in almost all areas where they have been applied.

The most important objective of this work is the experimental selection of the model SVM for the recognition of leaves. We used a one versus all approach for the selection and scanning of a large space of parameters in order to have an appreciable recognition rate.

Our thesis is structured in three chapters:

The first chapter of this thesis presents the general concept of the leaf recognition. By focusing on the characteristics of leaves as well as the different phases of recognition process. Finally, we presented some recent work carried out in the field of leaf recognition.

The second chapter represents using Generic Fourier Descriptor for features extraction and represents a state of the art on carrier vector machines in with a synthesis of the different approaches of the multi-class SVM present through a careful analysis of these approaches, highlighting their strengths and weaknesses.

We present in Chapter 3 a detailed description of our system of leaf recognition. As we presented the different experimental results by using SVM and the implementation of our search strategy to vary and select hyper parameters that give the right recognition rate. Finally, we present a discussion and evaluation of the results obtained.

CHAPTER1

Leaf recognition study

1.1 Introduction

Approximately 350,000 species of plants exist on earth, and they share a very close relationship to human beings. Plants play a major role in various areas, such as food, medical science, industry, and the environment. However, many species of plants are endangered because of environmental pollution due to the rapid development of human society. Therefore, it is very important to study automatic plant classification and recognition for plant protection.

Leaf recognition technology plays an important role in plant classification and its key issue lies in whether selected features are stable and have good ability to discriminate different kinds of leaves.

Many recent studies exist on plant classification and recognition based on plant components such as flowers, leaves, and barks. To handle such volumes of information, realization of a quick and efficient classification method has become an area of active study [14-16]. In particular, it is well known that the correct way to extract plant features involves plant recognition based on leaf images. Two features, which are widely used for plant recognition based on leaf image, are color and shape [14], [15-16]. In the color-based conventional study, a simple color similarity between two images can be measured by comparing their color histogram. Also in the shape based-conventional study, they used region and contour-based simple features and features could be considered time domain data.

However, the recognition performance was limited due to leaf color was affected by theseasons and there is a problem that user to directly specify both ends of the leaves.

1.2 plants

Plants are living things that are made up of cells. They need air, water, soil, and sunlight to live. They cannot move from place to place, but their leaves move to catch the sun and their roots move to-wards water. Their seeds can be carried by animals or blown by the wind.

We get food from all different parts of the plant: flowers, fruits, vegeta-bles, seeds, nuts, stems, and leaves. Grass gives us a cool, soft place to walk. Some plants give us medicine, and trees are used to make paper and furniture.

Over 270,000 species of plants have been identified and classified, but scientists believe that there are millions more waiting to be discovered. [1]

1.2.1 The plant kingdom

The Plant Kingdom is a way to classify (or organize) plants. They are divided into groups based on the traits they have in common. Scientists change the way plants are classified from time to time, when they discover new types of plants or learn new things about plants.

The two main groups are vascular plants (plants that use stems and veins to transport food and water), and non-vascular plants (plants with no roots, stems, or leaves).

Vascular plants can be divided into smaller groups, one of which is seed plants. This group includes flowering and non-flowering plants. Flowering plants include monocots (one seed leaf) and dicots (two seed leaves). The non-flowering plants can also be divided into several groups, including cycads, conifers, and ginkgo.[1]

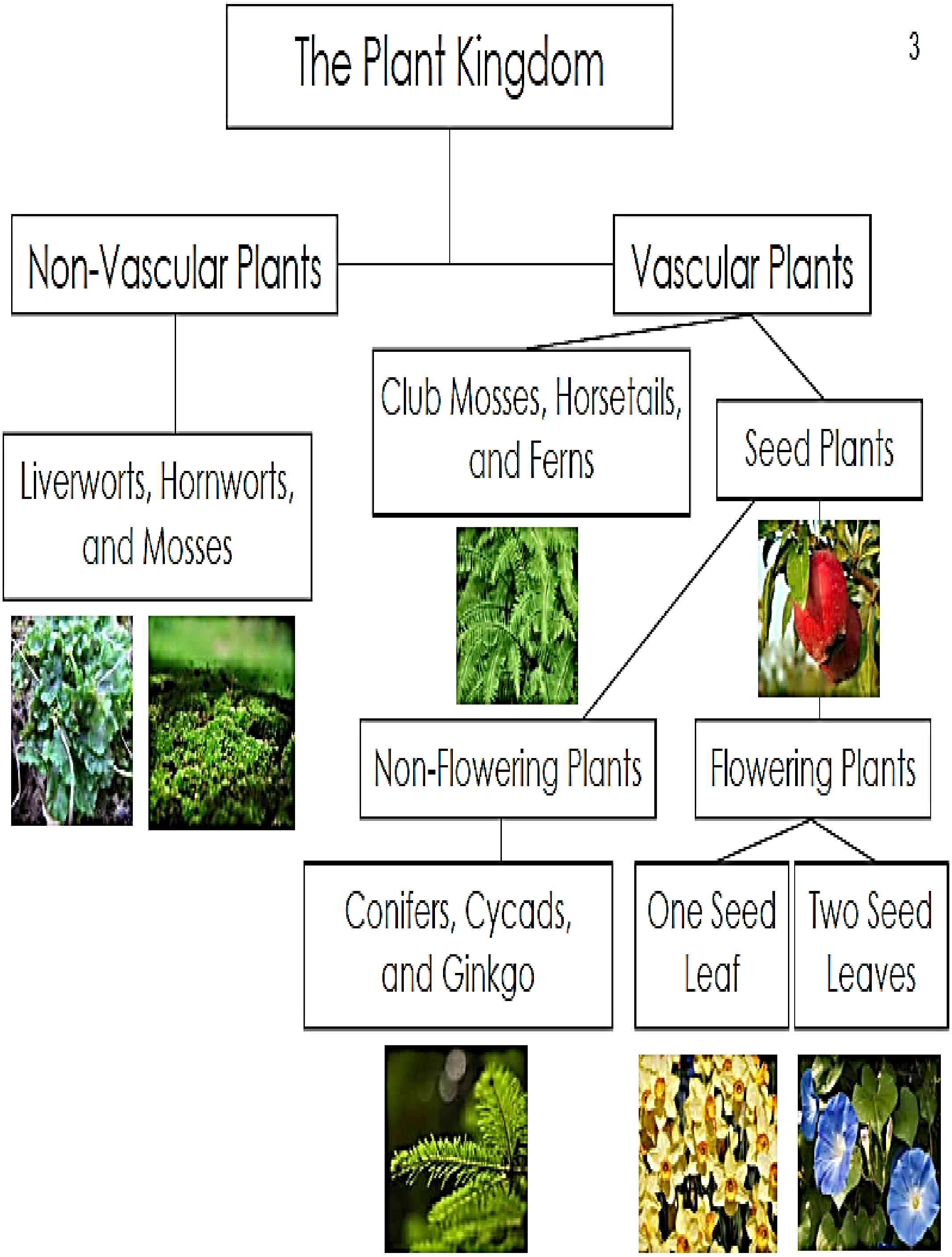


Figure1.1 :The plant kingdom

1.2.2 Trees

There are two different types of trees: non-flowering trees that have seeds that are not en closed, and flowering trees that have seeds that are enclosed. An example of a non-flowering tree would be a pine tree. An example of a flowering tree would be a fruit tree, such as peach or orange. [1]

Flowering trees are deciduous; that is, they shed their leaves every year. Other trees are conifers; they grow new leaves before shed-ding old ones, and stay green all year round ("evergreen").

Figure1.2: Trees

Trees consist of roots, trunk (stem), branch-es, twigs, and leaves. The tallest trees in the world are the redwoods of California, which can grow to be 379 ft (115.55 m) in height.

1.2.3 Leaves

Leaves are the primary food-producing organs of a plant. They are designed to be efficient in collecting light and using that light energy to produce food. [1]

1.2.3.1 Parts of Leaves

The main light-collecting structure on a leaf is a large, broad, flat surface called the leaf blade. The blade has many layers that not only help the plant move but also help it store materials and byproducts of photosynthesis. The blade is held away from the stem and supported by the petiole. The petiole is not exactly like a stem, but it does have xylem and phloem that transport water and sugar. The blade is supported by a system of veins that also contain both xylem and phloem. These veins prevent the blade from collapsing under its own weight. A leaf is often organized with one main vein running down the middle of the blade. This vein is called the midrib. All of the veins, the petiole, and the midrib help position the blade so that it is facing the light source. [1]

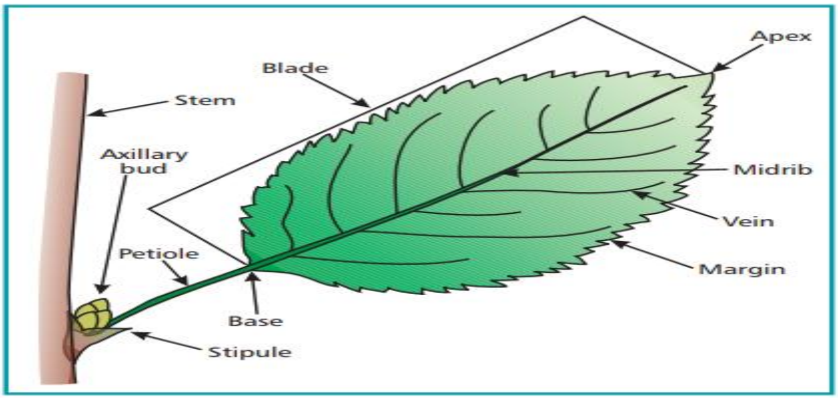


Figure1.3: The major parts of a leaf

1.2.3.2 Veins

Veins of flowering plants are found in several patterns. Monocots and dicots have differing patterns. Monocots have leaves with parallel veins. While the veins may not be parallel in a strict mathematical sense, none of the veins on a monocot leaf cross. They may look as if they are fused together at the top or bottom of the blade. Corn and grass plants are good examples of monocots. [1]

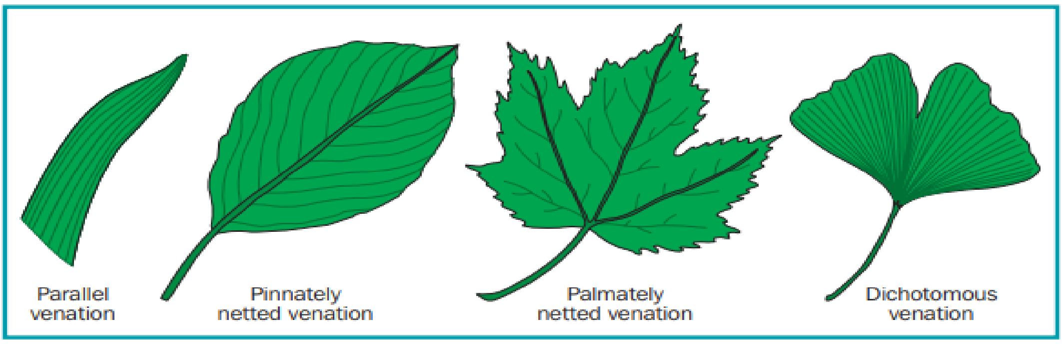


Figure1.4 : Leaf venation patterns

Dicots have leaves with veins that connect and branch from each other. Veins in a branching pattern are called netted veins. A leaf with netted veins sometimes has several smaller veins branching out of a dominant midrib, a condition known as pinnately netted. Elms and oaks have pinnate netting. A leaf may sometimes have several dominant veins branching out from the petiole. This condition, known as palmately netted, is common with maples and redbud. A few plants have a spreading vein pattern called dichotomous venation. A gingko leaf has veins of this type. [1]

1.2.3.3 Types Of Leaves

There are many different types of leaves. Some leaves are adapted to hot, dry climates by being able to store water or being smaller. Some leaves have very large blades to collect the maximum light in a shady location. The blades of some leaves are broken into three or more sections. A leaf that has only one blade on its petiole is called a simple leaf. Most plants have simple leaves. A leaf that has multiple blades, or leaflets, is called a compound leaf. There are different kinds of compound leaves. Two common types are the palmately compound leaf and the pinnately compound leaf. A palmately compound leaf has all its leaflets attached to a common point. A pinnately compound leaf has multiple leaflets attached along a rachis, or axis. [1]

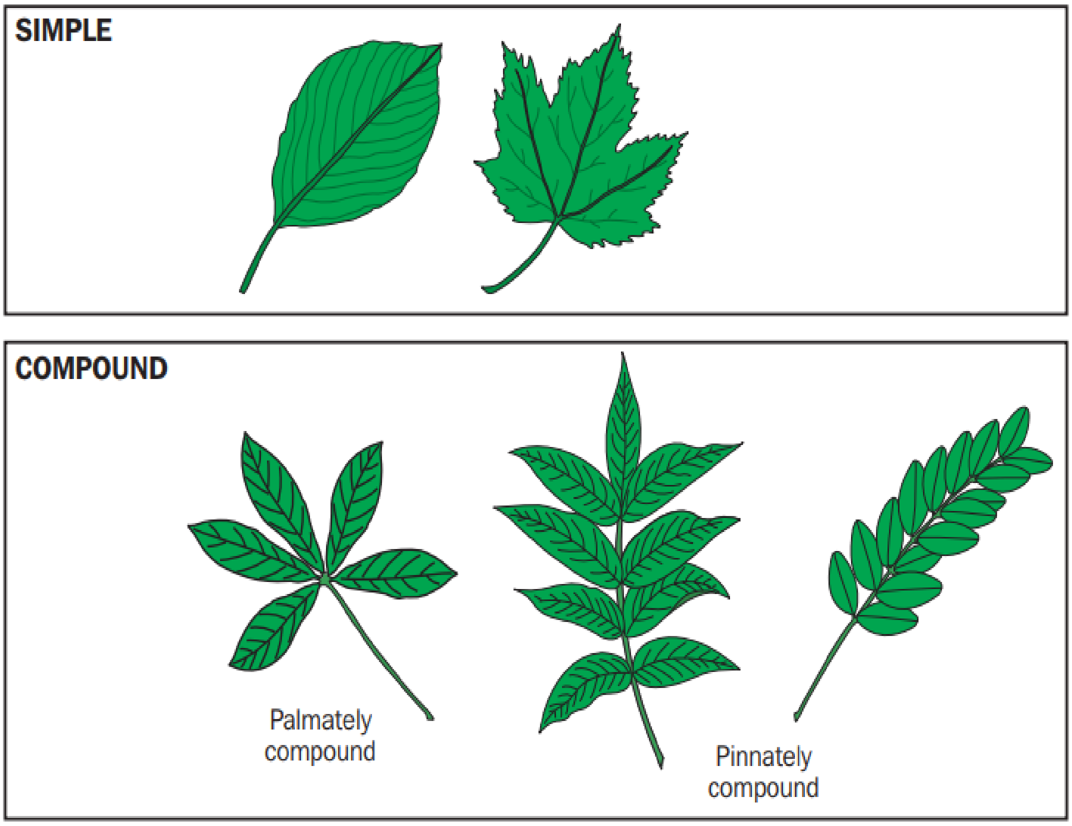


Figure1.5: Types of leaves

1.2.3.4 What is the shape of the leaf?

There are many terms used to describe leaf shape. Only seven of the more common leaf shapes are presented here (Figure 1.6), but guides mentioned in “Additional Resources” provide additional terms and descriptions. Elliptic leaves are broadest in the middle and narrower at either end. Linear leaves are long and narrow with the sides being close to parallel to each other. Lanceolate leaves are much longer than wide, with the widest point below the middle of the leaf. Spatulate leaves look kind of like a spatula, with the tip being rounded and gradually tapering to the base. Ovate leaves are egg-shaped while oval leaves are round to oval, lacking a pointed tip. Cordate leaves are heartshaped. [1]

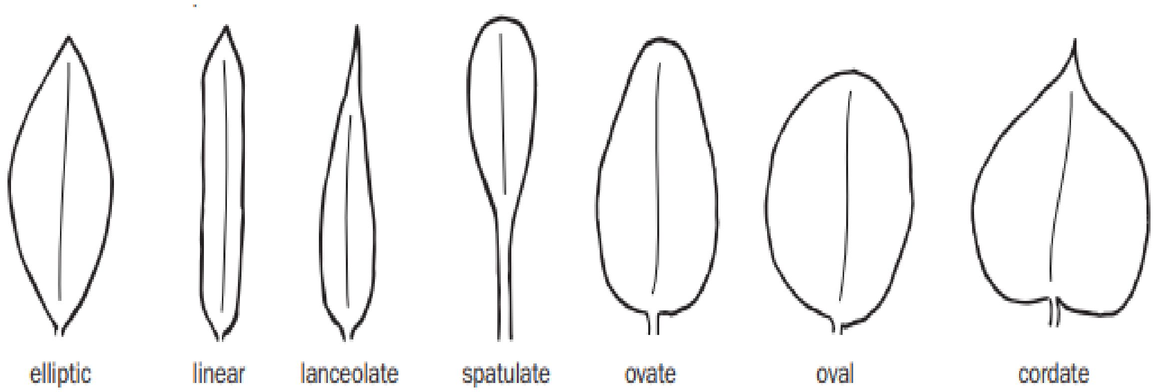


Figure1.6:Leafshapes

1.2.3.5 What other leaf characteristics are important?

The margin, or edge of the leaf, can also assist in plant identification. As with leaf shape, there are many ways to describe leaf margins. Only three major descriptions are provided here (Figure 1.7), but check the “Additional Resources” for more information. Entire margins are smooth and do not have any teeth, notches, or divisions. Leaves that have a toothed or saw-like margin are called dentate or serrate. Lobed leaves have indentations along the margin that cut inward toward the leaf midvein. Another leaf characteristic that can help with plant identification is whether the leaf is petiolate, sessile, or clasping (Figure 1.8). Petiolate leaves have a stalk (petiole) that attaches them to the stem. Sessile leaves do not have a petiole and are attached directly to the stem. Clasping leaves are sessile (i.e. do not have a petiole)and have a base that wholly or partly wraps around the stem. The leaves of Dalmatian toadflax (Linaria dalmatica) provide a good example of clasping leaves. Whitetop (Cardaria spp.) leaves are petiolate on the lower portion of the stem and clasping on the upper portion of the stem. [1]

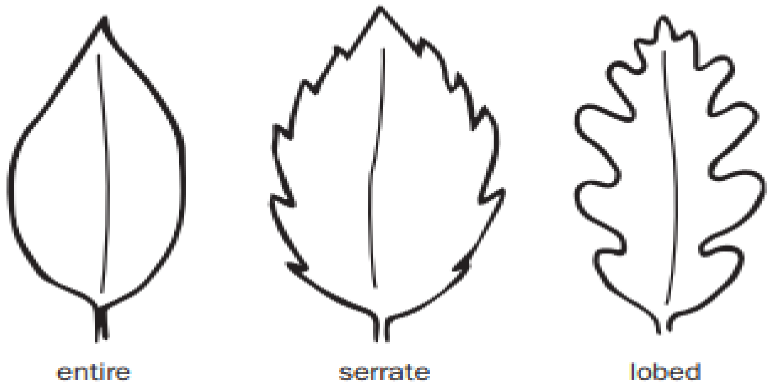


Figure 1.7: Leaf margins

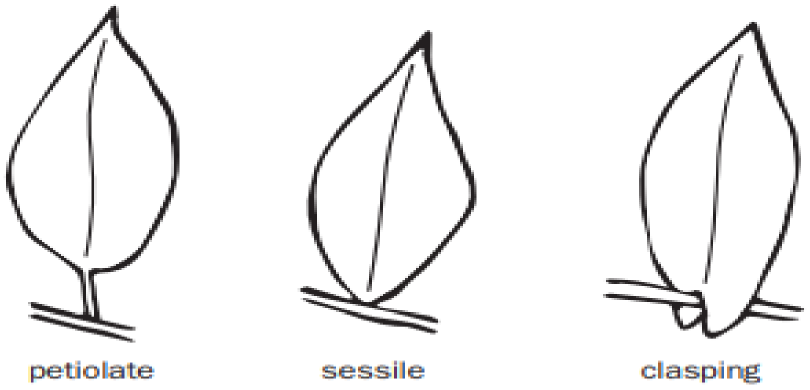


Figure 1.8:Leaf attachments

1.3

Computer Vision

Image recognition is the basic task in the areas of computer vision and pattern recognition. The field of computer vision is concerned with the automated processing of any images from the real world to extract characteristics through computers and interpret information on a real time and user requirements basis. Today, computers are playing key role in our daily lives. Most significant developments in computer hardware and software have contributed to the development of many real world research applications, namely, finger print recognition, digital signature recognition, automatic medical diagnosis and treatment, optical character recognition, document processing, video processing and plant image recognition etc. In addition to these, we also find many commercial applications, namely, commerce, sports, entertainment, business, education, food science, medical science, industrial and automation. An effort has also been carried in forestry, farming, and botany and in many fields that are more allied. The crop growth and health analysis, soil fertility, pest or disease detection, weed detection and post-harvest operations are the most common applications. Ultimately, all the applications exhibit machine intelligence in the respective fields.

Figure 1.9: Medicinal plants parts used as extracts of medicine

Figure 1.10: People searching for the collection of medicinal plants

In recent years, an automatic content extraction, analysis and retrieval for medicinal plants based on visual trait studies has become more important than ever before. This is because manual identification and processing of herbal drugs suffers from two major issues, namely, the scarcity of experts and subjectivity arising with the individuals in the field of Ayurveda. Therefore, the Ayurveda practitioners and biologists are in need of efficient computer software to automatically extract and analyze significant content of images of medicinal plants. Hence, in order to supplement the human visual system with machine vision, medicinal plant recognition is considered a research issue.

The general taxonomy provides important information for the identification and classification of medicinal plants. All the information about the plant like, locality of collection, date, morphological characters of plant like leaf and flower morphology, branching, venation, inflorescence, fruit and seed character are obtained. The Ayurveda researchers and botanists require such information for identification. Instead of manual system, a database of medicinal plants is created and retrieval methodologies based on text and visual information becomes useful. However, the design of a machine vision system to recognize, classify and retrieve medicinal plants is a challenging task due to the factors like illumination levels, weather conditions, shadows, occluded regions and cluttered background. Hence, the authors have chosen to help people living in forests, rural areas and those who practice Ayurveda medicines. In the present work, the authors have attempted to design an efficient approach for the identification and classification of images of Indian medicinal plants through digital image processing techniques. Finally, an image search engine for database of medicinal plants is developed for retrieval based on text and visual information. The Figure (1.11) gives the samples of images of medicinal plants with their scientific names. The different leaves with varying shapes and margins with their botanical terminologies.

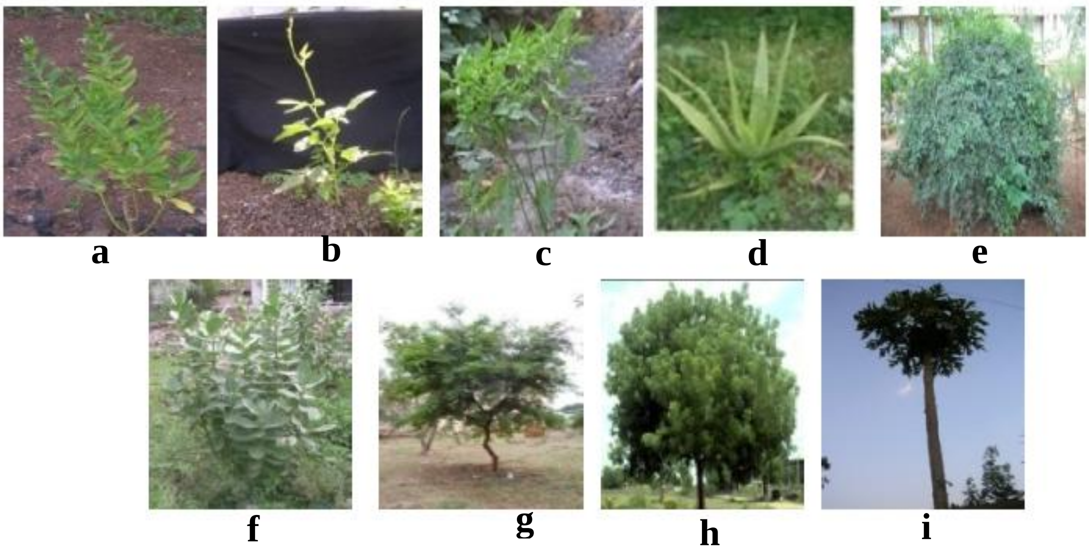


Figure1.11:Samplemedicinalplantimages(a)Catharanthusroseus (b)Vignaunguiculata(c)Capsicumannum(d)Aloebarbadensis (e)Saptachakra(f)Calotropisgigantea(g)Acaciacatechu (h)Azadirachitaindica(i)Caricapapaya

1.4 SystemOverview

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. [13]

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 plants.

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 (1.12).

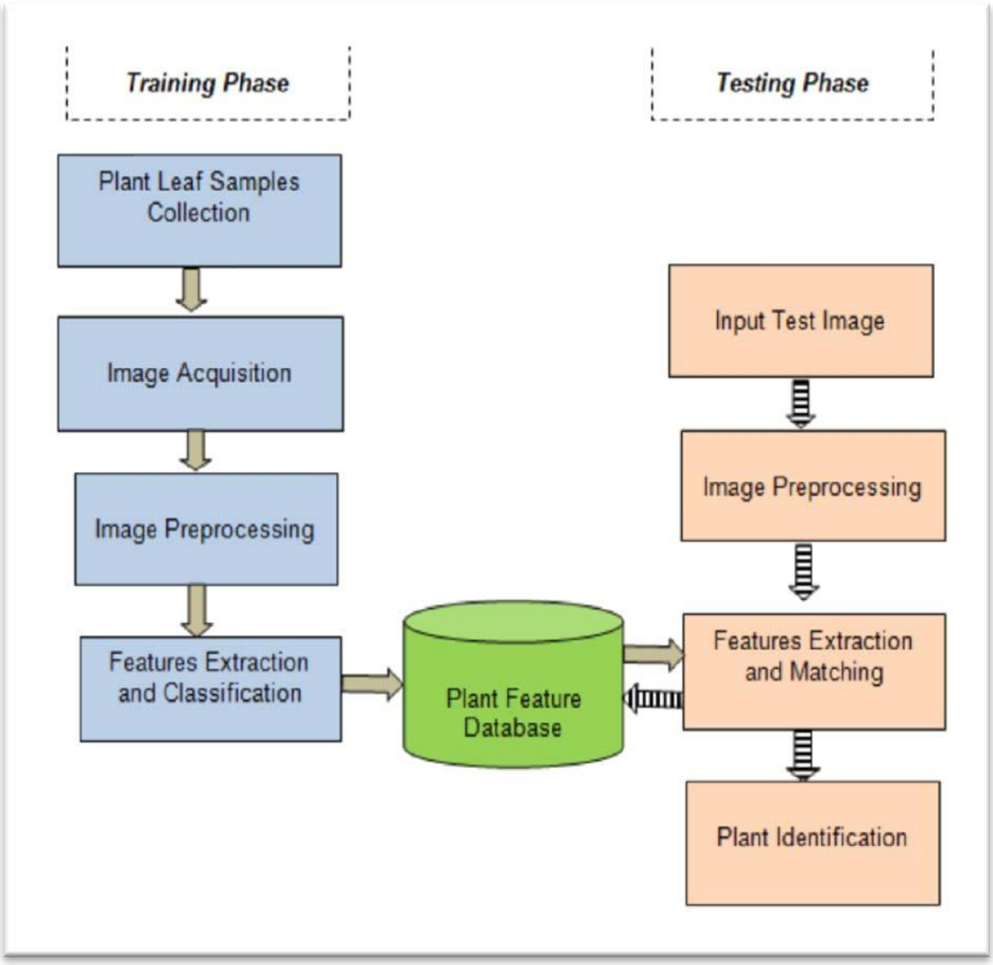


Figure1.12 : Main pattern recognition steps

1.5 Image Preprocessing

Image processing is a recent science that has emerged with the advent of the latest computer graphic technologies. This science aims to process and analyze digital images through a set of techniques and methods, in order to derive information that can be used to answer important questions and solve problems related to a specific field.

