VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belgaum – 590 018

A project report on

"Classification of Selected Plants Leaf based on Characteristics as Medicinal"

submitted in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

INFORMATION SCIENCE & ENGINEERING

by

K P Ramanath (1CR16IS035)
Girish K J (1CR16IS034)
Dikshith S Rai (1CR16IS031)
Amaresh Hubballi (1CR16IS012)

Under the guidance of

Mrs. Akhilaa
Assistant Professor
Dept. of ISE, CMRIT, Bengaluru

CMR INSTITUTE OF TECHNOLOGY

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

#132, AECS Layout, IT Park Road, Bengaluru-560037

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belgaum – 590 018

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

Certificate

This is to certify that the project entitled, **“Classification of Selected Plants Leaf based on Characteristics as Medicinal”**, is a bonafide work carried out by **K P Ramanath (1CR16IS035), Girish K J (1CR16IS034), Dikshith S Rai (1CR16IS031), and Amaresh Hubballi (1CR16IS012)** in partial fulfillment of the award of the degree of Bachelor of Engineering in Information Science & Engineering of Visvesvaraya Technological University, Belgaum, during the year 2019-20. It is certified that all corrections/suggestions indicated during reviews have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the said Degree.

Name & Signature of Guide	Name & Signature of HOD	Signature of Principal
Mrs. Akhilaa	Dr. M Farida Begam	Dr. Sanjay Jain

External Viva

Name of the Examiners	**Signature with date**
1.	
2.	

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belgaum – 590 018

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

Certificate

We, **K P Ramanath (1CR16IS035), Girish K J (1CR16IS034), Dikshith S Rai (1CR16IS031), and Amaresh Hubballi (1CR16IS012),** bonafide students of CMR Institute of Technology, Bangalore, hereby declare that the report entitled **“Classification of Selected Plants Leaf based on Characteristics as Medicinal”** has been carried out by us under the guidance of **Mrs Akhilaa, Assistant Professor,** CMRIT Bangalore, in partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering in **Information Science Engineering**, of the Visvesvaraya Technological University, Belgaum during the academic year 2019-2020. The work done in this dissertation report is original and it has not been submitted for any other degree in any university.

K P Ramanath (1CR16IS035)

Girish K J (1CR16IS034)

Dikshith S Rai (1CR16IS031)

Amaresh Hubballi (1CR16IS012)

ABSTRACT

Plants are considered as one of the greatest assets in the field of Indian Science of Medicine called Ayurveda. Some plants have its medicinal values apart from serving as the source of food. The innovation in the allopathic medicines has degraded the significance of these therapeutic plants. People failed to have their medications at their door step instead went behind the fastest cure unaware of its side effects. One among the reasons is the lack of knowledge about identifying medicinal plants among the normal ones. As the days pass it is becoming difficult for the people to identify the existence of the medicinal plants. Many are unaware of these plants. So, there is an urgent need for us to identify them and regrow them for the use of future generations. Leaf Identification by mechanical means often leads to wrong identification. Due to growing illegal trade and malpractices in the crude drug industry on one hand and lack of sufficient experts on the other hand, an automated and reliable identification and classification mechanism in order to handle the bulk of data and to curb the malpractices is needed. This project aims at implementing such system using image processing with images of the plant leaves as a basis of classification.

ACKNOWLEDGMENT

The satisfaction and euphoria that accompany a successful completion of any task would be incomplete without the mention of people who made it possible, success is the epitome of hard work and perseverance, but steadfast of all is encouraging guidance.

So, it is with gratitude that we acknowledge all those whose guidance and encouragement served as beacon of light and crowned our effort with success.

We would like to thank **Dr. Sanjay Chitnis**, Principal, CMRIT, Bangalore, for providing an excellent academic environment in the college and his never-ending support for the B.E program.

We would like to express our gratitude towards **Dr. M Farida Begam**, Assoc Professor and HOD, Department of Information Science and Engineering CMRIT, Bangalore, who provided guidance and gave valuable suggestions regarding the project.

We consider it a privilege and honor to express my sincere gratitude to our guide **Mrs. Akhilaa**, Assistant Professor, Department of Information Science & Engineering, for her valuable guidance throughout the tenure of this review.

We would also like to thank all the faculty members who have always been very Co-operative and generous. Conclusively, we also thank all the non-teaching staff and all others who have done immense help directly or indirectly during our project.

K P Ramanath (1CR16IS035)
Girish K J (1CR16IS034)
Dikshith S Rai (1CR16IS031)
Amaresh Hubballi (1CR16IS012)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

PREAMBLE

1.1 Introduction

Plants are an indispensable part of our ecosystem and the dwindling number of plant varieties is a serious concern. To conserve plants, their rapid identification by botanists is a must, thus a tool is needed which could identify plants using easily available information. There is a growing scientific consensus that plant habitats have been altered and species are disappearing at rates never witnessed before. The biodiversity crisis is not just about the perilous state of plant species but also of the specialists who know them.

This initially requires data about various plant varieties, so that they could be monitored, protected and can be used for future. Plants form the backbone of Ayurveda and today's Modern-day medicine and are a great source of revenue. Due to Deforestation and Pollution, lot of medicinal plant leaves have almost become extinct. So, there is an urgent need for us to identify them and regrow them for the use of future generations. Leaf Identification by mechanical means often leads to wrong identification. Due to growing illegal trade and malpractices in the crude drug industry on one hand and lack of sufficient experts on the other hand, an automated and reliable identification and classification mechanism in order to handle the bulk of data and to curb the malpractices is needed. To overcome this problem, automatic identification and classification of medicinal plants is needed for greater benefit to humankind. In today’s era, the automatic identification and classification of medicinal plants is an active research area in the field of image processing. Feature extraction and classification are the main steps in identification of medicinal plants and classification process which affect the overall accuracy of the classification system. Plants are identified based on leaves, flowers, bark, seed, fruits, roots, stem and other parameters like height, region of its growth and environmental factors.

For identification of plants many authors consider only leaves of the plants, because leaves are of two-dimensional nature and are available at all the time. But less research is done in identification of medicinal plants using flower and fruits/seeds because they are three-dimensional in nature and available only in specific seasons. This paper will give brief review about medicinal plants identification and classification using different technologies used in preprocessing, feature extraction and classification phases.

1.2 Existing System

- Several studies have been conducted in order to develop tools for the identification of plants during the last 10 years. One of the most authoritative works in the field of plant classification has been done by Wu et al.
- Using a different dataset but the same classifier, Hossain and Amin (2010) achieved a similar level of accuracy with similar features.
- Using machine learning techniques, Adams Begue et al. proposed a method for automatic identification of medicinal plants [3] using leaf features. Five different classifiers are used for classification purpose. Among them 90.1% accuracy is obtained from random forest classifier than k-nearest neighbour, naive Bayes (NB), SVM and neural networks.

1.2.1 Drawbacks

- The primary disadvantage of existing machine learning methods is that they cannot extract the optimized feature points, because the learning and classification processes are performed independently.
- It is still not very robust to rotation.

1.3 Proposed System

- The first step of training the data is preprocessing the input data one by one. The main purpose of pre-processing the data is to enhance the visual appearance of the image and to improve the manipulation of datasets.
- Various machine learning models can be trained to propose an accurate model based on features of leaf and other related data available on the dataset.
- The data will be used to train the model to predict the species of plant, classify whether it is medicinal or not and if it is medicinal what is the use of it.

1.4 Plan of Implementation

- Five species are selected and the images of those species are collected from different websites.
- The first step will be extracting and cleaning the dataset. We will use preprocessing techniques to deal with any missing or erroneous values.

- Once the data has been preprocessed, we will perform data analysis. This will help in finding relationships between the various attributes in the dataset, giving us an idea about the importance of each attribute.
- After pre-processing and analysis of the data, we will test the dataset with different machine learning models by training them, attempting to find a solution which gives the highest possible accuracy.
- Once the model is trained, the model will be developed as software for identification and classification of plants leaf based on their medicinal behaviour.

1.5 Problem Statement

Medicinal plants have received much attention since they are generally perceived as safe and accessible for human utilization. However, the proper identification of plant species has major benefits for a wide range of stakeholders. An attempt is made to use machine learning algorithms to identify and classify plants leaf based on their medicinal behaviour and hence provide better plant leaf classification and behavioural analysis system.

1.6 Objective of the Project

- The main objective of the project is to design a system which is capable of identifying and classifying plants leaf based on their medicinal behaviour.
- Data for training and prediction are to be obtained from a dummy dataset having 5 different species of plant leaf images.
- Different machine learning models is to be trained and tested to obtain as high accuracy as possible.

Chapter 2

LITERATURE SURVEY

Dahigaonkar, Tejas D., and Rasika T. Kalyane. "Identification of ayurvedic medicinal plants by image processing of leaf samples." ***International Research Journal of Engineering and Technology (IRJET), 5 (5)*** **(2018): 351-355.**

The process would go as follows. The given image is first preprocessed. It involves sharpening that was done by using a 2-D spatial unsharp filter and segmentation by thresholding technique. After preprocessing, the image will be ready for feature extraction. From this preprocessed image, parameter values of different features will be extracted. After preprocessing is done, the geometric features namely Extent, Solidity, Eccentricity, Equivalent Diameter are calculated. After this color features namely mean and standard deviation are calculated. Followed by that, texture analysis is done from the Gray Level Co-occurrence matrix of the image and contrast, correlation and entropy are calculated. Lastly, the shape of the image is calculated by using erosion technique.

Gopal, A., Reddy, S.P. and Gayatri, V., 2012, December. Classification of selected medicinal plants leaf using image processing. In ***2012 International Conference on Machine Vision and Image Processing (MVIP)*** **(pp. 5-8). IEEE.**

This paper proposes and implements a system for automatic identification of selected medicinal plant leaves from a stored data base of the leaves by processing an image of their leaves. The boundarybased features, moment features & colour of the leaves are used for the purpose of identification of leaf varieties. Such automated classification mechanisms can be useful for efficient classification of plant leaf species.

Venkataraman, D., and N. Mangayarkarasi. "Computer vision based feature extraction of leaves for identification of medicinal values of plants." ***2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*****. IEEE, 2016.**

This paper proposes that to identify a plant first we consider the leaves of that plant to classify them. Leaves can be classified based on various features like texture, shape and colour. The texture based feature classification has been discussed in a paper where a statistical approach has been used which gives the quantitative measure of the pixel arrangements in a region. So, the GLCM method has been used and the dissimilarity between the leaves has been found and leaves are classified.

Bandara, Madhuri, and Lochandaka Ranathunga. "Texture Dominant Approach for Identifying Ayurveda Herbal Species using Flowers." ***2019 Moratuwa Engineering Research Conference (MERCon)*****. IEEE, 2019.**

This paper presents a methodology to identify herbal species using their flower images. There are studies have been done to identify plants that most of them are done by using leaves and those are not specifically for Ayurveda plants. The approach extracts color, shape and different texture features from flower images and creates three different feature vectors with Haralick, Tamura and Gabor textures. Then classify each with different classifiers such as SVM, Decision Trees and K-NN. The results are compared with each other's individual performances in order to use them for identifying Ayurveda plants.

Chapter 3

PROPOSED METHODOLOGY

Figure3.1 illustrate the General model of medicinal plants identification and classification system. Following are the steps used in image processing technique for identification of medicinal plants.

- Plants images are captured from digital cameras, scanners, smart phones and plant dataset. Collected images are considered as input image to the system.
- Input image is pre-processed to remove noise, enhance, segment, filtering, cropping and resizing the images before applying classification techniques. The objective of this step is to focus only interest parts of the image and remove unrelated data from the picture. This will increase the system performance and computational speed.
- Next step is to extract the features from original plants images and then feed into the classifier for recognition purpose. Features extraction and classification is done by many image processing techniques few of them are listed in figure 3.1.
- For plants identification purpose many of the authors used texture, shape, color and edge features of leaves, flowers and fruits/seeds. Different classifiers are used for identification purpose. The comparative study of classifiers and feature extraction techniques are shown in table 1.

