VISVESVARAYA TECHNOLOGICAL UNIVERSITY - BELAGAVI

A PROJECT REPORT

On

"Botonical Vision : Revealing Secrets Of Medicinal Plants Through Image Processing"

Submitted in partial fulfilment of the requirement for the award of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE & ENGINEERING

Submitted by

MADHUSHREE D N [4MG20CS019]
NISARGA N L [4MG20CS027]
SHREYA R PISSAY [4MG20CS039]
SIRI R PISSAY [4MG20CS041]

Under the guidance of

Mrs. PADMAVATI K
HOD & Professor
Dept. of CS&E, GMIT

G.MADEGOWDA INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Bharathinagara , Maddur Tq ,Mandya District Karnataka - 571422

2023-2024

G.MADEGOWDA INSTITUTE OF TECHNOLOGY

Bharathinagara, Maddur Taluk, Mandya Dist-571422

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the Project Work entitled "Botonical Vision : Revealing Secrets Of Medicinal Plants Through Image Processing" is carried out by Madhushree DN[4MG20CS019] Nisarga N L[4MG20CS027] Shreya R Pissay[4MG20CS039] Siri R Pissay[4MG20CS041] a bonifide students of G.Madegowda Institute Of Technology, Bharathinagara, in partial fulfilment for the award of Bachelor of Engineering in Computer Science & Engineering Branch of Visvesvaraya Technological University Belagavi during the year 2023-2024 as per the University requirements for the VIII semester academic curriculum. The Project report has approved as it satisfies the requirement with respect to the Project work prescribed for Bachelor of Engineering degree.

Signature of the Guide:	Signature of the HOD:	Signature of the Principal:
Mrs. Padmavati K	**Mrs. Padmavati K**	**Dr. Chandan B R**
HOD & Professor	HOD & Professor	Principal
Dept. of CS&E,	Dept.ofCS&E,	GMIT,
GMIT, Bharathinagara	GMIT, Bharathinagara	Bharathinagara

PRINCIPAL
G. Madegowda Institute of Technology
Bharathinagar, Maddur Taluk,
Mandya District-571 422.

EXTERNAL VIVA

Name of the Examiner	Signature with Date
1. Dr. Shivashankar.	1.
2. Vinayashree A S	2. 27/05/24

DECLARATION

Madhushree D N[4MG20CS019], Nisarga N L[4MG20CS027], Shreya R Pissay [4MG20CS039], Siri R Pissay [4MG20CS041], a bonifide students of Bachelor Of Engineering in Computer Science & Engineering at G.Madegowda Institute Of Technology, Bharathinagara, hereby declare that the project work entitled **"Botonical Vision: Revealing Secrets Of Medicinal Plants Through Image Processing"** has been independently carried out by us under the guidance of **Mrs. Padmavati K**, HOD & Professor, Department of Computer Science & Engineering, submitted in the partial fulfilment of the requirement for the award of degree **Bachelor Of Engineering in Computer Science & Engineering of the Visvesvaraya Technological University Belagavi.**

we also declare that we have not submitted this project work to any other university for the award of any degree.

Place: Bharathinagara

Date: 27-05-2024

ACKNOWLEDGEMENT

This space is dedicates to acknowledge and express our gratitude to the following people for their magnificent support and contribution to our journey of BE project work.

The content and euphoria that entails the successful completion of any task be incomplete without applauding and thanking the people who made it possible, because success is the epitome of hard work, perseverance, undeterred missionary zeal, steadfast determination and paramount of all "Encouraging Guidance".

We express our sincere gratitude to our Principal **Dr Chandan B R**, for providing us means of attaining our most cherished goals.

We express our sincere regards and thanks to our HOD, **Mrs. Padmavati K, HOD & Professor, Computer Science and Engineering** for her sterling efforts, amenable assistance and inspiration in all phases of our project work.

We are very much thankful to our guide **Mrs. Padmavati K, HOD & Professor , Computer Science and Engineering,** for her constant guidance, confidence and her precious suggestion throughout this project work and thus timely completion of this project work.

We acknowledge the encouragement, support and instructions given by all the teaching and non-teaching Staffs of **Computer Science & Engineering Department.**

We are thankful to our parents. Last but not the least we are grateful to our friends for their assistance and encouragement which has made us to undertake this project work.

Project associates

Madhushree D N[4MG20CS019]
Nisarga N L [4MG20CS027]
Shreya R Pissay [4MG20CS039]
Siri R Pissay [4MG20CS041]

ABSTRACT

Plants Leaf play a vital role in human life, they provides oxygen, food, shelter, medicine, fuel, gums and environmental protection. Many plants are rich in medicinal values and contain active ingredients for medicinal use. Because of factor such as global warming, increasing population, professional secrecy, lack of Government support for research activities and lack of awareness about medicinal plants, many useful plants species are now becoming extinct and are getting destroyed. Manual identification of medicinal plants leaf is a time consuming process and need the help of experts for plants identification. 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 leaf 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. This paper presents literature on image processing techniques used in identification and classification of medicinal plants and also the importance and benefits of medicinal plants in recent years.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Ayurveda is an ancient medicinal system that is practiced in India and has its origins in the Vedic times, approximately 5000 years ago. Ayurveda is considered to be the oldest healing science. In the Sanskrit language, Ayurveda means "The Science of Life". In spite of being suppressed during years of foreign occupation, Ayurveda is being widely used in its native land and throughout the world. The main ingredients of ayurvedic medicines are plant leaves, roots, bark, fruits, seeds, etc. It is said that about 8000 plants of Indian origin are known to possess medicinal attributes. The fundamentals on which this Ayurvedic system is built are always true for all ages and can be easily adapted from generation to generation. This, in turn, makes Ayurveda one of the very few medicinal systems that were developed in ancient times and are still implemented to date. Ancient Tibetan and Traditional Chinese Medicine and Early Greek medicine have accepted many concepts that were described in the ancient Ayurvedic literature dating back to several thousands of years. This widespread and extensive use in the field of medicine makes Ayurveda the 'mother of healing'. These Ayurvedic plants are used for preparing medicines on a commercial basis. This has resulted in the production and marketing of Ayurvedic medicines to become a thriving industry with its annual turnover exceeding Rs 4000 crores. Because of this, the number of licensed Ayurvedic medicine manufacturers in India now exceeds 8500. Because of this increased commercialization of the Ayurvedic sector, several issues regarding the raw material quality used for their preparation need to be focused. These plants are usually collected by tribal masses that are not professionally trained in the work of identifying the correct plants. Even the manufacturing units, at times, receive improper or substituted medicinal plants. Most of these manufacturing units do not have proper quality control mechanisms that can screen these plants. In addition to this, confusion due to the different local names of these plants makes the matters worse. Because of the usage of improper raw materials, Ayurvedic medicine becomes inefficient. It may result in certain unpredictable side effects too. As a result, an intelligent system needs to be developed that can properly identify the ayurvedic plant based on the leaf samples. This will, in turn, improve the quality of the medicine and will also maintain its credibility.

1.2 Problem Statement

Deep learning is one of the major subfields of the machine learning framework. Machine learning is the study of design of algorithms, inspired by the model of the human brain. Deep learning is becoming more popular in data science fields like robotics, artificial intelligence (AI), audio & video recognition and image recognition. An artificial neural network is the core of deep learning methodologies. Deep learning is supported by various libraries such as Theano, TensorFlow, Caffe, Mx net etc., Keras is one of the most powerful and easy to use python library, which is built on top of popular deep learning libraries like TensorFlow, Theano, etc., for creating deep learning models. Detection of correct medicinal leaves can help botanists, taxonomists and drug manufacturers to make quality drug and can reduce the side effects caused by the wrong drug delivery. To identify the leaves of the plants, a type of artificial neural network called convolutional neural network (CNN) is used. The architecture we used here is Densenet121, which is a convolutional neural network that is a powerful model capable of achieving high accuracies on challenging datasets.

1.3 Objective

- Plant identification using a user-uploaded photograph would be one major goal of the proposed system.
- To create a graphical user interface that is easy to use. To ensure that the user interface is simple to grasp and intuitive to use.
- To provide instructions on how to care for the plant.
- To deliver the most accurate results possible.
- To give farmers a cost-effective and time-saving option.
- For the sake of providing a dependable and efficient system.

1.4 Motivation

Medicinal plants have been utilized in curing human diseases for thousands of centuries and are the source of a significant percentage of medicines. Medicinal plant species classification is critical for medicine production and conservation. Local peoples are not enough knowledgeable of their urban medicinal plants and their usages. Therefore, classifying the medicinal plant image using Convolutional Neural Network by high accuracy image classification model could be useful to identify different types of species.

CHAPTER 2

LITERARURE SURVEY

Title 1: A Novel Approach to Classification of Ayurvedic Medicinal Plants using Neural Networks
Author: Sameer A Kyalkond, Sudhanva S Aithal
Methodology: Creating an automated categorization system for medicinal plants is a time-consuming and challenging task. India is a nation with a varied variety of plant species, each with its unique set of therapeutic qualities. Because it is hard for humans to recall the names of all plant species and their applications, previous knowledge is essential for manual identification and categorization. The preservation of these therapeutic plants is crucial as it will help a broad variety of sectors, including medicine, botanic research, and plant taxonomy studies, among others.
Limitation: Less data sets.