Before the operations, some of the leaf images are rotated manually for helping the program to arrange leaf direction to the right side. After wards, automatic preprocessing techniques are applied to all of the leaf images. The preprocessing steps are illustrated.[2]

1.5.1 Definition of real image

Before discussing the scanned image, it is good to talk about the notion of image. In the etymological sense, the word "image", derived from the Latin world "imago", refers to the visual representation of an object by different means or supports (drawing, photography, painting, sculpture…). An image is the projection or analog representation of the actual intensities of a 3D scene on a 2D plane. It is also defined as structured setoff information that, after being displayed on the screen, has meaning for the human eye.

Mathematically, it can be described as a function I (x, y) of continuous analog brightness, defined in a bounded domain, the x and y are the spatial coordinates of a point of the image and I ids the function of light intensity and color. In the aspect, the image is unusable by the machine, which requires it is digitization.

As a definition of the realimages of a leaf in nature. It is a real model distinct in a certain way there are many plant species that have different leaves form and this in fact may distinguish this type from that. The leaf actually has a color that is often green. There are technical means that allow us to study them so the beginning is to take real pictures using the camaraderie of nature.

Figure 1.13: real image of leaves

1.5.2 Definition of leaf image on gray:

Most leaves have generally green color, while the color of leaves is changed by season or environmental factors. The color change of leaf image can cause decline of recognition performance or non-recognized problem. The color converting process on input image is the first step for leaf contour extraction, and it can set foundation to improve recognition performance irrelevant to the leaf color change.

Therefore, we convert the input color leaf image to gray scale image as follows:

$$\boldsymbol{Gray = 0.299 * R + 0.587 * G + 0.114 * B} \quad \textbf{(1.1)}$$

The converted gray scale leaf image is converted to a binary image once again. The threshold conversion is performed as follows:

$$\boldsymbol{B(x,y) = \begin{cases} 0 & if\ \ f(x,y) \le T \\ 255 & if\ \ f(x,y) > T \end{cases}} \quad \textbf{(1.2)}$$

Where, *B(x, y)*and *f(x, y)*are the intensity values of the gray scale image and the binary image, respectively, at position *(x, y)*, and*T* is the threshold value [10]. Figure (1.14) shows an example of leaf contour extraction.

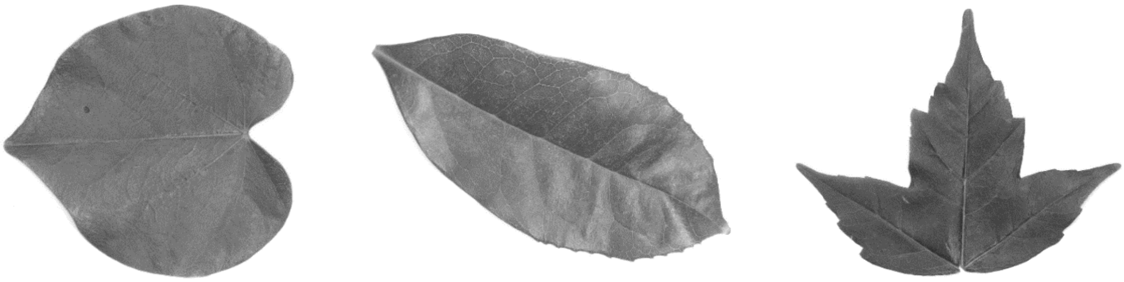


Figure1.14: image on gray

1.5.3 Definition of a digital leaf image

Unlike the leaf image obtained using an analog camera, or drawn on paper, the digital leaf image is stored on a computer in binary form (bit sequence of 0 and 1) after its acquisition. The storage of the scanned image can take place in different compressions formats (jpeg, bmp, png, gif, ...) whose decompression givesan image of the same size as the original image. The latter is represented by a matrix of elements called pixels (term resulting from the contraction of the English words "Picture" and "Element"). The value of each pixel represents a discrete intensity of light.

A digital leaf image is also defined as a discrete plane derived from an analog image after it has been digitized.

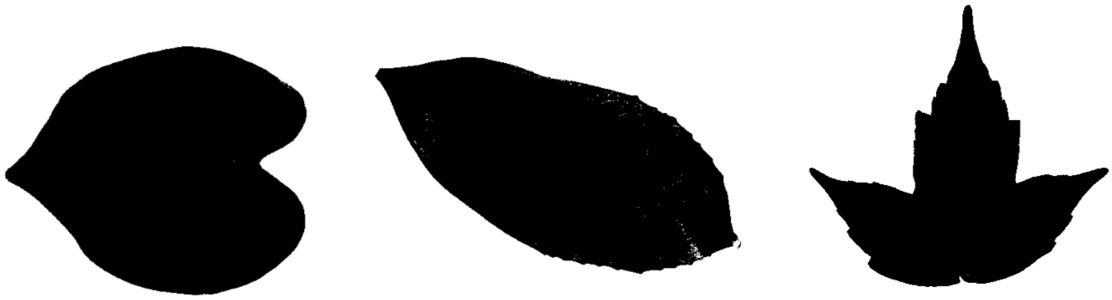


Figure 1.15: Digital leaves image

1.6 Image segmentation Techniques

The main aim of segmentation is simplification i.e. representing an image into meaningful and easily analyzable way. Image segmentation is necessary first step in image analysis. The goal of image segmentation is to divide an image into several parts/segments having similar features or attributes. The basic applications of image segmentation are: Content-based image retrieval, Medical imaging, Object detection and Recognition Tasks, Automatic traffic control systems and Video surveillance, etc. The image segmentation can be classified into two basic types: Local segmentation (concerned with specific part or region of image) and Global segmentation (concerned with segmenting the whole image, consisting of large number of pixels). The image segmentation approaches can be used many methods based on properties of image.

1.6.1 Thresholding Method

Thresholding methods are the simplest methods for image segmentation. These methods divide the image pixels with respect to their intensity level. These methods are used over images having lighter objects than background. The selection of these methods can be manual or automatic i.e. can be based on prior knowledge or information of image features. There are basically three types of thresholding [3] [4]:

Global Thresholding: This is done by using any appropriate threshold value/T. This value of T will be constant for whole image. On the basis of T the output image can be obtained from original image as:

$$\boldsymbol{q(x,y)} = \begin{cases} \mathbf{1} \;\; \boldsymbol{if} \;\; \boldsymbol{q(x,y) > T} \\ \mathbf{0} \;\; \boldsymbol{if} \;\; \boldsymbol{q(x,y) \leq T} \end{cases} \qquad \textbf{(1.3)}$$

1) Variable Thresholding: In this type of thresholding, the value of T can vary over the image. This can further be of two types:
    - Local Threshold: In this the value of T depends upon the neighborhood of x and y.
    - Adaptive Threshold: The value of T is a function of x and y.
2) Multiple Thresholding: In this type of thresholding, there are multiple threshold values like T0 and T1. By using these output image can be computed as:

$$\boldsymbol{q(x,y)} = \begin{cases} \boldsymbol{m \ \ if \ \ q(x,y) > T1} \\ \boldsymbol{n \ \ if \ \ q(x,y) \leq T1} \\ \boldsymbol{0 \ \ if \ \ q(x,y) \leq T0} \end{cases} \qquad \textbf{(1.2)}$$

The values of thresholds can be computed with the help of the peaks of the image histograms. Simple algorithms can also be generated to compute these.

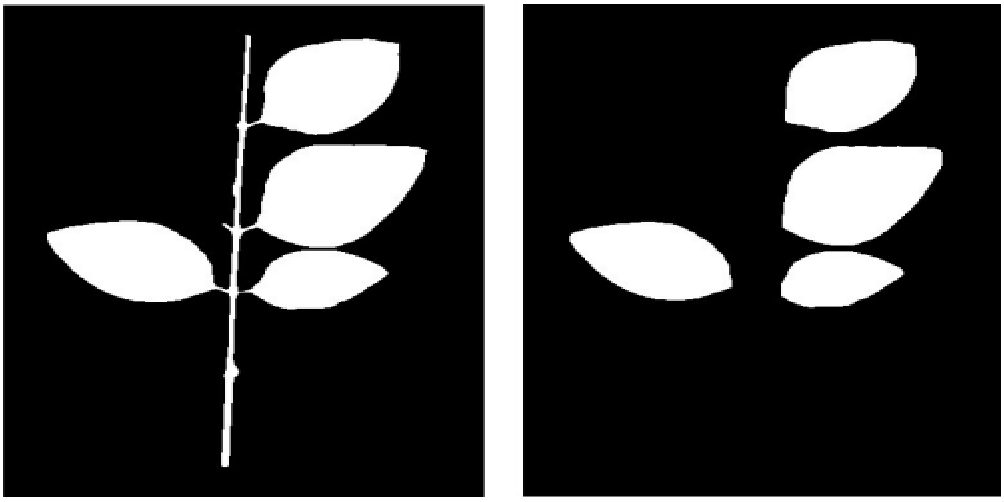


Figure 1.16: Thresholding segmentation Method

1.6.2 Edge Based Segmentation Method

The edge detection techniques are well developed techniques of image processing on their own. The edge based segmentation methods are based on the rapid change of intensity value in an image because a single intensity value does not provide good information about edges. Edge detection techniques locate the edges where either the first derivative of intensity is greater than a particular threshold or the second derivative has zero crossings. In edge based segmentation methods, first of all the edges are detected and then are connected together to form the object boundaries to segment the required regions. The basic two edge based segmentation methods are: Gray histograms and Gradient based methods. To detect the edges one of the basic edge detection techniques like sobel operator, canny operator and Robert's operator …etc can be used. Result of these methods is basically a binary image. These are the structural techniques based on discontinuity detection [5].

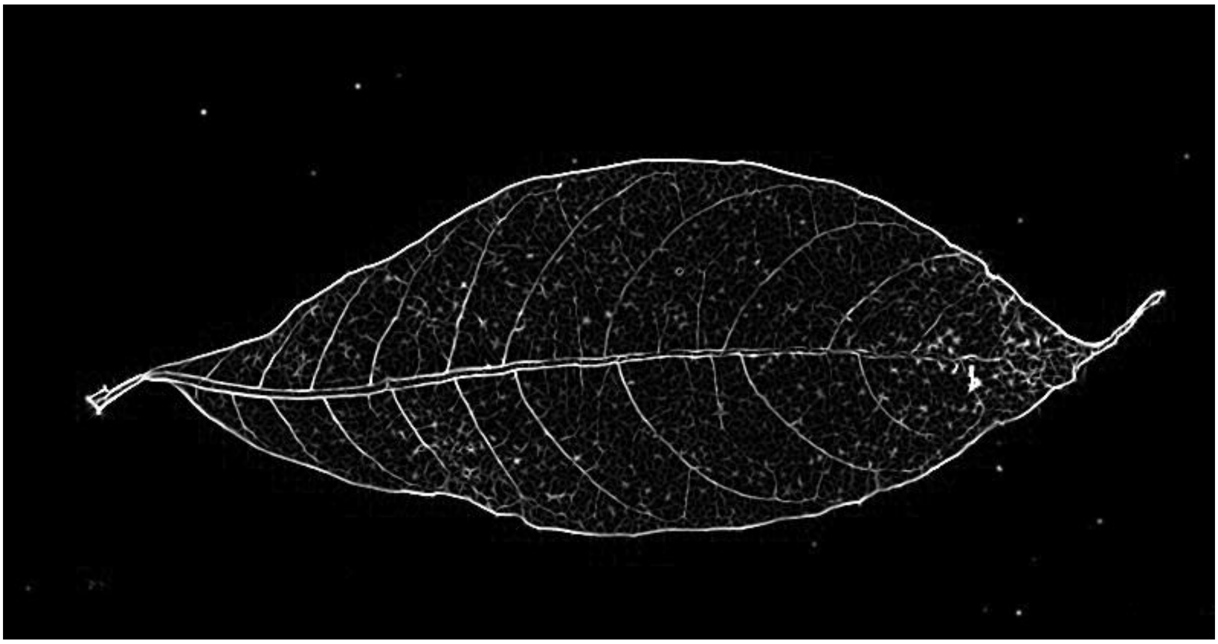


Figure1.17: Edge Based Segmentation Method

1.6.3 Region Based Segmentation Method

The region based segmentation methods are the methods that segments the image into various regions having similar characteristics.[6] [7] [8].

Figure1.18: Region Based Segmentation Method

1.6.4 Clustering Based Segmentation Method

The clustering based techniques are the techniques, which segment the image into clusters having pixels with similar characteristics. Data clustering is the method that divides the data elements into clusters such that elements in same cluster are more similar to each other than others. There are two basic categories of clustering methods: Hierarchical method and Partition based method. The hierarchical methods are based on the concept of trees. In this the root of the tree represents the whole database and the internal nodes represent the clusters. On the other side the partition based methods use optimization methods iteratively to minimize an objective function. In between these two methods there are various algorithms to find clusters. There are basic two types of clustering [9] [10].

1) Hard Clustering: Hard clustering is a simple clustering technique that divides the image into set of clusters such that one pixel can only belong to only one cluster. In other words, it can be said that each pixel can belong to exactly one cluster.

These methods use membership functions having values either 1 or 0 i.e. one either certain pixel can belong to particular cluster or not. An example of a hard clustering based technique is one k-means clustering based technique known as HCM.

In this technique, first of all the centers are computed then each pixel is assigned to nearest center. It emphasizes on maximizing the intra cluster similarity and also minimizing the inter cluster equality.

2) Soft clustering: The soft clustering is more natural type of clustering because in real life exact division is not possible due to the presence of noise. Thus soft clustering techniques are most useful for image segmentation in which division is not strict. The example of such type of technique is fuzzy c-means clustering. In this technique pixels are partitioned into clusters based on partial membership i.e. one pixel can belong to more than one clusters and this degree of belonging is described by membership values. This technique is more flexible than other techniques [9].

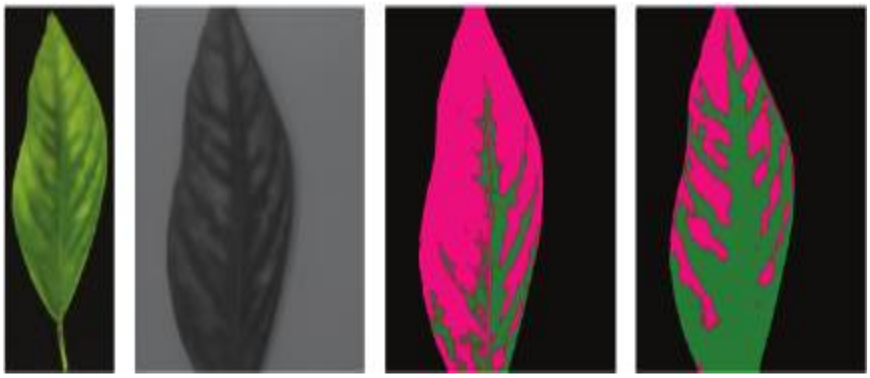


Figure1.19: Clustering Based Segmentation Method

1.6.5 Leaf segmentation challenges

In our case, the similarity between the background and the object of in-terest, and the difficulty to avoid adjacent and overlapping leaves constitute a prohibitive obstacle to the use of unconstrained active contours (Figure 1.20). The idea of using a template to represent the leaves is complicated by the fact that there is much more variety in shapes than for eyes or mouths. The only solution to overcome the aforementioned problems is however to take advantage of the prior knowledge we may have on leaf shapes to design a very flexible time-efficient model to represent leaves.

Figure 1.20: Segmentation issues with unconstrained region-based active contours

1.6.6 Comparison Of Various Segmentation Techniques

Table 1.1 shows a comparison between various segmentation techniques by specifying a brief description of every method each with its advantages and disadvantages[11]

Segmentation technique	Description	Advantages	Disadvantages
Thresholding Method	Based on the histogram peaks of the image to find particular threshold values.	No need of previous information, simplest method.	Highly dependent on peaks, spatial details are not considered.
Edge Based Method	Based on discontinuity detection.	Good for images having better contrast between objects.	Not suitable for wrong detected ortoo many edges.
Region Based Method	Based on partitioning image into homogeneous regions.	More immune to noise, useful when it is easy to define similarity criteria.	Expensive method in terms of time and memory.
Clustering Method	Based on division into homogeneous clusters.	Fuzzy uses partial membership therefore more useful for real problems.	Determining membership function is not easy.

Table1.1: comparison between various segmentation technique.

1.7 Plant Features

Human beings identify the plants by their stems, leaves, flowers and fruits. These become the features in their automatic recognition and differ from one plant to the other. Medicinal plants are not the exceptions. The classification of medicinal plants depends upon features uniqueness of the plant. The Scientists and Ayurveda practitioners and plant experts have collected data on innumerable plant species by studying the plants in their natural habitats and recorded information on their characteristics in scientific literature and databases for future reference. Consequently, taxonomists have defined some nomenclature and botanical terminologies through which plant and its parts are recognized. But, a machine vision system for recognition of medicinal plants requires capture of these features through either division of new approach for feature extraction or adoption of existing methods in the literature.

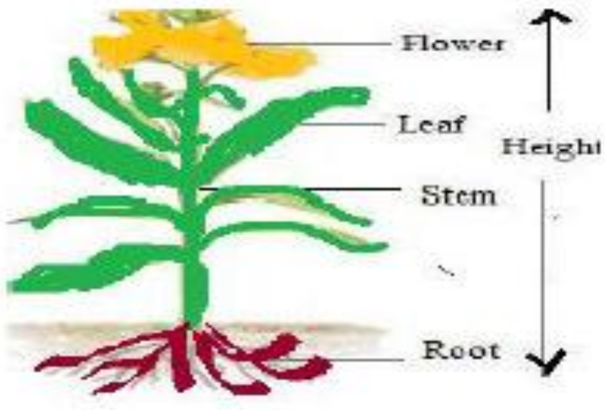


Figure1.21: features of plant

1.7.1 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.[13]

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 Ratioof Major Axis Length: This feature is denoted as ratio of minor axis length to major axis length. It isreverse of the aspect ratio that is used in the literature…etc.

1.7.2 Shape Features

We used the morphological features in [12] as the shape features, which are common shape features used in the literature. There are five fundamental features: the longest distance between any two points on a leaf border (L), the length of main vein (lengthwise- Lυ), the widest distance of a leaf (crosswise-W), the leaf area (A) and the leaf perimeter (P). Then, twelve features are constructed using these five fundamental features by some mathematical operations:

–smoothness of a leaf image

–aspect ratio (Lυ/W)
–form factor, the difference between a leaf and a circle (4 π A / P²)
–rectangularity (Lυ W / A)
–narrow factor (L/ Lυ)
–ratio of perimeter to longest distance (P / L)
–ratio of perimeter to the sum of main vein length and widest distance(P /(Lυ + W))
–and five structural features obtained by applying morphological opening ongrayscale image.

1.7.3 Color Features

In our experiments, we realized that some leaf images in Flavia dataset [12] have very similar shapes. Thus classification accuracy is greatly affected by this simi-larity. However, even shapes are similar i n some leaves, there are some differences in colors of leaves. Therefore, in addition to the shape features, we extracted color based features from leaf images. When calculating these features, we eliminated background color of leaf images.

We defined two sets of color features. In the first set, we used mean (μ) and standard deviation (σ) of intensity values of red, green, blue channels and average of these channels. So that, the first set of color features contain eight features.

Mean and standard deviation of each component are calculated as follows:

$$\mu = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N} Pxy \qquad \textbf{(1.5)}$$

$$\sigma = \sqrt{\frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}(Pxy-\mu)^2} \qquad \textbf{(1.6)}$$

In (1) and (2), M and N are dimensions of an image and Pxy is the intensity value of pixel at (x, y) coordinate.

The second set of color features consists of color histograms in red, green, and blue channels. RGB histograms provide us an efficient representation of color distribution. Thus, we are able to effectively analyze color information in an image by using these histograms. After studying several bin sizes, we obtained the best results with 10 bins in each histogram. Since we are calculating three histograms for red, green and blue channels, there are thirty new features in the second set of color features.

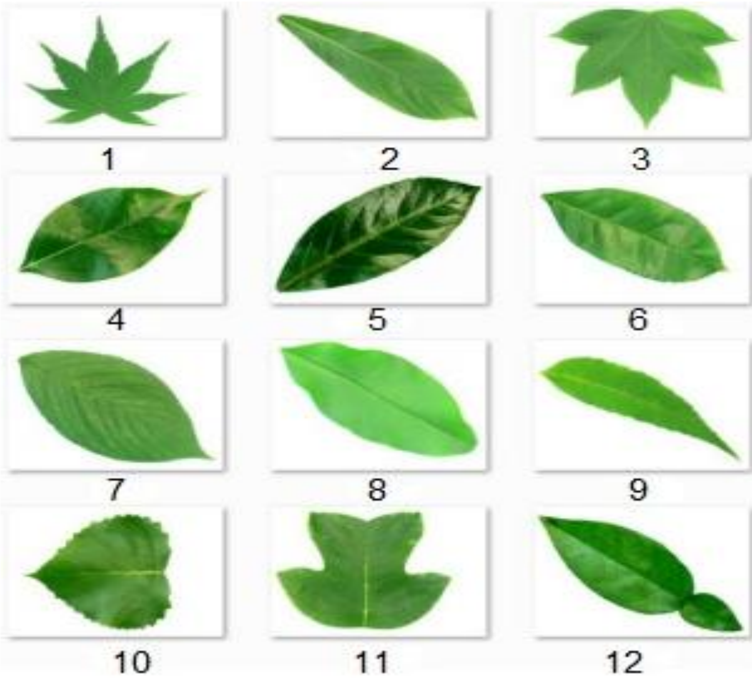


Figure1.22: Plant Recognition Approach Using deferent Shape and graduate Color Features

1.7.4 Texture Features

Guiying Li (2012) defined texture is a repeated pattern of information or arrangement of the structure with regular intervals. In a general sense, texture refers to surface characteristics and appearance of an object given by the size, shape, density, arrangement, proportion of its elementary parts. A basic stage to collect such features through texture analysis process is called as texture feature extraction. Due to the signification of texture information, texture feature extraction is a key function in various image processing applications like remote sensing, medical imaging and content-based image retrieval.

There are four major application domains related to texture analysis namely texture classification, segmentation, synthesis and shape from texture.

Texture classification produces a classified output of the input image where each texture region is identified with the texture class it belongs.

Texture segmentation makes a partition of an image into a set of disjoint regions based on texture properties, so that each region is homogeneous with respect to certain texture characteristics.

Texture synthesis is a common technique to create large textures from usually small texture samples, for the use of texture mapping in surface or scene rendering applications.

The shape from texture reconstructs three dimensional surface geometry from texture information. For all these techniques, texture extraction is an inevitable stage.

Figure1.23: texture of leaves

1.7.4.1 Texture Feature Extraction

Neville et al (2003) discussed texture features can be extracted using several methods such as statistical, structural, model based and transform information.

1.7.4.1.1 Structural based Feature Extraction

Structural approaches represent texture by well-defined primitives and a hierarchy of spatial arrangements of those primitives. The description of the texture needs the primitive definition. The advantage of the structural method based feature extraction is that it provides a good symbolic description of the image; however, this feature is more useful for image synthesis than analysis tasks. This method is not appropriate for natural textures because of the variability of micro-texture and macro-texture.

1.7.4.1.2 Statistical based Feature Extraction

Statistical methods characterize the texture indirectly according to the non-deterministic properties that manage the relationships between the gray levels of an image. Statistical methods are used to analyze the spatial distribution of gray values by computing local features at each point in the image and deriving a set of statistics from the distributions of the local features. The statistical methods can be classified into first order (one pixel), second order (pair of pixels) and higher order (three or more pixels) statistics.

The first order statistics estimate properties (e.g. average and variance) of individual pixel values by waiving the spatial interaction between image pixels. The second order and higher order statistics estimate properties of two or more pixel values occurring at specific locations relative to each other. The most popular second order statistical features for texture analysis are derived from the co-occurrence matrix.

1.7.4.1.3 Model based Feature Extraction

Model based texture analysis such as fractal model and Markov model are based on the structure of an image that can be used for describing texture and synthesizing it. These methods describe an image as a probability model or as a linear combination of a set of basic functions. The Fractal model is useful for modeling certain natural textures that have a statistical quality of roughness at different scales and self-similarity, and also for texture analysis and discrimination.

There are different types of models based feature extraction technique depending on the neighbor-hood system and noise sources. The different types are one-dimensional time-series models, Auto Regressive (AR), Moving Average (MA) and Auto Regressive Moving Average (ARMA). Random field models analyze spatial variations in two dimensions.

Global random field models treat the entire image as a realization of a random field, and local random field models assume relationships of intensities in small neighbor-hoods. Widely used class of local random field models are Markov models, where the conditional probability of the intensity of a given pixel depends only on the intensities of the pixels in its neighbor-hood.

1.7.4.1.4 Transform based Feature Extraction

Transform methods, such as Fourier, Gabor and Wavelet Transforms represent an image in space whose co-ordinate system has an interpretation that is closely related to the characteristics of a texture. Methods based on Fourier transforms have a weakness in a spatial localization so these do not perform well. Gabor filters provide means for better spatial localization but their usefulness is limited in practice because there is usually no single filter resolution where one can localize a spatial structure in natural textures. These methods involve transforming original images by using filters and calculating the energy of the transformed images. These are based on the process of the whole image that is not good for some applications which are based on one part of the input image.