Because of environmental factors and lack of awareness about medicinal plants in human beings, plants are becoming extinct and rare. Botanist identifies the medicinal plants based on biological characteristics. This is lengthy process and consumes more time to identify plants species because one plant may have a similar kind of morphological features with another plant. Incorrect identification will create a bad impression about Ayurvedic medicines and produce unexpected side effects in human beings. It is difficult for a person to remember names of every medicinal plant hence it is very much essential to build an automatic identification and classification system for greater benefit. The purpose of automatic identification and classification of medicinal plants is to educate and provide correct knowledge to common people and farmers, which will help to increase the cultivation of medicinal plants.

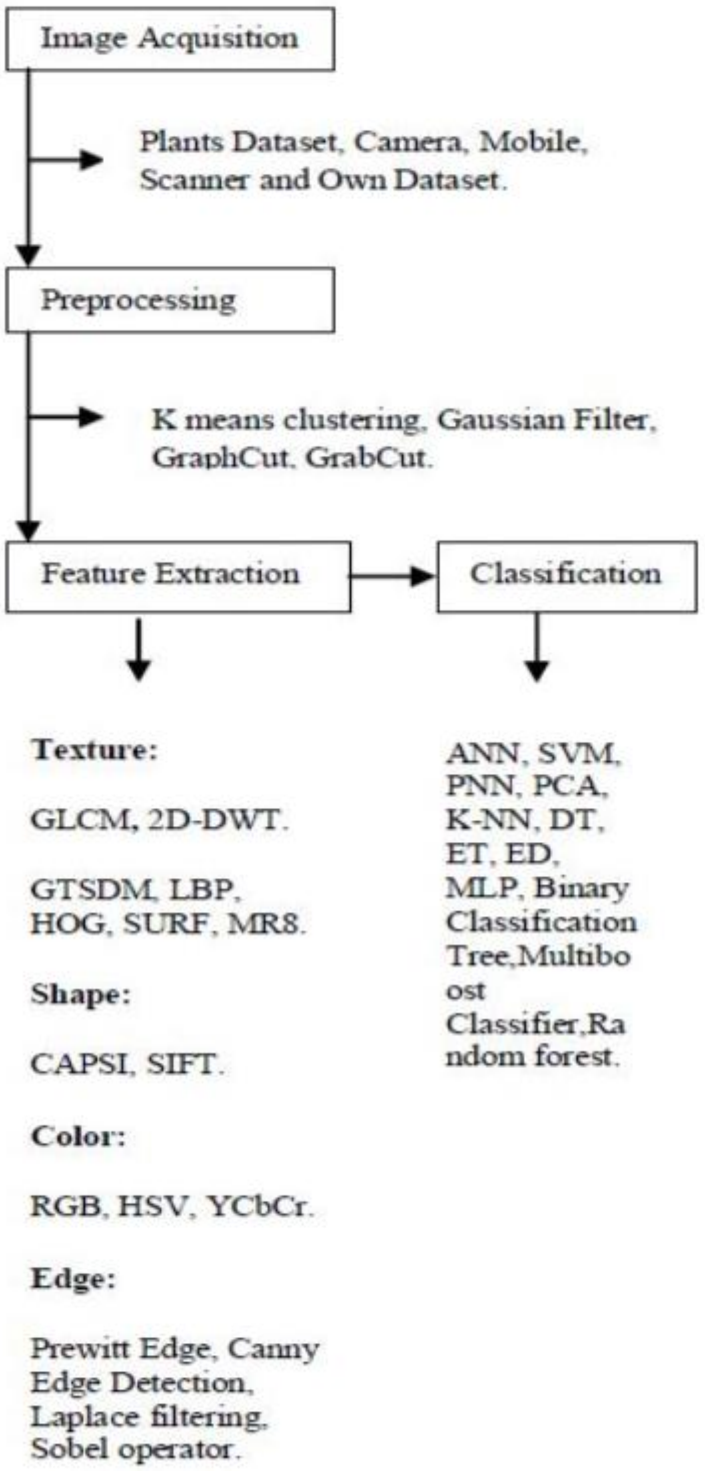
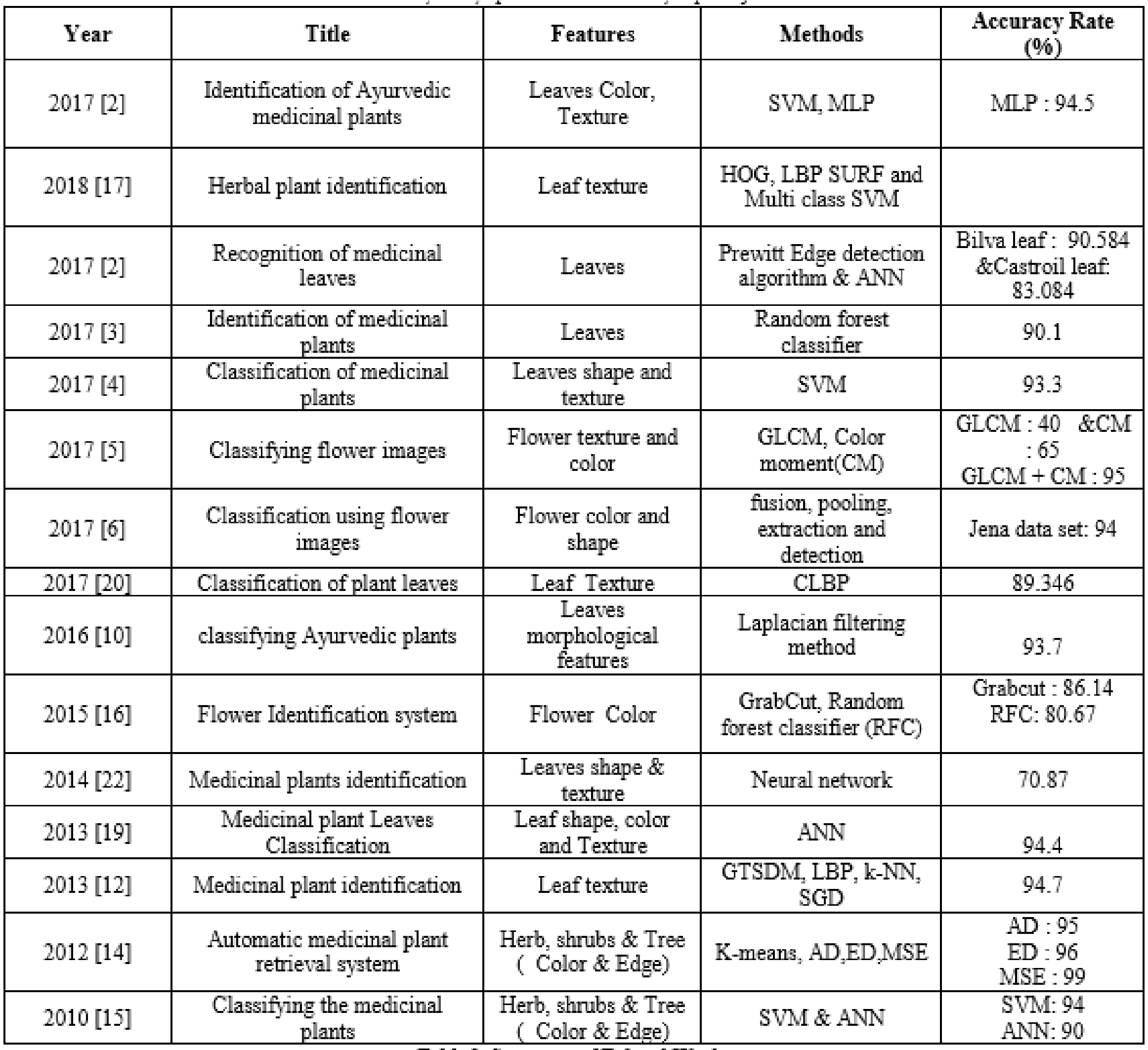


Image Acquisition
Plants Dataset, Camera, Mobile, Scanner and Own Dataset.
Preprocessing
K means clustering, Gaussian Filter, GraphCut, GrabCut.
Feature Extraction
Classification
Texture:
GLCM, 2D-DWT.
GTSDM, LBP, HOG, SURF, MR8.
Shape:
CAPSI, SIFT.
Color:
RGB, HSV, YCbCr.
Edge:
Prewitt Edge, Canny Edge Detection, Laplace filtering, Sobel operator.
ANN, SVM, PNN, PCA, K-NN, DT, ET, ED, MLP, Binary Classification Tree,Multiboost Classifier,Random forest.

Fig.3.1 A Block diagram for identification and classification of medicinal plants



Year	Title	Features	Methods	Accuracy Rate (%)
2017 [2]	Identification of Ayurvedic medicinal plants	Leaves Color, Texture	SVM, MLP	MLP : 94.5
2018 [17]	Herbal plant identification	Leaf texture	HOG, LBP SURF and Multi class SVM	
2017 [2]	Recognition of medicinal leaves	Leaves	Prewitt Edge detection algorithm & ANN	Bilva leaf : 90.584 &Castroil leaf: 83.084
2017 [3]	Identification of medicinal plants	Leaves	Random forest classifier	90.1
2017 [4]	Classification of medicinal plants	Leaves shape and texture	SVM	93.3
2017 [5]	Classifying flower images	Flower texture and color	GLCM, Color moment(CM)	GLCM : 40 &CM : 65 GLCM + CM : 95
2017 [6]	Classification using flower images	Flower color and shape	fusion, pooling, extraction and detection	Jena data set: 94
2017 [20]	Classification of plant leaves	Leaf Texture	CLBP	89.346
2016 [10]	classifying Ayurvedic plants	Leaves morphological features	Laplacian filtering method	93.7
2015 [16]	Flower Identification system	Flower Color	GrabCut, Random forest classifier (RFC)	Grabcut : 86.14 RFC: 80.67
2014 [22]	Medicinal plants identification	Leaves shape & texture	Neural network	70.87
2013 [19]	Medicinal plant Leaves Classification	Leaf shape, color and Texture	ANN	94.4
2013 [12]	Medicinal plant identification	Leaf texture	GTSDM, LBP, k-NN, SGD	94.7
2012 [14]	Automatic medicinal plant retrieval system	Herb, shrubs & Tree (Color & Edge)	K-means, AD,ED,MSE	AD : 95 ED : 96 MSE : 99
2010 [15]	Classifying the medicinal plants	Herb, shrubs & Tree (Color & Edge)	SVM & ANN	SVM: 94 ANN: 90

Table 1. Summary of Related Works

Image Acquisition:

Image Acquisition Image Acquisition refers to acquiring an image by means of camera of a leaf. Image of a leaf can be taken with the help of a digital camera. Other methods can also be used.

Image Pre-Processing:

Image Pre -processing refers to working on images in order to convert it in a suitable form on which the algorithm can be worked. The captured images will be cropped and be resized so that it can be effectively tested. In Digital image processing, computer algorithms are applied to execute image processing on digital images.

Pre-processing consists of

- Resize Image

- Filter Image
- Segment Image
- Crop Image
- Binarization

K-Means based clustering:

Assume the number of clusters, K, is known. The partitioning of data is done such that the average spread or variance of partition is minimized. Let μk (n) denote the k th cluster center at the n th iteration and Rk denote the region of k th cluster at a given iteration. Initially, we assign arbitrary values to μk (0). At the nth iteration take one of the data points xi and assign it to the cluster whose center is closest to it, that is,

$$x_i \in R_k \Leftrightarrow d\left(x_i, \mu_k(n)\right) = \min_{j=1,\ldots,K} [d(x_i, \mu_j(n))] \quad (1)$$

Where d(x,y) is the distance measure used. Re compute the cluster centers by finding the point that minimizes the distance for elements within each cluster. Thus,

$$\mu_k(n+1): \sum_{x_i \in R_k} d(x_i, \mu_k(n+1) = \min_{y} \sum_{x_i} d(x_i, y), \quad k = 1, \ldots.. K \quad (2)$$

The procedure is repeated for each xi , one at a time, until the cluster and their centers remain unchanged. If d(x,y) is the Euclidean distance, then a cluster center is simply the mean location of its elements. If K is not known, we start with a large value of K and then merge to K-1, K-2 ... clusters by suitable cluster distance measure.