Title 2: Recognition of Ayurvedic Medicinal Plants from Leaves: A Computer Vision Approach
Author: Amala Sabu, Sreekumar K
Methodology: Plants are an indispensable part of our ecosystem and India has a long history of using plants as a source of medicine. Since the advent of modern allopathic medicine, the use of traditional medicine declined to a considerable extent. However, in recent years, traditional medicine has made a comeback for a variety of reasons they are inexpensive, nontoxic, and do not impact any side effect.
Limitations: Collecting the data set is difficult process.

Title 3: Real-Time Identification of Medicinal Plants using Machine Learning Techniques.
Author: Sivaranjani.C, Lekshmi Kalinathan
Methodology: Ayurveda is being practiced in India since the Vedic times for the well-being of the people. It is one of the oldest systems of Medicinal science that is used even today. Being used extensively in medicine, proper identification of the appropriate medicinal plants used for the manufacture of medicine is very important in Ayurvedic medicinal industry. Identification of the appropriate medicinal plants is usually done by the skilled practitioners who have achieved expertise in this field.
Limitation: we should not increase the depth of our model

Title 4: Medicinal Plant Identification Using Deep Learning

Author: R.Geerthana1 , P.Nandhini2 , R.Suriyakala3

Methodology: We use CNN architecture to train our data and develop the system with high accuracy. Several model architectures were trained, with the best performance reaching a 96.67% success rate in identifying the corresponding medicinal plant.

Limitations: we should not increase the depth of our model

Title 5: Identification of Medicinal Plants using Deep Learning

Author: R. Upendar Rao1 , M. Sai Lahari2 , K. Pavana Sri 3 , K. Yaminee Srujana4 , D. Yaswanth5

Methodology: In this project we explore feature vectors from both the front and back side of a green leaf along with morphological features to arrive at a unique optimum combination of features that maximizes the identification rate. A database of medicinal plant leaves is created from scanned images of front and back side of leaves of commonly used medicinal plants.

Limitations: Less Data Sets

Title 6: Identification of Indian Medicinal Leaves using Convolutional Neural Networks

Author: Bhargavi Jahagirdar1, Divya Munot2, Niranjan Belhekar3, Dr. K. Rajeswari4

Methodology: In this paper, we have described the implementation of Convolutional Neural Networks (CNN) for the identification of Indian medicinal leaves.

Limitations: Collecting the data set is difficult process.

CHAPTER 3

SYSTEM ANALYSIS

3.1 Existing System

Several studies have been conducted in order to develop tool for the identification of plants during last 10 year. One of the most authoritative works in the field of plant classification has been done by DestaSandyaPrasvita. MedLeaf - a mobile application for medicinal plant identification based on leaf image. The application consists of two main functionalities, i.e. medicinal plant identification and document searching of medicinal plant. MedLeaf is computer-aided medicinal plant recognition system that use technology of computer vision and intelligent information processing techniques. They used Local Binary Pattern to extract leaf texture and Probabilistic Neural Network (PNN) to classify the image.

3.2 Disadvantages Of Existing System

Limited Accuracy: The existing system has limited accuracy in detecting medicinal plants, leading to potential misidentification and incorrect results. Achieved an average accuracy of 90.3% with the Flavia dataset.

Slow Processing: The current system takes a significant amount of time to process and analyze images, resulting in delays in obtaining results.

Limited Plant Species Recognition: The current system has a limited capability to recognize a wide range of plant species, hindering its effectiveness in identifying medicinal plants accurately.

3.2 Proposed System

The proposed system for medicinal plant detection utilizes image processing techniques to accurately identify and classify different types of medicinal plants. The system incorporates advanced image processing algorithms to analyze plant images and extract relevant features for classification. All the images in the dataset are read, processed, and feature extracted. Raw data is loaded for classification of input image. The proposed system contains the four stages, namely, Image acquisition, Image pre-processing, Feature Extraction and Classification. The system follows deep learning and it is completed in two phases, training and testing.

CHAPTER 4

SYSTEM REQUIREMENTS

4.1 Requirement Analysis

Requirement analysis, also called requirement engineering, is the process of determining user expectations for a new modified product. It encompasses the tasks that determine the need for analyzing, documenting, validating and managing software or system requirements. The requirements should be documentable, actionable, measurable, testable and traceable related to identified business needs or opportunities and define to a level of detail, sufficient for system design.

4.2 Functional Requirements

Useful requirements describe the product's internal activities: that is, the technical subtleties, monitoring and handling of data and other specific functionality demonstrating how to satisfy the use cases. They are upheld by non-utilitarian prerequisites that force the plan or execution of imperatives.

- System should process the data.
- System should segment the medicinal leaf images.
- System should detect the medicinal leaf image.
- System should predict medicinal images using medicinal leaf images.

4.3 Non-Functional Requirements

Unnecessary prerequisites are requirements that suggest parameters that can be used to assess a framework's operation rather than specific activities. This should be distinguished from useful necessities indicating explicit behavior or capabilities. Reliability, flexibility, and price are common non-practical necessities. The architecture should be created in order to incorporate new modules and functionalities, thereby promoting application development. The cost should be small as a result of programming packages being freely accessible.

- Usability: System Should be user Friendly
- Reliability: The system should be Reliable
- Performance: The system Should not take excess time in detecting the coffee leaf disease
- Supportability: System should be easily updatable for future enhancement.

4.4 Hardware Requirements

- System : Intel i3/i5 2.4 GHz.
- Hard Disk : 500 GB
- Ram : 4/8 GB

4.5 Software Requirements

- Operating system : Windows XP/ Windows 10
- Software Tool : Open CV
- Coding Language : Python
- Toolbox : Image processing toolbox.

4.6 Language Specification

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This tutorial gives enough understanding on Python programming language.

4.7 APPLICATION OF PYTHON:

- **Easy-to-learn** : Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** : Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** : Python's source code is fairly easy-to-maintain.
- **Interactive Mode** : Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** : Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** : You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** : Python provides interfaces to all major commercial databases.
- **Scalable** : Python provides a better structure and support for large programs than shell scripting.

CHAPTER 5

SYSTEM DESIGN

5.1 Design Overview

Design overview explains the architecture that would be used for developing a software product. It is an overview of an entire system, identifying the main components that would be developed for the product and their interfaces.

5.2 System Architecture

A Deep-CNN is type of a DNN consists of multiple hidden layers such as convolutional layer, RELU layer, Pooling layer and fully connected a normalized layer. CNN shares. weights in the convolutional layer reducing the memory footprint and increases the performance of the network. The important features of CNN lie with the 3D volumes of neurons, local connectivity and shared weights. A feature map is produced by convolution layer through the convolution of different sub-regions of the input image with a learned kernel. Then, a non-linear activation function is applied through ReLu layer to improve the convergence properties when the error is low. In pooling layer, a region of the image/feature map is chosen and the pixel with the maximum value among them or average values is chosen as the representative.

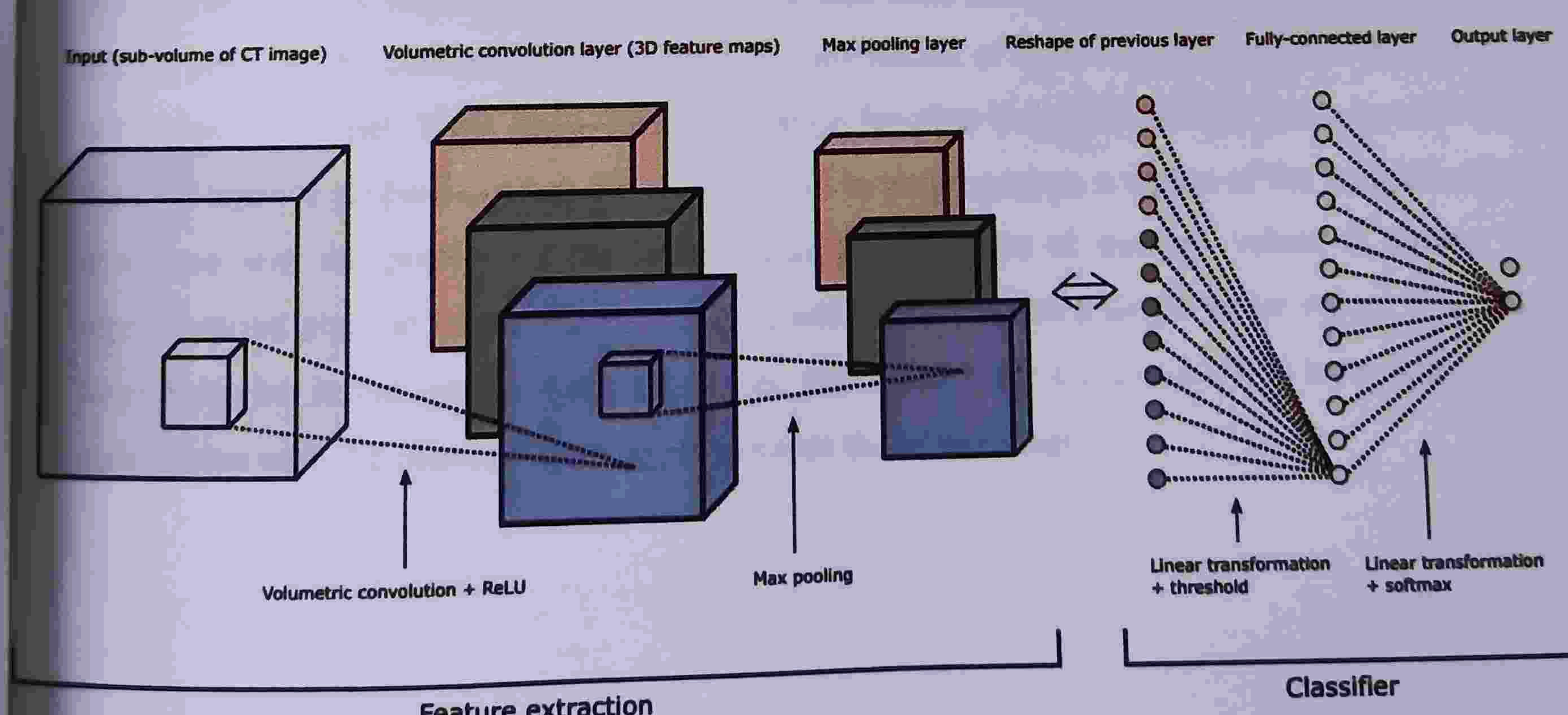

Fig 5.1: Deep-Convolutional Neural Network Architecture

The proposed method follows these stages:

Data Set: The dataset for training is obtained from Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI). LIDC and IDRI consist of 1000 CT scans of both large and small tumors saved in Digital Imaging and Communications in Medicine (DICOM) format.