1.8 Classification (Recognition)

Once the features have been extracted, and 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, and 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.[13]

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 eachclass of the training data. Recognition:

    1. Pre-processing

    2. Feature extraction: (both steps are same as above)

    3. Classification: Compare feature vectors tothe 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.

1.9 General Literature Survey

The authors have carried out literature survey to know the state of the art applications of computer vision and digital image processing techniques in the real world, more specifically connected with plant recognition. Following is the gist of the works cited in the literature.

Abdul kadir et. al., have proposed a method using combined features such as, polar Fourier transform, color moments and vein features to retrieve images of leaves. The method is very useful for the recognition of foliage plants. The system has been tested on Flavia andFolia leaf data sets. The retrieval accuracy of 93.13% and 90.13% are observed for Flavia and Folia data sets respectively.[42]

Mahmood R. Golzarian and Ross A. Frick have developed a method for classification of images of three grasses, namely, wheat, ryegrass and brome grass species at early growth stages. A combination of color, texture and shape features is used. The features are reduced to three descriptors using Principal Component Analysis. Three components are able to distinguish three grasses with a classification accuracy of 85% and 88% for ryegrass and brome grass respectively. The study helps for weed management .[43]

Mahmood R. Golzarian has investigated an adaptive learning for segmentation of plant images into plant and non-plant regions. The Kohenen's self-organizing map (SOM) algorithm is deployed for segmentation of plant images. Nine color features of three color space models are used as features. The method worked well even in the presence of noise. [43]

Faisal Ahmed et. al., have investigated the use of Support Vector Machine (SVM) and Bayesian classifier as machine learning algorithms for the effective classification of crops and weeds in digital images. From the performance comparison, it is reported that SVM classifier has outperformed Bayesian classifier. Young plants that did not mutually overlap with other plants are used in the study .[44]

B. Sathyabama et al.,has presented Content Based Leaf Image Retrieval (CBLIR) for ecommerce application. The Log-Gabor wavelet and Scale Invariant Feature Transform (SIFT) are deployed for leaf image texture and shape features respectively. The retrieval accuracyof 97.5% is observed .[45]

Ulrich Weiss et. al., have contributed a method for recognition of plant species by robots for weed detection and nursery plant detection. The plant species are distinguished using 3D

LIDAR sensor and supervised learning. Different learning methods like, logistic regression functions, Support Vector Machines and Neural Networks are found to be suitable. The average classification accuracy on six types of plant species, twenty images of each class is found to be 98% .[46]

Yovel Y et. al. have presented a new method for Plant classification from bat-like echolocation signals. In this work, a plant is considered as a three-dimensional array of leaves emitted bat call. The plants are classified based on signals from a database of plant echoes that are created by plants with a frequency-modulated bat-like ultrasonic pulse. The algorithm uses the spectrogram of a single echo from which it uses features that are accessible to bats. The Support Vector Machine (SVM) learning is used to automatically extract suitable linear combinations of time and frequency cues from the spectrograms. The classification is reported to be high.[47]

1.10 Conclusion

Leaves are the primary food-producing organs of a plant. The main light-collecting structure on a leaf is a large, broad, flat surface called the leaf blade. The blade is held away from the stem and supported by the petiole.

A leaf that has only one blade on its petiole is called a simple leaf. A leaf that has multiple blades is called a compound leaf. Two common types of compound leaves are the palmately compound leaf and the pinnately compound leaf.

Leaves are arranged along a stem in one of four major ways. They may be opposite, alternate, subopposite, or whorled.

The literature survey has reveals that the computers are used in many automation tasks related to plant domain. Still there are enormous applications connected to plant image recognition. We have observed plantidentification and classification in various fields such as, agriculture, weed classification, plant growth analysis, horticulture, forestry biomass prediction and vegetable recognition. Some works are also reported on plant species recognition based on their parts such as leaves, flowers and bark. But, the plant identification with respect to full image is very much scarce. Hence, from the literature survey, to the best of our knowledge, it has been observed that no considerable work has been cited in the literature on the development of machine vision systems for identification, classification and retrieval of medicinal plants images in the Indian context.

The automatic identification of medicinal plants with their relevant information such as plant names in different languages and medicinal usage is very much necessary in the present era. The methodologies help the development of a search engine for the database of Ayurveda medicinal plants in information retrieval through plant snap. This is the motivation for taking up this work.

CHAPTER1

Leaf recognition study

1.1 Introduction

Approximately 350,000 species of plants exist on earth, and they share a very close relationship to human beings. Plants play a major role in various areas, such as food, medical science, industry, and the environment. However, many species of plants are endangered because of environmental pollution due to the rapid development of human society. Therefore, it is very important to study automatic plant classification and recognition for plant protection.

Leaf recognition technology plays an important role in plant classification and its key issue lies in whether selected features are stable and have good ability to discriminate different kinds of leaves.

Many recent studies exist on plant classification and recognition based on plant components such as flowers, leaves, and barks. To handle such volumes of information, realization of a quick and efficient classification method has become an area of active study [14-16]. In particular, it is well known that the correct way to extract plant features involves plant recognition based on leaf images. Two features, which are widely used for plant recognition based on leaf image, are color and shape [14], [15-16]. In the color-based conventional study, a simple color similarity between two images can be measured by comparing their color histogram. Also in the shape based-conventional study, they used region and contour-based simple features and features could be considered time domain data.

However, the recognition performance was limited due to leaf color was affected by theseasons and there is a problem that user to directly specify both ends of the leaves.

1.2 plants

Plants are living things that are made up of cells. They need air, water, soil, and sunlight to live. They cannot move from place to place, but their leaves move to catch the sun and their roots move to-wards water. Their seeds can be carried by animals or blown by the wind.

We get food from all different parts of the plant: flowers, fruits, vegeta-bles, seeds, nuts, stems, and leaves. Grass gives us a cool, soft place to walk. Some plants give us medicine, and trees are used to make paper and furniture.

Over 270,000 species of plants have been identified and classified, but scientists believe that there are millions more waiting to be discovered. [1]

1.2.1 The plant kingdom

The Plant Kingdom is a way to classify (or organize) plants. They are divided into groups based on the traits they have in common. Scientists change the way plants are classified from time to time, when they discover new types of plants or learn new things about plants.

The two main groups are vascular plants (plants that use stems and veins to transport food and water), and non-vascular plants (plants with no roots, stems, or leaves).

Vascular plants can be divided into smaller groups, one of which is seed plants. This group includes flowering and non-flowering plants. Flowering plants include monocots (one seed leaf) and dicots (two seed leaves). The non-flowering plants can also be divided into several groups, including cycads, conifers, and ginkgo.[1]

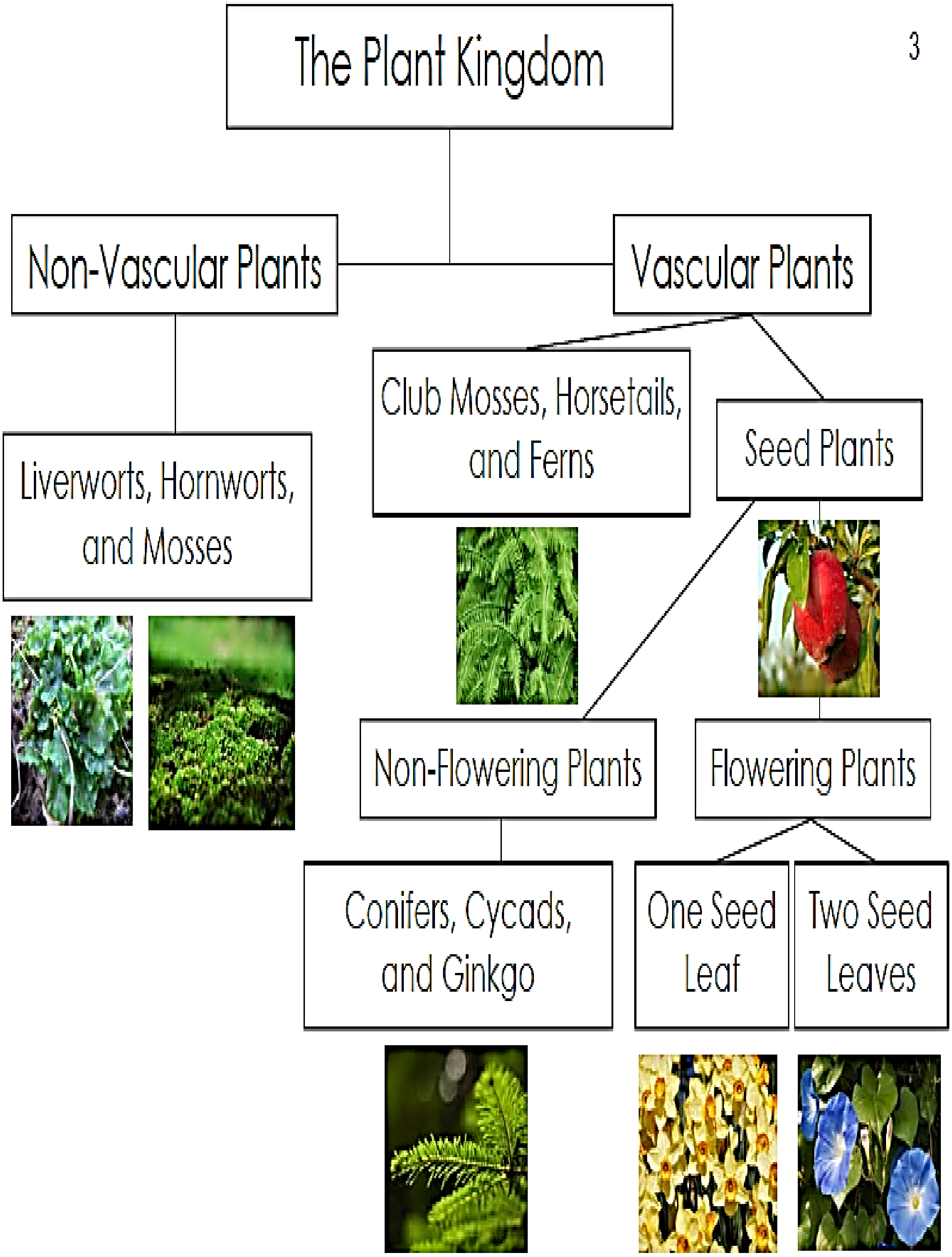


Figure1.1 :The plant kingdom

1.2.2 Trees

There are two different types of trees: non-flowering trees that have seeds that are not en closed, and flowering trees that have seeds that are enclosed. An example of a non-flowering tree would be a pine tree. An example of a flowering tree would be a fruit tree, such as peach or orange. [1]

Flowering trees are deciduous; that is, they shed their leaves every year. Other trees are conifers; they grow new leaves before shed-ding old ones, and stay green all year round ("evergreen").

Figure1.2: Trees

Trees consist of roots, trunk (stem), branch-es, twigs, and leaves. The tallest trees in the world are the redwoods of California, which can grow to be 379 ft (115.55 m) in height.

1.2.3 Leaves

Leaves are the primary food-producing organs of a plant. They are designed to be efficient in collecting light and using that light energy to produce food. [1]

1.2.3.1 Parts of Leaves

The main light-collecting structure on a leaf is a large, broad, flat surface called the leaf blade. The blade has many layers that not only help the plant move but also help it store materials and byproducts of photosynthesis. The blade is held away from the stem and supported by the petiole. The petiole is not exactly like a stem, but it does have xylem and phloem that transport water and sugar. The blade is supported by a system of veins that also contain both xylem and phloem. These veins prevent the blade from collapsing under its own weight. A leaf is often organized with one main vein running down the middle of the blade. This vein is called the midrib. All of the veins, the petiole, and the midrib help position the blade so that it is facing the light source. [1]

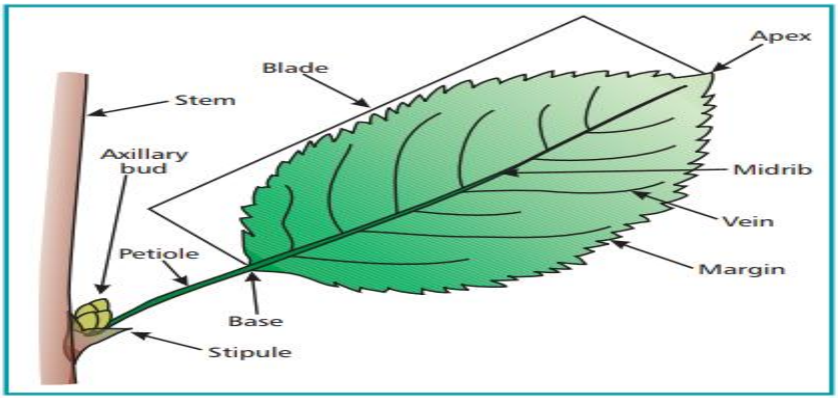


Figure1.3: The major parts of a leaf

1.2.3.2 Veins

Veins of flowering plants are found in several patterns. Monocots and dicots have differing patterns. Monocots have leaves with parallel veins. While the veins may not be parallel in a strict mathematical sense, none of the veins on a monocot leaf cross. They may look as if they are fused together at the top or bottom of the blade. Corn and grass plants are good examples of monocots. [1]

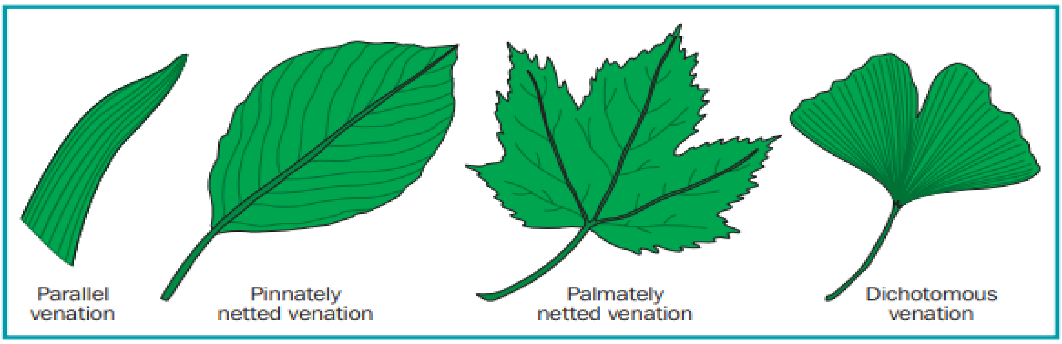


Figure1.4 : Leaf venation patterns

Dicots have leaves with veins that connect and branch from each other. Veins in a branching pattern are called netted veins. A leaf with netted veins sometimes has several smaller veins branching out of a dominant midrib, a condition known as pinnately netted. Elms and oaks have pinnate netting. A leaf may sometimes have several dominant veins branching out from the petiole. This condition, known as palmately netted, is common with maples and redbud. A few plants have a spreading vein pattern called dichotomous venation. A gingko leaf has veins of this type. [1]

1.2.3.3 Types Of Leaves

There are many different types of leaves. Some leaves are adapted to hot, dry climates by being able to store water or being smaller. Some leaves have very large blades to collect the maximum light in a shady location. The blades of some leaves are broken into three or more sections. A leaf that has only one blade on its petiole is called a simple leaf. Most plants have simple leaves. A leaf that has multiple blades, or leaflets, is called a compound leaf. There are different kinds of compound leaves. Two common types are the palmately compound leaf and the pinnately compound leaf. A palmately compound leaf has all its leaflets attached to a common point. A pinnately compound leaf has multiple leaflets attached along a rachis, or axis. [1]

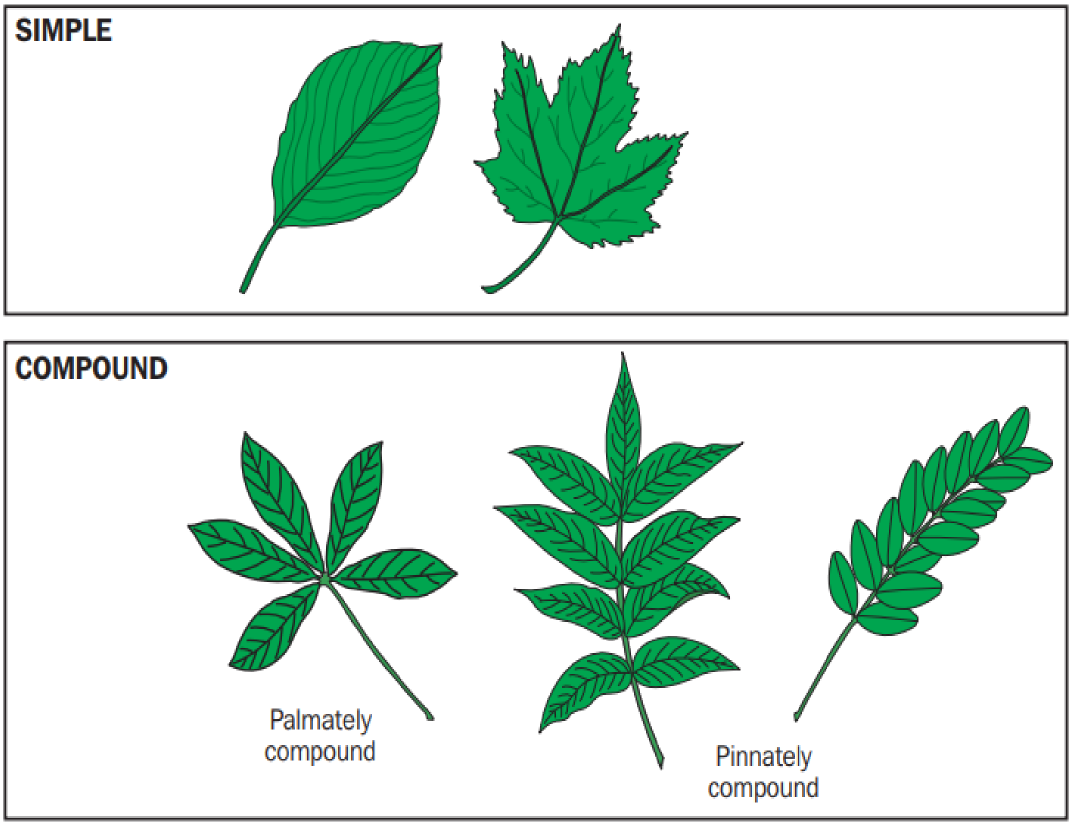


Figure1.5: Types of leaves

1.2.3.4 What is the shape of the leaf?

There are many terms used to describe leaf shape. Only seven of the more common leaf shapes are presented here (Figure 1.6), but guides mentioned in “Additional Resources” provide additional terms and descriptions. Elliptic leaves are broadest in the middle and narrower at either end. Linear leaves are long and narrow with the sides being close to parallel to each other. Lanceolate leaves are much longer than wide, with the widest point below the middle of the leaf. Spatulate leaves look kind of like a spatula, with the tip being rounded and gradually tapering to the base. Ovate leaves are egg-shaped while oval leaves are round to oval, lacking a pointed tip. Cordate leaves are heartshaped. [1]

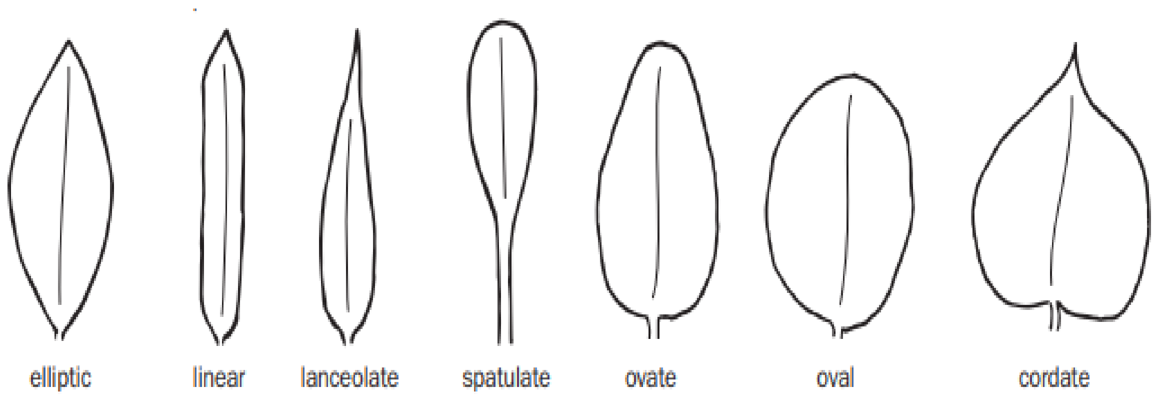


Figure1.6:Leafshapes

1.2.3.5 What other leaf characteristics are important?

The margin, or edge of the leaf, can also assist in plant identification. As with leaf shape, there are many ways to describe leaf margins. Only three major descriptions are provided here (Figure 1.7), but check the “Additional Resources” for more information. Entire margins are smooth and do not have any teeth, notches, or divisions. Leaves that have a toothed or saw-like margin are called dentate or serrate. Lobed leaves have indentations along the margin that cut inward toward the leaf midvein. Another leaf characteristic that can help with plant identification is whether the leaf is petiolate, sessile, or clasping (Figure 1.8). Petiolate leaves have a stalk (petiole) that attaches them to the stem. Sessile leaves do not have a petiole and are attached directly to the stem. Clasping leaves are sessile (i.e. do not have a petiole)and have a base that wholly or partly wraps around the stem. The leaves of Dalmatian toadflax (Linaria dalmatica) provide a good example of clasping leaves. Whitetop (Cardaria spp.) leaves are petiolate on the lower portion of the stem and clasping on the upper portion of the stem. [1]

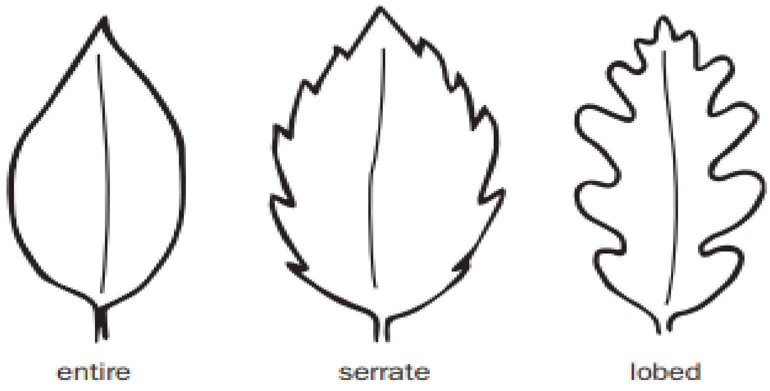


Figure 1.7: Leaf margins

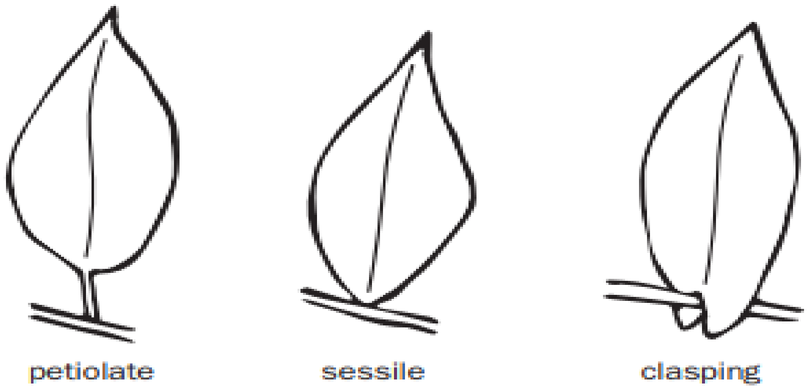


Figure 1.8:Leaf attachments

1.3 Computer Vision

Image recognition is the basic task in the areas of computer vision and pattern recognition. The field of computer vision is concerned with the automated processing of any images from the real world to extract characteristics through computers and interpret information on a real time and user requirements basis. Today, computers are playing key role in our daily lives. Most significant developments in computer hardware and software have contributed to the development of many real world research applications, namely, finger print recognition, digital signature recognition, automatic medical diagnosis and treatment, optical character recognition, document processing, video processing and plant image recognition etc. In addition to these, we also find many commercial applications, namely, commerce, sports, entertainment, business, education, food science, medical science, industrial and automation. An effort has also been carried in forestry, farming, and botany and in many fields that are more allied. The crop growth and health analysis, soil fertility, pest or disease detection, weed detection and post-harvest operations are the most common applications. Ultimately, all the applications exhibit machine intelligence in the respective fields.

Figure 1.9: Medicinal plants parts used as extracts of medicine

Figure 1.10: People searching for the collection of medicinal plants

In recent years, an automatic content extraction, analysis and retrieval for medicinal plants based on visual trait studies has become more important than ever before. This is because manual identification and processing of herbal drugs suffers from two major issues, namely, the scarcity of experts and subjectivity arising with the individuals in the field of Ayurveda. Therefore, the Ayurveda practitioners and biologists are in need of efficient computer software to automatically extract and analyze significant content of images of medicinal plants. Hence, in order to supplement the human visual system with machine vision, medicinal plant recognition is considered a research issue.

The general taxonomy provides important information for the identification and classification of medicinal plants. All the information about the plant like, locality of collection, date, morphological characters of plant like leaf and flower morphology, branching, venation, inflorescence, fruit and seed character are obtained. The Ayurveda researchers and botanists require such information for identification. Instead of manual system, a database of medicinal plants is created and retrieval methodologies based on text and visual information becomes useful. However, the design of a machine vision system to recognize, classify and retrieve medicinal plants is a challenging task due to the factors like illumination levels, weather conditions, shadows, occluded regions and cluttered background. Hence, the authors have chosen to help people living in forests, rural areas and those who practice Ayurveda medicines. In the present work, the authors have attempted to design an efficient approach for the identification and classification of images of Indian medicinal plants through digital image processing techniques. Finally, an image search engine for database of medicinal plants is developed for retrieval based on text and visual information. The Figure (1.11) gives the samples of images of medicinal plants with their scientific names. The different leaves with varying shapes and margins with their botanical terminologies.