Following are the steps in k-means clustering:

1. Read input image of leaf.
2. Convert RGB color space to L* a* b* color space: This conversion enables to quantify the visual difference present in the RGB image.
3. Classify the colors in a* b* space using k-means clustering. K-means clustering treats each object as having a location in space. K-means finds partition such that objects within each cluster are as close to each other as possible and as far from objects in other clusters as possible. Since the color information exists in the a* b* space. The objects of interest are the pixels with „a*" and „b*" values.
4. Label every pixel in the image using the results from k-means. For every object in the input image, k-means returns an index corresponding to a cluster and label every pixel in the image with its cluster index.

5. Create image that segment the original image by color. This step will result in k number of images of each of which is a segment of the original image that is partitioned by color. When segmentation is completed, one of the clusters contains the diseased spots being extracted [8].

Feature Extraction:

Extraction of features of an image is a property in image processing where the major attributes which have to be analyzed are extracted.

Percentage Infection Calculation (P)

After calculation of the total leaf area (AT) as well as the diseased area (AD) of the leaf, the percentage infection (P) is calculated by using the following equation:

$$P = A_D / A_T \times 100$$

Classification:

In order to identify plants that are unfamiliar and to differentiate the medicinal plants with similar features, it is necessary to classify them with distinguishing features. The large variety in medicinal plants has a very diverse range of identifying features or properties which are used for grouping them. We used easy-to-see features, physical characteristics called morphological features for identification. The features described in Chapter 4 give possible plant morphology structural features for identification. The identification of medicinal plants is an incremental process. We have felt that hierarchical identification and classification is necessary for proper medicinal plant species identification in Ayurveda. In order to explore the application of computer vision in automating different activities of 161 Indian System of Medicine called Ayurveda, we have carried out the connected literature survey to know the state of the art.

3.1 Open CV

It is a library of programming1functions mainly aimed at real-time1computer vision. It is developed by Intel research center and subsequently supported by1Willow Garage and now maintained by itseez. It is written in C++ and its primary interface is also in C++. Its binding is in Python, Java, and Mat lab. OpenCV runs on a variety of platform i.e. Windows, Linux, and MacOS, openBSD in desktop and Android, IOS and Blackberry in mobile. It is used in diverse purpose for facial recognition, gesture recognition, object identification, mobile robotics, segmentation etc. It is a com-

-bination of OpenCV C++ API and Python language. In our project we are using OpenCV version 2 OpenCV is used to gesture control to open a camera and capture the image. It is also used in the image to text and voice conversion technique.

3.2 Tensor Flow:

TensorFlow [1] is an interface for expressing machine learning algorithms, and an implementation for executing such algorithms. A computation expressed using TensorFlow can be executed with little or no change on a wide variety of heterogeneous systems, ranging from mobile devices such as phones and tablets up to large-scale distributed systems of hundreds of machines and thousands of computational devices such as GPU cards. The system is flexible and can be used to express a wide variety of algorithms, including training and inference algorithms for deep neural network models, and it has been used for conducting research and for deploying machine learning systems into production across more than a dozen areas of computer science and other fields, including speech recognition, computer vision, robotics, information retrieval, natural language processing, geographic information extraction, and computational drug discovery. This paper describes the TensorFlow interface and an implementation of that interface that we have built at Google. The TensorFlow API and a reference implementation were released as an open-source package under the Apache 2.0 license in November, 2015 and are available at www.tensorflow.org.

In a TensorFlow graph, each node has zero or more inputs and zero or more outputs, and represents the instantiation of an operation. Values that flow along normal edges in the graph (from outputs to inputs) are tensors, arbitrary dimensionality arrays where the underlying element type is specified or inferred at graph-construction time. Special edges, called control dependencies, can also exist in the graph: no data flows along such edges, but they indicate that the source node for the control dependence must finish executing before the destination node for the control dependence starts executing. Since our model includes mutable state, control dependencies can be used directly by clients to enforce happens before relationships. Our implementation also sometimes inserts control dependencies to enforce orderings between otherwise independent operations as a way of, for example, controlling the peak memory usage.

Chapter 4

SYSTEM REQUIREMENT SPECIFICATION

A System Requirement Specification (SRS) is basically an organization's understanding of a customer or potential client's system requirements and dependencies at a particular point prior to any actual design or development work. The information gathered during the analysis is translated into a document that defines a set of requirements. It gives the brief description of the services that the system should provide and also the constraints under which, the system should operate. Generally, SRS is a document that completely describes what the proposed software should do without describing how the software will do it. It's a two-way insurance policy that assures that both the client and the organization understand the other's requirements from that perspective at a given point in time.

SRS document itself states in precise and explicit language those functions and capabilities a software system (i.e., a software application, an ecommerce website and so on) must provide, as well as states any required constraints by which the system must abide. SRS also functions as a blueprint for completing a project with as little cost growth as possible. SRS is often referred to as the "parent" document because all subsequent project management documents, such as design specifications, statements of work, software architecture specifications, testing and validation plans, and documentation plans, are related to it.

Requirement is a condition or capability to which the system must conform. Requirement Management is a systematic approach towards eliciting, organizing and documenting the requirements of the system clearly along with the applicable attributes. The elusive difficulties of requirements are not always obvious and can come from any number of sources.

4.1 Functional Requirements

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality. Following are the functional requirements on the system:

• Dataset will be cleaned and vectorized and sent to ML models.

• Model will be trained based on our own dataset

• Model will be able to predict whether the input leaf has some medicinal characteristics or not.

• If the leaf has some medicinal characteristics than those characteristics will be displayed.

4.1.1 Dataset

We have created our dataset as the dataset available on the internet is mix of many species. These data have been downloaded from various websites which provide dataset. Few of the important websites are Google Dataset, Kaggle, UCI Machine Learning Repository and Dataset for You. The database consists of 5 different species of plants leaf. We have around 50 pictures of each species.

4.2 Non-Functional Requirements

Non-functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviours. They may relate to emergent system properties such as reliability, response time and store occupancy. Non-functional requirements arise through the user needs, because of budget constraints, organizational policies and the need for interoperability with other software and hardware systems.

4.2.1 Product Requirements

- Correctness: It followed a well-defined set of procedures and rules to engage a conversation with the user and a pre-trained classification model to compute also rigorous testing is performed to confirm the correctness of the data.
- Modularity: The complete product is broken up into many modules and well-defined interfaces are developed to explore the benefit of flexibility of the product.
- Robustness: This software is being developed in such a way that the overall performance is optimized and the user can expect the results within a limited time with utmost relevance and correctness. Non-functional requirements are also called the qualities of a system. These qualities can be divided into execution quality and evolution quality. Execution qualities are

security and usability of the system which are observed during run time, whereas evolution quality involves testability, maintainability, extensibility or scalability.

Non-functional requirements are also called the qualities of a system. These qualities can be divided into execution quality and evolution quality. Execution qualities are security and usability of the system which are observed during run time, whereas evolution quality involves testability, maintainability, extensibility or scalability.

4.2.2 Basic Operational Requirements

The customers are those that perform the eight primary functions of systems engineering, with special emphasis on the operator as the key customer. Operational requirements will define the basic need and, at a minimum, will be related to these following points:

- Mission profile or scenario: It describes the procedures used to accomplish mission objective. It also finds out the effectiveness or efficiency of the system.
- Performance and related parameters: It point out the critical system parameters to accomplish the mission.
- Utilization environments: It gives a brief outline of system usage. Finds out appropriate environments for effective system operation.
- Operational life cycle: It defines the system lifetime.

4.2.3 System Configurations

Hardware system Configuration

- Processor: 1.8 GHz or faster processor. Dual-core(minimum)
- RAM: 8 GB (for development)
- Hard Disk: 20-50 GB of free space
- Display: 1024x768 minimum screen resolution.

Software system Configuration

- Operating System: Windows 10/8.1/Windows Server (for development) or Linux
- Coding Language: Python

- Tools and frameworks
    - Numpy
    - Matplotlib
    - OpenCV
    - Glob
    - Keras

Chapter 5

SYSTEM ANALYSIS

Analysis is the process of finding the best solution to the problem. System analysis is the process by which we learn about the existing problems, define objects and requirements and evaluates the solutions. It is the way of thinking about the organization and the problem it involves, a set of technologies that helps in solving these problems. Feasibility study plays an important role in system analysis which gives the target for design and development.

5.1 Feasibility Study

All systems are feasible when provided with unlimited resource and infinite time. But unfortunately this condition does not prevail in practical world. So it is both necessary and prudent to evaluate the feasibility of the system at the earliest possible time. Months or years of effort, thousands of rupees and untold professional embarrassment can be averted if an ill-conceived system is recognized early in the definition phase. Feasibility & risk analysis are related in many ways. If project risk is great, the feasibility of producing quality software is reduced. In this case three key considerations involved in the feasibility analysis are

• ECONOMICAL FEASIBILITY
• TECHNICAL FEASIBILITY
• SOCIAL FEASIBILITY

5.1.1 Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

5.1.2 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

5.1.3 Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

5.2 Analysis

5.2.1 Performance Analysis

For the complete functionality of the project work, the project is run with the help of healthy networking environment. Performance analysis is done to find out whether the proposed system. It is essential that the process of performance analysis and definition must be conducted in parallel.

5.1.2 Technical Analysis

System is only beneficial only if it can be turned into information systems that will meet the organization’s technical requirement. Simply stated this test of feasibility asks whether the system will work or not when developed & installed, whether there are any major barriers to implementation. Regarding all these issues in technical analysis there are several points to focus on:

Changes to bring in the system: All changes should be in positive direction, there will be increased level of efficiency and better customer service.

Required skills: Platforms & tools used in this project are widely used. So the skilled manpower is readily available in the industry.

Acceptability: The structure of the system is kept feasible enough so that there should not be any problem from the user's point of view.

5.1.3 Economical Analysis

Economic analysis is performed to evaluate the development cost weighed against the ultimate income or benefits derived from the developed system. For running this system, we need not have any routers which are highly economical. So the system is economically feasible enough.

Chapter 6

SYSTEM DESIGN

Design is a meaningful engineering representation of something that is to be built. It is the most crucial phase in the developments of a system. Software design is a process through which the requirements are translated into a representation of software. Design is a place where design is fostered in software Engineering. Based on the user requirements and the detailed analysis of the existing system, the new system must be designed. This is the phase of system designing. Design is the perfect way to accurately translate a customer's requirement in the finished software product. Design creates a representation or model, provides details about software data structure, architecture, interfaces and components that are necessary to implement a system. The logical system design arrived at as a result of systems analysis is converted into physical system design.

6.1 System development methodology

System development method is a process through which a product will get completed or a product gets rid from any problem. Software development process is described as a number of phases, procedures and steps that gives the complete software. It follows series of steps which is used for product progress. The development method followed in this project is waterfall model.

6.1.1 Model phases

The waterfall model is a sequential software development process, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Requirement initiation, Analysis, Design, Implementation, Testing and maintenance.

Requirement Analysis: This phase is concerned about collection of requirement of the system.This process involves generating document and requirement review.

System Design: Keeping the requirements in mind the system specifications are translated in to a software representation. In this phase the designer emphasizes on: algorithm, data structure, software architecture etc.

Coding: In this phase programmer starts his coding in order to give a full sketch of product. In other word system specifications are only converted in to machine readable compute code.

Implementation: The implementation phase involves the actual coding or programming of the software. The output of this phase is typically the library, executables, user manuals and additional software documentation

Testing: In this phase all programs (models) are integrated and tested to ensure that the complete system meets the software requirements. The testing is concerned with verification and validation.

Maintenance: The maintenance phase is the longest phase in which the software is updated to fulfill the changing customer need, adapt to accommodate change in the external environment, correct errors and oversights previously undetected in the testing phase, enhance the efficiency of the software.