Image Segmentation: The segmentation of photographs is the phase where the visual image is partitioned into several parts. This normally helps to identify artifacts and boundaries. The aim of segmentation is to simplify the transition in the interpretation of a picture into the concrete picture that can be clearly interpreted and quickly analyzed.

Pre-Processing: In preprocessing stage, the median filter is used to restore the image under test by minimizing the effects of the degradations during acquisition. Various preprocessing and segmentation techniques of lung nodules are discussed in. The median filter simply replaces each pixel value with the median value of its neighbors including itself. Hence, the pixel values which are very different from their neighbors will be eliminated.

5.3 Convolutional Neural Networks

A CNN is type of a DNN consists of multiple hidden layers such as convolutional layer, RELU layer, Pooling layer and fully connected a normalized layer. CNN shares weights in the convolutional layer reducing the memory footprint and increases the performance of the network. The important features of CNN lie with the 3D volumes of neurons, local connectivity and shared weights. A feature map is produced by convolution layer through convolution of different sub regions of the input image with a learned kernel. Then, anon- linear activation function is applied through ReLu layer to improve the convergence properties when the error is low. In pooling layer, a region of the image/feature map is chosen and the pixel with maximum value among them or average values is chosen as the representative pixel so that a 2x2 or 3x3 grid will be reduced to a single scalar value. This results a large reduction in the sample size. Sometimes, traditional Fully-Connected (FC) layer will be used in conjunction with the convolutional layers towards the output stage.

A CNN is composed of several kinds of layers:

- **Convolutional layer:** creates a feature map to predict the class probabilities for each feature by applying a filter that scans the whole image, few pixels at a time.

- **Pooling layer (down-sampling):** scales down the amount of information the convolutional layer generated for each feature and maintains the most essential information (the process of the convolutional and pooling layers usually repeats several times).

- **Fully connected input layer:** flattens the outputs generated by previous layers to turn them into a single vector that can be used as an input for the next layer.

- **Fully connected layer:** Applies weights over the input generated by the feature analysis to predict an accurate label.

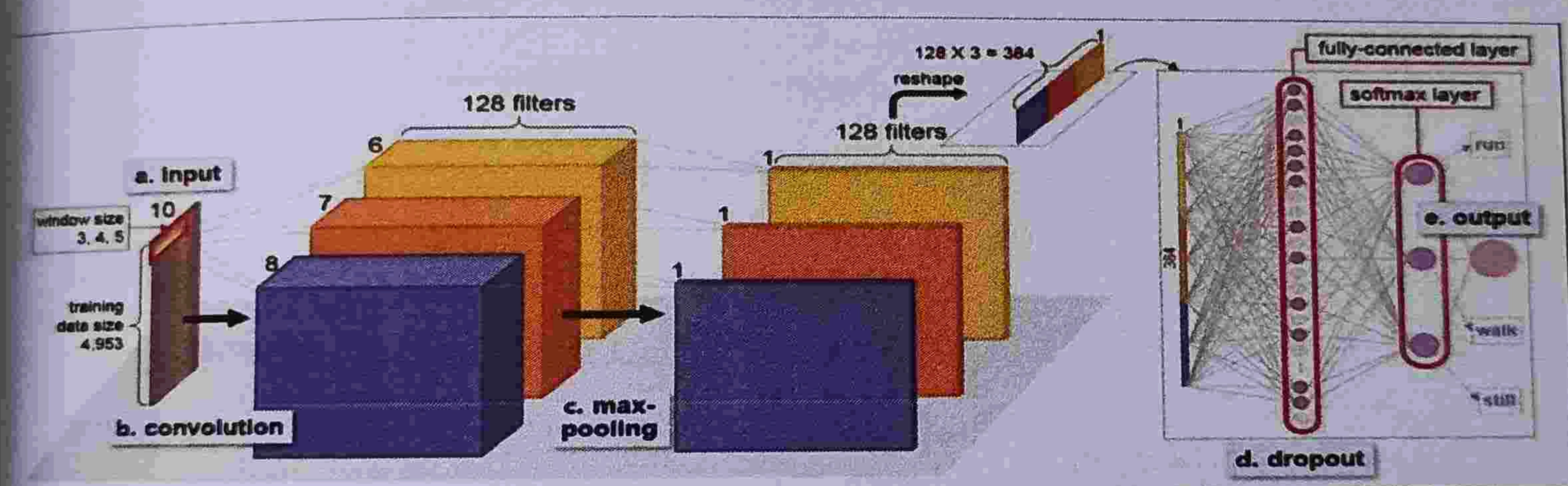

Fig 5.2: Convolutional Neural Network General Architecture

CHAPTER 6

DIAGRAMS

6.1 Data flow diagram

A dataflow outline is a tool for referring to knowledge progression from one module to the next module as shown in Fig 4.3 This graph gives the data of each module's info and yield. The map has no power flow and there are no circles at the same time.

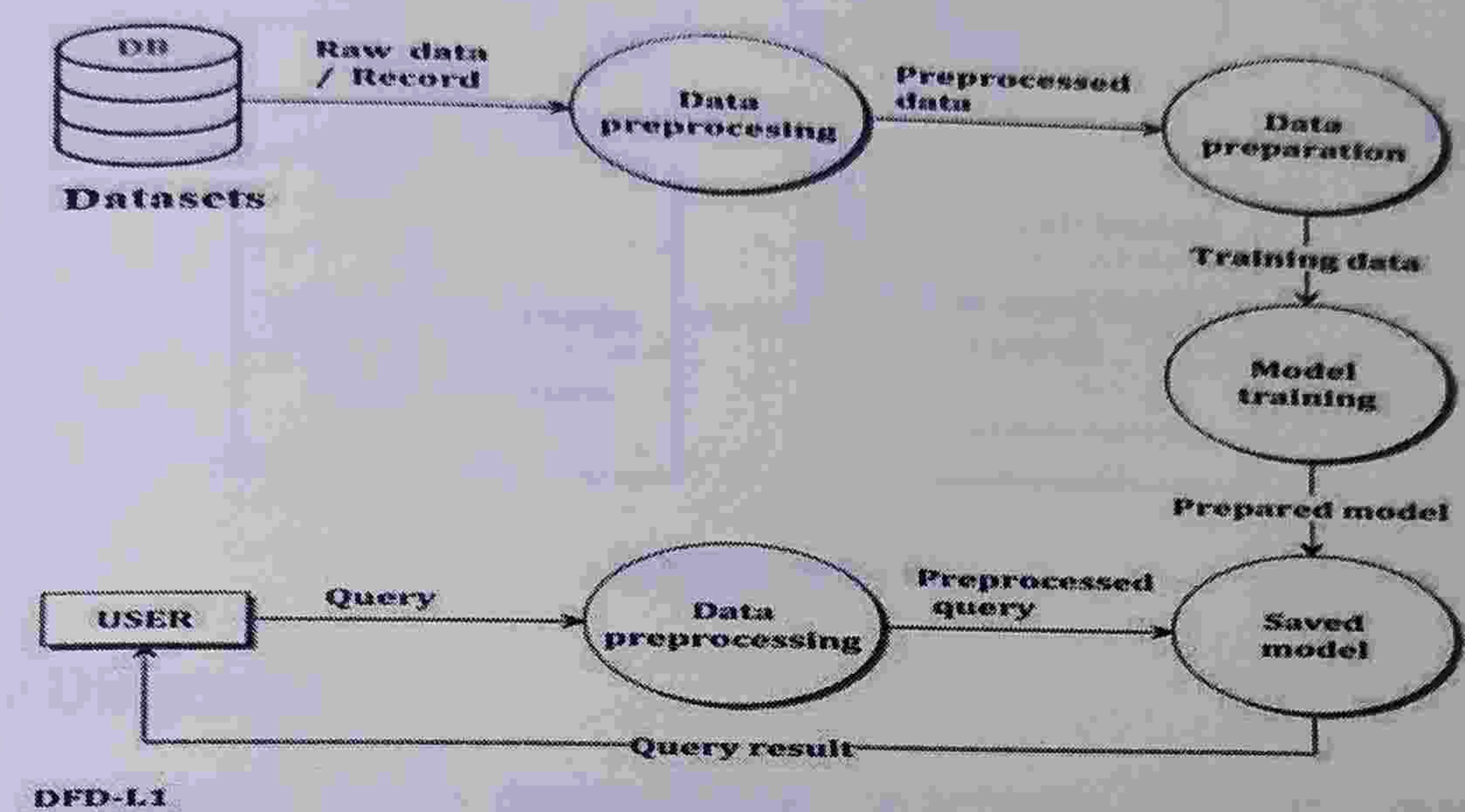

Fig 6.1: Data Flow Diagram

6.2 Use Case Diagram

Use case diagram is the boundary, which defines the system of interest in relation to the world around it. The actors, usually individuals involved with the system defined according to their roles. The use cases, which are the specific roles played by the actors within and around the system.