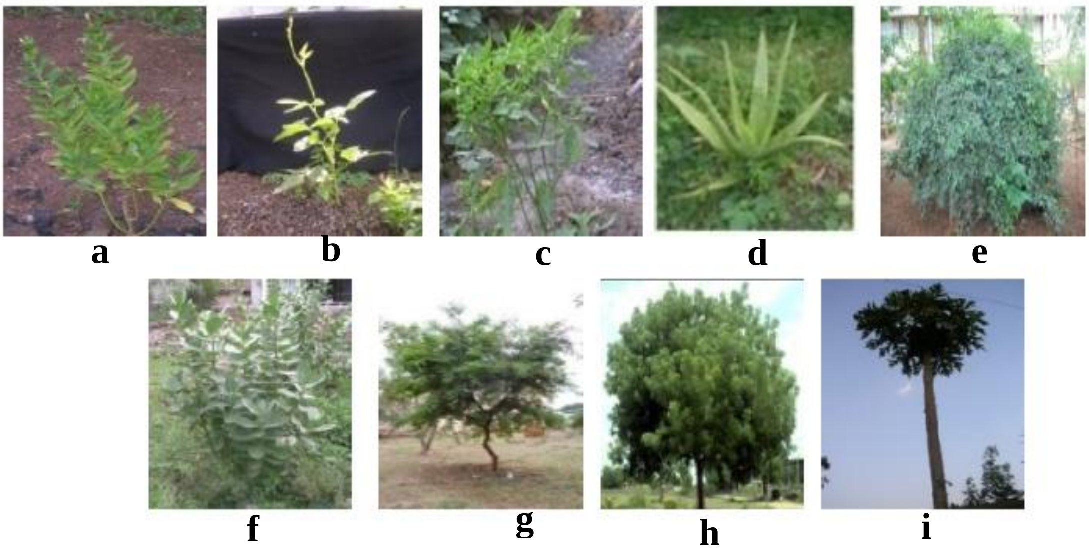


Figure1.11:Samplemedicinalplantimages(a)Catharanthusroseus (b)Vignaunguiculata(c)Capsicumannum(d)Aloebarbadensis (e)Saptachakra(f)Calotropisgigantea(g)Acaciacatechu (h)Azadirachitaindica(i)Caricapapaya

1.4 SystemOverview

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. [13]

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 plants.

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 (1.12).

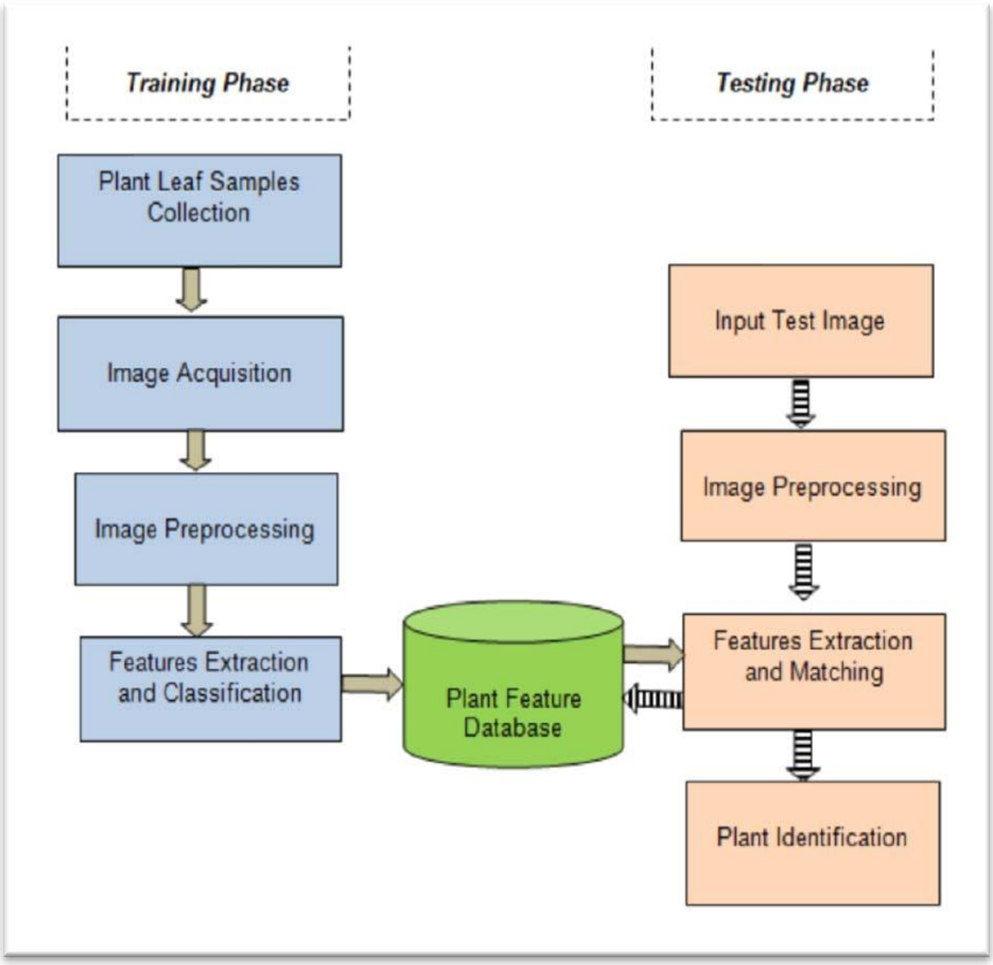


Figure1.12 : Main pattern recognition steps

1.5 Image Preprocessing

Image processing is a recent science that has emerged with the advent of the latest computer graphic technologies. This science aims to process and analyze digital images through a set of techniques and methods, in order to derive information that can be used to answer important questions and solve problems related to a specific field.

Before the operations, some of the leaf images are rotated manually for helping the program to arrange leaf direction to the right side. After wards, automatic preprocessing techniques are applied to all of the leaf images. The preprocessing steps are illustrated.[2]

1.5.1 Definition of real image

Before discussing the scanned image, it is good to talk about the notion of image. In the etymological sense, the word "image", derived from the Latin world "imago", refers to the visual representation of an object by different means or supports (drawing, photography, painting, sculpture…). An image is the projection or analog representation of the actual intensities of a 3D scene on a 2D plane. It is also defined as structured setoff information that, after being displayed on the screen, has meaning for the human eye.

Mathematically, it can be described as a function I (x, y) of continuous analog brightness, defined in a bounded domain, the x and y are the spatial coordinates of a point of the image and I ids the function of light intensity and color. In the aspect, the image is unusable by the machine, which requires it is digitization.

As a definition of the realimages of a leaf in nature. It is a real model distinct in a certain way there are many plant species that have different leaves form and this in fact may distinguish this type from that. The leaf actually has a color that is often green. There are technical means that allow us to study them so the beginning is to take real pictures using the camaraderie of nature.

Figure 1.13: real image of leaves

1.5.2 Definition of leaf image on gray:

Most leaves have generally green color, while the color of leaves is changed by season or environmental factors. The color change of leaf image can cause decline of recognition performance or non-recognized problem. The color converting process on input image is the first step for leaf contour extraction, and it can set foundation to improve recognition performance irrelevant to the leaf color change.

Therefore, we convert the input color leaf image to gray scale image as follows:

$$\boldsymbol{Gray = 0.299 * R + 0.587 * G + 0.114 * B} \quad \textbf{(1.1)}$$

The converted gray scale leaf image is converted to a binary image once again. The threshold conversion is performed as follows:

$$\boldsymbol{B(x,y) = \begin{cases} 0 & if \ \ f(x,y) \le T \\ 255 & if \ \ f(x,y) > T \end{cases}} \quad \textbf{(1.2)}$$

Where, *B(x, y)*and *f(x, y)*are the intensity values of the gray scale image and the binary image, respectively, at position *(x, y),* and*T* is the threshold value [10]. Figure (1.14) shows an example of leaf contour extraction.

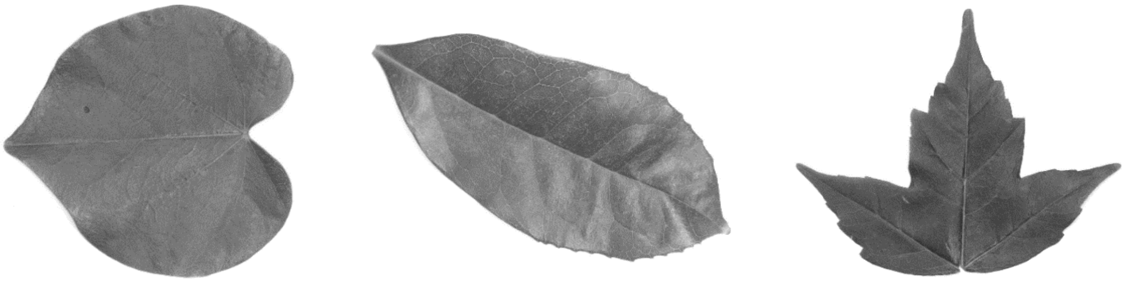


Figure1.14: image on gray

1.5.3 Definition of a digital leaf image

Unlike the leaf image obtained using an analog camera, or drawn on paper, the digital leaf image is stored on a computer in binary form (bit sequence of 0 and 1) after its acquisition. The storage of the scanned image can take place in different compressions formats (jpeg, bmp, png, gif, ...) whose decompression givesan image of the same size as the original image. The latter is represented by a matrix of elements called pixels (term resulting from the contraction of the English words "Picture" and "Element"). The value of each pixel represents a discrete intensity of light.

A digital leaf image is also defined as a discrete plane derived from an analog image after it has been digitized.

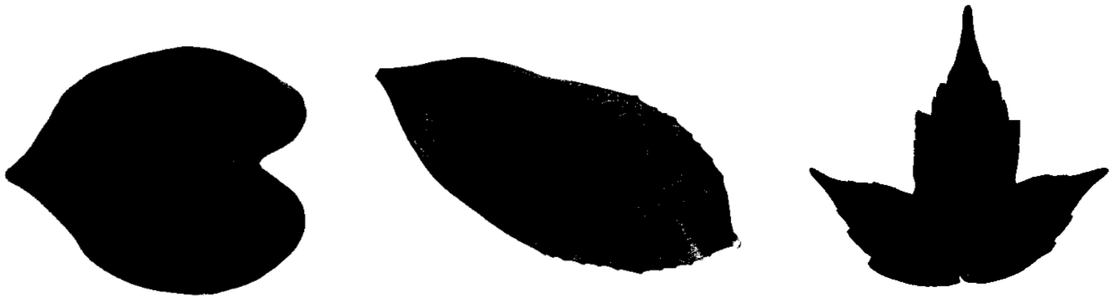


Figure 1.15: Digital leaves image

1.6 Image segmentation Techniques

The main aim of segmentation is simplification i.e. representing an image into meaningful and easily analyzable way. Image segmentation is necessary first step in image analysis. The goal of image segmentation is to divide an image into several parts/segments having similar features or attributes. The basic applications of image segmentation are: Content-based image retrieval, Medical imaging, Object detection and Recognition Tasks, Automatic traffic control systems and Video surveillance, etc. The image segmentation can be classified into two basic types: Local segmentation (concerned with specific part or region of image) and Global segmentation (concerned with segmenting the whole image, consisting of large number of pixels). The image segmentation approaches can be used many methods based on properties of image.

1.6.1 Thresholding Method

Thresholding methods are the simplest methods for image segmentation. These methods divide the image pixels with respect to their intensity level. These methods are used over images having lighter objects than background. The selection of these methods can be manual or automatic i.e. can be based on prior knowledge or information of image features. There are basically three types of thresholding [3] [4]:

Global Thresholding: This is done by using any appropriate threshold value/T. This value of T will be constant for whole image. On the basis of T the output image can be obtained from original image as:

$$\boldsymbol{q(x,y)} = \begin{cases} \mathbf{1} \;\; \boldsymbol{if} \;\; \boldsymbol{q(x,y) > T} \\ \mathbf{0} \;\; \boldsymbol{if} \;\; \boldsymbol{q(x,y) \leq T} \end{cases} \qquad \textbf{(1.3)}$$

1) Variable Thresholding: In this type of thresholding, the value of T can vary over the image. This can further be of two types:
   - Local Threshold: In this the value of T depends upon the neighborhood of x and y.
   - Adaptive Threshold: The value of T is a function of x and y.
2) Multiple Thresholding: In this type of thresholding, there are multiple threshold values like T0 and T1. By using these output image can be computed as:

$$\boldsymbol{q(x,y)} = \begin{cases} \boldsymbol{m \ \ if \ \ q(x,y) > T1} \\ \boldsymbol{n \ \ if \ \ q(x,y) \leq T1} \\ \boldsymbol{0 \ \ if \ \ q(x,y) \leq T0} \end{cases} \qquad \textbf{(1.2)}$$

The values of thresholds can be computed with the help of the peaks of the image histograms. Simple algorithms can also be generated to compute these.

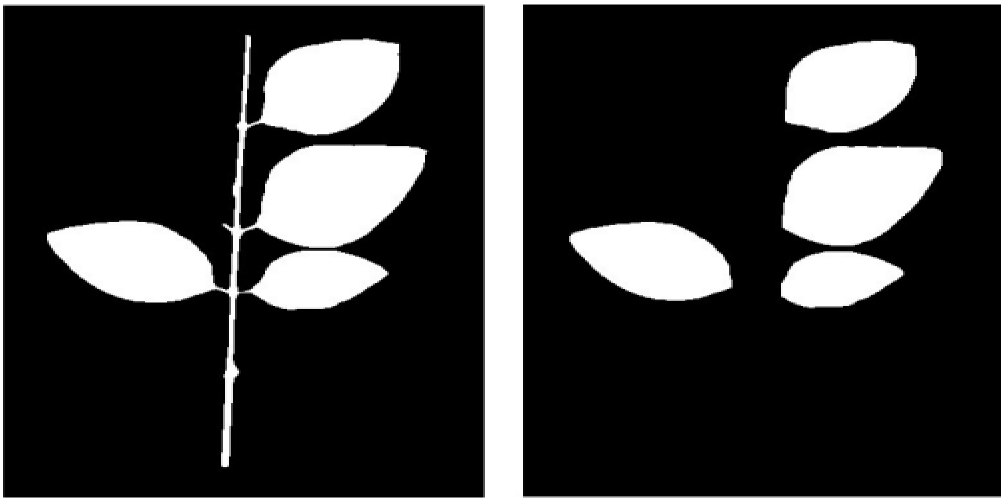


Figure 1.16: Thresholding segmentation Method

1.6.2 Edge Based Segmentation Method

The edge detection techniques are well developed techniques of image processing on their own. The edge based segmentation methods are based on the rapid change of intensity value in an image because a single intensity value does not provide good information about edges. Edge detection techniques locate the edges where either the first derivative of intensity is greater than a particular threshold or the second derivative has zero crossings. In edge based segmentation methods, first of all the edges are detected and then are connected together to form the object boundaries to segment the required regions. The basic two edge based segmentation methods are: Gray histograms and Gradient based methods. To detect the edges one of the basic edge detection techniques like sobel operator, canny operator and Robert's operator …etc can be used. Result of these methods is basically a binary image. These are the structural techniques based on discontinuity detection [5].

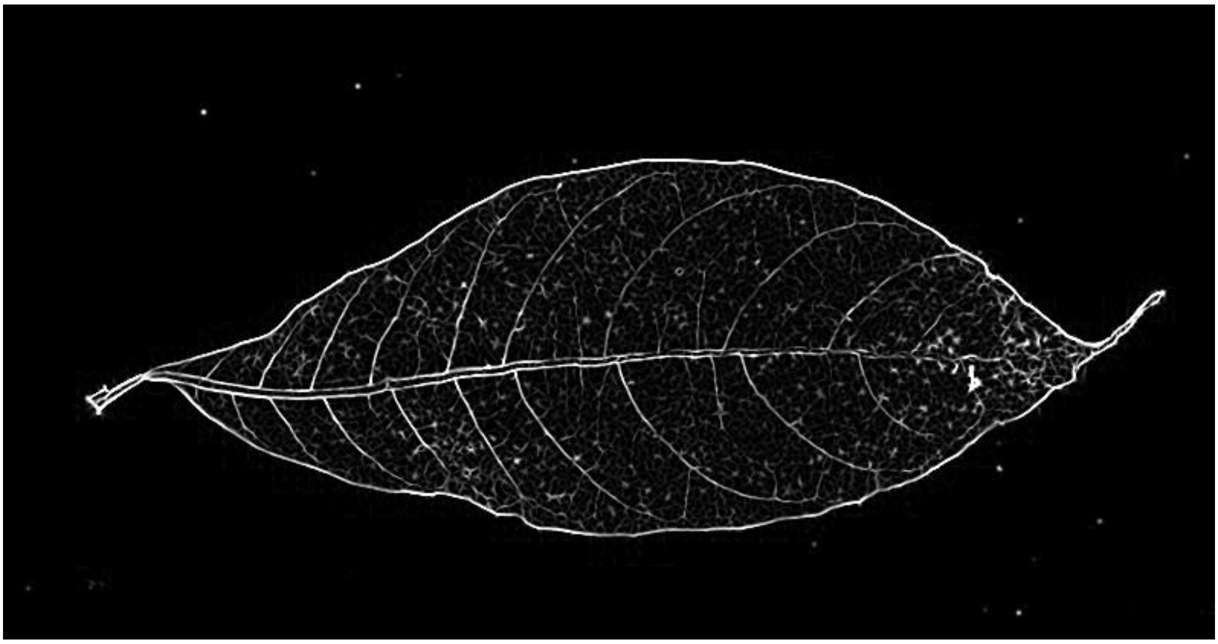


Figure1.17: Edge Based Segmentation Method

1.6.3 Region Based Segmentation Method

The region based segmentation methods are the methods that segments the image into various regions having similar characteristics.[6] [7] [8].

Figure1.18: Region Based Segmentation Method

1.6.4 Clustering Based Segmentation Method

The clustering based techniques are the techniques, which segment the image into clusters having pixels with similar characteristics. Data clustering is the method that divides the data elements into clusters such that elements in same cluster are more similar to each other than others. There are two basic categories of clustering methods: Hierarchical method and Partition based method. The hierarchical methods are based on the concept of trees. In this the root of the tree represents the whole database and the internal nodes represent the clusters. On the other side the partition based methods use optimization methods iteratively to minimize an objective function. In between these two methods there are various algorithms to find clusters. There are basic two types of clustering [9] [10].

1) Hard Clustering: Hard clustering is a simple clustering technique that divides the image into set of clusters such that one pixel can only belong to only one cluster. In other words, it can be said that each pixel can belong to exactly one cluster.

These methods use membership functions having values either 1 or 0 i.e. one either certain pixel can belong to particular cluster or not. An example of a hard clustering based technique is one k-means clustering based technique known as HCM.

In this technique, first of all the centers are computed then each pixel is assigned to nearest center. It emphasizes on maximizing the intra cluster similarity and also minimizing the inter cluster equality.

2) Soft clustering: The soft clustering is more natural type of clustering because in real life exact division is not possible due to the presence of noise. Thus soft clustering techniques are most useful for image segmentation in which division is not strict. The example of such type of technique is fuzzy c-means clustering. In this technique pixels are partitioned into clusters based on partial membership i.e. one pixel can belong to more than one clusters and this degree of belonging is described by membership values. This technique is more flexible than other techniques [9].

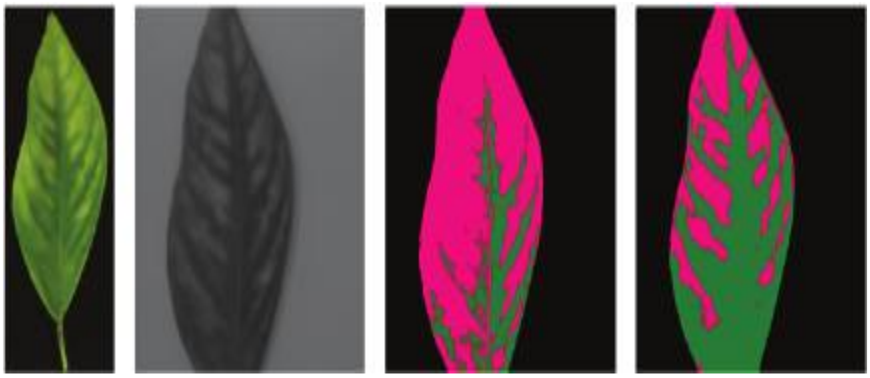


Figure1.19: Clustering Based Segmentation Method

1.6.5 Leaf segmentation challenges

In our case, the similarity between the background and the object of in-terest, and the difficulty to avoid adjacent and overlapping leaves constitute a prohibitive obstacle to the use of unconstrained active contours (Figure 1.20). The idea of using a template to represent the leaves is complicated by the fact that there is much more variety in shapes than for eyes or mouths. The only solution to overcome the aforementioned problems is however to take advantage of the prior knowledge we may have on leaf shapes to design a very flexible time-efficient model to represent leaves.

Figure 1.20: Segmentation issues with unconstrained region-based active contours

1.6.6 Comparison Of Various Segmentation Techniques

Table 1.1 shows a comparison between various segmentation techniques by specifying a brief description of every method each with its advantages and disadvantages[11]

Segmentation technique	Description	Advantages	Disadvantages
Thresholding Method	Based on the histogram peaks of the image to find particular threshold values.	No need of previous information, simplest method.	Highly dependent on peaks, spatial details are not considered.
Edge Based Method	Based on discontinuity detection.	Good for images having better contrast between objects.	Not suitable for wrong detected ortoo many edges.
Region Based Method	Based on partitioning image into homogeneous regions.	More immune to noise, useful when it is easy to define similarity criteria.	Expensive method in terms of time and memory.
Clustering Method	Based on division into homogeneous clusters.	Fuzzy uses partial membership therefore more useful for real problems.	Determining membership function is not easy.

Table1.1: comparison between various segmentation technique.

1.7 Plant Features

Human beings identify the plants by their stems, leaves, flowers and fruits. These become the features in their automatic recognition and differ from one plant to the other. Medicinal plants are not the exceptions. The classification of medicinal plants depends upon features uniqueness of the plant. The Scientists and Ayurveda practitioners and plant experts have collected data on innumerable plant species by studying the plants in their natural habitats and recorded information on their characteristics in scientific literature and databases for future reference. Consequently, taxonomists have defined some nomenclature and botanical terminologies through which plant and its parts are recognized. But, a machine vision system for recognition of medicinal plants requires capture of these features through either division of new approach for feature extraction or adoption of existing methods in the literature.

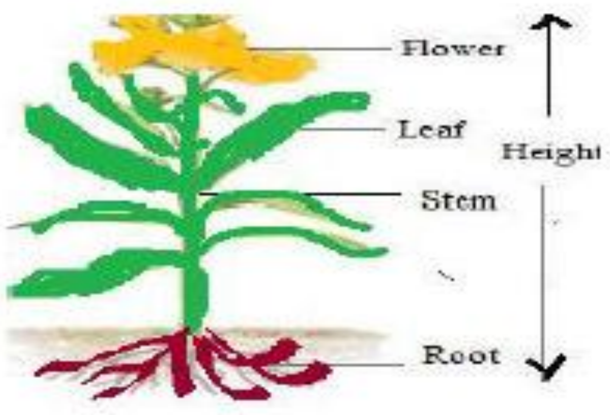


Figure1.21: features of plant

1.7.1 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.[13]

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 Ratioof Major Axis Length: This feature is denoted as ratio of minor axis length to major axis length. It isreverse of the aspect ratio that is used in the literature…etc.

1.7.2 Shape Features

We used the morphological features in [12] as the shape features, which are common shape features used in the literature. There are five fundamental features: the longest distance between any two points on a leaf border (L), the length of main vein (lengthwise- Lυ), the widest distance of a leaf (crosswise-W), the leaf area (A) and the leaf perimeter (P). Then, twelve features are constructed using these five fundamental features by some mathematical operations:

–smoothness of a leaf image

–aspect ratio (Lυ/W)

–form factor, the difference between a leaf and a circle (4 π A / P^2)

–rectangularity (Lυ W / A)

–narrow factor (L/ Lυ)

–ratio of perimeter to longest distance (P / L)

–ratio of perimeter to the sum of main vein length and widest distance(P /(Lυ + W))

–and five structural features obtained by applying morphological opening ongrayscale image.

1.7.3 Color Features

In our experiments, we realized that some leaf images in Flavia dataset [12] have very similar shapes. Thus classification accuracy is greatly affected by this simi-larity. However, even shapes are similar i n some leaves, there are some differences in colors of leaves. Therefore, in addition to the shape features, we extracted color based features from leaf images. When calculating these features, we eliminated background color of leaf images.

We defined two sets of color features. In the first set, we used mean (μ) and standard deviation (σ) of intensity values of red, green, blue channels and average of these channels. So that, the first set of color features contain eight features.

Mean and standard deviation of each component are calculated as follows:

$$\mu = \frac{1}{MN}\sum_{x=1}^{M} \sum_{y=1}^{N} Pxy \tag{1.5}$$

$$\sigma = \sqrt{\frac{1}{MN}\sum_{x=1}^{M} \sum_{y=1}^{N}(Pxy - \mu)^2} \tag{1.6}$$

In (1) and (2), M and N are dimensions of an image and Pxy is the intensity value of pixel at (x, y) coordinate.

The second set of color features consists of color histograms in red, green, and blue channels. RGB histograms provide us an efficient representation of color distribution. Thus, we are able to effectively analyze color information in an image by using these histograms. After studying several bin sizes, we obtained the best results with 10 bins in each histogram. Since we are calculating three histograms for red, green and blue channels, there are thirty new features in the second set of color features.

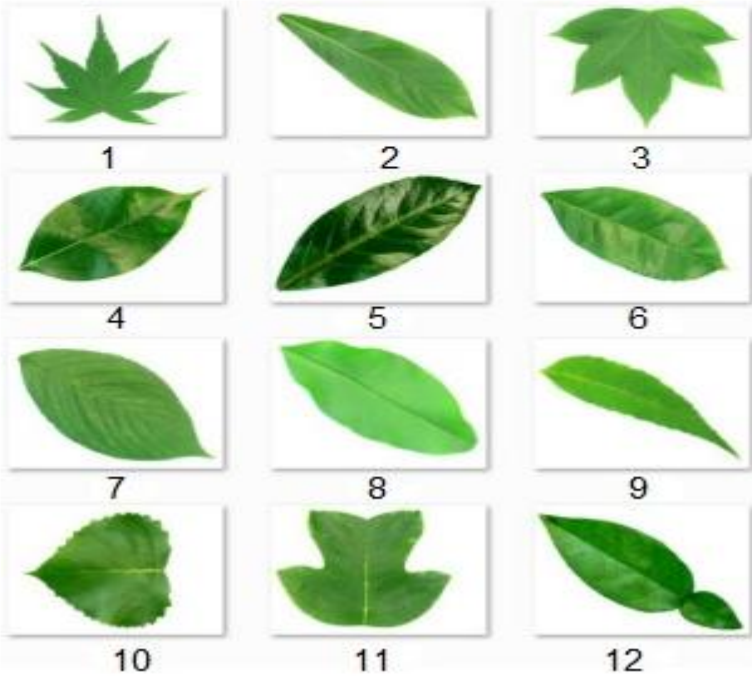


Figure1.22: Plant Recognition Approach Using deferent Shape and graduate Color Features

1.7.4 Texture Features

Guiying Li (2012) defined texture is a repeated pattern of information or arrangement of the structure with regular intervals. In a general sense, texture refers to surface characteristics and appearance of an object given by the size, shape, density, arrangement, proportion of its elementary parts. A basic stage to collect such features through texture analysis process is called as texture feature extraction. Due to the signification of texture information, texture feature extraction is a key function in various image processing applications like remote sensing, medical imaging and content-based image retrieval.