6.1.2 Advantages of the Waterfall Model

- Clear project objectives.
- Stable project requirements.
- Progress of system is measurable.
- Strict sign-off requirements.
- Helps you to be perfect.
- Logic of software development is clearly understood.
- Production of a formal specification
- Better resource allocation.
- Improves quality. The emphasis on requirements and design before writing a single line of code ensures minimal wastage of time and effort and reduces the risk of schedule slippage.
- Less human resources required as once one phase is finished those people can start working on to the next phase.
- A schedule can be set with deadlines for each stage of development and a product can proceed through the development process like a car in a car-wash, and theoretically, be delivered on time.
- It allows for departmentalization and managerial control. Simple and easy to understand and use.
- Easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.

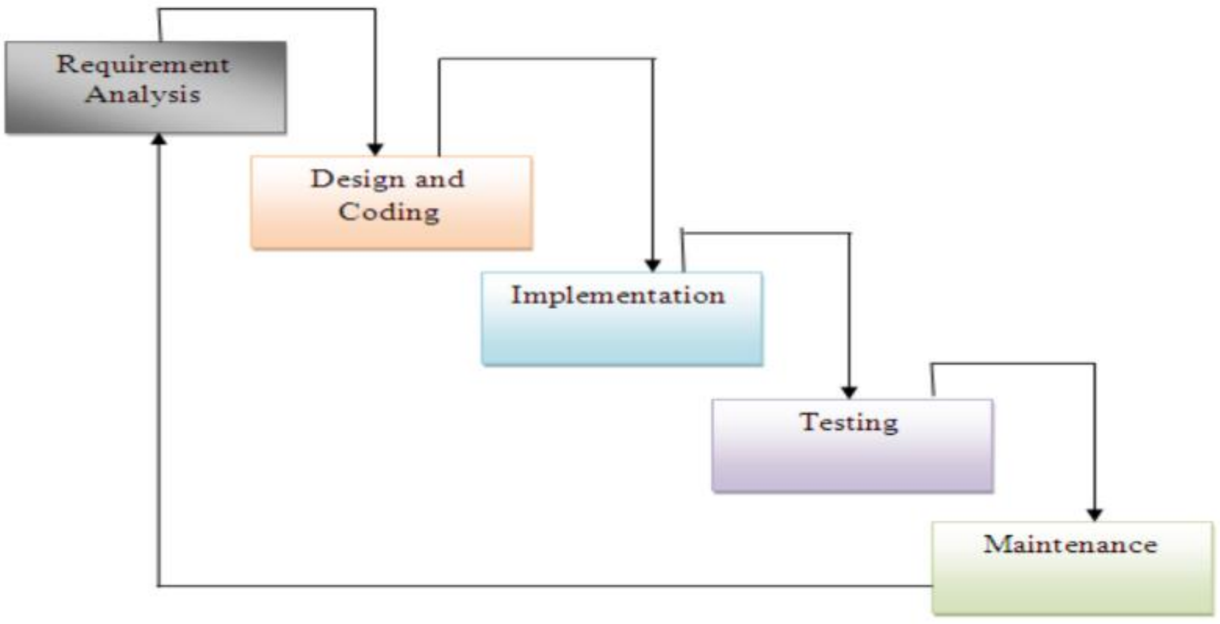
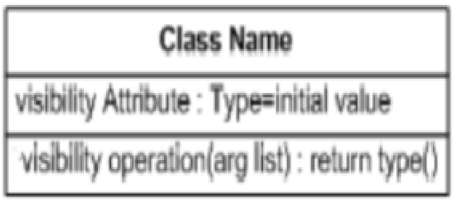
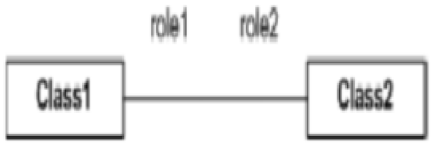
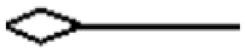
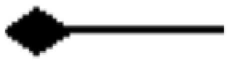
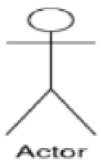


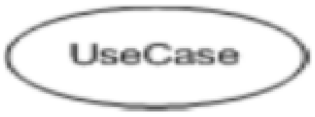
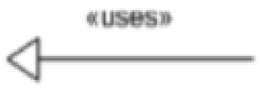
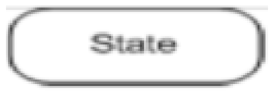
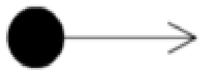
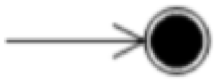
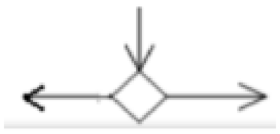
Fig 6.1: Waterfall model

6.2 Design Using UML

Designing UML diagram specifies, how the process within the system communicates along with how the objects with in the process collaborate using both static as well as dynamic UML diagrams since in this ever-changing world of Object Oriented application development, it has been getting harder and harder to develop and manage high quality applications in reasonable amount of time. As a result of this challenge and the need for a universal object modeling language every one could use, the Unified Modeling Language (UML) is the Information industries version of blue print. It is a method for describing the systems architecture in detail. Easier to build or maintains system, and to ensure that the system will hold up to the requirement changes.

A picture is worth a thousand words, this idiom absolutely fits describing UML. Object-oriented concepts were introduced much earlier than UML. At that point of time, there were no standard methodologies to organize and consolidate the object-oriented development. It was then that UML came into picture. There are a number of goals for developing UML but the most important is to define some general-purpose modeling language, which all modelers can use and it also needs to be made simple to understand and use.

Sl. No	Symbol Name	Symbol	Description
1	Class	 Class Name visibility Attribute : Type=initial value visibility operation(arg list) : return type()	Classes represent a collection of similar entities grouped together.
2	Association		Association represents a static relation between classes.
3	Aggregation		Aggregation is a form of association. It aggregates several classes into a single class.
4	Composition		Composition is a special type of aggregation that denotes a strong ownership between classes.
5	Actor		Actor is the user of the system that reacts with the system.

6	Use Case		A use case is an interaction between system and the external environment.
7	Relation (Uses)		It is used for additional purpose communication.
8	Communication		It is the communication between use cases.
9	State		It represents the state of process. Each state goes through various flows.
10	Initial State		It represents initial state of object.
11	Final State		It represents final state of object.
12	Control Flow		It represents decision making process for object.
13	Decision Box		It represents the decision making process from a constraint.

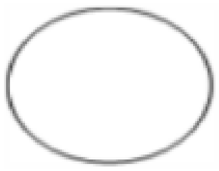
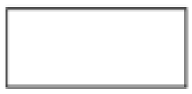
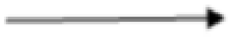
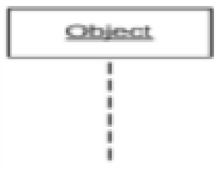
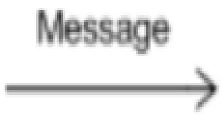
14	Data Process/ State		A circle in a DFD represents a state or process which has been triggered due to some other event or action.
15	External Entity		It represents external entity such as Keyboard, sensors, etc which are used in the system.
16	Transition		It represents any communication that occurs between processes.
17	Object Lifeline	 	Object lifeline represents the vertical dimension that object communicates.
18	Message	 	It represents messages exchanged.

Table 6.1: Symbols used in UML

6.3 Data Flow Diagram

The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data is generated by the system.

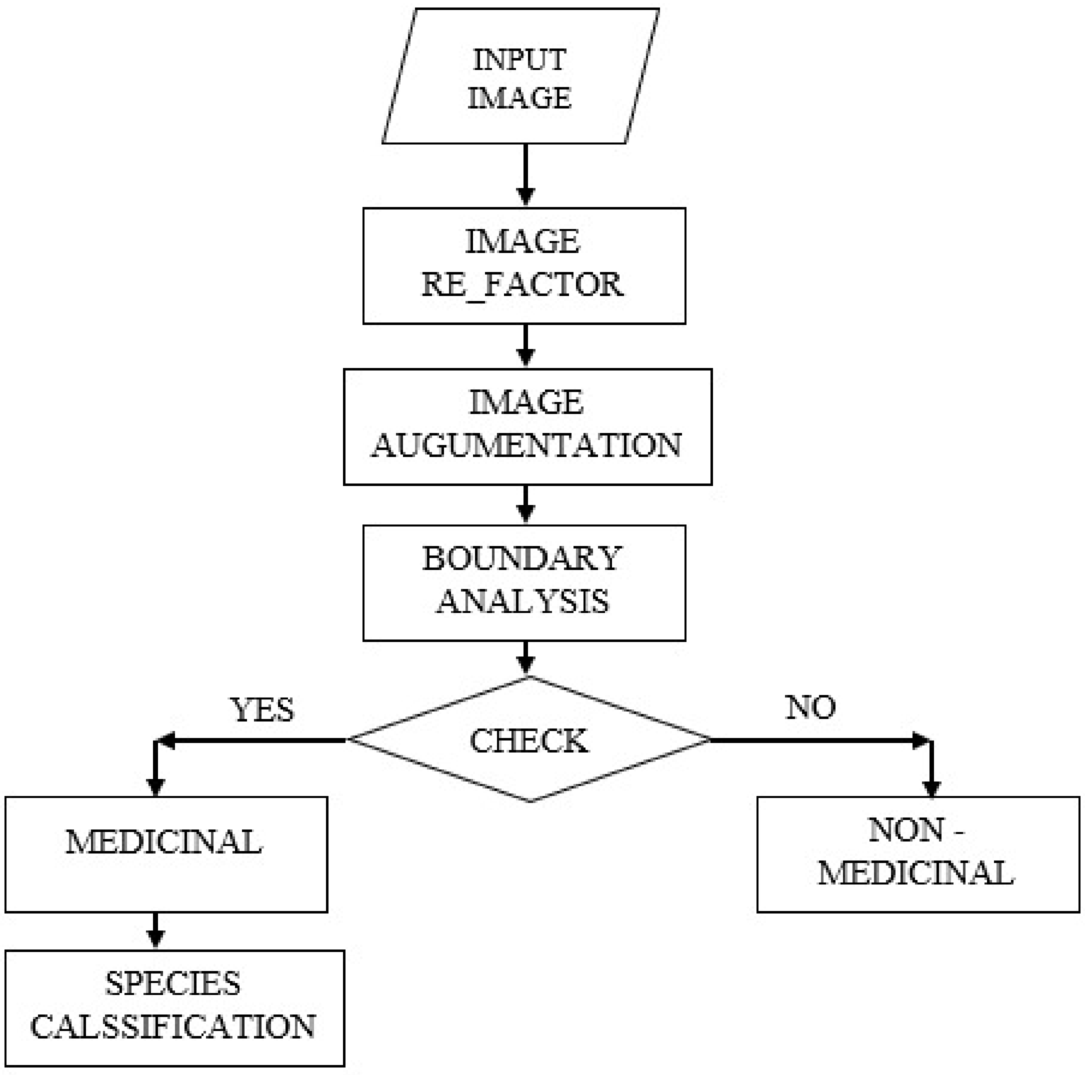


Fig 6.2 Data Flow Diagram

6.4 Component Diagram

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitra-

-rily complex systems.

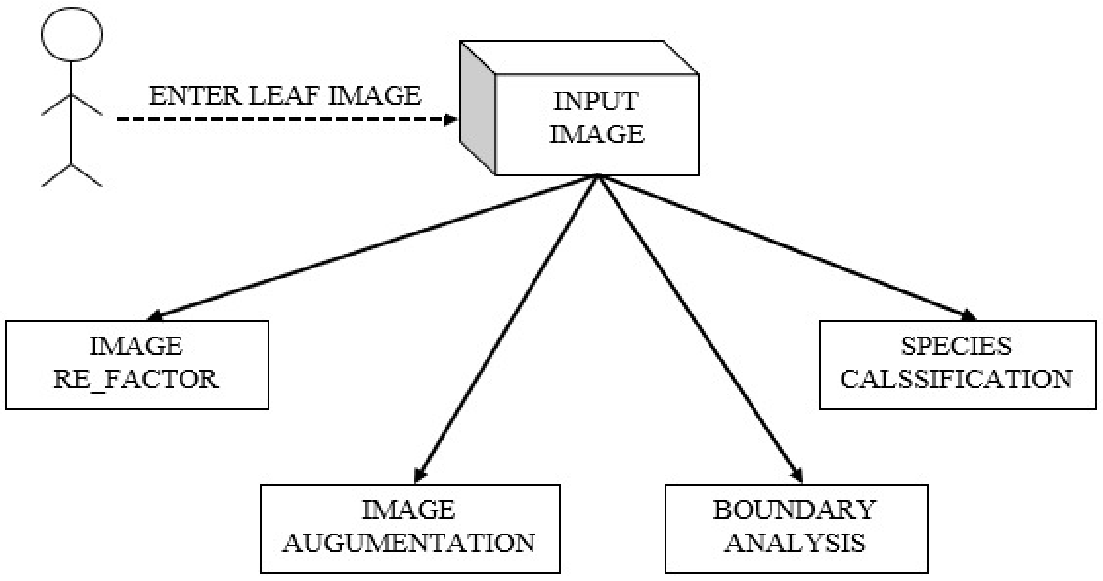


Fig 6.3 Component Diagram

The component diagram for the decentralized system ideally consists of different modules that are represented together via a common module for the user. The user is required to have the input files in the current folder where the application is being used.