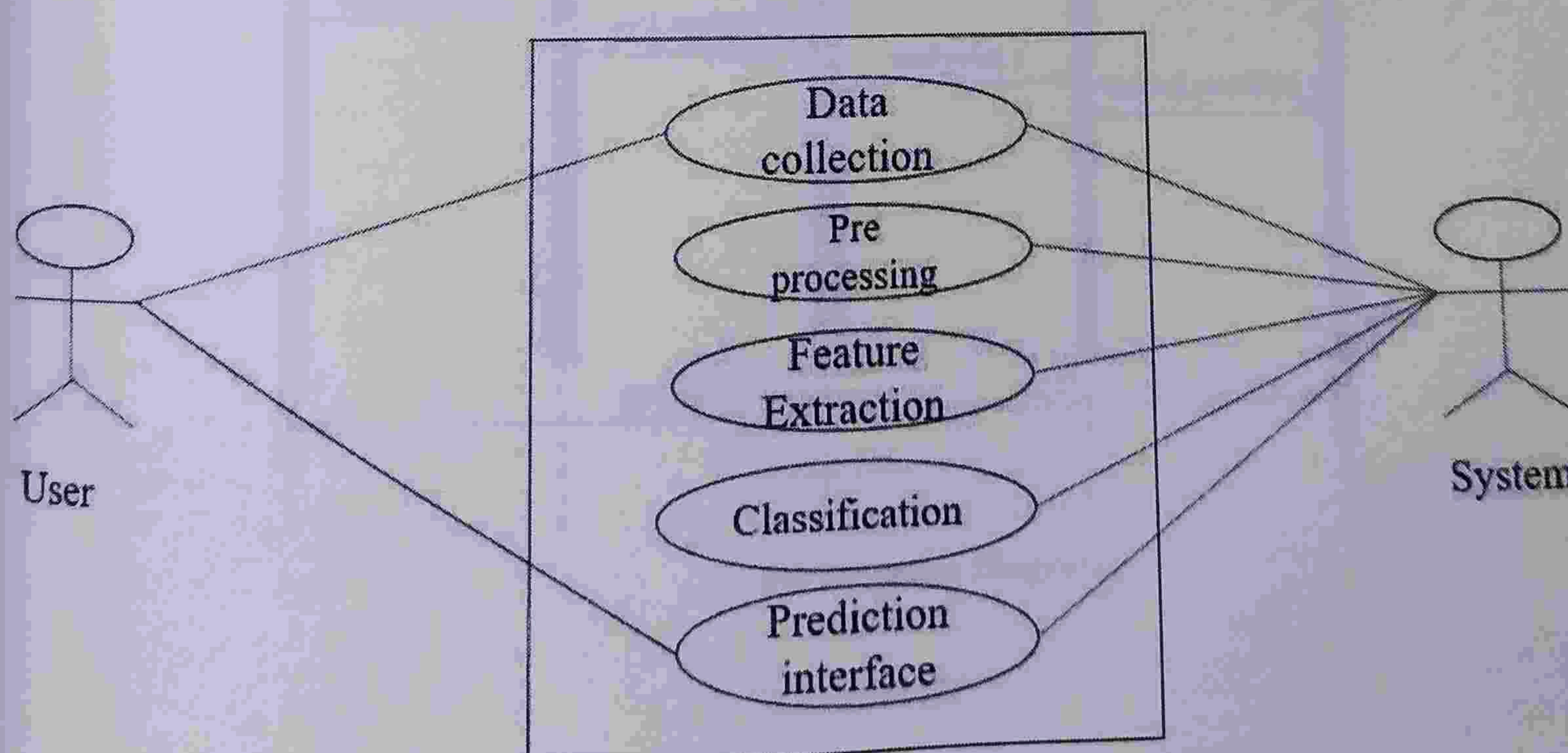

Fig 6.2:Use Case Diagram

6.3 Class Diagrams

Class diagrams are the main building block in object-oriented modeling. They are used to show the different objects in a system, their attributes, their operations and the relationships among them as shown in the Fig