There are four major application domains related to texture analysis namely texture classification, segmentation, synthesis and shape from texture.

Texture classification produces a classified output of the input image where each texture region is identified with the texture class it belongs.

Texture segmentation makes a partition of an image into a set of disjoint regions based on texture properties, so that each region is homogeneous with respect to certain texture characteristics.

Texture synthesis is a common technique to create large textures from usually small texture samples, for the use of texture mapping in surface or scene rendering applications.

The shape from texture reconstructs three dimensional surface geometry from texture information. For all these techniques, texture extraction is an inevitable stage.

Figure1.23: texture of leaves

1.7.4.1 Texture Feature Extraction

Neville et al (2003) discussed texture features can be extracted using several methods such as statistical, structural, model based and transform information.

1.7.4.1.1 Structural based Feature Extraction

Structural approaches represent texture by well-defined primitives and a hierarchy of spatial arrangements of those primitives. The description of the texture needs the primitive definition. The advantage of the structural method based feature extraction is that it provides a good symbolic description of the image; however, this feature is more useful for image synthesis than analysis tasks. This method is not appropriate for natural textures because of the variability of micro-texture and macro-texture.

1.7.4.1.2 Statistical based Feature Extraction

Statistical methods characterize the texture indirectly according to the non-deterministic properties that manage the relationships between the gray levels of an image. Statistical methods are used to analyze the spatial distribution of gray values by computing local features at each point in the image and deriving a set of statistics from the distributions of the local features. The statistical methods can be classified into first order (one pixel), second order (pair of pixels) and higher order (three or more pixels) statistics.

The first order statistics estimate properties (e.g. average and variance) of individual pixel values by waiving the spatial interaction between image pixels. The second order and higher order statistics estimate properties of two or more pixel values occurring at specific locations relative to each other. The most popular second order statistical features for texture analysis are derived from the co-occurrence matrix.

1.7.4.1.3 Model based Feature Extraction

Model based texture analysis such as fractal model and Markov model are based on the structure of an image that can be used for describing texture and synthesizing it. These methods describe an image as a probability model or as a linear combination of a set of basic functions. The Fractal model is useful for modeling certain natural textures that have a statistical quality of roughness at different scales and self-similarity, and also for texture analysis and discrimination.

There are different types of models based feature extraction technique depending on the neighbor-hood system and noise sources. The different types are one-dimensional time-series models, Auto Regressive (AR), Moving Average (MA) and Auto Regressive Moving Average (ARMA). Random field models analyze spatial variations in two dimensions.

Global random field models treat the entire image as a realization of a random field, and local random field models assume relationships of intensities in small neighbor-hoods. Widely used class of local random field models are Markov models, where the conditional probability of the intensity of a given pixel depends only on the intensities of the pixels in its neighbor-hood.

1.7.4.1.4 Transform based Feature Extraction

Transform methods, such as Fourier, Gabor and Wavelet Transforms represent an image in space whose co-ordinate system has an interpretation that is closely related to the characteristics of a texture. Methods based on Fourier transforms have a weakness in a spatial localization so these do not perform well. Gabor filters provide means for better spatial localization but their usefulness is limited in practice because there is usually no single filter resolution where one can localize a spatial structure in natural textures. These methods involve transforming original images by using filters and calculating the energy of the transformed images. These are based on the process of the whole image that is not good for some applications which are based on one part of the input image.

1.8 Classification (Recognition)

Once the features have been extracted, and 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, and 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.[13]

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 eachclass of the training data. Recognition:

    1. Pre-processing

    2. Feature extraction: (both steps are same as above)

    3. Classification: Compare feature vectors tothe 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.

1.9 General Literature Survey

The authors have carried out literature survey to know the state of the art applications of computer vision and digital image processing techniques in the real world, more specifically connected with plant recognition. Following is the gist of the works cited in the literature.

Abdul kadir et. al., have proposed a method using combined features such as, polar Fourier transform, color moments and vein features to retrieve images of leaves. The method is very useful for the recognition of foliage plants. The system has been tested on Flavia andFolia leaf data sets. The retrieval accuracy of 93.13% and 90.13% are observed for Flavia and Folia data sets respectively.[42]

Mahmood R. Golzarian and Ross A. Frick have developed a method for classification of images of three grasses, namely, wheat, ryegrass and brome grass species at early growth stages. A combination of color, texture and shape features is used. The features are reduced to three descriptors using Principal Component Analysis. Three components are able to distinguish three grasses with a classification accuracy of 85% and 88% for ryegrass and brome grass respectively. The study helps for weed management .[43]

Mahmood R. Golzarian has investigated an adaptive learning for segmentation of plant images into plant and non-plant regions. The Kohenen's self-organizing map (SOM) algorithm is deployed for segmentation of plant images. Nine color features of three color space models are used as features. The method worked well even in the presence of noise. [43]

Faisal Ahmed et. al., have investigated the use of Support Vector Machine (SVM) and Bayesian classifier as machine learning algorithms for the effective classification of crops and weeds in digital images. From the performance comparison, it is reported that SVM classifier has outperformed Bayesian classifier. Young plants that did not mutually overlap with other plants are used in the study .[44]

B. Sathyabama et al.,has presented Content Based Leaf Image Retrieval (CBLIR) for ecommerce application. The Log-Gabor wavelet and Scale Invariant Feature Transform (SIFT) are deployed for leaf image texture and shape features respectively. The retrieval accuracyof 97.5% is observed .[45]

Ulrich Weiss et. al., have contributed a method for recognition of plant species by robots for weed detection and nursery plant detection. The plant species are distinguished using 3D

LIDAR sensor and supervised learning. Different learning methods like, logistic regression functions, Support Vector Machines and Neural Networks are found to be suitable. The average classification accuracy on six types of plant species, twenty images of each class is found to be 98% .[46]

Yovel Y et. al. have presented a new method for Plant classification from bat-like echolocation signals. In this work, a plant is considered as a three-dimensional array of leaves emitted bat call. The plants are classified based on signals from a database of plant echoes that are created by plants with a frequency-modulated bat-like ultrasonic pulse. The algorithm uses the spectrogram of a single echo from which it uses features that are accessible to bats. The Support Vector Machine (SVM) learning is used to automatically extract suitable linear combinations of time and frequency cues from the spectrograms. The classification is reported to be high.[47]

1.10 Conclusion

Leaves are the primary food-producing organs of a plant. The main light-collecting structure on a leaf is a large, broad, flat surface called the leaf blade. The blade is held away from the stem and supported by the petiole.

A leaf that has only one blade on its petiole is called a simple leaf. A leaf that has multiple blades is called a compound leaf. Two common types of compound leaves are the palmately compound leaf and the pinnately compound leaf.

Leaves are arranged along a stem in one of four major ways. They may be opposite, alternate, subopposite, or whorled.

The literature survey has reveals that the computers are used in many automation tasks related to plant domain. Still there are enormous applications connected to plant image recognition. We have observed plantidentification and classification in various fields such as, agriculture, weed classification, plant growth analysis, horticulture, forestry biomass prediction and vegetable recognition. Some works are also reported on plant species recognition based on their parts such as leaves, flowers and bark. But, the plant identification with respect to full image is very much scarce. Hence, from the literature survey, to the best of our knowledge, it has been observed that no considerable work has been cited in the literature on the development of machine vision systems for identification, classification and retrieval of medicinal plants images in the Indian context.

The automatic identification of medicinal plants with their relevant information such as plant names in different languages and medicinal usage is very much necessary in the present era. The methodologies help the development of a search engine for the database of Ayurveda medicinal plants in information retrieval through plant snap. This is the motivation for taking up this work.

CHAPTER 2
GFD and SVM

2.1 Introduction

Large-scale data storage is feasible due to the large capacity and low cost of hard drives; huge image databases are becoming more prevalent. We need to be able to search these databases with a textual description of the image we desire in order for a large database to have any use. Manually entering this searchable information (metadata) is tedious and impractical when the number of images is large. One way to quickly extract and assign information contained in images is using Fourier descriptors to recognize shapes.

Fourier descriptors are a classical method to shape recognition and they have grown into a general method to encode various shape signatures. Previous experiments have used Fourier descriptors to recognize different types of marine life, product deformations, and tree leaves. I chose to implement a tree leaf identification program using Matlab because of a personal interest in nature and a database of leaf images is easy to create. There are many unique leaf shapes such as the oak tree leaf and the maple tree leaf. These leaves are easily recognized by anyone who has grown up around these trees. However, there are also more subtle differences between leaf shapes. A red maple leaf has notched lobes; a sugar maple leaf has smooth lobes. Because a good method for shape recognition needs to detect these subtleties, leaf shapes are a great example to test the limits of Fourier descriptor methods.

2.2 Generic Fourier Descriptor

2.2.1 One Dimensional Fourier Descriptor

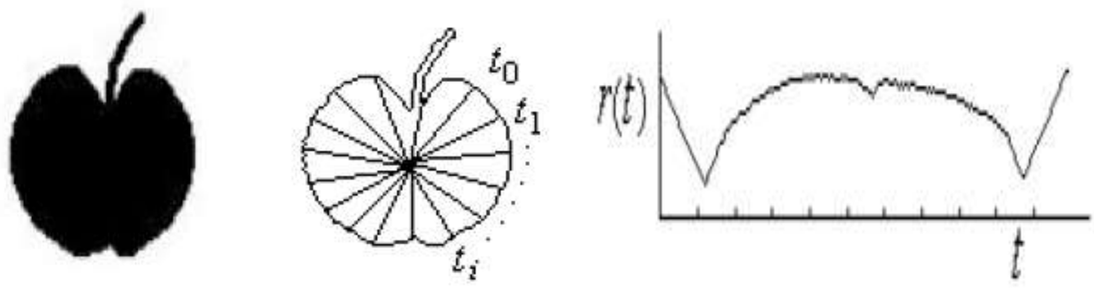
One dimensional FD has been successfully applied to many shape representation applications, especially to character recognition. The nice characteristics of FD, such as simple derivation, simple normalization, simple to do matching, robust to noise, perceptually meaningful, compact and hierarchical coarse to fine representation, make it a popular shape descriptor [[17],[18],[19],[20],[21], [22], [23],[24],[25]]. Generally, 1-D FD is obtained through Fourier transform (FT) on a shape signature function derived from shape boundary coordinates {(x(t), y(t)), t=0, 1, …, N-1}. A typical shape signature function is the centroid distance function which is given by the distance of the boundary points from the centroid (x_c, y_c) of the shape

$$\mathbf{r(t) = ([x(t) - x_c]^2 + [y(t) - y_c]^2)^{\frac{1}{2}}, \quad t = 0,1,\ldots, N\text{-}1 \quad (2.1)}$$

where

$$x_c = \frac{1}{N}\sum_{t=0}^{N-1} x(t) \; y_c = \frac{1}{N}\sum_{t=0}^{N-1} y(t) \qquad \textbf{(2.2)}$$

An example of centroid distance function of an apple shape is shown in Figure 2.1



(a) (b) (c)

Figure 2.1: (a) An apple shape; (b) the contour of (a); (c) centroid distance function of (a)

One dimensional FT is then applied on r(t) to obtain the Fourier transformed coefficients

$$a_n = \frac{1}{N}\sum_{t=0}^{N-1} r(t)\exp\left(\frac{-2j\pi nt}{N}\right), \qquad n = 0, 1, \ldots, N-1 \qquad \textbf{(2.3)}$$

The magnitudes of the coefficients an (n=0, 1… N-1) normalized by the magnitude of the first coefficient a0 are used as shape descriptors, called Fourier descriptors. The acquired FDs are translation, rotation and scale invariant. It has been shown that shape representation using Fourier descriptor (FD) outperforms many other contour shapes descriptors [22], [26]. However, all these methods assume the knowledge of shape boundary information which may not be available in general situations. For example, it is difficult to derive 1-D FD for the shape in Figure 2.2 (a), because the contour of the shape is not available. Furthermore, 1-D FD cannot capture shape interior content which is important for shape discrimination. For example, FD is not able to discriminate the shape in Figure 2.2(b) from the shape in Figure 2.2(c). The drawbacks limit the application of 1-D FD.

(a) (b) (c)

Figure 2.2: (a) A shape without contour; (b)(c) two shapes with same contour but with different interior content

2.2.2 Polar Fourier Transform

Fourier transform has been widely used for image processing and analysis. The advantage of analyzing image in spectral domain over analyzing shape in spatial domain is that it is easy to overcome the noise problem which is common to digital images. Besides, the spectral features of an image are usually more concise than the features extracted from spatial domain. One dimensional FT has been successfully applied to contour shape (which is usually represented by a shape signature derived from the shape boundary coordinates) to derive FD. The application of one dimensional FT on shape assumes the knowledge of shape boundary information. There is no reported work on region based FD. In this section we introduce generic FD derived from 2-D PFT.

The continuous and discrete 2-D Fourier transform of a shape image f (x, y) ($0 \le x < M$, $0 \le y < N$) are given by (2.4) and (2.5) respectively.

$$\boldsymbol{F(u,v) = \int_x \int_y f(x,y) \exp[-j2\pi(ux+vy)]\, dxdy} \qquad \textbf{(2.4)}$$

$$\boldsymbol{F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \exp[-j2\pi(ux/M + uy/N)]} \qquad \textbf{(2.5)}$$

The u and v in (2.6) are the u-th and v-th spatial frequency in horizontal and vertical direction respectively. 2-D FT can be directly applied to any shape image without assuming the knowledge of boundary information. However, direct applying 2-D FT on a shape image in Cartesian space to derive FD is not practical because the features captured by 2-D FT are not rotation invariant. Rotation invariance of a shape is important because similar shapes can be under different orientations. For example, the two patterns (shapes) in Figure 2.3(a) and (b) are similar patterns (shapes), however, their Fourier spectra distributions (Figure 2.3(c) and (d)) on frequency plane are different. The difference of feature distributions makes it impractical to match the two patterns, especially online.

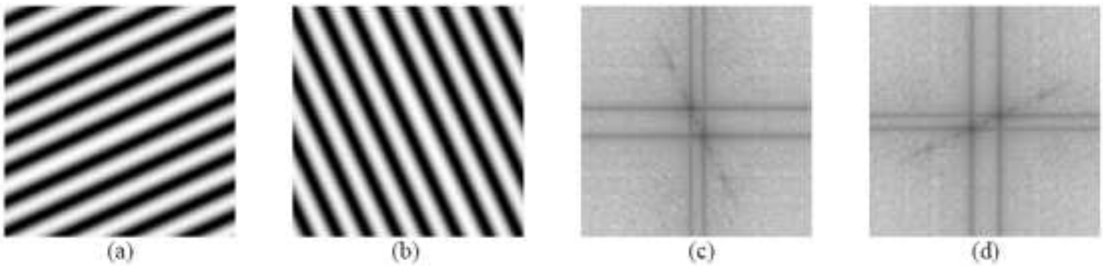


Figure 2.3 : (a) a pattern; (b) pattern (a) rotated by 90 degrees; (c) Fourier spectra of (a); (d) Fourier spectra of (b).

Therefore, we consider shape image in polar space and applying polar Fourier transform (PFT) on shape image. The PFT produces rotation-invariant data particularly well-suited for accurate extraction of orientation features. In the following, we study and describe two PFTs. The study is necessary, because theoretically sound method may not readily applicable for implementation.

To derive PFT, both the data f(x, y) and the spectra F(u, v) are put into polar space, that is, let

$$\mathbf{x = r.cos\boldsymbol{\theta}, \quad x = r.sin\boldsymbol{\theta}} \qquad \textbf{(2.6)}$$

$$\mathbf{u = \boldsymbol{\rho}.cos\boldsymbol{\psi}, \quad v= x = \boldsymbol{\rho}.sin\boldsymbol{\psi}} \qquad \textbf{(2.7)}$$

(r, θ) is the polar coordinates in image plane and (ρ,ψ) is the polar coordinates in frequency plane. The definition of (r, θ) and (ρ,ψ) is the same as that in (2.1). The differentials of x and y are:

$$\mathbf{Dx = cos\boldsymbol{\theta}\ dr + rsin\boldsymbol{\theta d\theta}} \qquad \textbf{(2.8)}$$

$$\boldsymbol{Dy = sin\ \theta\ dr + rsin\ \theta\ d\theta} \qquad \textbf{(2.9)}$$

The Jacobian of (2.8) is *r*. By replacing (2.6) and (2.8) into (2.4) we have the polar Fourier transform (PFT1):

$$\boldsymbol{PF_1(\rho,\psi) = \int_r \int_\theta rf(r,\theta)}\mathbf{exp}\boldsymbol{[-j2\pi r\rho}\mathbf{sin}\boldsymbol{(\theta+\psi)]drd\theta}\textbf{(2.10)}$$

The discrete PFT1 is then obtained as

$$\boldsymbol{PF_1(\rho_l, \psi_m) = \sum_p \sum_i f(r_p, \theta_i).r_p.\exp[-j2\pi r_p \rho_l \sin(\theta_i + \psi_m)]} \quad (2.11)$$

Wherer_p = p/R, Θ_i = i(2π/T) (0≤ i<T); ρ_lρl = l (0≤ l<R) and ψ_m = mΘ_i. R and T are the resolution of radial frequencyand angular frequency respectively. The acquired polar Fourier coefficients F(ρ, ψ) are used to derive normalized FD for shape representation.

PFT1 is the direct result from the polar transform of (2.5). However, due to the presence of ψ_m within sin function sin(Θ_i +ψ_m), the physical meaning of $\psi_m$is not the m-th angular frequency. The features captured by the PFT1 lost physical meaning in circular direction. To overcome the problem, a modified polar FT (PFT2) is derived by treating the polar image in polar space as a normal two-dimensional rectangular image in Cartesian space. Figure 2.4 demonstrate the rectangular polar image. Figure 2.4(a) is the original shape image in polar space, Figure 2.4(b) is the rectangular polar image plotted into Cartesian space.

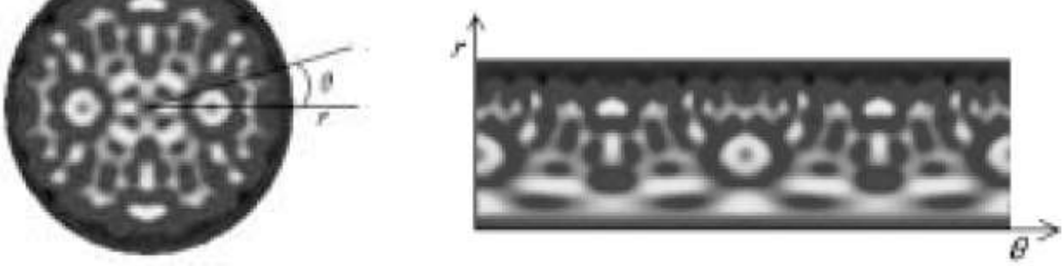


Figure 2.4: (a) original shape image in polar space; (b) polar image of (a) plotted into Cartesian space.

The polar image of Figure 4(b) is the normal rectangular image. Therefore, if we apply 2-D FT on this rectangular image, the polar FT has the similar form to the normal 2-D discrete FT of (2.6) in Cartesian space. Consequently, the modified polar FT is obtained as

$$\boldsymbol{PF_2(\rho, \emptyset) = \sum_r \sum_i f(r, \theta_i)\exp[j2\pi(\rho\frac{r}{R} + \frac{2\pi i}{T}\emptyset)]} \quad (2.12)$$

where 0≤ r <R and θ_i = i(2π/T) (0≤ i <T); 0≤ρ<R, 0≤φ<T. R and T are the radial and angular resolutions. PFT2 has a simpler form than ZMD and PFT1. There is no need to constrain the shape into a unit circle (the constraint requires a extra scale normalization in spatial domain) as required in the implementation of ZMD (because Zernike moment is defined in a unit circle). And the physical meaning of ρ and φ is similar to u and v in (2.6).

The ρ and φ are simply the number of radial frequencies selected and the number of angular frequencies selected. The determination of ρ and φ is physically achievable, because shape features are usually captured by the few low frequencies.

Figure 2.5(a)(b) shows the polar images of the two patterns in Figure 2.3(a)(b) and their polar Fourier spectra are shown in Figure (c) and (d). It can be observed from Figure 5 that rotation of pattern in Cartesian space results in circular shift in polar space. The circular shift does not change the spectra distribution on polar space. This is demonstrated in Figure 2.5(c) and (d). The polar Fourier spectra is more concentrated around the origin of the polar space. This is particularly well-suited for shape representation, because for efficient shape representation, the number of spectra features selected to describe the shape should not be large. Since f(x, y) is a real function, the spectra is circular symmetric, only one quarter of the spectra features are needed to describe the shape. The acquired polar Fourier coefficients F(ρ, φ) are used to derive normalized FD for shape representation.

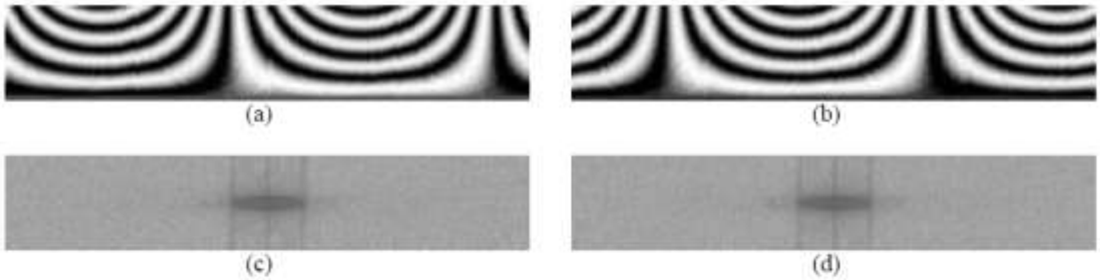


Figure 2.2: (a)(b) polar images of the two patterns in Figure 3(a) and (b); (c) Fourier spectra of (a); (d) Fourier spectra of (b).

2.2.3 Derivation of Generic FD

In this section, the derivation of FD using the above described VZM and PFT is given in details. The VZM and the two polar FTs: PFT1 and PFT2 are all implemented in the experiments to derive FD in the purpose to determine which is the most appropriate for shape retrieval.

Given a shape image I = {f(x, y); 0≤x<M, 0≤y<N}. To apply VZM and PFT, the shape image is converted from Cartesian space to polar space I_p = {f(r, θ); 0≤r<R, 0≤θ<2π}, R is the maximum radius of the shape. The origin of the polar space is set to be the centroid of the shape, so that the shape is translation invariant. The centroid (x_c, y_c) is given by

$$x_c = \frac{1}{M}\sum_{x=0}^{N-1} x,$$
$$y_c = \frac{1}{M}\sum_{y=0}^{N-1} y \qquad (2.13)$$

And

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2}, \theta = arctan\frac{y-y_c}{x-x_c} \qquad (2.14)$$

The VZM and PFTs are applied on I_p. The acquired coefficients of the three transform are translation invariant due to the use of centroid as polar space origin. Rotation invariance is achieved by ignoring the phase information in the coefficients and only retaining the magnitudes of the coefficients. To achieve scale invariance, the first magnitude value is normalized by the area of the circle (area) in which the polar image resides or the mass of the shape (mass), and all the other magnitude values are normalized by the magnitude of the first coefficient. The translation, rotation and scale normalized PFT coefficients are used as the shape descriptors. To summarize, the shape descriptor derived from VZM and the FD derived from PFT1 and PFT2 are VZMD, FD1 and FD2 respectively, they are shown as following

$$\mathbf{VZMD} = \left\{\frac{|VF(0)|}{mass}, \frac{|VF(1)|}{|VF(0)|}, \dots, \frac{|VF(n)|}{|VF(0)|}\right\} \qquad (2.15)$$

$$\mathbf{FD1} = \left\{\frac{|PF_1(0)|}{mass}, \frac{|PF_1(0,1)|}{|PF_1(0,0)|}, \dots, \frac{|PF_1(0,n)|}{|PF_1(0,0)|}, \dots, \frac{|PF_1(m,0)|}{|PF_1(0,0)|}, \dots, \frac{|PF_1(m,n)|}{|PF_1(0,0)|}\right\} \qquad (2.16)$$

$$\mathbf{FD2} = \left\{\frac{|PF_2(0)|}{area}, \frac{|PF_2(0,1)|}{|PF_2(0,0)|}, \dots, \frac{|PF_2(0,n)|}{|PF_2(0,0)|}, \dots, \frac{|PF_2(m,0)|}{|PF_2(0,0)|}, \dots, \frac{|PF_2(m,n)|}{|PF_2(0,0)|}\right\} \qquad (2.17)$$

Where m is the maximum number of the radial frequencies selected and n is the maximum number of angular frequencies selected. m and n can be adjusted to achieve hierarchical coarse to fine representation requirement. For efficient shape description, only a small number of the acquired descriptors are selected for shape representation. The selected descriptors form a feature vector which is used for indexing the shape. For two shapes represented by their Fourier descriptors, the similarity between the two shapes is measured by the Euclidean distance between the two feature vectors of the shapes. Therefore, the online matching is efficient and simple.

2.2.4 Implementation of GFD

The implementation of GFD can be summarized into 4 steps, translation normalization, polar Fourier transform, rotation normalization and scale normalization. The algorithm of deriving GFD using PFT2. The algorithms of deriving VZMD and GFD using PFT1 are similar, with only difference in the basis calculation of polar Fourier transform step and scaling normalization step.

2.3 Support Vector Machine (SVM)

The classification problem can be restricted to consideration of the two-class problem without loss of generality. In this problem the goal is to separate the two classes by a function which is induced from available examples. The goal is to produce a classifier that will work well on unseen examples, i.e. it generalizes well. Consider the example in Figure (2.6). Here there are many possible linear classifiers that can separate the data, but there is only one that maximizes the margin (maximizes the distance between it and the nearest data point of each class). This linear classifier is termed the optimal separating hyperplane. Intuitively, we would expect this boundary to generalize well as opposed to the other possible boundaries.