6.5 Use case Diagram:

A use case defines a goal-oriented set of interactions between external entities and the system under consideration. The external entities which interact with the system are its actors. A set of use cases describe the complete functionality of the system at a particular level of detail and it can be graphically denoted by the use case diagram.

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

In software and systems engineering, a use case is a list of steps, typically defining interactions between a role (known in Unified Modeling Language (UML) as an "actor") and a system, to achieve a goal. The actor can be a human, an external system, or time.

In systems engineering, use cases are used at a higher level than within software engineering, often representing missions or stakeholder goals. The detailed requirements may then be captured in Systems Modeling Language (SysML) or as contractual statements.

The Seqeunce of activities that are carried out are the same as the other diagrams .Use case for this module indicates the users interaction with the system as a whole rather than individual modules .All the encryption mechanisms are carried out via the login page that redirects the user to the particular functionality that he or she wishes to implement.

6.6 Activity Diagram

An activity diagram shows the sequence of steps that make up a complex process. An activity is shown as a round box containing the name of the operation. An outgoing solid arrow attached to the end of the activity symbol indicates a transition triggered by the completion.

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams are intended to model both computational and organisational processes (i.e. workflows). Activity diagrams show the overall flow of control.

Activity diagrams are constructed from a limited number of shapes, connected with arrows. The most important shape types:

- rounded rectangles represent actions;
- diamonds represent decisions;
- bars represent the start (split) or end (join) of concurrent activities;
- a black circle represents the start (initial state) of the workflow;
- an encircled black circle represents the end (final state).

The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

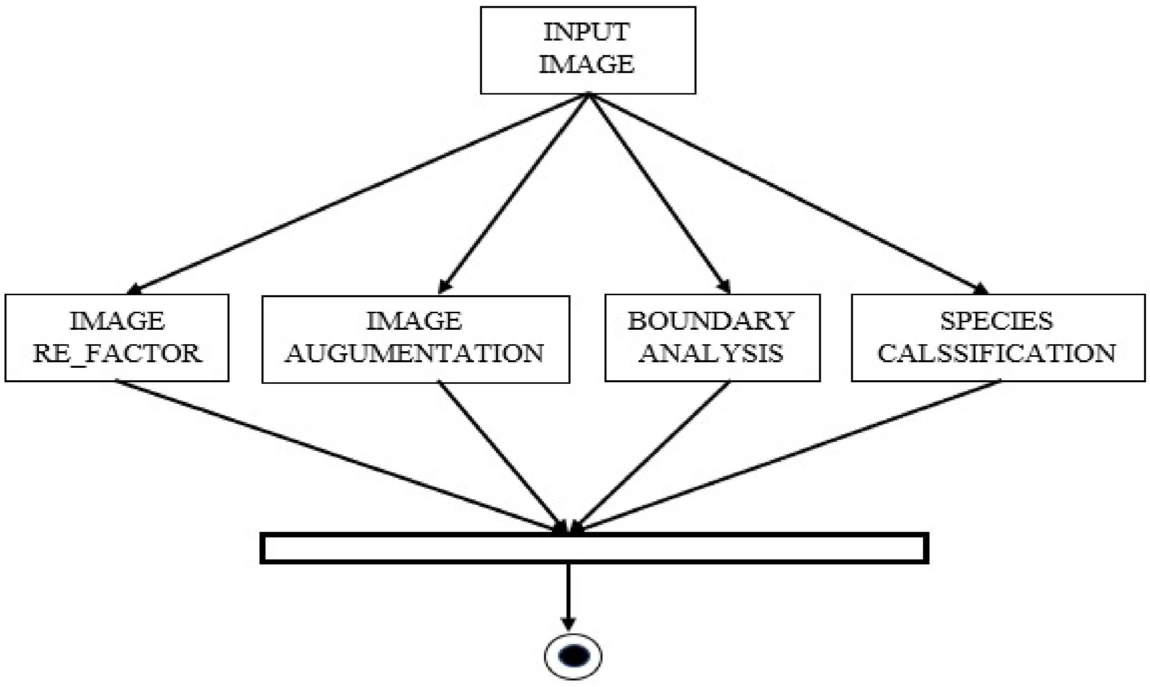


Fig 6.4 Activity Diagram

Chapter 7

IMPLEMENTATION

7.1 Flow Diagram:

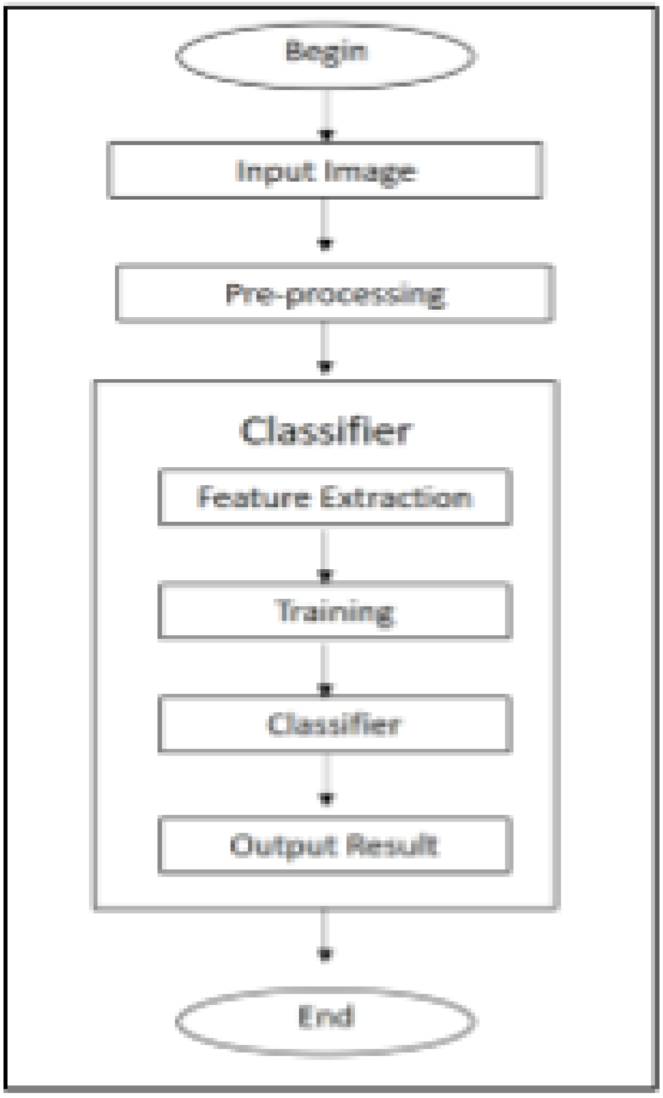


Fig 7.1. Identification and classification of Medicinal plants

7.2 Image Processing in Leaf:

Leaf pattern recognition usually follows the steps as shown in Figure 1. The most challenging part of this study is to extract distinctive features of leaves for plant species recognition. In this case, different classifiers using high performance statistical approaches have been used to perform leaf features extraction and classification. The advancement in computer vision and artificial intelligence have greatly assisted researchers to classify plants through statistical modeling-T

The pre-processing step consists of image re-orientation, cropping, gray scaling, binary thresholding, noise removal, contrast stretching, threshold inversion, and edge recognition. Image reorientation is aligning the input image to a standardized position, with the leaf aligned to either the x-axis or y-axis. For leaves that have the greater width: Length ratio, the length is preferably placed in the vertical or upright position [10]. To decrease the amount of computational load that is exerted upon the graphic processing unit, cropping the image is a necessary step to reduce the unnecessary fore-ground region of the prompt image. Turkoglu and Hanbay [11] suggested that leaf feature extraction could be done by dividing the leaf image into two or four parts, instead of extracting for the whole leaf. The proposed image processing techniques using color, vein, Fourier descriptors (FD), and gray-level co-occurrence matrix (GLCM) methods had proven to achieve 99.1% accuracy using the Flavia leaf dataset.

Gray scale conversion of the image into geometrical data is implemented to optimize the contrast and intensity of images. Later, the thresholding process creates a binary image from the gray scaled image to translate the value of the image to its closest threshold, and therefore having either one of two possible values for each pixel, as presented in Figure 2. Different types of noises, such as grains, and holes, could affect digital images; therefore erosion and dilation are a series of operations implemented in order to remove the background noises. The images are considered homogenous if they do not exhibit substantial divergences between one another in terms of contrast stretching. These images, when shown in histogram representation, exhibit very narrow peaks. In homogeneity is caused by the lack of uniform lighting upon the image. The image is normalized in order to stretch the narrow range to a more dynamic range. The binary images from the process are inverted during threshold conversion, to convert the background into black. Suzuki algorithm can be utilized to extract the contours of images and further refined by diminishing the contours with small lengths with regards to its largest contour [10]. This process is known as edge recognition.

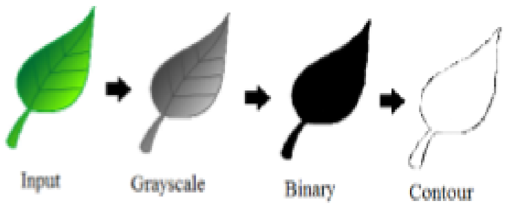


Fig 7.2 Image Pre-processing stage

7.3 Feature Extraction

Feature extraction module is a module that is used to perform feature extraction that produces identifier. Feature extraction technique used is the analysis of texture. In texture analysis obtained some identifier be a major element for the identification process. The identifier are: entropy, energy, contrast and homogeneity. Entropy measures the randomization of the intensity distribution of the image pixels. Energy is a feature to measure the concentration of the intensity in the pair co-occurrence matrix. The greater the energy value if the spouse is eligible matrix pixel intensities are concentrated in a few coordinates co-occurrences and shrink when lying spread. Another feature is the contrast that is used to measure the strength of the difference in intensity in the image. Values greater contrast if the variation in the image of high intensity and low intensity decreases when variations. The opposite of contrast is intensity variations in the image. Values greater homogeneity if the variation in the intensity of the image decreases and vice versa. The results of the database image feature extraction of the leaves is as shown in Figure

Proses Ekstraksi Citra

Fig 7.3 Feature Extraction

Mathematical Classifiers:

Upon the completion of leaf pattern extraction, the information which is also known as feature vectors are used for further inspection, comparison before being grouped into their particular classes. There are many mathematical classifiers being used by researchers. Each classifier may have its advantages and limitations.

7.4 Artificial Neural Network (ANN):

One of the relatively superior and competent classifiers is the ANN, especially in terms of its accuracy. This is because ANN is pertinent to resolve non-linear problems like leaf pattern recognition. However, previous research revealed that the leaf with oblong pattern could increase error rate of recognition possibly due to the uniform structure[26]. The basic structure of ANN is an interconnected set of nodes [28]. There are multiple layers of nodes connected each other to generate the desired output as illustrated in Figure 3.