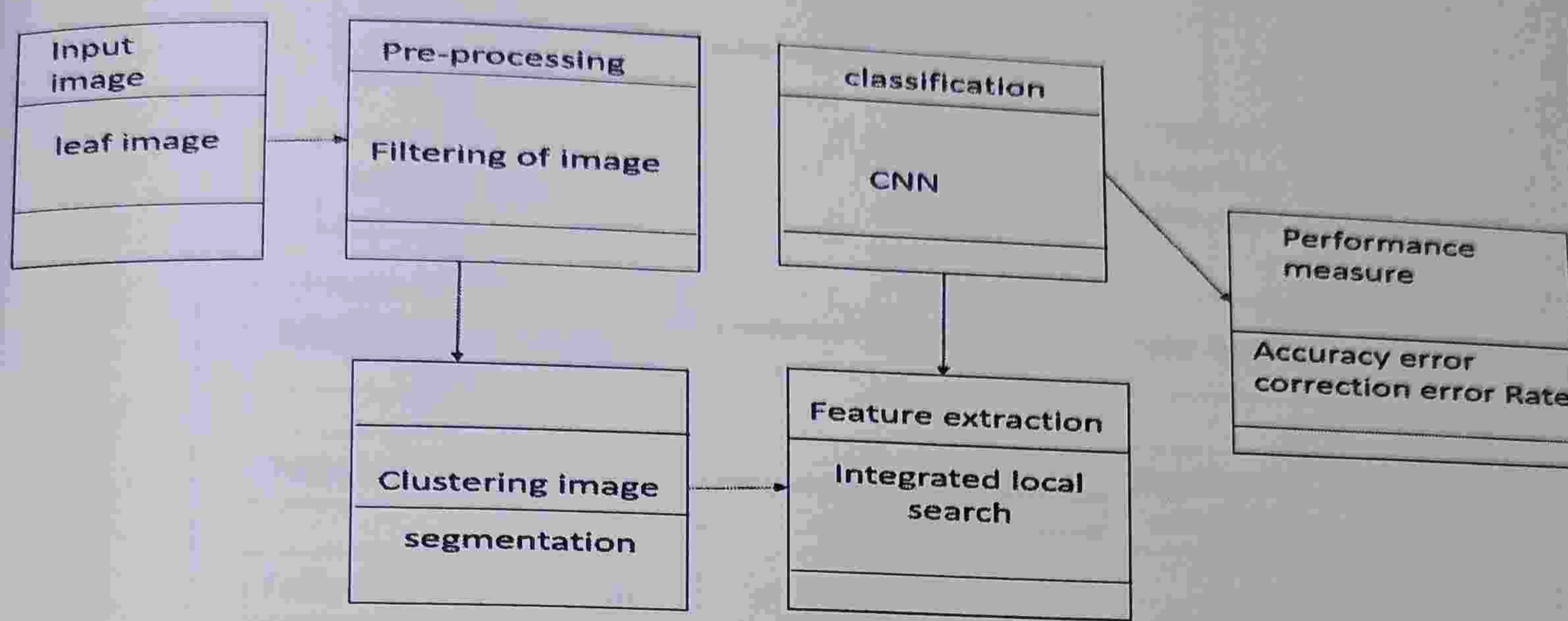

Fig 6.3 Class Diagram

6.4 Sequence Diagrams

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place as shown in Fig

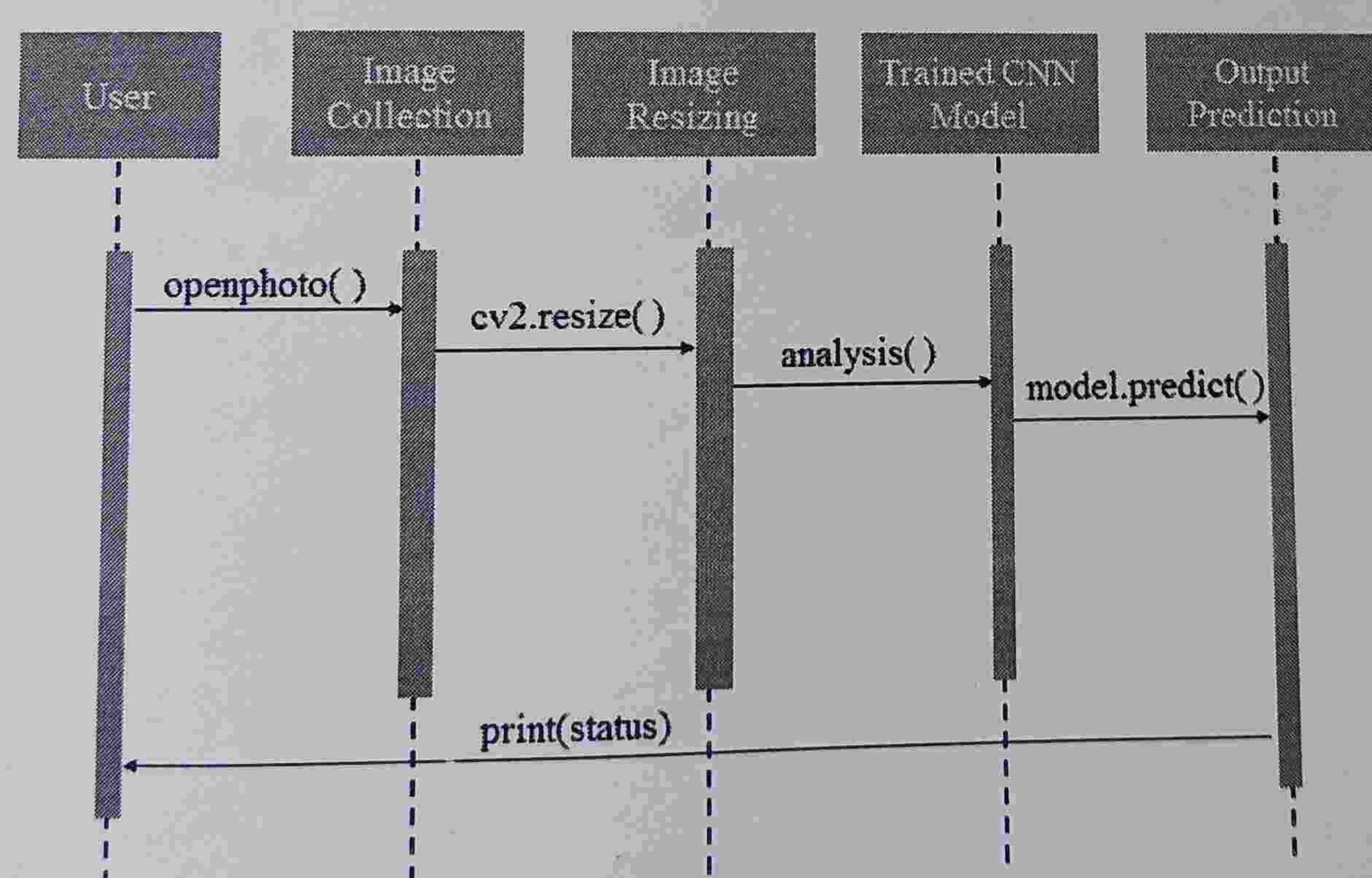

Fig 6.4 Sequence Diagram

6.3 Class Diagrams

Class diagrams are the main building block in object-oriented modeling. They are used to show the different objects in a system, their attributes, their operations and the relationships among them as shown in the Fig