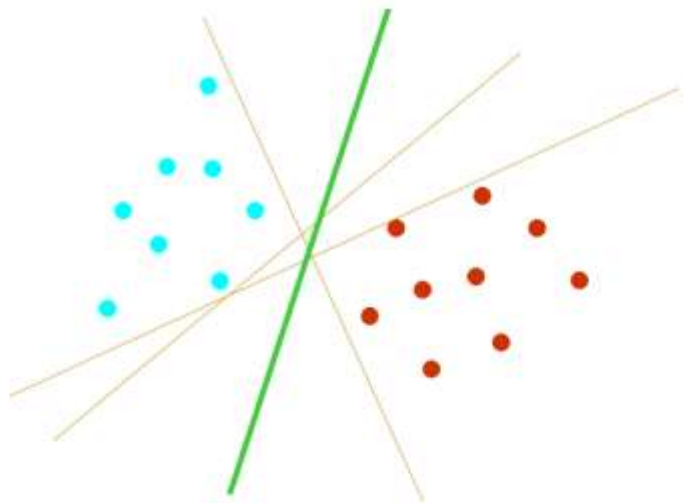


Figure 2.3: Optimal Separating Hyperplane

2.3.1 The Optimal Separating Hyperplane

Consider the problem of separating the set of training vectors belonging to two separate classes,

D = n (x 1, y1), . . . ,(xl , yl) o , x ∈ R n , y ∈ {−1, 1}, **(2.18)**

With a hyperplane,

(w, xi) + b = 0 **(2.19)**

The set of vectors is said to be optimally separated by the hyperplane if it is separated without error and the distance between the closest vector to the hyperplane is maximal. There is some redundancy in Equation 2.2, and without loss of generality it is appropriate to consider a canonical hyperplane [27], where the parameters w, b are constrained by,

$$\min_{i}\left|\left(w, x^{i}\right) + b\right| = 1 \qquad (2.20)$$

This incisive constraint on the parameterization is preferable to alternatives in simplifying the formulation of the problem. In words it states that: the norm of the weight vector should be equal to the inverse of the distance, of the nearest point in the data set to the hyperplane. The idea is illustrated in Figure 2.7, where the distance from the nearest point to each hyperplane is shown.

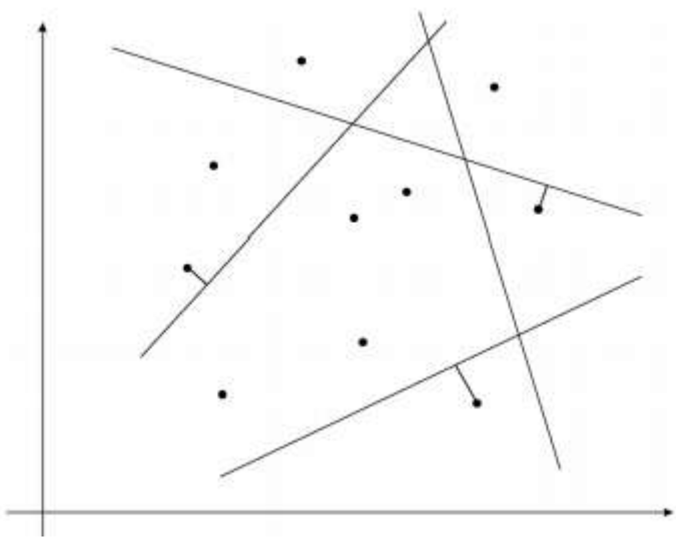


Figure 2.4: Canonical Hyperplanes

A separating hyperplane in canonical form must satisfy the following constraints,

$$\boldsymbol{y^i}\ \mathbf{[(w,}\boldsymbol{x^i}\mathbf{)+b\ ]\geq 1,}\qquad \mathbf{i = 1, \ldots, l.} \tag{2.21}$$

The distance d(w, b; x) of a point x from the hyperplane (w, b) is,

$$\mathbf{d(w, b; x)} = \frac{|(\boldsymbol{w},\boldsymbol{x^i})+\boldsymbol{b}|}{||\boldsymbol{w}||} \tag{2.22}$$

The optimal hyperplane is given by maximizing the margin, ρ, subject to the constraints of Equation 2.4. The margin is given by

$$\mathbf{p(w, b)} = \min_{\boldsymbol{x^i}:\boldsymbol{y^i}=-1} \boldsymbol{d}(\boldsymbol{w},\boldsymbol{b};\ \boldsymbol{x^i}) + \min_{\boldsymbol{x^i}:\boldsymbol{y^i}=1} \boldsymbol{d}(\boldsymbol{w},\boldsymbol{b};\ \boldsymbol{x^i})$$

$$= \min_{\boldsymbol{x^i}:\boldsymbol{y^i}=-1} \frac{|(\boldsymbol{w},\boldsymbol{x^i})+\boldsymbol{b}|}{||\boldsymbol{w}||} + \min_{\boldsymbol{x^i}:\boldsymbol{y^i}=1} \frac{|(\boldsymbol{w},\boldsymbol{x^i})+\boldsymbol{b}|}{||\boldsymbol{w}||}$$

$$= \frac{1}{||\boldsymbol{w}||}\left(\min_{\boldsymbol{x^i}:\boldsymbol{y^i}=-1} \left|(\boldsymbol{w},\ \boldsymbol{x^i})+\boldsymbol{b}\right| + \min_{\boldsymbol{x^i}:\boldsymbol{y^i}=1} \left|(\boldsymbol{w},\boldsymbol{x^i})+\boldsymbol{b}\right|\ \right) = \frac{2}{||\boldsymbol{w}||} \tag{2.23}$$

Hence the hyperplane that optimally separates the data is the one that minimizes

$$\mathbf{\Phi(w)} = \frac{1}{2}||\boldsymbol{w}||^2 \tag{2.24}$$

It is independent of b because provided Equation 2.4 is satisfied (i.e. it is a separating hyperplane) changing b will move it in the normal direction to itself. Accordingly the margin remains unchanged but the hyperplane is no longer optimal in that it will be nearer to one class than the other. To consider how minimizing Equation 2.7 is equivalent to implementing the SRM principle, suppose that the following bound holds,

$$||\boldsymbol{w}||^2 < \mathbf{A} \tag{2.25}$$

Then from Equation 2.4 and 2.5,

$$d\ (w,\ b;\ x) \geq \frac{1}{A} \qquad (2.26)$$

Accordingly, the hyperplanes cannot be nearer than 1 A to any of the data points and intuitively it can be seen in Figure 2.8 how this reduces the possible hyperplanes, and hence the capacity.

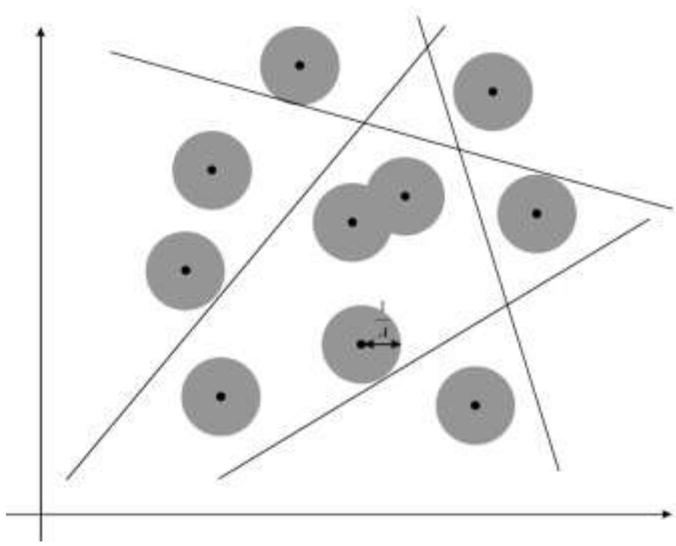


Figure 2.5: Constraining the Canonical Hyperplanes

The VC dimension, h, of the set of canonical hyperplanes in n dimensional space is bounded by,

$$\mathbf{h} \leq \ \min\ [R^2 A^2, \mathrm{n}] + 1, \qquad (2.27)$$

where R is the radius of a hypersphere enclosing all the data points. Hence minimizing Equation 2.7 is equivalent to minimizing an upper bound on the VC dimension. The solution to the optimization problem of Equation 2.7 under the constraints ofEquation 2.4 is given by the saddle point of the Lagrange functional (Lagrangian) [28],

$$\Phi\ (\mathbf{w}\ ,\ \mathbf{b},\alpha) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{l} \alpha_i\ \ (y^i\left[\left(w\ , x^i + b\right) - 1\right] \qquad (2.28)$$

where _ are the Lagrange multipliers. The Lagrangian has to be minimized with respect to w, b and maximized with respect to *w, b, x*. Classical Lagrangian duality enables the primal problem, Equation 2.11, to be transformed to its dual problem, which is easier
to solve. The dual problem is given by,

$$\boldsymbol{max_{\alpha}}\ \mathbf{W(\alpha) =} \boldsymbol{max_{\alpha}}\ \mathbf{(}\ \boldsymbol{max_{w,\ b}}\ \mathbf{\Phi\ (w, b, \alpha))} \tag{2.29}$$

The minimum with respect to w and b of the Lagrangian, Φ, is given by,

$$\frac{\partial \Phi}{\partial \Phi} = 0 \ \rightarrow \sum_{i=1}^{l} \alpha_i y_i = 0 \tag{2.30}$$

$$\frac{\partial \Phi}{\partial \Phi} = 0 \ \rightarrow w = \sum_{i=1}^{l} \alpha_i y_i x_i = 0 \tag{2.31}$$

Hence from Equations 2.11, 2.12 and 2.13, the dual problem is,

$$\max_{\alpha} w(\alpha) = \max_{\alpha} - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j (x_i x_j) + \sum_{k=1}^{l} \alpha_k, \tag{2.32}$$

and hence the solution to the problem is given by,

$$\alpha^* = \arg\min_{\alpha} \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j (x_i x_j) - \sum_{k=1}^{l} \alpha_k, \tag{2.33}$$

with constraints,

$$\alpha_i \ \geq 0 \ i = 1, \dots, l$$

$$\sum_{j=1}^{l} \alpha_j y_j \ = 0 \tag{2.34}$$

Solving Equation 2.15 with constraints Equation 2.16 determines the Lagrange multipliers, and the optimal separating hyperplane is given by,

$$\boldsymbol{w}^{*} = \sum_{i=1}^{l} \boldsymbol{\alpha}_i \boldsymbol{y}_i \boldsymbol{x}_i = \mathbf{0}$$

$$\boldsymbol{b}^{*} = -\frac{1}{2}(\boldsymbol{w}^{*}, \boldsymbol{x}_r + \boldsymbol{x}_s) \quad \textbf{(2.35)}$$

where x_r and x_s are any support vector from each class satisfying,

$$\boldsymbol{\alpha}_r, \boldsymbol{\alpha}_s > \mathbf{0}, \boldsymbol{y}_r = -\mathbf{1}, \boldsymbol{y}_s = \mathbf{1} \quad \textbf{(2.36)}$$

The hard classifier is then,

$$\boldsymbol{f}(\mathbf{x}) = \mathbf{sgn}((\boldsymbol{w}^{*}, \boldsymbol{x}) + \boldsymbol{b}) \quad \textbf{(2.37)}$$

Alternatively, a soft classifier may be used which linearly interpolates the margin,

$$\mathbf{f(x) = h}((\boldsymbol{w}^{*}, \boldsymbol{x}) + \boldsymbol{b})\textbf{where h(z)} = \begin{cases} -\mathbf{1} : & \boldsymbol{z} < -\mathbf{1} \\ \boldsymbol{z} : & -\mathbf{1} \le \boldsymbol{z} \le \mathbf{1} \\ +\mathbf{1} : & \boldsymbol{z} > \mathbf{1} \end{cases} \quad \textbf{(2.38)}$$

This may be more appropriate than the hard classifier of Equation 2.19, because it produces a real valued output between −1 and 1 when the classifier is queried within the margin, where no training data resides. From the Kuhn-Tucker conditions,

$$\boldsymbol{\alpha}_i(\boldsymbol{y}^{*}[(\boldsymbol{w}, \boldsymbol{x}^{*} + \boldsymbol{b}] - \mathbf{1}) = \mathbf{0}, \boldsymbol{i} = \mathbf{1}, \ldots, \mathbf{l}. \quad \textbf{(2.39)}$$

and hence only the points xi which satisfy,

$$\boldsymbol{y}^{\boldsymbol{i}}\,[(\mathbf{w}, \boldsymbol{x}^{\boldsymbol{i}}) + \boldsymbol{b}\,] = \mathbf{1} \quad \textbf{(2.40)}$$

Will have non-zero Lagrange multipliers. These points are termed Support Vectors (SV). If the data is linearly separable all the SV will lie on the margin and hence the number of SV can be very small. Consequently, the hyperplane is determined by a small subset of the training set; the other points could be removed from the training set and recalculating the hyperplane would produce the same answer. Hence, SVM can be used to summarize the information contained in a data set by the SV produced. If the data is linearly separable, the following equality will hold,

$$||\boldsymbol{w}||^2 \sum_{i=1}^{l} \alpha_i = \sum_{i \in SVs}^{l} \alpha_i = \sum_{i \in SVs}^{l} \sum_{j \in SVs}^{l} \alpha_i \alpha_j y_i y_j (x_i x_j) \qquad \textbf{(2.41)}$$

Hence from Equation 2.10 the VC dimension of the classifier is bounded by,

$$\mathbf{h} \leq \min\,[\, R^2 \sum_{i \in SVs}^{l} , n] + 1 \qquad \textbf{(2.42)}$$

2.3.2 The Generalized Optimal Separating Hyperplane

So far the discussion has been restricted to the case where the training data is linearly separable. However, in general this will not be the case, Figure 2.9. There are two approaches to generalizing the problem, which are dependent upon prior knowledge of the problem and an estimate of the noise on the data. In the case where it is expected (or possibly even known) that a hyperplane can correctly separate the data, a method of

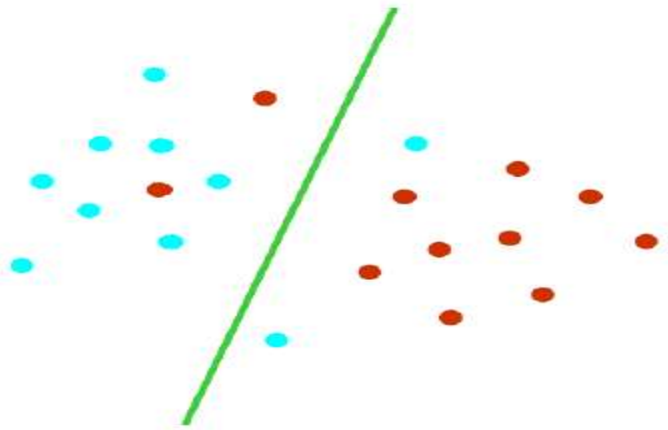


Figure 2.6: Generalized Optimal Separating Hyperplane

Introducing an additional cost function associated with misclassification is appropriate. Alternatively, a more complex function can be used to describe the boundary. To enable the optimal separating hyperplane method to be generalized, Cortes and [27] introduced non-negative variables, $\xi_i \geq 0$, and a penalty function,

$$F_\sigma(\xi) = \sum_i \xi_i^\sigma \; \sigma > 0, \tag{2.43}$$

where the , ξ_i are a measure of the misclassification errors. The optimization problem is now posed so as to minimize the classification error as well as minimizing the bound onthe VC dimension of the classifier. The constraints of Equation 2.4 are modified for thenon-separable case to,

$$y^i \left[(\mathrm{w}, x^i) + b \right] \geq 1 - \xi_i, \quad i = 1, \ldots, l. \tag{2.44}$$

where $\xi_i \geq 0$ The generalized optimal separating hyperplane is determined by the vector *w*, that minimizes the functional,

$$\Phi(\mathrm{w}, \xi) = \frac{1}{2}||w||^2 + C \sum_i \xi_i \tag{2.45}$$

(where C is a given value) subject to the constraints of Equation 2.27. The solution to the optimization problem of Equation 2.28 under the constraints of Equation 2.27 isgiven by the saddle point of the Lagrangian [28],

$$\Phi(\mathrm{w}, \mathrm{b}, \alpha, \xi, \beta) = \frac{1}{2}||w||^2 + C \sum_i \xi_i - \sum_{i=1}^{l} \alpha_i \left(y^i [w^T x^i + \mathrm{b}] - 1 + \xi_i \right) - \sum_{j=1}^{l} \beta_i \xi_i, \tag{2.46}$$

where α,β are the Lagrange multipliers. The Lagrangian has to be minimized with respect to w, b, x and maximized with respect to α,β. As before, classical Lagrangian duality enables the primal problem, Equation 2.29, to be transformed to its dual problem.The dual problem is given by,

$$\min_{\boldsymbol{\alpha}} \boldsymbol{w}(\boldsymbol{\alpha},\boldsymbol{\beta}) = \min_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\min_{\boldsymbol{w},\boldsymbol{b},\boldsymbol{\xi}} \Phi(\boldsymbol{w},\boldsymbol{b},\boldsymbol{\alpha},\boldsymbol{\xi},\boldsymbol{\beta})) \qquad \textbf{(2.47)}$$

The minimum with respect to w, b and _ of the Lagrangian, _, is given by,

$$\frac{\partial\Phi}{\partial b} = 0 \rightarrow \sum_{i=1}^{l} \alpha_i\, y^i = 0$$

$$\frac{\partial\Phi}{\partial \xi} = 0 \rightarrow \mathbf{w} = \sum_{i=1}^{l} \alpha_i\, y^i x^i$$

$$\frac{\partial\Phi}{\partial \xi} = 0 \rightarrow \quad \alpha_i + \beta_i = \mathrm{C}. \qquad \textbf{(2.48)}$$

Hence from Equations 2.29, 2.30 and 2.31, the dual problem is,

$$\min_{\boldsymbol{\alpha}} \boldsymbol{w}(\boldsymbol{\alpha}) = \max_{\boldsymbol{\alpha}} -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{j} \alpha_i\alpha_j y_i y_j\,(x_i,x_i) + \sum_{k=1}^{1} a_k, \qquad \textbf{(2.49)}$$

and hence the solution to the problem is given by,

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{j} \alpha_i\alpha_j y_i y_j\,(x_i,x_i) - \sum_{k=1}^{1} a_k, \qquad \textbf{(2.50)}$$

with constraints,

$$0 \le \alpha_i \le C \quad i = 1, \ldots, l$$

$$\sum_{i=1}^{j} \alpha_i y_i = 0 \qquad \textbf{(2.51)}$$

The solution to this minimization problem is identical to the separable case except for a modification of the bounds of the Lagrange multipliers. The uncertain part of Cortes'sapproach is that the coefficient C has to be determined. This parameter introduces additional capacity control within the classifier. C can be directly related to a regularization parameter [29];[30]. [31] uses a value of C = 5, but ultimately C must be chosen to reflect the knowledge of the noise on the data.

2.3.3 Multi-Class Support Vector Machine

the theoretical properties of the loss function of SimMSVM, including the Fisher consistency issue. We also give the experimental comparisons and find that the new simplified model achieves significant speed-up comparedwith its original model [32].

2.3.3.1 Indirect Multi-Class SVM

2.3.3.1.1 One-Versus-Rest Approach

The one-versus-rest (1VR) approach [33] constructs k separate binary classifiers for k-class classification. The m-th binary classifier is trained using the data from the m-th class as positive examples and the remaining k −1 classes as negative examples. During test, the class label is determined by the binary classifier that gives maximum output value. A major problem of the one-versus-rest approach is the imbalanced training set. Suppose that all classes have an equal size of training examples, the ratio of positive to negative examples in each individual classifier is $\frac{1}{k-1}$. In this case, the symmetry of the original problem is lost.

2.3.3.1.2 One-Versus-One Approach

Another classical approach formulti-class classification is the one-versus-one (1V1) or pairwise decomposition [34]. It evaluates all possible pairwise classifiers and thus induces k(k −1)/ 2 individual binary classifiers. Applying each classifier to a test example would give one vote to the winning class. A test example is labeled to the class with the most votes. The size of classifiers created by the one-versus-one approach is much larger than that of the one-versus-rest approach. However, the size of QP in each classifier is smaller, which makes it possible to train fast. Moreover, compared with the one-versus-rest approach, the one-versus-one method is more symmetric. Platt et al. [35] improved the one-versus-one approach and proposed a method called Directed Acyclic Graph SVM (DAGSVM) that forms a tree-like structure to facilitate the testing phase. As a result, it takes only k −1 individual evaluations to decide the label of a test example.

2.3.3.2 Direct Multi-Class SVM

2.3.3.2.1 Weston and Watkins' Multi-Class SVM

Instead of creating several binary classifiers, a more natural way is to distinguish all classes in one single optimization processing, as proposed by Vapnik [33],Weston and Watkins [36], and Bredensteiner and Bennett [37]. For a k-class problem, these methods design a single objective function for training all k-binary SVMs simultaneously and maximize the margins from each class to the remaining ones. Here, we take the method by Weston and Watkins as an example. Given a labeled training set represented by $\{(x_1,y_1) \dots,$ $(x_l, y_l)\}$ of cardinality l, where $x_i \in R^d$ and $y_i \in \{1, \dots, k\}$, the formulation proposed in [36] is given as follows:

$$\min_{W_m \in H, b \in R^k, \xi \in R^{l*k}} \frac{1}{2} \sum_{m=1}^{k} w_m^T w_m + C \sum_{i=1}^{l} \sum_{t \neq y_i} \xi_{i,t} \qquad \textbf{(2.52)}$$

$$\textbf{Subject to}\, w_{y_i}^T \varphi(x_i) + b_{y_i} \geq w_t^T \varphi(x_i) + b_t + 2 - \xi_{i,t} \qquad \textbf{(2.53)}$$

$$\xi_{i,t} \geq 0,$$

$$i = 1, \dots, l,\ t \in \{1, \dots, k\} \backslash y_i$$

The resulting decision function is

$$argmax_m f_m(X) = argmax_m f_m(w_t^T \varphi(x_i) + b_m). \qquad \textbf{(2.54)}$$

The main disadvantage of this approach is that the computational time may be very high due to the enormous size of the resulting QP.

2.3.3.2.2 Crammer and Singer's Multi-Class SVM

Crammer and Singer [32] presented another "all-together" approach by solving the following optimization problem:

$$\min_{W_m \in H,, \xi \in R^l} \frac{1}{2} \sum_{m=1}^{k} w_m^T w_m + C \sum_{i=1}^{l} \xi_i \tag{2.55}$$

Subject to

$$w_{y_i}^T \varphi(x_i) \text{-} w_t^T \varphi(x_i) \geq 1 - \delta_{y_i,t} + \xi_i \tag{2.56}$$

$$i = 1, \ldots, l, \; t \in \{1, \ldots, k\}$$

where$\delta_{i,j}$ is the Kronecker delta (defined as 1 for i = j and as 0 otherwise). The resulting decision function is

$$argmax_m f_m(X) = argmax_m w_t^T \varphi(x) \tag{2.57}$$

Note that the constraints $\xi_i \geq 0$, i = 1, …, l, are implicitly indicated in the margin constraints of (2.7) when t equals y_i. In addition, (2.55) focuses on classification rule (2.57) without the bias terms b_m, m = 1, . . . ,k. A nonzero $b_m$can be easily modeled by adding an additional constant feature to each x (see, e.g., [38]).

2.3.4 Decision DAGs

A Directed Acyclic Graph (DAG) is a graph whose edges have an orientation and no cycles. A Rooted DAG has a unique node such that it is the only node which has no arcs pointing into it. A Rooted Binary DAG has nodes which have either 0 or 2 arcs leaving them. We will use Rooted Binary DAGs in order to define a class of functions to be used in classification tasks. The class of functions computed by Rooted Binary DAGs is formally defined as follows.

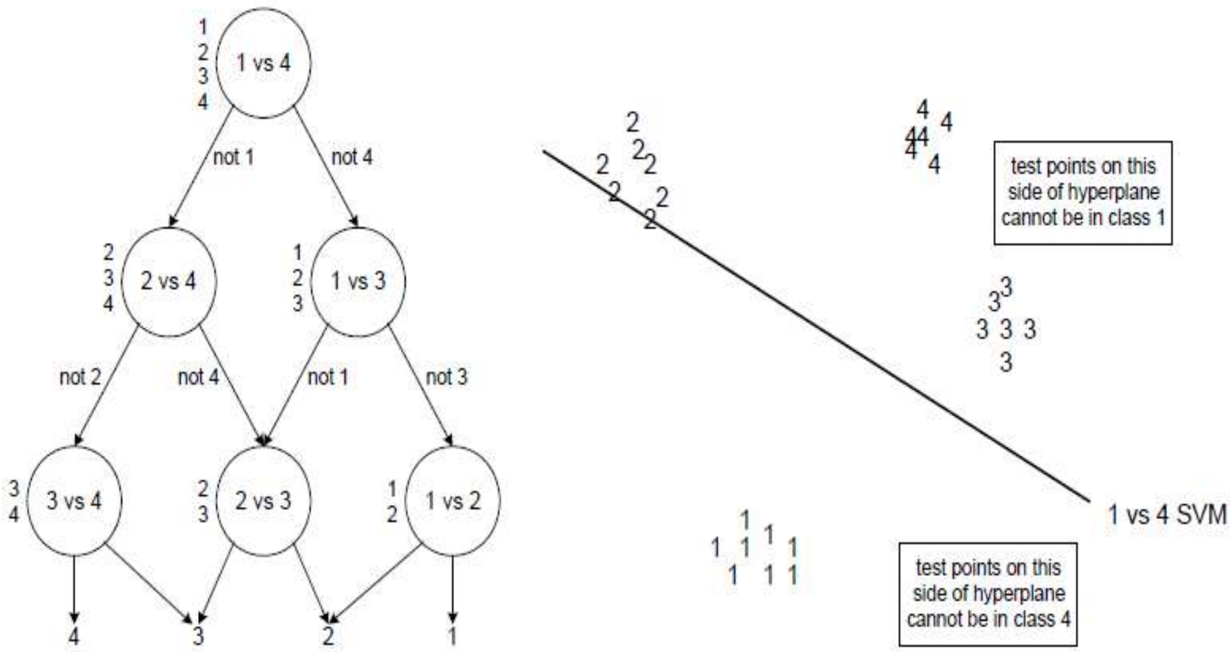
Definition

Decision DAGs (DDAGs). Given a space *X* and a set of Boolean functions
$F = \{f : X \rightarrow \{0, 1\}\}$, the class DDAG(F) of Decision DAGs on *N*classes over *F*are functions which can be Implemented using a rooted binary DAG with *N*leaves labeled by the classes where each of the $K = N(N-1)/2$ internal nodes is labeled with an element of *F*. The nodes are arranged in a triangle with the single root node at the top, two nodes in the second layer

and so on until the final layer of N leaves. The i-th node in layer j < N is connected to the i-th and (i+1)-st node in the (i+1)-st layer.

To evaluate a particular DDAG *G*on input $x\in X$, starting at the root node, the binary function at a node is evaluated. The node is then exited via the left edge, if the binary function is zero; or the right edge, if the binary function is one. The next node's binary function is then evaluated. The value of the decision function D *(x)* is the value associated with the final leaf node (see Figure 2.10). The path taken through the DDAG is known as the evaluation path. The input *x* reaches a node of the graph, if that node is on the evaluation path for *x*. We refer to the decision node distinguishing classes *i* and *j* as the *ij*-node. Assuming that the number of a leaf is its class, this node is the *i*-th node in the(*N*-*j*+*i*)-th layer provided $i < j$. Similarly, the *j*-nodes are those nodes involving class *j*, that is, the internal nodes on the two diagonals containing the leaf labeled by *j*.