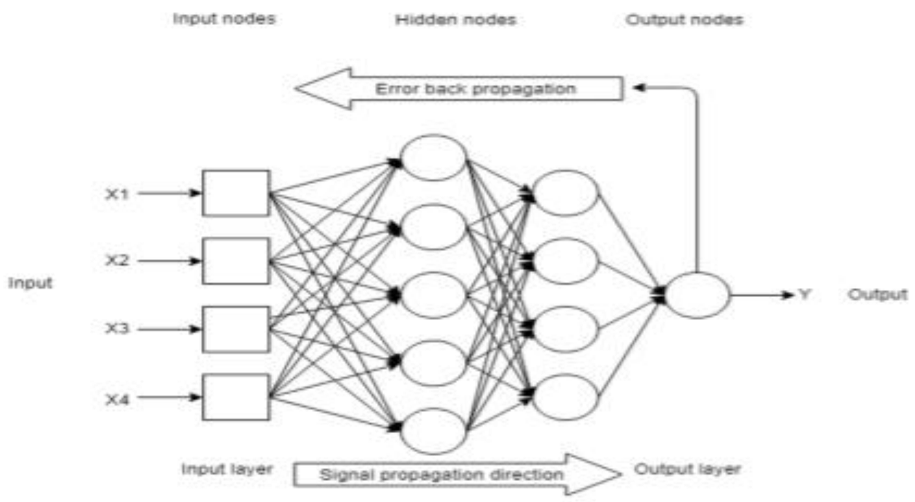


Fig 7.4 Artificial Neural Network

The data processing is directed from the front, when prompts information, to the back. When the system acknowledges the results, further training is performed and this new information is redirected to the front to reset weights to the front side of the neural units in order to increase its accuracy. This process is known as feed-forward back-propagation method, and thus increasing the accuracy of the classifier.

The input nodes of the system will be determined by the extracted features. The number of nodes of the output layers would be determined by the number of plant categories too. The classifier would be trained by utilizing back propagation method and the weight of the links will be altered to reduce the error between the expected and actual outputs.

7.5 Convolutional Neural Network (CNN):

CNN implements deep learning to machine image processing in order to classify pictures of leaf samples. Recent development of hardware and information processing technology has made deep learning a self-learning method that utilizes massive sum of data in a more feasible manner as shown in Figure 4.

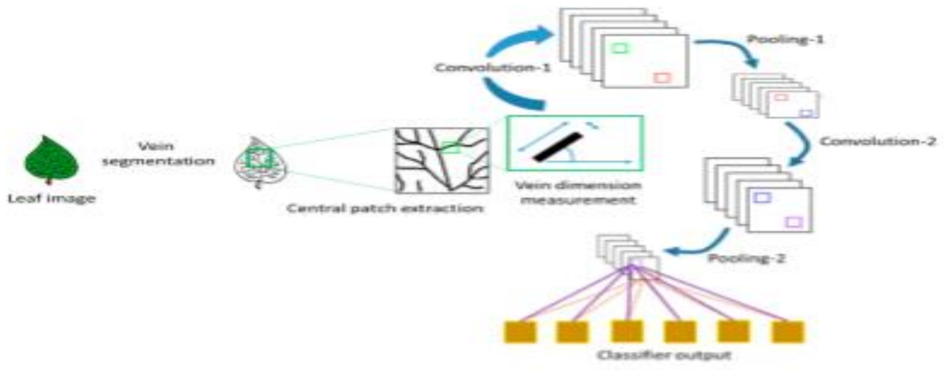


Figure 4. Leaf pattern detection with convolutional neural network.

The CNN initially aims to imitate the visual system of human. The retina identifies edges of an object by strong intensity of light compared to the whole object in the human visual system. This information is sent to lateral geniculate nucleus (LGN). The shape of the information is compressed and delivered to primary visual cortex (V1). The edge and contour of the image is interpreted at V1. Simultaneously, the image information from the retina of the left and the right eyes give depth perception which provides distance information. These data are sent to secondary visual cortex (V2) where recognition of overall shape and color perception of different segments of the image are taking place before being transferred to tertiary visual cortex (V3) to interpret the color of the whole object.

7.6 Support Vector Machine (SVM)

The idea of SVM is to define decision boundaries of feature vectors on decision plane which separates features unanimously. Since the distinction in feature between the images is evident, the images will be classified into their respective class with little to no complication. New sample is appointed in to one of the classes that match their characteristics in this training algorithm. The samples are then charted onto the same region of class where they shall be. A fine separation between classes is naturally attained by the hyperplane that marks the furthest distance to the nearest data point of each group. This is also due to the fact that the larger margin results in small generalization error of the classifier. Applying invariant feature to the SVM will vastly increase the accuracy of the classifier. The application of the invariant features goes with the contour-based descriptor. This classifier is exceptionally accurate because it is capable to recognize images without the problem of image rotation, translation, scale, and inversion or mirroring. The only drawback is that the system is not capable to differentiate leaves that have almost the same shape. In response to this situation, the principal component analysis (PCA) could be applied to enhance the performance of this system. PCA operates by converting data set into a new set of data and the variables are arranged according to their priorities. This causes the components to be uncorrelated and sorted in the order that the components with the largest variation occurs first, followed by the components with the smallest variation which is then terminated. Finally, one of the more superior classifiers would be SVM which integrates the Hu moment invariant algorithm to its model which could achieve up to 94.1% accuracy [45]. Zhang et al.[46] classified plant species using leaf shape and texture. They used SVM classifier for a dataset of 1900 leaves belonging to 32 different species. A new classification method based on the generation of space features by combining local texture features using wavelet decomposition, co-occurrence matrix statistics, and global shape features. The method was able to extract features of the plant leaf images and yielded an accuracy of over 93.8%.

A method of extracting15 features of leaf images via canny edge detector,and SVM as the classifier was reported by Salman et al. [47]. The extracted features were convex area, filled area, perimeter (P), eccentricity (E), solidity, perimeter ratio of diameter, orientation, narrow factor, extent, Euler number, diameter, circularity, rectangularity, perimeter ratio of length and width, complexity, and compactness. The study used 2220 images from 22 plant species in the Flavia dataset and achieved an accuracy of 85 to 87% only.

7.7 Probabilistic Neural Network (PNN):

Similar to CNN, PNN is another branch of ANN which incorporates additional algorithm. PNN utilizes radial basis function (RBF) which measures nonlinear variable in a shape of bell as illustrated in Figure 5.

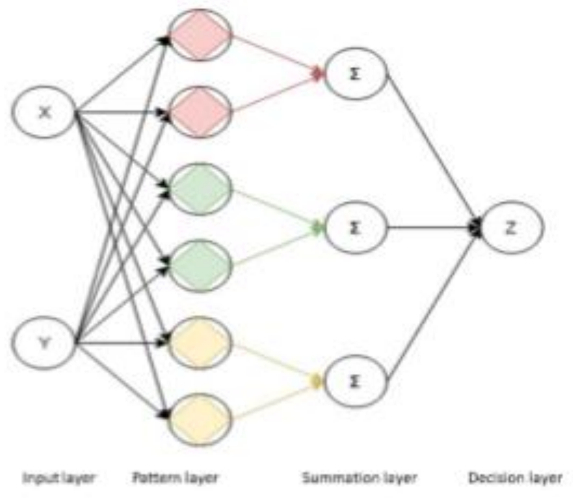


Figure 5. Mechanism of probabilistic neural network.

This PNN classifier trains the loaded feature vector with higher speed rate as compared to that of a back propagation system. Since the feature characteristic is predetermined, the classification step becomes more straightforward which makes this classifier robust to distortion. This trait also makes this classifier a simple training approach and structure.

Since the weight passes between the layers of nodes are established in the early stage, the pre-existing weights will not be manipulated. Nevertheless, new vectors are placed into weight matrices during the training phase which results in real-time viability. During the process of recognizing the leaf pattern, the feature vector of leaf is classified by the network into a particular class since the assigned class is assumed to have the highest probability to be accurate.

Previous study conducted by Stephen Gang Wu [3] managed to achieve a classification accuracy of 90% using PNN. With the same classifier also, an improved version of PNN produced better result with 93.7%accuracyintheshapefeatureextraction. When texture feature is in corporated, the accuracy of the classifier could achieve upto 98.3% [39]. Interms of network structure, PNN has the upper hand compared to ANN since it selects the nodes automatically. This also makes PNN be more flexible to determine the output [40]. Despite being convenient in classifying leaf features, this system possesses limitation where it is susceptible to data overfitting. The development of this neural network is also time consuming because it has intricate network structure.

7.8 K-Nearest Neighbor (KNN)

KNN algorithm classifies images by comparing the input images to the closest training samples from the feature space. This technique is considered as the simplest machine classifier where the training process of this algorithm only involves two fundamental steps of keeping the feature data and information of the training images. Ambiguous points are justified by assigning them to the information of KNN. Generally, the images are grouped with respect to the label of KNN by the majority of votes. Figure 6 describes the sample analysis with KNN.

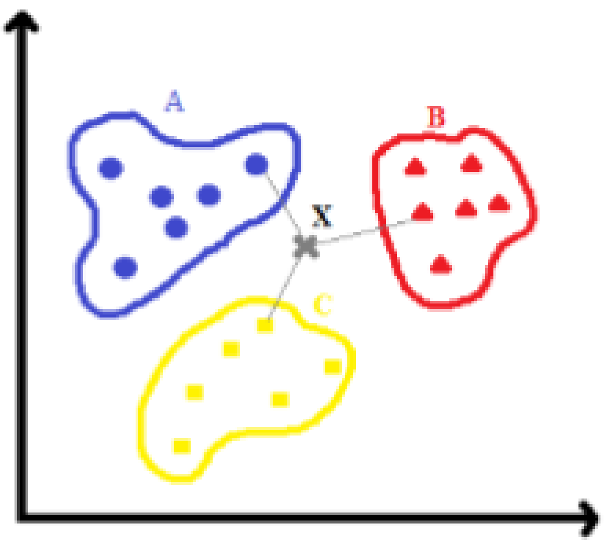


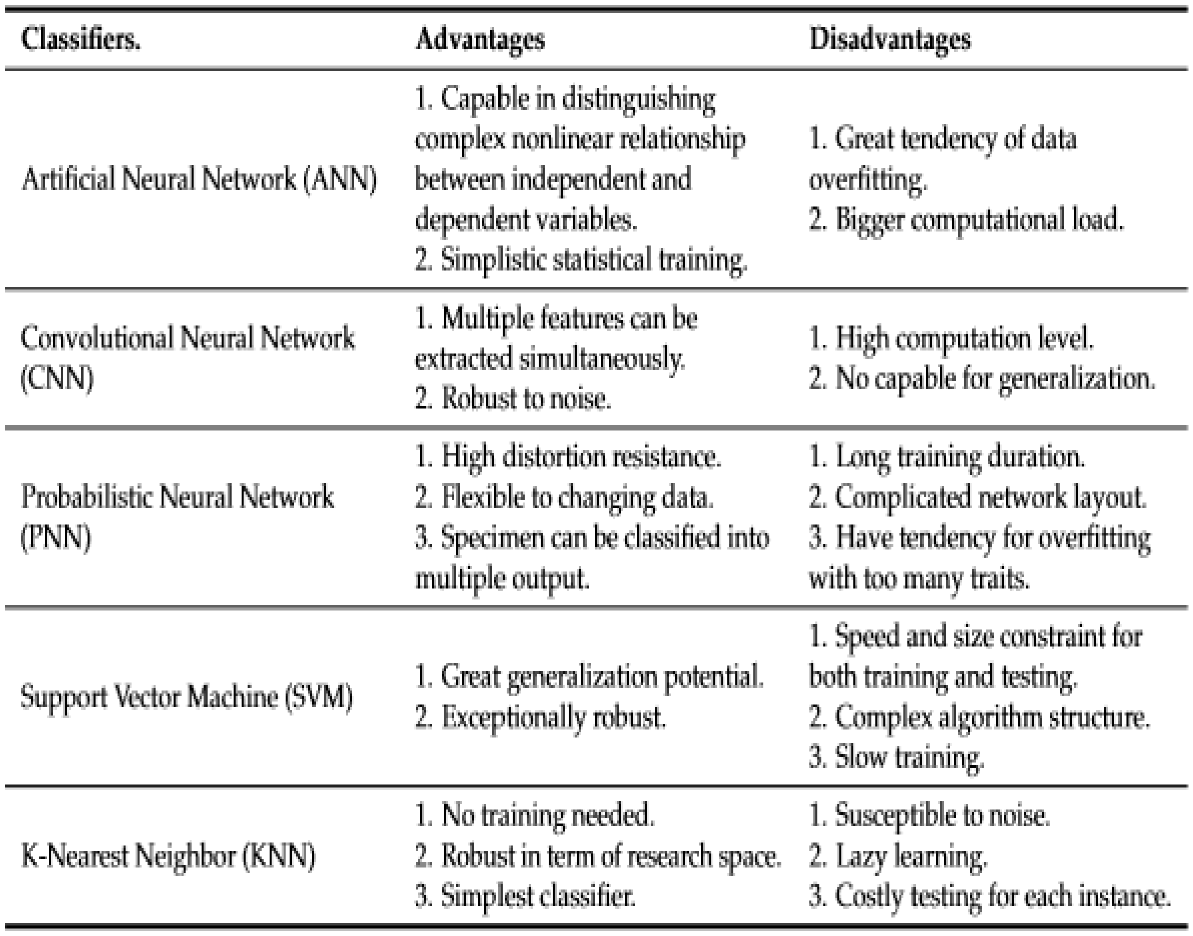
Figure 6. Sample analysis with k-nearest neighbor.