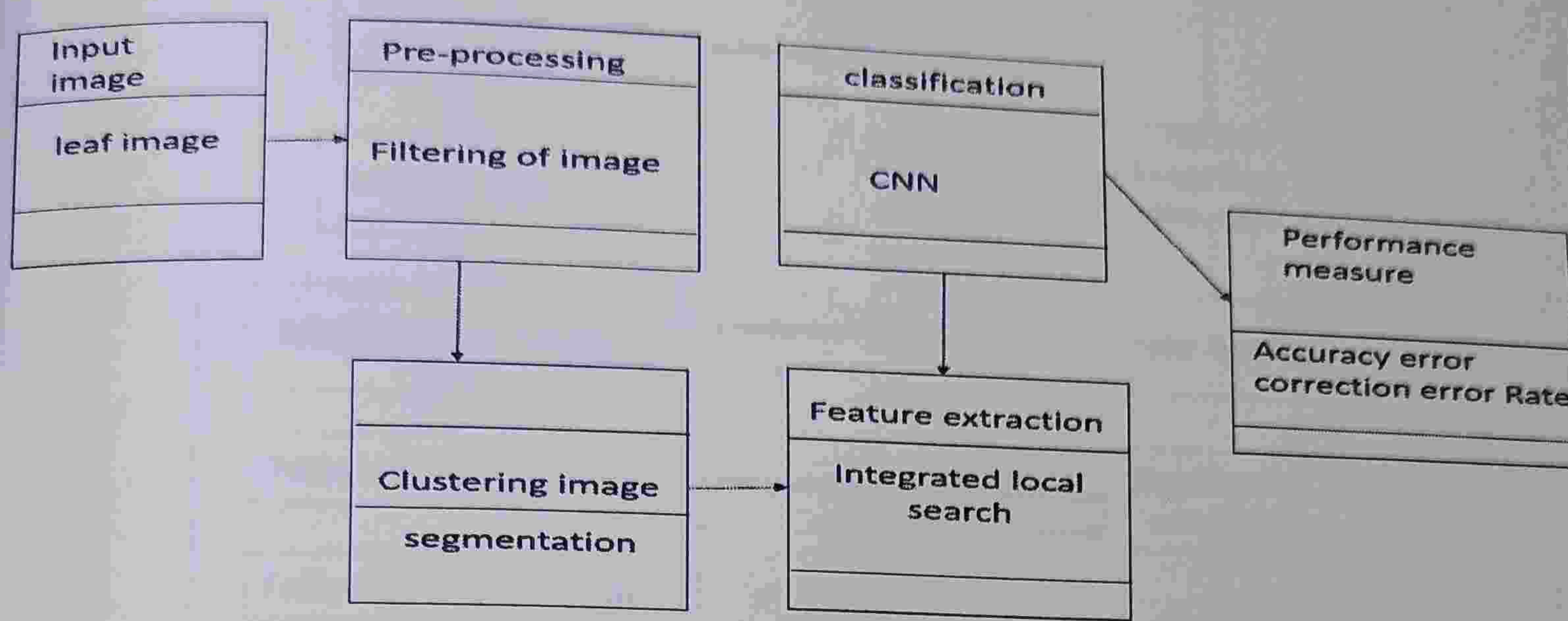

Fig 6.3 Class Diagram

6.4 Sequence Diagrams

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place as shown in Fig

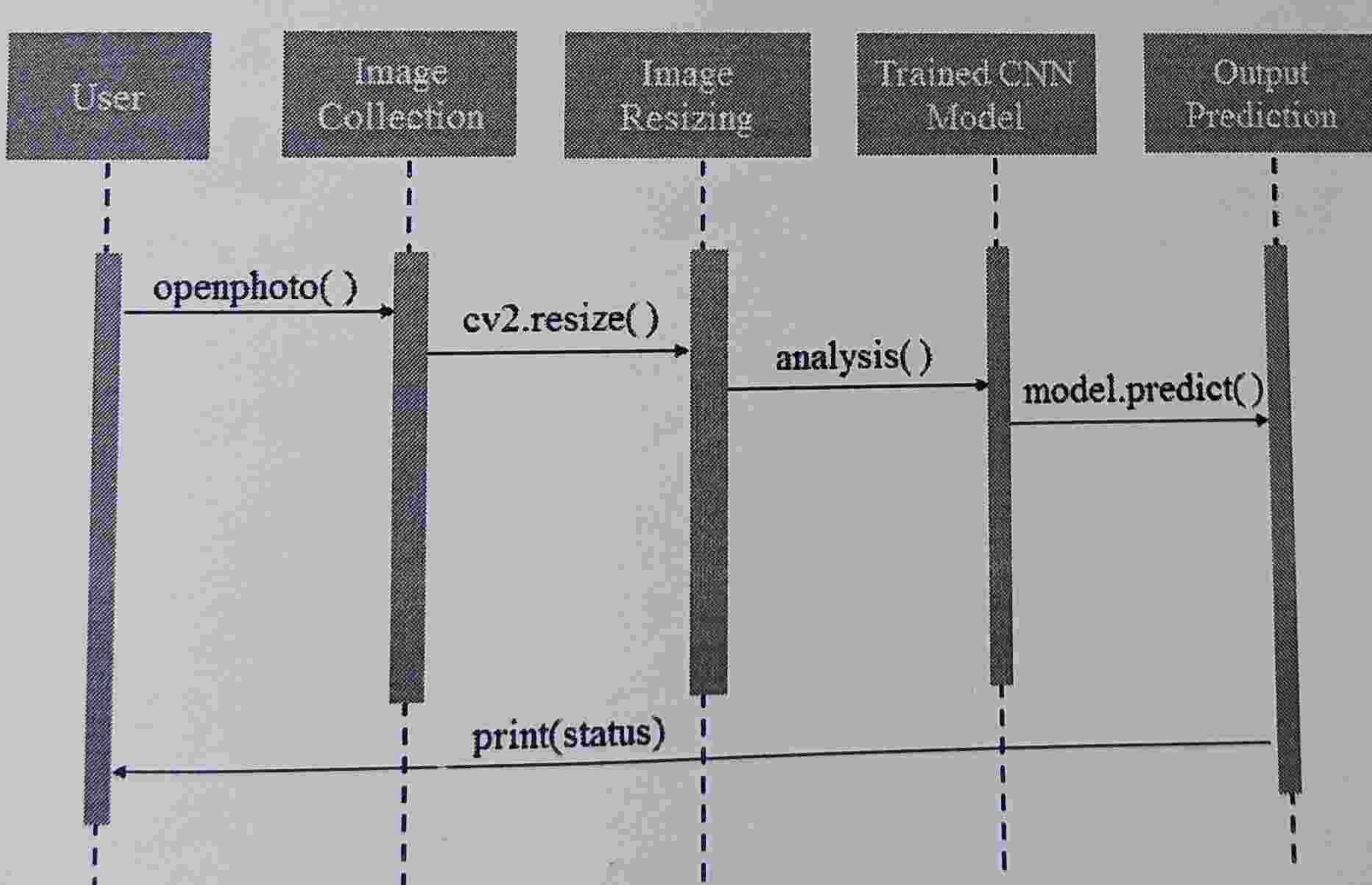

Fig 6.4 Sequence Diagram

6.5 Activity Diagram

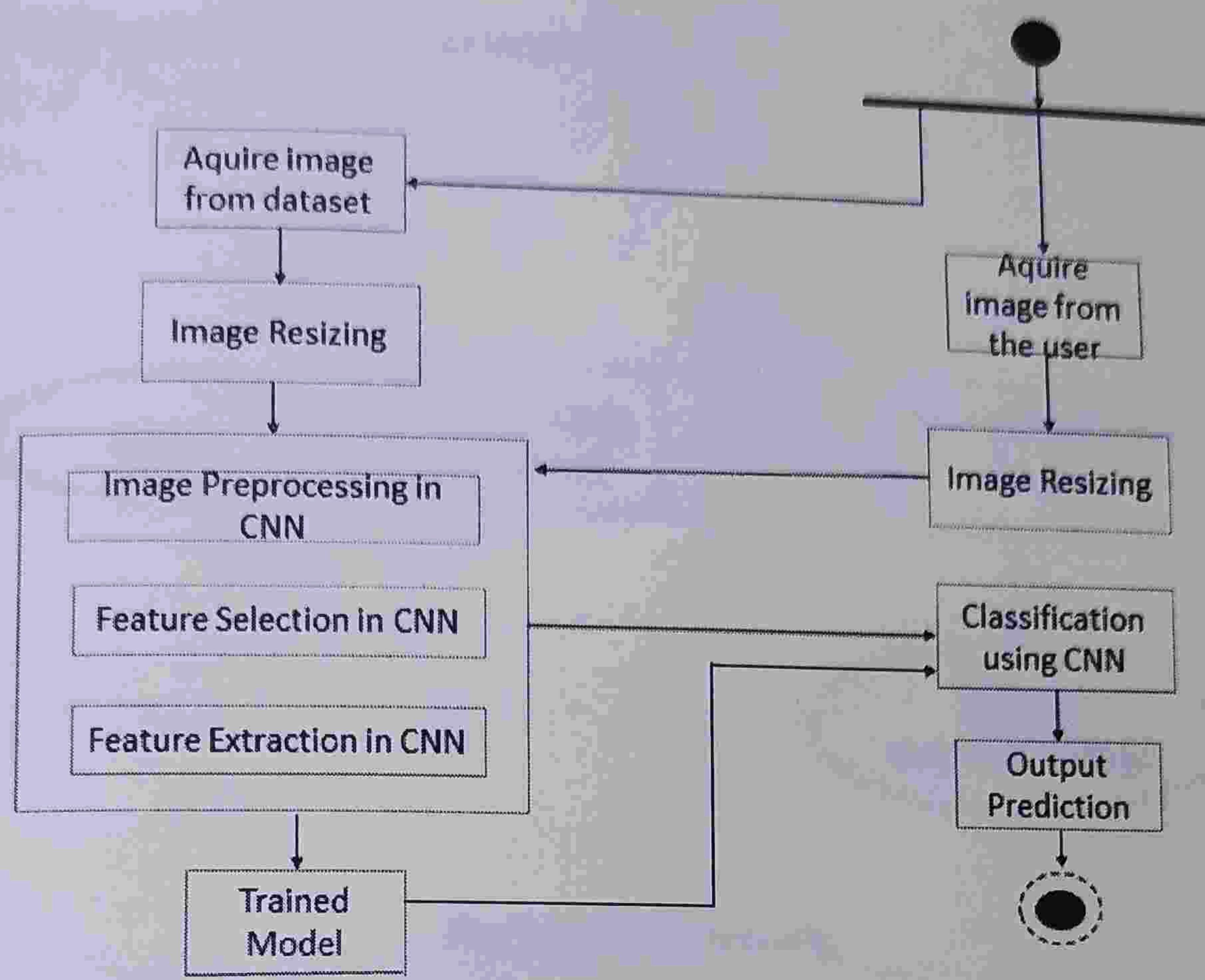

Fig 6.5 Activity Diagram

CHAPTER 7

DATA FLOW DIAGRAM

7.1 Data Flow Diagram for Pre-processing

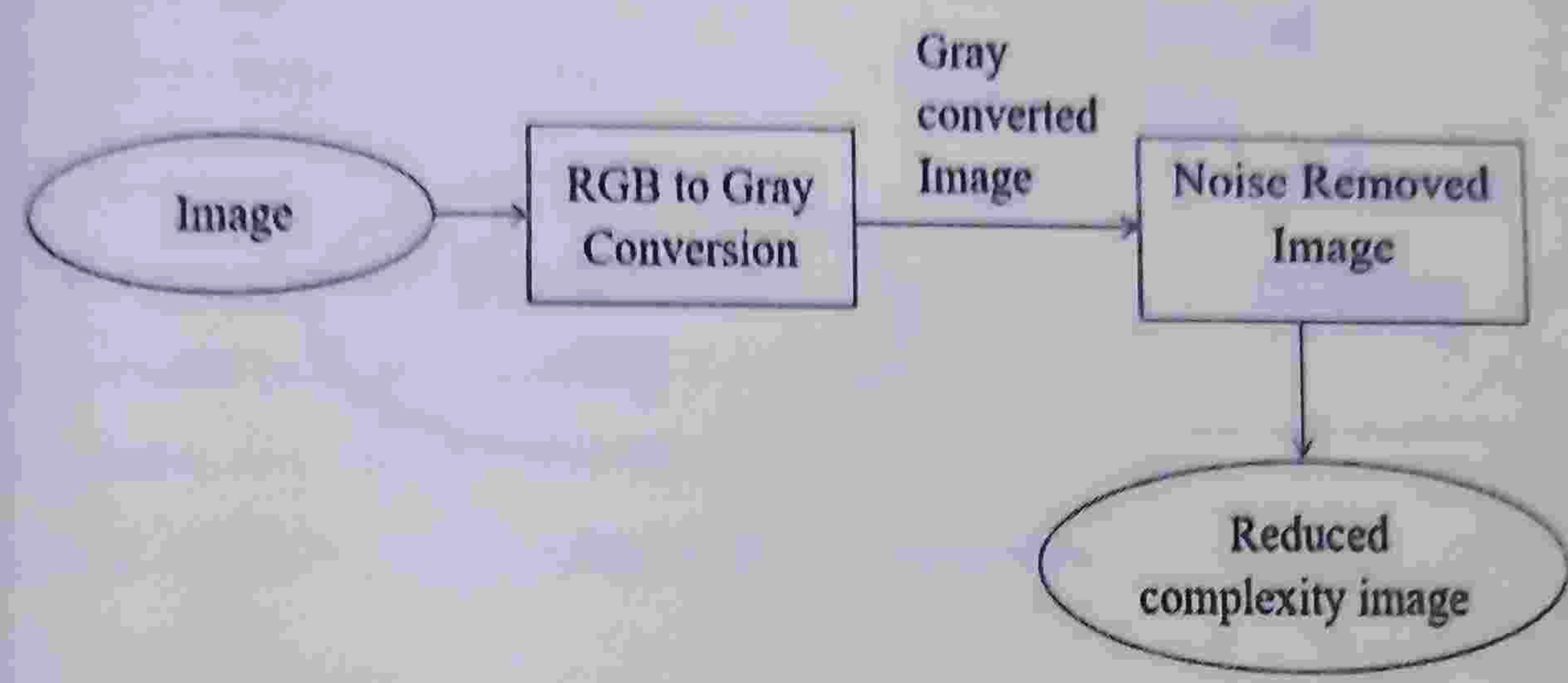

Fig 7.1: Pre-Processing

7.2 Data Flow Diagram for Identification

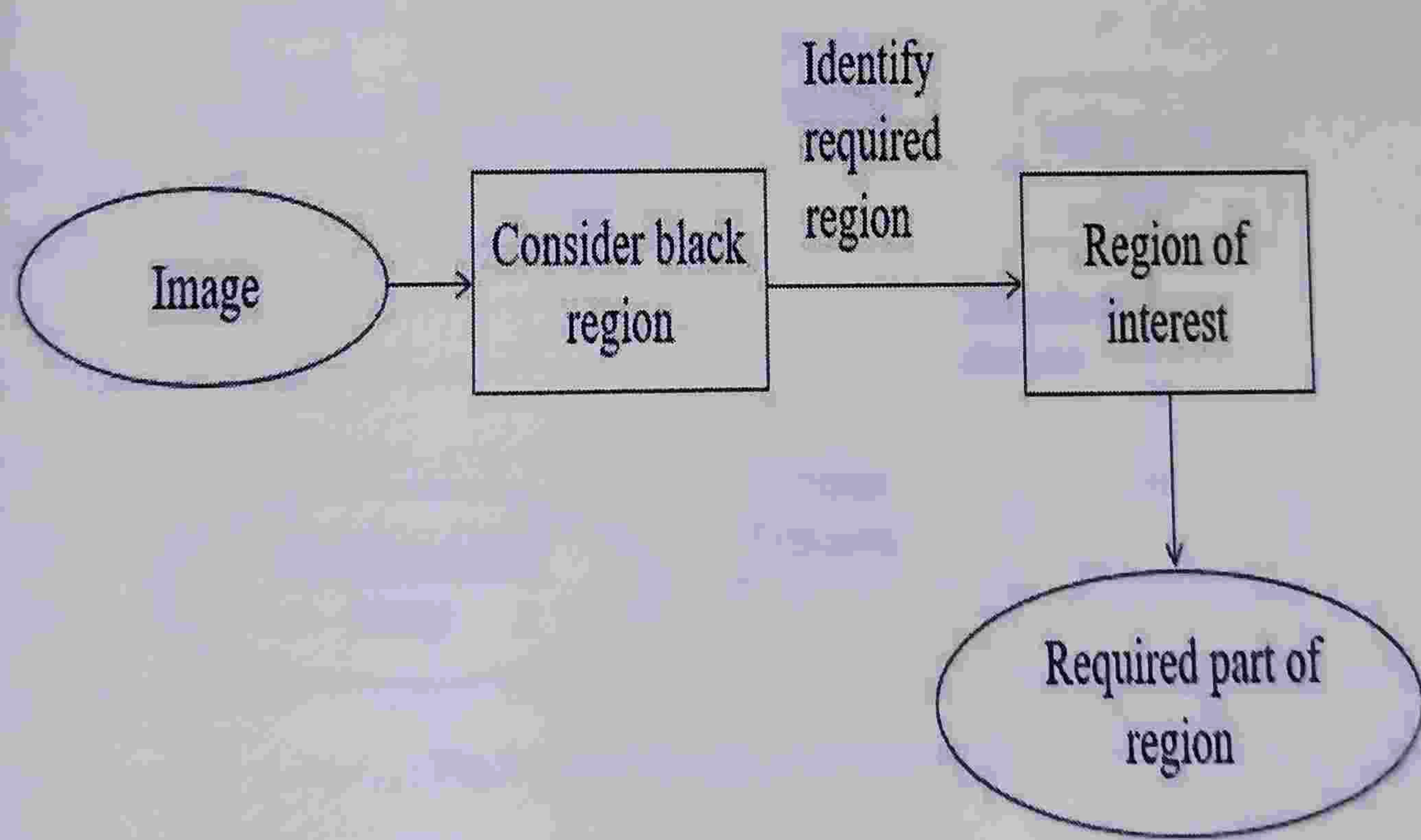

Fig 7.2: Identification