The DDAG is equivalent to operating on a list, where each node eliminates one class fro mthe list. The list is initialized with a list of all classes. A test point is evaluated against the decision node that corresponds to the first and last elements of the list. If the node prefers.



(a) (b)

Figure 6.10: (a) The decision DAG for finding the best class out of four classes.

One of the two classes, the other class is eliminated from the list, and the DDAG proceeds to test the first and last elements of the new list. The DDAG terminates when only one class remains in the list. Thus, for a problem with *N* classes, *N-1* decision nodes will be evaluated in order to derive an answer.

The current state of the list is the total state of the system. Therefore, since a list state is reachable in more than one possible path through the system, the decision graph the algorithm traverses is a DAG, not simply a tree.

Decision DAGs naturally generalize the class of Decision Trees, allowing for a more efficient representation of redundancies and repetitions that can occur in different branches of the tree, by allowing the merging of different decision paths. The class of functions implemented is the same as that of Generalized Decision Trees [39], but this particular representation presents both computational and learning-theoretical advantages.

2.3.5 The voting Algorithm

The different voting algorithms is used to describe below. Each algorithm takes an in ducerand a training set as input and runs the inducer multiple times by changing the distribution of training set instances. The generated classifiers are then combined to create a final classifier that is used to classify the test set.

The mechanism of the voting algorithm is making the max numbers of separating between each classes and the other one so that happening exactly when we separated the classes with support vector machine in both training and test phases.

In our dataset, we have many vectors of features of leave images, so for each class; the SVM used these vectors so we can name S set of all dataset. And S’ is set of each class from S so the voting algorithm makes all the numbers of votes from S for each S’ and at the end we just take the max voting numbers to S’ and we decide each class it is.

2.4 Conclusion

The proposed GFD satisfies all the six requirements set by MPEG-7 for shape representation, that is, good retrieval accuracy, compact features, general application, low computation complexity, robust retrieval performance and hierarchical coarse to fine representation. It has been tested on both MPEG-7 contour shape database and MPEG-7 region shape database. Comparisons have been made between GFD, 1-D FD, and MPEG-7 shape descriptors, results show that the proposed GFD outperforms these shape descriptors.

Support vector machines are one of the widely used machine learning algorithms for data classification.

In the simplest form, SVM uses a linear hyperplane to create a classifier with a maximal margin, in other cases, where the data is not linearly separable, the data is mapped into a higher dimension feature space.

In our project, the contribution of Generic Fourier Descriptor was on extracting of characteristic, then we did the recognition part using Support Vector Machine.

CHAPTER 3

Modeling, Experimental Results and Discussions

3.1 Introduction

In this chapter, we have presented the database, the working Environment and the different steps of a complete system of Leaf recognition were implemented with MATLAB 2013.

This chapter consist of the steps of our preprocessing like converting to RGB and centroid, feature extraction…etc. we have got a statistical results on feature extraction with GFD by changing the parameters of N and M and we have identified them by SVM..

3.2 Working Environment:

3.2.1 Hardware Environment:

We used in this work a laptop with (680GB, Intel(R) Core (TM) i5-3210M CPU @ 2.50 Ghz) with RAM 4.00 Go, used to perform our application of Recognition of Leaf Images Based on generic fourier descriptor Using a SVM.

3.2.2 Programming Language

a. MATLAB

The name MATLAB stands for MATRIX LABRATORY.

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

a. Math and computation;
b. Algorithm development;
c. Modeling, simulation, and prototyping;
d. Data analysis, exploration, and visualization;
e. Scientific and engineering graphics;
f. Application development, including graphical user interface building.

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations.

3.3 Proposed Methodology

A typical image based plant identification system is shown in figure3.1 and the major steps are explained in consecutive sub-sections.

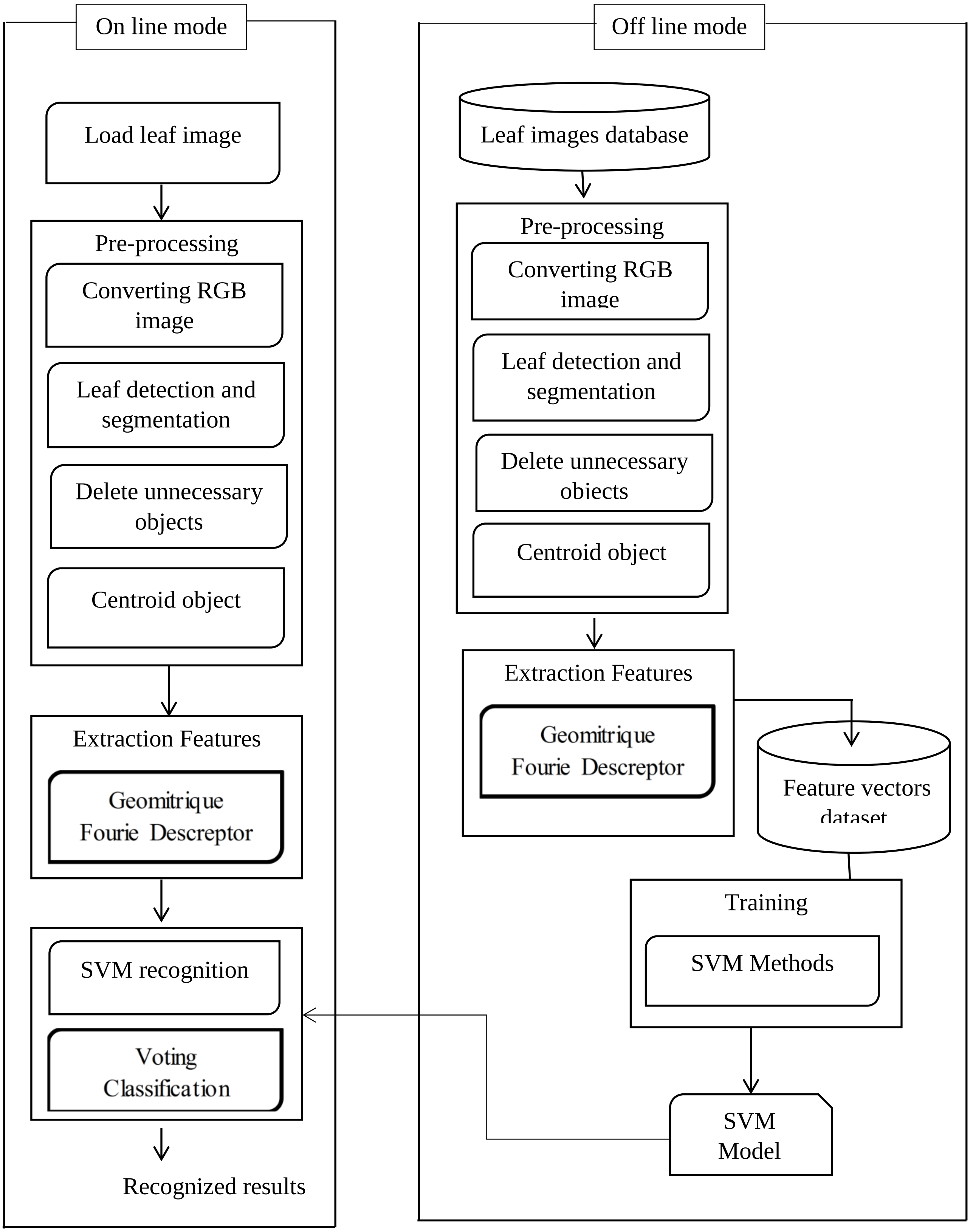


Figure3.1 : Flowchart of proposed system

3.3.1 Data Base (Leaf Screenshot)

A leaf image can be easily acquired using scanner or digital camera. The image can be of any size. However, for better results, the image should have preferably single color background with no petiole. The proposed system is tested on Flavia dataset, which contains 1907 RGB leaf images of 32 plants; each species has 50 to 77 sample leaves Figure 3.2. Each image in dataset is of 1600x 1200 resolution having white background and with no leafstalk. File names of all images are 4-digit numbers, followed by a ".jpg" suffix.

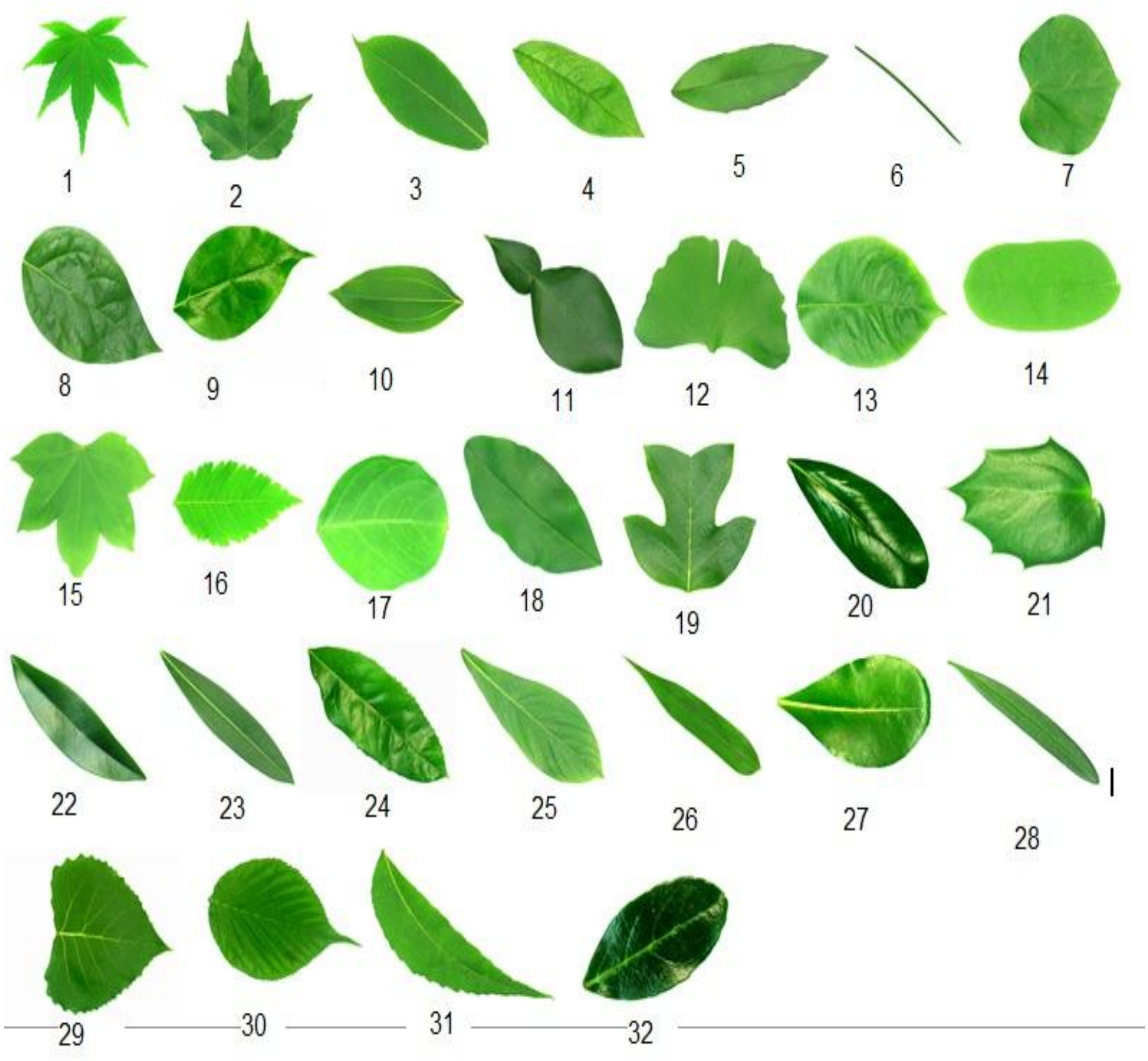


Figure3.2: Screenshot of all 32-leaf types

For that we create this table 3.1 it contains all plants leaves images with some information of Scientific Names and numbers of specimens available in each class

Class	Scientific Names	#	Class	Scientific Names	#
1	Acer Dalmatum	56	17	Lagerstroemia indica	61
2	Acer buergerianum	53	18	Ligustrum quihoui	55
3	Aesculus_chinensis	63	19	Liriodendron chinense	53
4	Amygdalus persica	54	20	Magnolia grandiflora	57
5	Berberis anhweiensis	65	21	Mahonia bealei	55
6	Cedrus deodara	77	22	Manglietia fordiana	52
7	Cercis chinensis	72	23	Nerium indcus	66
8	Chimonanthus praecox	52	24	Osmanthus fragrans	56
9	Cinnamomum camphora	65	25	Phoebe zhennan	62
10	Cinnamomum japonicum	55	26	Phyllestachys pubescens	59
11	Citrus reticulata	56	27	Pittosporum tobira	63
12	Ginkgo biloba	62	28	Podocarpus macrophyllus	60
13	Ilex macrocarpa	50	29	Populus canadensis Moench	64
14	Indigofera tinetoria	73	30	Prunus yedoensis	55
15	Kalopanax septemlobus	52	31	Toona sinensis	65
16	Koelreuteria paniculata	59	32	Viburnum awabuki	60

Table3.1: Leaf database: plant species (class) and number of specimens available (#)

3.4 Presentation of the application

3.4.1 Interface of application

This the interface of our application

1: Load the leaf image that the user want to recognize

2: Load all the vectors of features of 32 classes which means 1907 vectors

3: Take training number of vectors from each class that selected from dataset

4: Save a model created by training SVM

5: Classify all leaf images with white background

6: Classify all leaf images with different background after segmentation

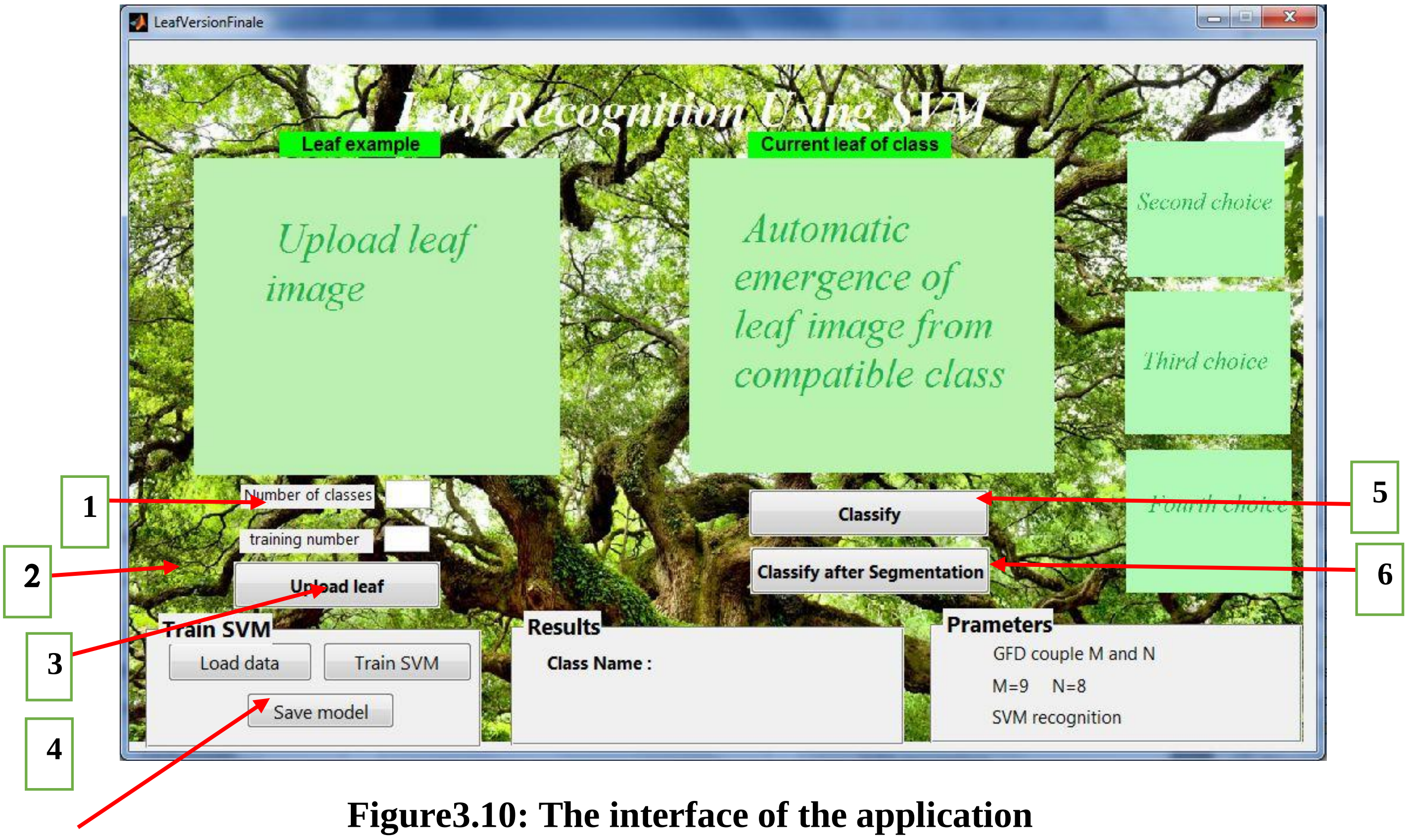


Figure3.10: The interface of the application

After uploading the leaf image and choosing the number of classes and the training number vectors from each class. Than after loading data we click on Train SVM bottom, which is shown in figure 3.9.

Figure3.11: training step

After clicking the classify bottom we wait a few seconds, than we notice the recognition on two ways, first the name of class in the result area, secondly example image from the right class, we also show the second three choices on the left according to the rest of recognition probabilities.

However, after uploading a leaf image with different background and click the same bottom we get a bad recognition result, in figure 3.11

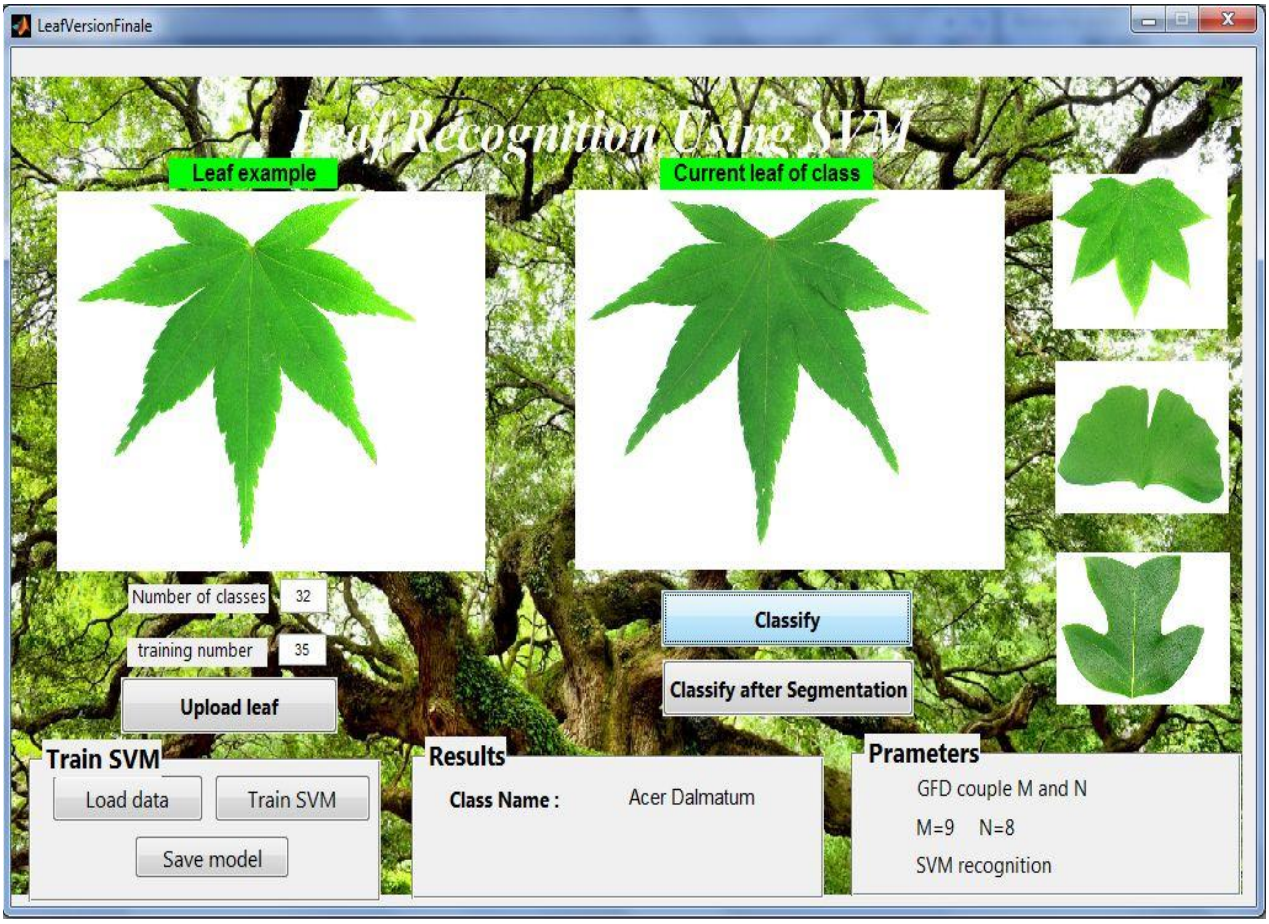


Figure3.12: classification step (1)

Figure3.13: classification step (2)

However, when we click classify after segmentation we get the right recognition result figure 3.12

Figure3.14: classification step (3)

3.4.2 Pre-processing

In order to extract any specific information, image pre-processing steps are carried out before the actual analysis of the image data. Preprocessing refers to the initial processing of input leaf image to eliminate the noise and correct the distorted or degraded data. Figure 3.1 illustrates techniques like grayscale conversion, binarization, smoothing, filtering, edge detection, etc. used for the enhancement of the leaf image.

3.4.2.1 Converting RGB image

In this phase, we convert the leaf image to the grayscale conversion, then to binarization conversion. The leaf example from class Acer Dalmatum Figure 3.3.

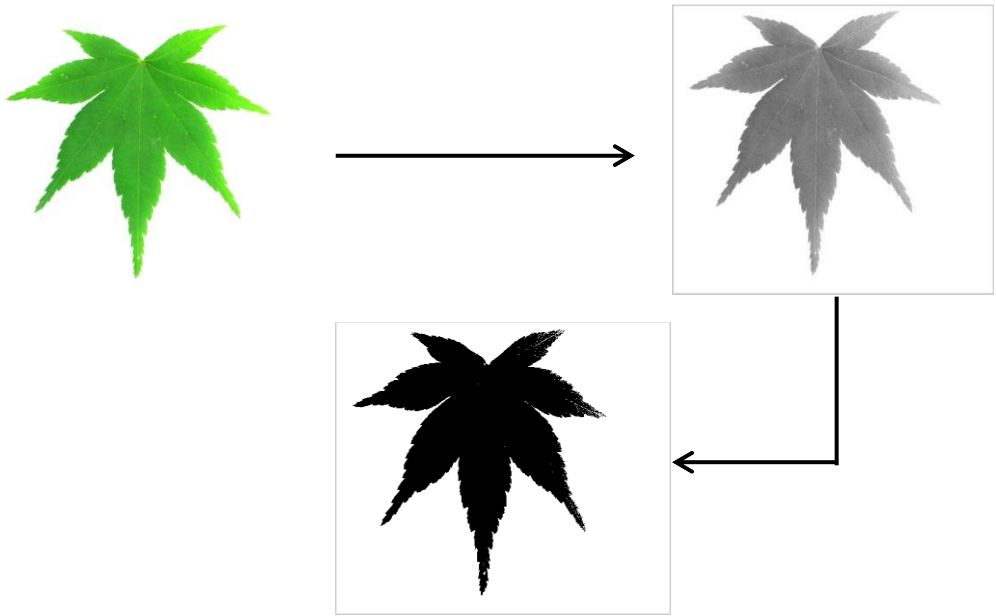


Figure3.15 : Converting leaf image to RGB

3.4.2.2 Leaf detection and segmentation

In figure 3.4 we show segmentation phase, on leaf image with different background from the ones on our database.

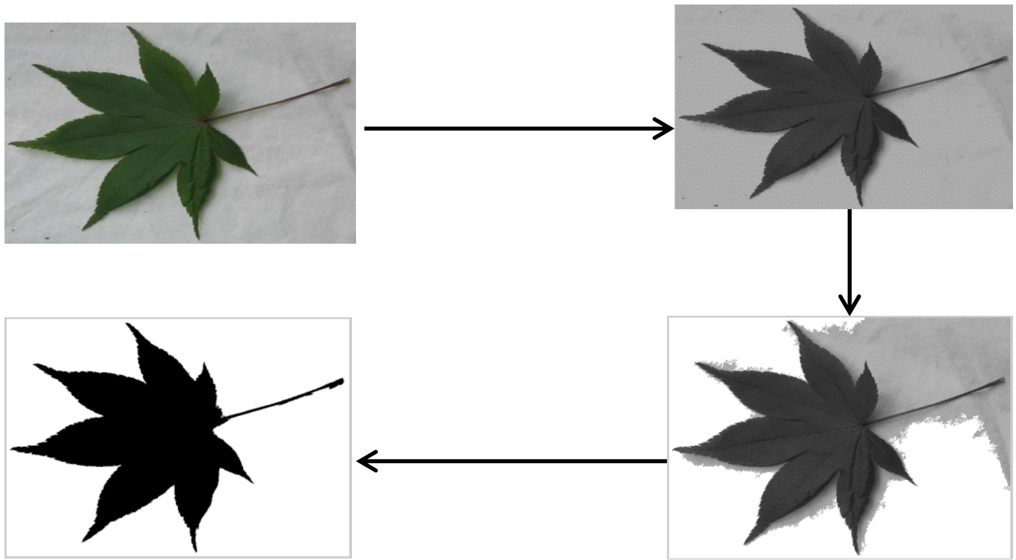


Figure3.16: segmentation phase

3.4.2.3 Delete unnecessary objects

Therefore, after the majority of preprocessing, we remove all content that is less than 4,000 pixels, these unnecessary objects may be spots in the background of the leaf; to get the process we need just one object.

3.4.2.4 Centroid the required object

For many descriptors shapes (e.g. Fourier descriptors etc.) in the image feature it is necessary to have the leaf center in the image center i.e the center of mass of the object lies in the middle of the leaf image. From the class Aesculus chinensis Figure 3.5.