K is always an odd integer since there are only two classes. For example, images that are subjected to test will be directly classified as the class of the image close to it, if k = 1. In some cases, when testing for multi-class classification, tied outcome could be encountered when k is an odd integer. This issue could be fixed by measuring the Euclidean distance after acquiring the vector of fixed-length of the image that had been converted to real number.

The image information from the pre-processing stage will determine the features that are going to be compared by rationing the characteristics of leaf to obtain the white area ratio, perimeter to hull, hull is a ration to get the ratio of area of leaf to the area of hull which will be used for vector computation and many others. These ratios are then normalized to have a value between zero and one.

Chronologically, the image that needs to be identified will go through the same process as the images that have been stored in the data base upto normalizing the ratios. The tested image is compared to each image from the database afterward. On the other hand, comparison by color histogram could be used to find out the identity of a new leaf if the dataset from previous processes are composed of leaves from different plants.

Table 1. Comparison between advantages and disadvantages of classifiers.



Classifiers.	Advantages	Disadvantages
Artificial Neural Network (ANN)	1. Capable in distinguishing complex nonlinear relationship between independent and dependent variables. 2. Simplistic statistical training.	1. Great tendency of data overfitting. 2. Bigger computational load.
Convolutional Neural Network (CNN)	1. Multiple features can be extracted simultaneously. 2. Robust to noise.	1. High computation level. 2. No capable for generalization.
Probabilistic Neural Network (PNN)	1. High distortion resistance. 2. Flexible to changing data. 3. Specimen can be classified into multiple output.	1. Long training duration. 2. Complicated network layout. 3. Have tendency for overfitting with too many traits.
Support Vector Machine (SVM)	1. Great generalization potential. 2. Exceptionally robust.	1. Speed and size constraint for both training and testing. 2. Complex algorithm structure. 3. Slow training.
K-Nearest Neighbor (KNN)	1. No training needed. 2. Robust in term of research space. 3. Simplest classifier.	1. Susceptible to noise. 2. Lazy learning. 3. Costly testing for each instance.

KNN has advantages of using all input leaf images as test subjects before storing the collected information to the database. Increasing the number of images for testing will definitely increase the accuracy of the classifier. In other words, adding plant species to the database can also reduce the accuracy of the system. Somehow, Kherkhah and Asghari [57] reported that applying Cosine KNN classifier and PCA algorithm in gist feature vector was outperformed compared to the approaches of Patternnet neural network and SVM. This classifier is sensitive to inadequate lighting which will affect the accuracy of the system. The feature extraction is too simple with basic characteristic requirement. Future work could incorporate the algorithm that would make the system more stable to light disparity. Hence, simple classifier like KNN could obtain 83.5% accuracy. Although the feature extraction process is simple and quick, the low accuracy result is relatively weak to be accepted. This classifier is not stable enough to handle distortion from the extracted samples and would lead to inaccuracy during the classification process. A method incorporated with prescribed color histogram had proven to increase the reliability of this classifier up to 87.3% accuracy.

ANN is a unique classifier to identify the complex nonlinear relationship between independent and dependent variables. Since leaf pattern recognition is considered as a complex nonlinear problem, ANN would be able to interpret the variables effectively. ANN is also considered to be simple in terms of its statistical training. The iteration of the training mechanism is conducted one at a time, and the result is feed-forwarding into the system as a new information. Consequently, this system may cause data overfitting due to the fact that it iterates a large sum of data. This will end with huge computational load to sustain the feed-forward back-propagated data.

CNN may involve multiple features extraction, and at the same time, providing detail and quick detection. Therefore, this mechanism makes CNN be more resilient towards unnecessary noise. This classifier also utilizes a high computation level to extract multiple features. This system is not appropriate in generalizing features. Similar to CNN, PNN has a high noise resistance too. With its capability to foresee potential information, it is flexible to do data changing. Since the weight of the vector is predetermined, the specimen could be classified into multiple outputs depending on the closeness of the sample to the vector. The drawback for this classifier is the needs for long duration of training when new weight vector is applied. This system has an intricate layout structure. In case excessive traits are involved, it is prone to overfitting also.

SVM is different from other neural networks, it has great generalization potential and is exceptionally robust. The disadvantages of this system are its speed and size which are restricted for both training and testing. This classifier has a complicated algorithm structure to determine the feature vector of each sample individually. Since its generalization potential is great, it takes time for the training process before it could classify a particular set of samples. The research team of Shah et al. [60] utilized a dual-path deep CNN to learn leaf image features and optimize for classification in order to classify the leaf based on marginalized shape context and shape-texture dual-pathdeep CNN. The study revealed that the dual-path CNN method outperformed other CNN methods such as uni-patch

CNN, texture-patch CNN, marginalized shape context with SVM classifier, multiscale distance matrix with SVM classifier, curvature histogram. The method gave near-perfect-top-1-match result using Flavia dataset and top-three-match results for other datasets. A more distinct approach brought by KNN classifier is to emphasize the differences between features and compare the close feature vector, and thus make training absolutely unnecessary. This is also robust in terms of research space. The characteristics make KNN the simplest classifier. KNN classifier is very susceptible to noise which may alter the results during the classification process. Lack of training makes this classifier likely to be a lazy learning system. The prediction step in KNN is rather expensive, since every prediction is made, the classifier determines the nearest neighbor by comparing to the entire dataset.

Chapter 8

TESTING

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that all the system elements have been properly integrated and perform allocated functions. The testing process is actually carried out to make sure that the product exactly does the same thing what is supposed to do. In the testing stage following goals are tried to achieve:

- To affirm the quality of the project.
- To find and eliminate any residual errors from previous stages.
- To validate the software as a solution to the original problem.
- To provide operational reliability of the system.

8.1 Testing Methodologies

There are many different types of testing methods or techniques used as part of the software testing methodology. Some of the important testing methodologies are:

8.1.1 White box testing

White box testing (clear box testing, glass box testing, and transparent box testing or structural testing) uses an internal perspective of the system to design test cases based on internal structure. It requires programming skills to identify all paths through the software. The tester chooses test case inputs to exercise paths through the code and determines the appropriate outputs. While white box testing is applicable at the unit, integration and system levels of the software testing process, it is typically applied to the unit. While it normally tests paths within a unit, it can also test paths between units during integration, and between subsystems during a system level test.

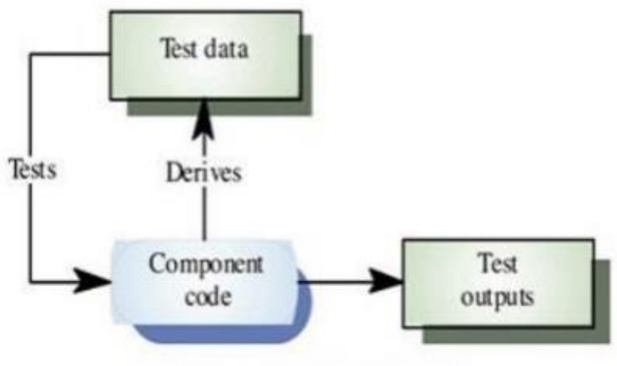


Fig 8.1.1: White Box Testing

Though this method of test design can uncover an overwhelming number of test cases, it might not detect unimplemented parts of the specification or missing requirements, but one can be sure that all paths through the test object are executed. Using white box testing we can derive test cases that:

- Guarantee that all independent paths within a module have been exercised at least once.
- Exercise all logical decisions on their true and false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Execute internal data structure to assure their validity

8.1.1.1 Advantages of White Box Testing

- To start the white box testing of the desired application there is no need to wait for user face (UI) to be completed. It covers all possible paths of code which will ensure a thorough testing.
- It helps in checking coding standards.
- Tester can ask about implementation of each section, so it might be possible to remove unused/deadlines of codes helps in reducing the number of test cases to be executed during the black box testing.
- As the tester is aware of internal coding structure, then it is helpful to derive which type of input data is needed to test the software application effectively.
- White box testing allows you to help in code optimization

8.1.1.2 Disadvantages of White Box Testing

- To test the software application a highly skilled resource is required to carry out testing who has good knowledge of internal structure of the code which will increase the cost.

- Updating the test script is required if there is change in requirement too frequently.
- If the application to be tested is large in size, then exhaustive testing is impossible.
- It is not possible for testing each and every path/condition of software program, which might miss the defects in code.
- White box testing is a very expensive type of testing.
- To test each paths or conditions may require different input conditions, so in order to test full application, the tester need to create range of inputs which may be a time consuming.

8.1.2 Black box testing

Black box testing focuses on the functional requirements of the software. It is also known as functional testing. It is a software testing technique whereby the internal workings of the item being tested are not known by the tester. For example, in a black box test on software design the tester only knows the inputs and what the expected outcomes should be and not how the program arrives at those outputs.

The tester does not ever examine the programming code and does not need any further knowledge of the program other than its specifications. It enables us to derive sets of inputs that will fully exercise all functional requirements for a program.

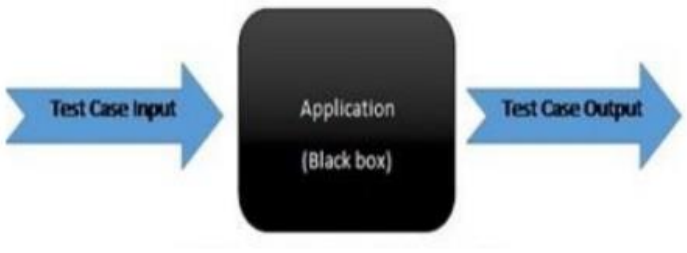


Fig 8.1.2: Black Box Testing

Black box testing is an alternative to white box technique. Rather it is a complementary approach that is likely to uncover a different class of errors in the following categories:

- Incorrect or missing function.
- Interface errors.
- Performance errors.
- Initialization and termination errors.
- Errors in objects.

8.1.2.1 Advantages of Black Box Testing

- The test is unbiased as the designer and the tester are independent of each other.
- The tester does not need knowledge of any specific programming languages.
- The test is done from the point of view of the user, not the designer.
- Test cases can be designed as soon as the specifications are complete.

8.1.2.2 Disadvantages of Black Box Testing

- The test inputs need to be from large sample space. That is, from a huge set of data this will take time.
- Also it is difficult to identify all possible inputs in limited testing time. So writing test cases is slow and difficult.
- Chances are more that there will be unidentified paths during this testing.

8.2 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

8.3 System Testing

This information contributes towards reducing the ambiguity about the system. For example, when deciding whether to release a product, the decision makers would need to know the state of the product including aspects such as the conformance of the product to requirements, the usability of the product, any known risks, the product's compliance to any applicable regulations,

Software testing enables making objective assessments regarding the degree of conformance of the system to stated requirements and specifications.