7.3 Data Flow Diagram for Feature Extraction

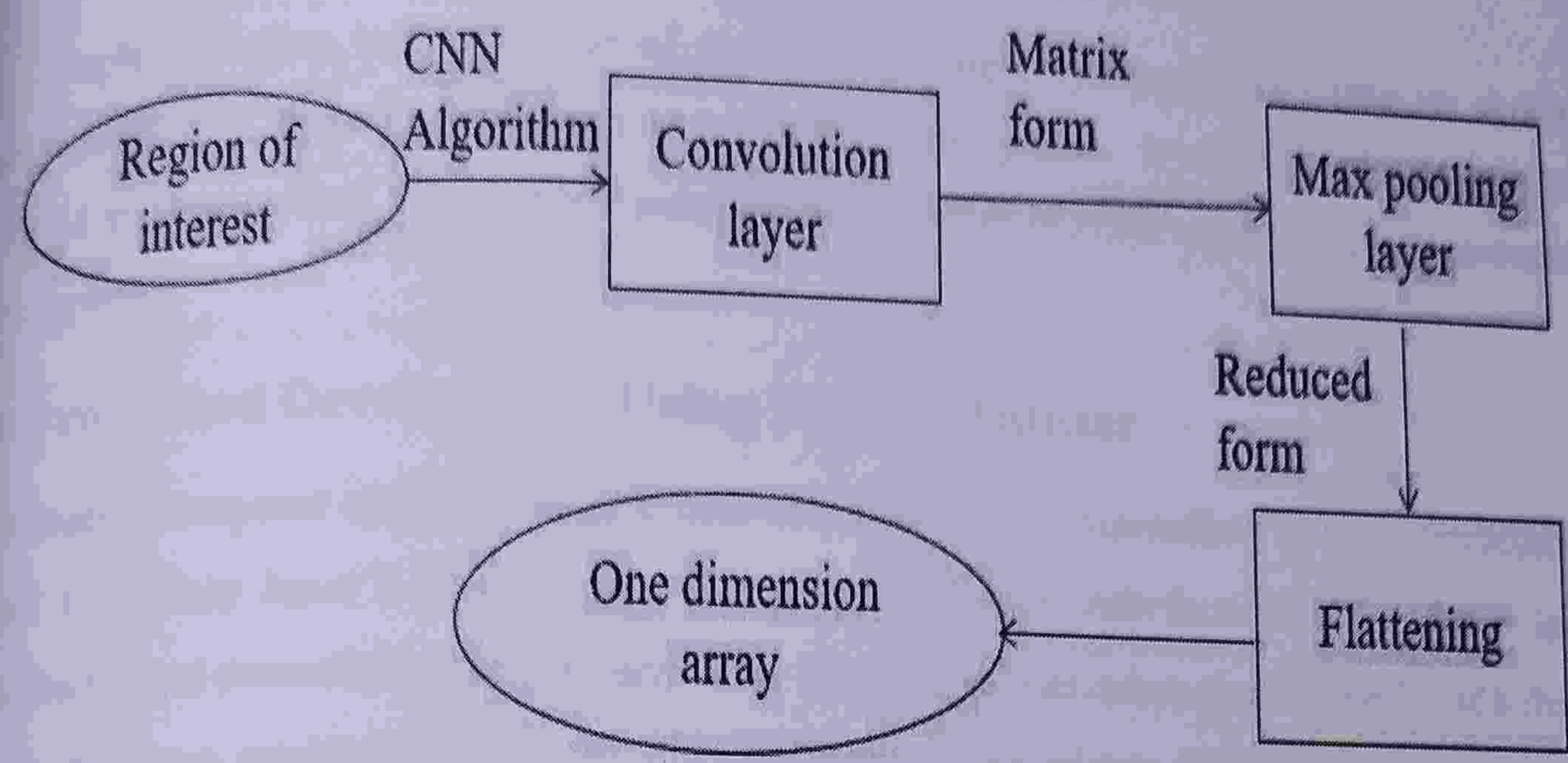

Fig 7.3: Feature Extraction

7.4 Data Flow Diagram for Classification and Detection

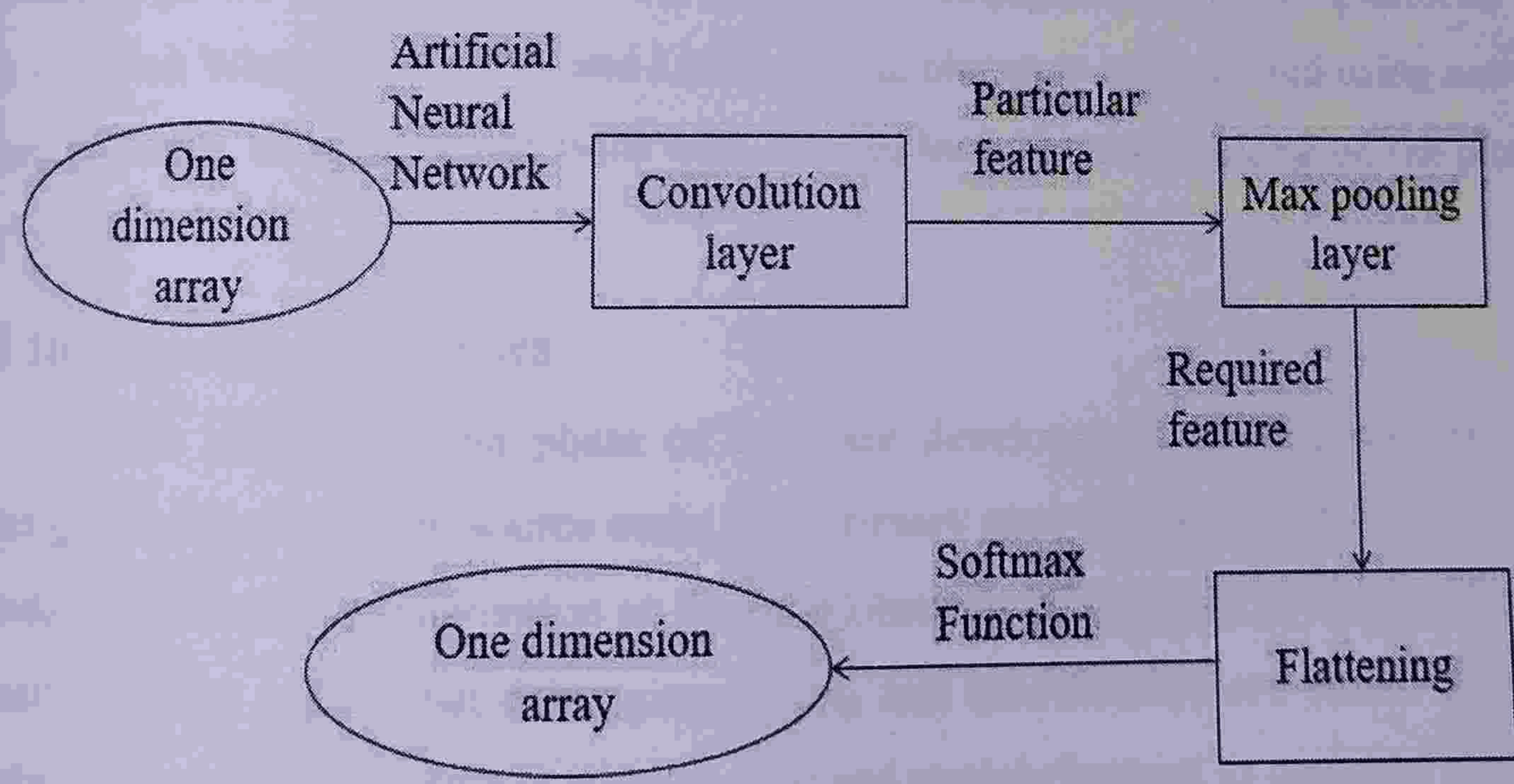

Fig 7.4 Classification and Detection

CHAPTER 8

IMPLEMENTATION

Implementation is the process of converting a new system design into an operational one. It is the key stage in achieving a successful new system. It must therefore be carefully planned and controlled. The implementation of a system is done after the development effort is completed