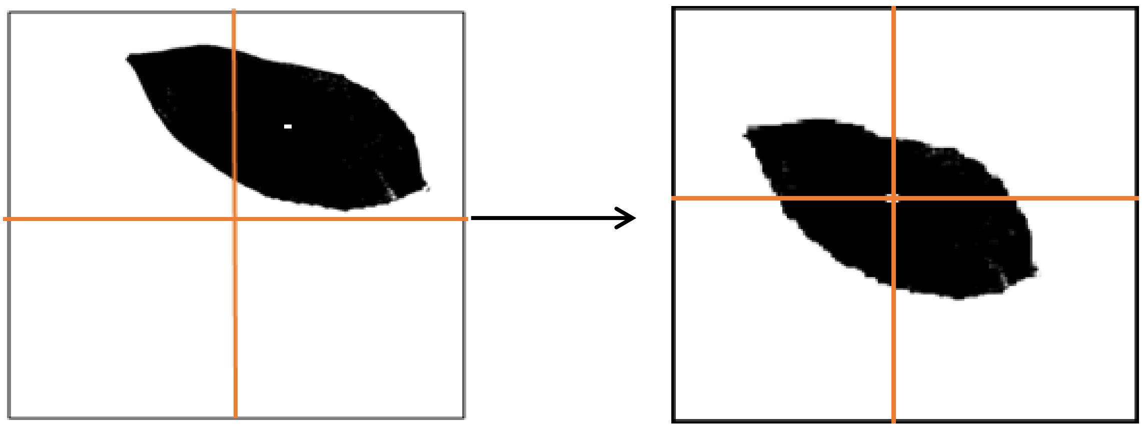


Figure3.17 : Get the object in the center of image

3.4.3 Feature extraction

Our method takes into account the shape features of the leaf. Leaves of different plants are invariably similar in shape therefore a single feature alone may not produce expected results; so in this work we use the GFD technic for extraction of features and we have established the dataset; Depending on this algorithm:

3.4.3.1 Algorithm of deriving GFD:

Calculate the generic Fourier descriptors of an object in a binary image:[41]
1. Input shape image data *f*(*x*, *y*);

2. Get centroid of the shape (*xc*, *yc*);

3. Set the centroid as the origion; /* translation normalization */

4. Get the maximum radius of the shape image (*maxRad*);

5. Polar Fourier transform

For radial frequency (*rad*) from zero to maximum radial frequency (*m*)

For angular frequency (*ang*) from zero to maximum angular frequency (*n*)

For *x* from zero to width of the shape image

For *y* from zero to height of the shape image

{

radius = square root[$(x - maxRad)^2$ + $(y - maxRad)^2$];

theta = arctan2[(*y*-*maxRad*)/(*x*-*maxRad*)]; /* *theta* falls within [–π, +π] */

if(theta<0) theta += 2π; /* extend *theta* to [0, 2π] */

FR[*rad*][*ang*] += *f*(*x*,*y*)×cos[2π×*rad*×(*radius*/*maxRad*) + *ang*×*theta*]; /* real part of spectra */

FI[*rad*][*ang*] −= *f*(*x*,*y*) ×sin[2π×*rad*×(*radius*/*maxRad*) + *ang*×*theta*]; /* imaginary part of spectra */

}

6. Calculate FD

For rad from zero to m

For ang from zero to n

{

/* rotation and scale normalization */

If (rad=0 & ang=0)

FD[0] = square root[(FR2[0][0] + FR2[0][0])/(π×maxRad2)];

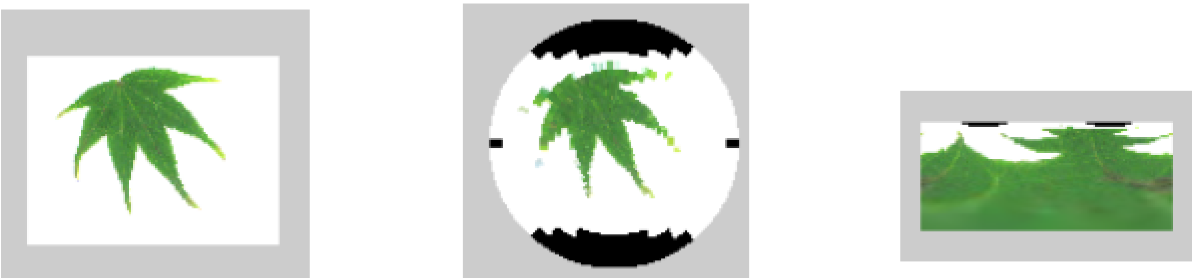
Else

FD[rad×n+ang] = square root[(FR2[rad][ang] + FI2[rad][ang])/ FD[0]];

}

7. Output feature vector **FD**.

We show an example of an leaf image with GFD in figure 3.6



(a) (b) (c)

Figure 3.18 implementation of GDF on a leaf image :(a) leaf image (b) leaf image in polar space (c) polar image of (b) plotted into Cartesian space.

3.4.4 SVM phase

3.4.4.1 SVM

In this phase on the SVM ,one to one approach, in all classes for create the model that contain the part of training, in the same way the recognition based on SVM model. After that, we worked on the voting classification method.

3.5 Test and results

In this part of our work we change a lot in parameters for getting good results; we change the parameter in the function GFD, we get some deferent types of vectors. All this vectors have the deferent length Table, the values and all this will be showed at the table 3.2.

GFD parameters		Size Vector	GFD parameters		Size Vector	Other different GFD parameters		Size Vector
M	**N**	**byte**	**M**	**N**	**Byte**	**M**	**N**	**byte**
6	**6**	**49**	**3**	**8**	**36**	**7**	**5**	**48**
	8	**63**	**6**		**63**	**9**	**6**	**70**
	10	**77**	**9**		**90**	**9**	**4**	**50**
	12	**91**	**12**		**117**	**3**	**6**	**28**
						10	**10**	**121**

(a) **(b)** **(c)**

Table3.2 : Size Vector of different values of GFD (M, N)

In part (a) M=6 is proposed and N is changed, part (b) propose N=8 and change M, part (c) change M and N randomly in the table 3.3 we will show you deferent percentages of SVM recognition and we noticed that the longest length of victors does not means best result on the 13 examination.

The same way we change the number of vectors in the training and we stable it in the number 35 vectors from each class of our dataset.

3.5.1 SVM application

In this phase on SVM training, we use one to one approach we chose 35 vectors from each class. So that the Equivalent between. 45% and 70% for all classes for create the model that contain the training result; secondly in the SVM recognition is on the off line mode by the other part from dataset in off line mode or by the leaf image screenshot. After this research, we get a real results and statistics about our work and what we have reached; Table 3.3.

GFD parameters		SVM	GFD parameters		SVM	Other different GFD parameters		SVM
M	**N**	**%**	**M**	**N**	**%**	**M**	**N**	**%**
6	**6**	**78.81**	**3**	**8**	**79.56**	**7**	**5**	**78.80**
	8	**79.72**	**6**		**79.12**	**9**	**6**	**78.59**
	10	**79.12**	**9**		**80.16**	**9**	**4**	**77.57**
	12	**79.04**	**12**		**78.08**	**3**	**6**	**79.28**
						10	**10**	**78.77**

(a) (b) (c)

Table3.3 : Different results (%) of SVM Classification

3.6 Results and Analysis

This is all statistics percentage obtained after 13 different tests, and in each test we changed the parameters M and N in GFD; at the time we record the obtained results of Table 3.3.

Then we have to translate the data on the Table 3.3 to curves subject to fixe each of parameters M and N, and we get the curves Figure 3.5 and Figure 3.6 apply to the *MATLAB* in figure 3.5 we observe a good percentage is correspond of value N=8. For that we get this interested value and fix it. Then we change the value of M parameter and the curve in Figure 3.6 and we get other highest percentage.

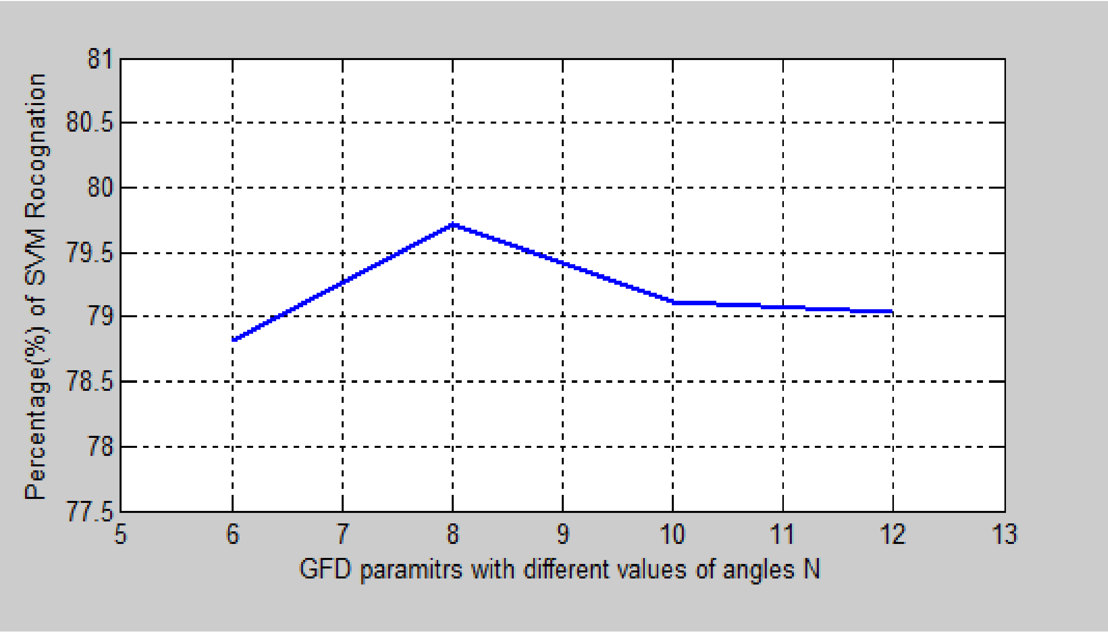


Figure3.7: SVM recognition after extraction of features with GFD parameter M=6

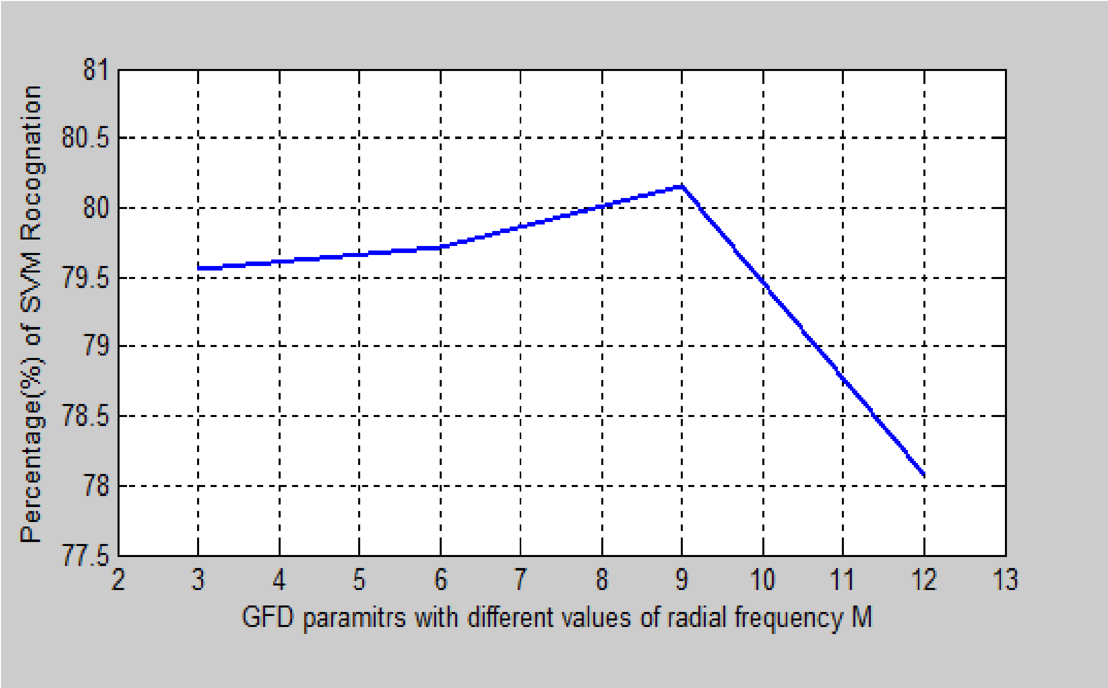


Figure3.8 : SVM recognition after extraction of features with GFD parameter N=8

After observing the table and curves we have to choose the highest value between all the results in Figure 3.5. So this value is 80.16% from good SVM recognition with the couple M=9 and N=8.

3.6.1 Best result

After choosing the highest percentage we analyze and select the best and the worst observations. Firstly we begin with the confusion matrix for linear discriminant analysis of the Descriptor of the Fourier Descriptors and all information are in Table 3.4.

We observed on the confusion matrix all information in the SVM recognition for all the classes in our dataset; and we get the diagonal of the confusion matrix. Then we have read the very accurate results recorded on the confusion matrix, after that we conclude in recent table 3.4

3.6.2 Analyses of information

After the observation, we can create another table including the stats related to the number of classes on each interval ratios percentage of SVM recognition Table 3.4.

Intervals **(%)**	100	[90,100[	[80,90[	[70,80[	[60,70[	[50,60[	Less than 50
N° classes	1,6,7, 15	2,11,14,17,1 9,21,22, 23,25,27,29	3,12,18 ,26,28	8,13,24	9,10,16 ,31	4,20,30 ,32	5

Table3.4 : numbers of classes in different interval

In Table 3.4, we have noticed that the values of the classes are located within the top ratios, So we have 20 classes more than 80% , 7 classes between 60% and 80%, a few numbers of classes takes less than 60%, but the problem is that one class is miss-classified, so that a problem has happened with the two other classes

3.6.2.1 Analyses negative classification

The weaker class takes just 20% of the class “Berberis anhweiensis” and the other class that there was a problem with it is “Manglietia fordiana” and “Nerium indcus” Figure 3.7.

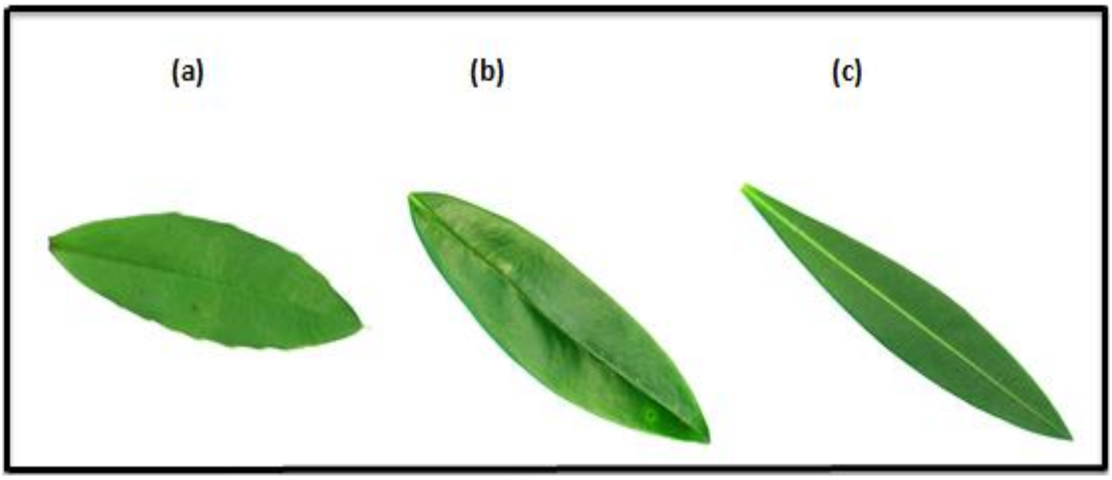


Figure3.9: examples leaves of the three interested classes (a) Berberis anhweiensis (b) Manglietia fordiana (c) Nerium indcus

But the opposite is not true. So for that we create Table 3.5 that contains the ratios of convergence between the three classes mentioned above.

Class	Berberis anhweiensis	Manglietia fordiana	Nerium indcus
Berberis anhweiensis	20%	30%	35%
Manglietia fordiana	3.70%	96.3%	0.0%
Nerium indcus	4.88%	0.0%	95.12%

Table3.5: Statistics of correct and wrong between classes

The question is why this mistake of miss-classified classes is in one way and not in the other way around?

Our proposed answer is that the class “Berberis anhweiensis” is a little similar with the two other classes “Manglietia fordiana” and “Nerium indcus” so this similarity is in the shape. So may be if we consider taking other features for example the other features (for example texture) it will be other results.

3.7 Conclusion

In this chapter, we have presented a detailed description of our leaf recognition system with the performance evaluation of the segmentation phase and recognition.

We have developed a leaf recognition system that has many stages: preprocessing, segmentation, feature extraction, creating an SVM model, classification and recognition.

Our database has many classes with a similar shape leaves and that a challenge for the recognition.

Anyway, with Generic Fourier Descriptor (GFD) we get a good dataset of features, this helped us by using the SVM method to get a well recognition results.

General Conclusion

The field of leaves examination is one of the widest scopes of scientific research. This field is highly important and greatly beneficial; a high percentage of drugs industry depends on the substances extracted from the plants. As the field of agriculture is witnessing a remarkable development, too many countries are totally depending on the field of leaves examination and scientific research. Computer science as a field has a further addition in providing ways of leaves examination and classification.

In our work we have worked on different type of leaves with different shapes, so from many methods of extracting features, generic Fourier descriptor (GFD), this method depends on the value of radial frequency and the value of angels for extracting the features.

We also have relied on the linear SVM because it given us a good recognition results with our dataset.

Our system has been tested on a base containing 32 classes, each class consist of more than 50 images.

Our system has given very good results in terms of recognition, which shows that the efficiency of the method used to extract primitives and the strategy used to searching and selecting parameters for multi-class SVM models.

This has been an interesting and encouraging experience, but there are also opportunities of future extensions remain possible:

- Checking other types of features to further improve the recognition.
- Segmentation of leaf images with different background.

Bibliography

[1] All_About_Plants_Printable_Book. Montessori For Everyone 2018.

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

[3] Y. J. Zhang, “An Overview of Image and Video Segmentation in the last 40 years”, Proceedings of the 6th International Symposium on Signal Processing and Its Applications, pp. 144-151, 2001.

[4] T. Lindeberg and M. X. Li, "Segmentation and classification of edges using minimum description length approximation and complementary junction cues", Computer Vision and Image Understanding, vol. 67, no.1, 1997.

[5] S. Saleh, N. V. Kalyankar and S. Khamitkar, “Image segmentation by using edge detection”, (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 03, 2010.

[6]M. R. Khokher, A. Ghafoor and A. M. Siddiqui, “Image segmentation using multilevel graph cuts and graph development using fuzzy rule-based system”, IET image processing, 2012.

[7] N. Senthilkumaran and R. Rajesh, “Edge Detection Techniques for Image. Segmentation – A Survey of Soft Computing Approaches”, International Journal of Recent Trends in Engineering, Vol. 1, No. 2, May 2009

[8] S. Angelina, L. Padma Suresh and S. H. Krishna Veni, “Image Segmentation Based On Genetic Algorithm for Region Growth and Region Merging”, International Conference on Computing, Electronics and Electrical Technologies (ICCEET), 2012.

[9] M. Yambal and H. Gupta, “Image Segmentation using Fuzzy C Means Clustering: A survey”, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 2, Issue 7, July 2013.

[10] V. K. Dehariya and S. K. Shrivastava, R. C. Jain, “Clustering of Image Data Set Using K-Means and Fuzzy K- Means Algorithms”, International conference on CICN, pp. 386- 391, 2010.

[11] H.P. Narkhede, “Review on Image Segmentation Techniques ” International Journal of Science and Modern Engineering, Vol 1, Issue 8,July 2013.

[12] Wu, S.G., Bao, F.S., Xu, E.Y., Wang, Y.X., Chang, Y.F., Xiang, Q.I.: A leaf recognitionalgorithm for plant classification using probabilistic neural network.

In: IEEE 7th International Symposium on Signal Processing and Information Technology(2007)

[13] San Diego State University.Leaf Recognition andd Matching with MATLAB.VijayaBylaiah Spring 2014

[14] S. G. F. Wu, S. Bao, E. Y. Y. Xu, X. Y. Wang, F. Chang and Q. L. Xiang, "A Leaf Recognition Algorithm for Plant Classification using Probabilistic Neural Network", The 7th IEEE International Symposium on Signal Processing and Information Technology, (2007) Cairo, Egypt.

[15] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta and R. Jain, "Content-based image retrieval at the end of the early years", IEEE Trans. Pattern Anal. Mach., vol. 22, (2000)., vol. 74, (2011).

[16] S. Zhang and Y. K. Lei, "Modified locally linear discriminant embedding for plant leaf recognition", Neurocomputing

[17] EricPersoon and King-sun Fu. Shape Discrimination Using Fourier Descriptors. IEEE Trans. OnSystems, Man and Cybernetics, Vol.SMC-7(3):170-179, 1977.

[18] Charles T. Zahn and Ralph Z. Roskies. Fourier Descriptors for Plane closed Curves. IEEE Trans. On Computer, c-21(3):269-281, 1972.

[19] K. Arbter, W. E. Snyder, H. Burkhardt and G. Hirzinger. Application of Affine-Invariant Fourier Descriptors to Recognition of 3-D Objects. IEEE Trans. PAMI 12(7):640-647, 1990.

[20] P. J. van Otterloo. A contour-Oriented Approach to Shape Analysis. Prentice Hall International (UK) Ltd. C1991, pp.90-108.

[21] T. W. Rauber. Two-Dimensional Shape Description. Technical Report: GR UNINOVA-RT-10-94, University Nova de Lisboa, Portugal, 1994.

[22] HannuKauppinen, TapioSeppanen and MattiPietikainen. An Experimental Comparison of Autoregressive and Fourier-Based Descriptors in 2D Shape Classification. IEEE Trans. PAMI-17(2):201-207, 1995

[23] B.M.Mehtre, Mohan S. Kankanhalli and Wing Foon Lee. Shape Measures for Content based Image Retrieval: A Comparison. Information Processing & Management, 33(3): 319-337, 1997.

[24] C.-L. Huang and D.-H. Huang. A Content-based image retrieval system. Image and Vision

Computing, 16:149-163, 1998.

[25] D. S. Zhang and G. J. Lu. A Comparative Study on Shape Retrieval Using Fourier Descriptors
with Different Shape Signatures. In Proc. Int. Conference on Multimedia and Distance Education. Fargo, ND, USA, June 2001, pp.1-9.

[26] D. S. Zhang and G. Lu Content-Based Shape Retrieval Using Different Shape Descriptors: A Comparative Study. In Proc. of IEEE International Conference on Multimedia and Expo (ICME2001), August 22-25, 2001, Tokyo, Japan, pp.317-320.

[27] Vapnik. The Nature of Statistical Learning Theory. Springer, N.Y., 1995. ISBN 0-387-94559-8.

[28] Minoux. Mathematical Programming: Theory and Algorithms. John Wiley and Sons, 1986.

[29] Girosi. An equivalence between sparse approximation and Support Vector Machines. A.I. Memo 1606, MIT Artificial Intelligence Laboratory, 1997.

[30] Girosi. An equivalence between sparse approximation and Support Vector Machines. A.I. Memo 1606, MIT Artificial Intelligence Laboratory, 1997.

[31] Blanz, B. Sch¨olkopf, H. B¨ulthoff, C. Burges, V. Vapnik, and T. Vetter. Comparison of view–based object recognition algorithms using realistic 3D models. In C. von der Malsburg, W. von Seelen, J. C. Vorbr¨uggen, and B. Sendhoff, editors, Artificial Neural Networks — ICANN'96, pages 251 – 256, Berlin, 1996. Springer Lecture Notes in Computer Science, Vol. 1112.

[32] Crammer, K., Singer, Y.: On the algorithmic implementation of multiclass kernel-based vectormachines. J. Mach. Learn. Res. 2, 265–292 (2001)

[33] Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)

[34] Knerr, S., Personnaz., L., Dreyfus, G.: Single-layer learning revisited: a stepwise procedurefor building and training neural network. In: Fogelman, J. (ed.) Neurocomputing: Algorithms, Architectures and Applications. Springer, Berlin (1990)

[35] Platt, J., Cristianini, N., Shawe-Taylor, J.: Large margin dags for multiclass classification.In: Advances in Neural Information Processing Systems, vol. 12, pp. 547–553. MIT Press, Cambridge (2000)

[36] Weston, J., Watkins, C.: Multi-class support vector machines. In: Proceedings of ESANN99 (1999)

[37] Bredensteiner, E., Bennett, K.: Multicategory classification by support vector machines.Comput. Optim. Appl. **12**, 53–79 (1999)

[38] Mangasarian, O., Musicant, D.: Lagrangian support vector machines. J. Mach. Learn. Res. 1, 161–177 (2001)

[39] K. Bennett, N. Cristianini, J. Shawe-Taylor, and D. Wu. Enlarging the margin in perceptron decision trees. *Machine Learning (submitted.*

[40] Javed Hossain, M. Ashraful Amin, "Leaf Shape Identification Based Plant Biometrics", Proceedings of 13th International Conference on Computer and Information Technology (ICCIT), 2010, pp. 458-463.

[41] Dengsheng Zhang and Guojun LuShape Based Image Retrieval Using Generic Fourier DescriptorsGippsland School of Computing and Info. Tech. MonashUniversity

[42] Abdul kadir et. al Leaf Classification Using Shape, Color, and Texture Features , International Journal of Computer Trends and Technology- July to Aug Issue 2011

[43] Mahmood R. Golzarian Classification of images of wheat, ryegrass and brome grass species at early growth stages using principal component analysis 2011 Sep 24

[44] Faisal Ahmed Performance Analysis of Support Vector Machine and Bayesian Classifier for Crop and Weed Classification from Digital Images World Applied Sciences Journal 12 (4): 432-440, 2011

[45] B. Sathyabama et al Content Based Leaf Image Retrieval (CBLIR) Intended for e-commerce International Journal of Information Technology Convergence and Services (IJITCS) Vol.1, No.2, April 2011

[46] Ulrich Weiss et. al Plant Species Classification using a 3D LIDAR Sensor and Machine Learning Stefan Laible, Karsten Bohlmann and Andreas Zell Wilhelm-Schickard-Institute for Computer Science (WSI), University of Tuebingen, Tuebingen, Germany

[47] Yovel Y et. al Plant Classification from Bat-Like Echolocation Signals Matthias Otto Franz, Peter Stilz, Hans-Ulrich Schnitzler March 21, 2008