System testing checks complete end-end scenarios, as a user would exercise the system. The system has to be tested for correctness of the functionality by setting it up in a controlled environment. System testing includes testing of functional and nonfunctional requirements. It helps to verify and validate the system. All components of system should have been successfully unit tested and then checked for any errors after integration.

8.4 Quality Assurance

Quality assurance consists of the auditing and reporting functions of management. The goal of quality assurance is to provide management with the data necessary to be informed about product quality, thereby gaining insight and confident that the product quality is meeting its goals. This is an "umbrella activity" that is applied throughout the engineering process. Software quality assurance encompasses:

- Analysis, design, coding and testing methods and tools
- Formal technical reviews that are applied during each software engineering
- Multi-tiered testing strategy
- Control of software documentation and the change made to it.
- A procedure to ensure compliance with software development standards.
- Measurement and reporting mechanisms.

8.4.1 Quality Factors

An important objective of quality assurance is to track the software quality and assess the impact of methodological and procedural changes on improved software quality. The factors that affect the qua-

-lity. The factors that affect the quality can be categorized into two broad groups:

- Factors that can be directly measured.
- Factors that can be indirectly measured
- These factors focus on three important aspects of a software product
- Its operational characteristics
- Its ability to undergo changes
- Its adaptability to a new environment.
- Effectiveness or efficiency in performing its mission
- Duration of its use by its customer.

8.5 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input :	Identified classes of valid input must be accepted.
Invalid Input :	identified classes of invalid input must be rejected.
Functions :	identified functions must be exercised.
Output :	identified classes of application outputs must be exercised.
Systems/Procedures:	Interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

Chapter 9

RESULT AND PERFORMANCE ANALYSIS

The result of this research can be seen from accuracy that calculated from several combinations of features. Table 9.1 shows classification accuracy of this research. When single feature is used, shape feature has the lowest accuracy of 51%.

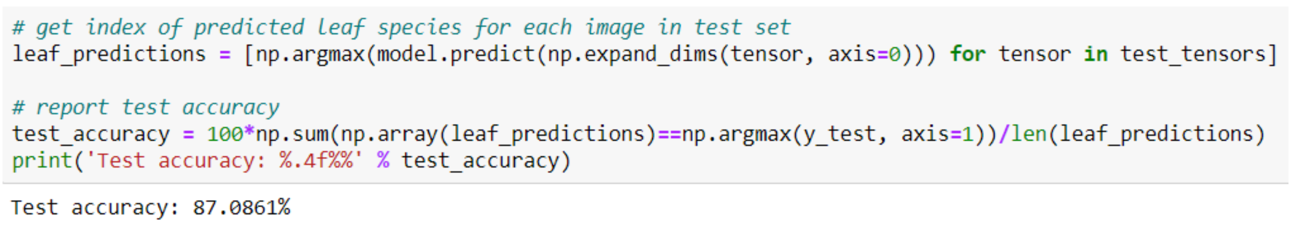
Features	Accuracy
SHAPE	51 %
MARGIN	67 %
TEXTURE	78.3 %
MARGIN + TEXTURE	84.5 %
MARGIN + SHAPE	81 %
SHAPE + TEXTURE	82.6 %
MARGIN + SHAPE + TEXTURE	87 %

Table 9.1 Classification Accuracy

On the other hand, margin feature results in 67% accuracy because every leaf has different margin or border. The next feature to look out for is Texture. Texture has the highest accuracy compared to all the other individual features. So if we combine the best two features which are margin and texture we will get an test accuracy of 84.5%. The other two combinations are margin and shape, shape and texture. Both will provide an accuracy of 81% and 82.6% respectively. But if we consider all the three features we will get an accuracy of 87%. Hence we decided to go with all the feature of the image.

9.1 Model Accuracy

We have used all the three features namely shape, margin and texture to train our model. The model accuracy we got is around 87%. Below we have included the model accuracy results snapshot:



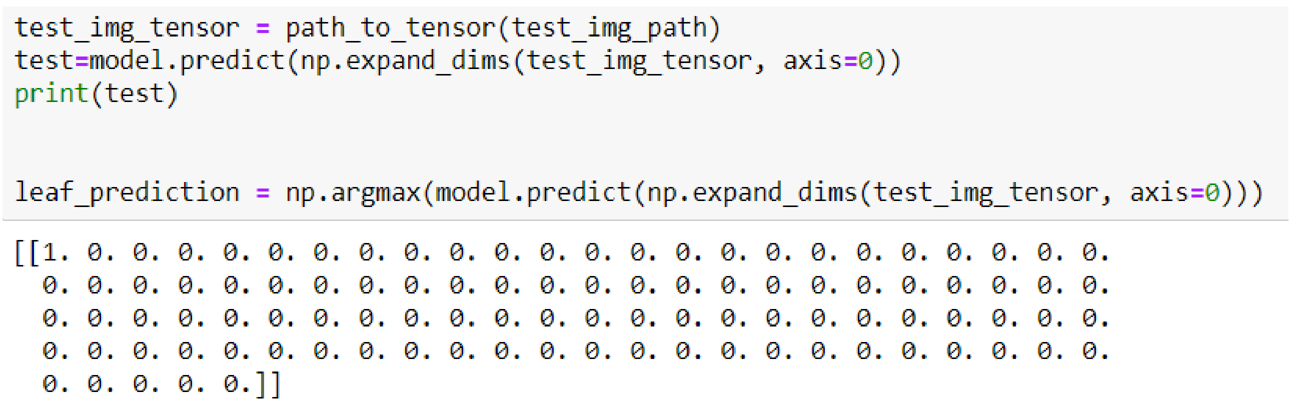
```
# get index of predicted leaf species for each image in test set
leaf_predictions = [np.argmax(model.predict(np.expand_dims(tensor, axis=0))) for tensor in test_tensors]

# report test accuracy
test_accuracy = 100*np.sum(np.array(leaf_predictions)==np.argmax(y_test, axis=1))/len(leaf_predictions)
print('Test accuracy: %.4f%%' % test_accuracy)
```

```
Test accuracy: 87.0861%
```

Fig 9.1: Model Accuracy

9.2 Predicted Array List

The prediction returns an list of array where the array elements corresponds to the leaf species. The predicted leaf species index element will be one and the rest will be zeros. By that we will be able to get the prediction of the leaf.



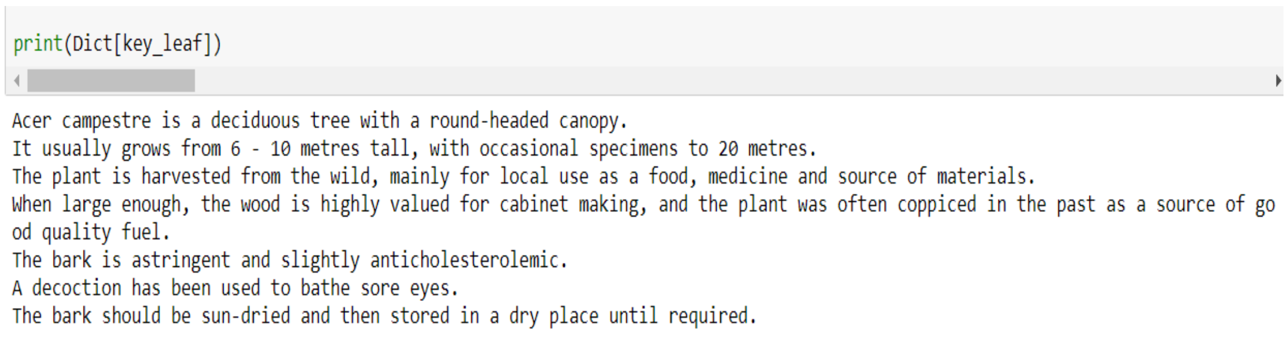
```
test_img_tensor = path_to_tensor(test_img_path)
test=model.predict(np.expand_dims(test_img_tensor, axis=0))
print(test)


leaf_prediction = np.argmax(model.predict(np.expand_dims(test_img_tensor, axis=0)))
```

```
[[1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0.]]
```

Fig 9.2 Predicted Array List

9.3 Features of the Resultant leaf

We have created a dictionary which contains all the details or medicinal values of the medicinal leaves. If the leaf is not matched with any of the leaves in our dictionary than it will be classified as non-medicinal leaf. If it is a medicinal leaf the we will match the leaf name with the keys of dictionary and get the value of that perticular key. That value corresponds to the medicinal value of the predicted leaf. Here is the snapshot of the above section:



```
print(Dict[key_leaf])
```

```
Acer campestre is a deciduous tree with a round-headed canopy.
It usually grows from 6 - 10 metres tall, with occasional specimens to 20 metres.
The plant is harvested from the wild, mainly for local use as a food, medicine and source of materials.
When large enough, the wood is highly valued for cabinet making, and the plant was often coppiced in the past as a source of go
od quality fuel.
The bark is astringent and slightly anticholesterolemic.
A decoction has been used to bathe sore eyes.
The bark should be sun-dried and then stored in a dry place until required.
```

Fig 9.3: Features of resultant leaf

Summary

This chapter gives a graphic view if the execution of the system. The output screens show each of the execution stages for each module. Based on the analysis the proposed project performs well in terms of accuracy, dependency and usability.

Chapter 10

CONCLUSION AND FUTURE SCOPE

10.1 Conclusion

The objective of this study is to reduce manual work and increase the efficiency by the automatic identification of medicinal plants using image processing techniques. From the literature survey the majority of the researchers used leaf features for the classification of medicinal plants and less research work was done in classification of medicinal plants using flowers and fruits/seeds.

Our work is to enhance the research in identification and classification of medicinal plants as herbs, shrubs and trees using flowers and fruits/seeds features including leaf. Automatic identification and classification of medicinal plants will provide medicinal knowledge to common people and farmers which help in increasing production of such essential plants. This automatic classification system also helps botanists, consumers, forestry services, taxonomists, pharmaceutical companies and Ayurveda practioners to identify and classify the medicinal plants without any human assistance.

10.2 Feature scope:

The medicinal plants are closely connected with the traditional knowledge of its use. During the early periods, the knowledge of the medicinal properties of the plants was transferred from one generation to another generation orally and no documentation of the medicinal plants has been maintained. Even though, the herbal formulations are regaining their momentum, the major problem behind the herbal medicine is that there is a lack of standard protocol for their standardization and problem behind carrying out clinical trials. It is essential to evaluate the herbal plant scientifically and documents should be made to know their medicinal properties. To revitalize Indian medicinal heritage, through creative application of the traditional health sciences for the enhancement the quality of health care in rural and urban India, extensive research on plants for natural leads is very essential.

REFERENCES

[1] Manojkumar P., Surya C. M., and Varun P. Gopi, "Identification of Ayurvedic Medicinal Plants by Image Processing of Leaf Samples", 2017 Third International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), pp 978-1- 5386-1931-5.

[2] Mr. K.Nithiyanandhan and Prof.T. Bhaskara Reddy, "Analysis of the Medicinal Leaves by using Image Processing Techniques and ANN", Vol 8, No. 5, ISSN No. 0976-5697, May-June 2017.

[3] Adams Begue, Venitha Kowlessur, Fawzi Mahomoodally, Upasana Singh and Sameerchand, "Automatic Recognition of Medicinal Plants using Machine Learning Techniques", International Journal of Advanced Computer Science and Applications, Vol. 8, No. 4, 2017.

[4] H. X. Kan, L. Jin, and F. L. Zhou," Classification of Medicinal Plant Leaf Image Based on Multi-Feature Extraction", Pattern Recognition and Image Analysis, Vol. 27, No. 3, 2017, pp. 581–587, 1054-6618. © Pleiades Publishing, Ltd.

[5] Riddhi H. Shaparia, Dr. Narendra M. Patel and Prof. Zankhana H. Shah," Flower Classification using Texture and Color Features", International Conference on Research and Innovations in Science, Engineering &Technology, Volume 2, 2017, Pages 113–118.

[6] https://hackr.io/tutorials/learn-machine-learning-ml

[7] https://machinelearningmastery.com/start-here/#getstarted