8.1 Front-End Development Using Python Tkinter

Modern computer applications are user-friendly. User interaction is not restricted to console-based I/O. They have a more ergonomic graphical user interface (GUI) thanks to high-speed processors and powerful graphics hardware. These applications can receive inputs through mouse clicks and can enable the user to choose from alternatives with the help of radio buttons, dropdown lists, and other GUI elements.

8.2 Tkinter Programming

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object- oriented interface to the Tk GUI toolkit. Tkinter has several strengths. It's cross- platform, so the same code works on Windows, macOS, and Linux. Visual elements are rendered using native operating system elements, so applications built with Tkinter look like they belong on the platform where they're run.

8.3 Implementation Issues

The implementation phase of software development is concerned with translating design specifications into source code. The primary goal of implementation is to write source code and internal documentation so that conformance of the code to its specifications can be easily verified and so that debugging testing and modification are eased. This goal can be achieved by making the source code as clear and straightforward as possible. Simplicity clarity and elegance are the hallmarks of good programs and these characteristics have been implemented in each program module.

The goals of implementation are as follows.

- Minimize the memory required.
- Maximize output readability.
- Maximize source text readability.
- Minimize the number of source statements.
- Minimize development time

8.4 Module specification:

Module Specification is the way to improve the structural design by breaking down the system into modules and solving it as an independent task. By doing so the complexity is reduced and the modules can be tested independently. The number of modules for our model is three, namely preprocessing, identification, feature extraction and detection. So each phase signify the functionalities provided by the proposed system. In the data pre-processing phase noise removal using median filtering is done.

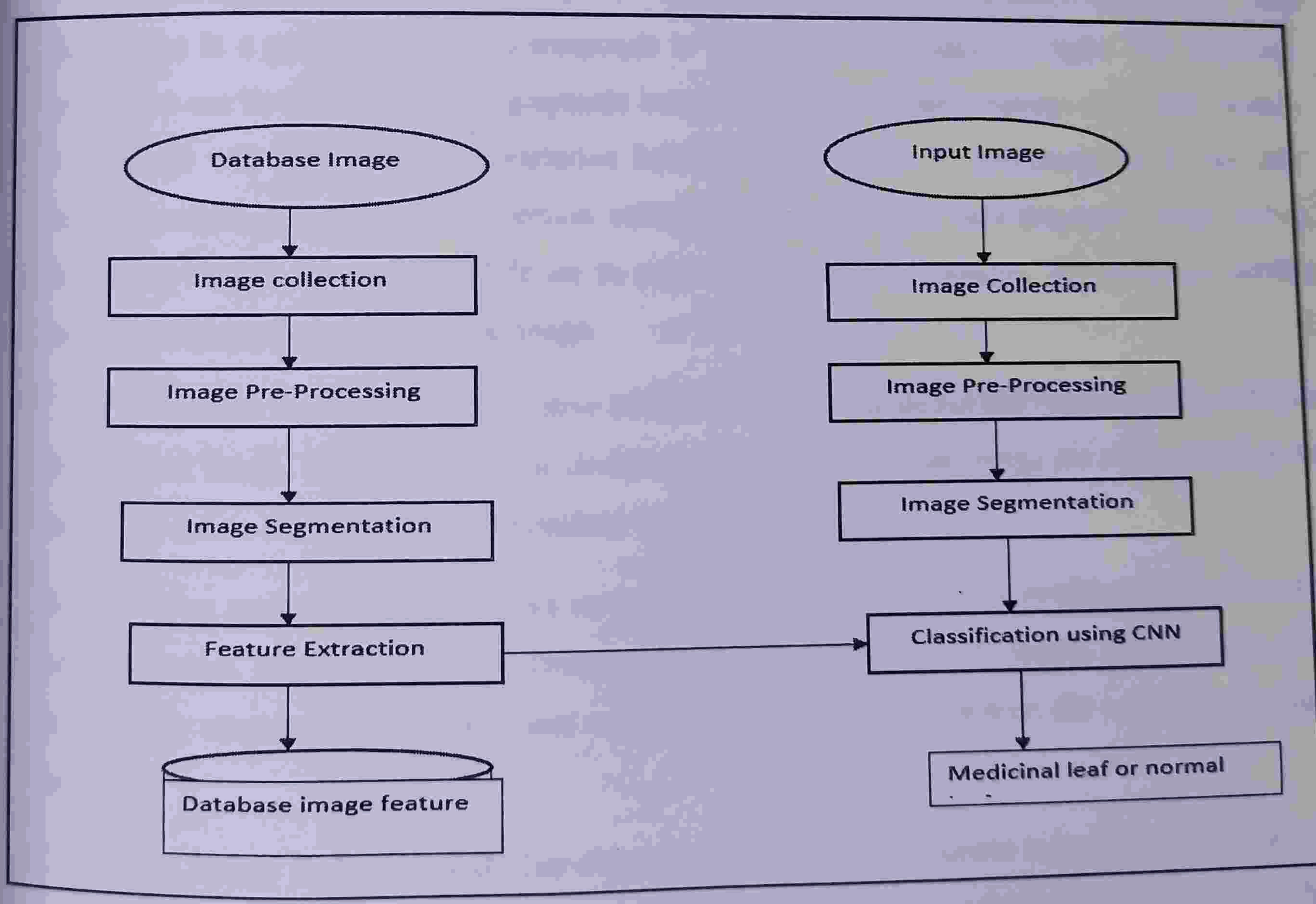

Fig 8.4: Data flow Diagram of Training and Testing Phase

The System design mainly consists of

1. Image Collection
2. Image Preprocessing
3. Image Segmentation
4. Feature Extraction
5. Training
6. Classification

1. Image Collection

The dataset that we have used in this project is available publicly on the internet

2. Image Preprocessing

The goal of pre-processing is an improvement of image data that reduces unwanted distortions and enhances some image features important for further image processing. Image pre-processing involves three main things a) Grayscale conversion b) Noise removal c) Image enhancement

a) **Grayscale conversion**: Grayscale image contains only brightness information. Each pixel value in a grayscale image corresponds to an amount or quantity of light. The brightness graduation can be differentiated in grayscale image. Grayscale image measures only light intensity 8 bit image will have brightness variation from 0 to 255 where '0' represents black and '255' represent white. In grayscale conversion color image is converted into grayscale image shows. Grayscale images are easier and faster to process than colored images. All image processing technique are applied on grayscale image.

b) **Noise Removal**: The objective of noise removal is to detect and remove unwanted noise from digital image. The difficulty is in deciding which features of an image are real and which are caused by noise. Noise is random variations in pixel values.

We are using median filter to remove unwanted noise. Median filter is nonlinear filter, it leaves edges invariant. Median filter is implemented by sliding window of odd length. Each sample value is sorted by magnitude, the centermost value is median of sample within the window, is a filter output.

c) **Image Enhancement**: The objective of image enhancement is to process an image to increase visibility of feature of interest. Here contrast enhancement is used to get better quality result.

3. Image Segmentation :

Image segmentation are of many types such as clustering, threshold, neutral network based and edge based. In this implementation we are using the clustering algorithm called mean shift clustering for image segmentation. This algorithm uses the sliding window method for converging to the Centre of maximum dense area. This algorithm makes use of many sliding windows to converge the maximum dense region. Mean shift clustering Algorithm This algorithm is mainly used for detecting highly dense region.

4. Feature Extraction:

There are many features of an image mainly color, texture, and shape. Here we are considering three features that are color histogram, Texture which resembles color, shape, and texture.

5. Training

Training dataset was created from images of known Cancer stages. Classifiers are trained on the created training dataset. A testing dataset is placed in a temporary folder. Predicted results from the test case, Plots classifiers graphs and add feature-sets to test case file, to make image processing models more accurate

6. Classification

The binary classifier which makes use of the hyper-plane which is also called as the decision boundary between two of the classes is called as Convolution Neural Network. Some of the problems are pattern recognition like texture classification makes use of CNN. Mapping of nonlinear input data to the linear data provides good classification in high dimensional space in CNN. The marginal distance is maximized between different classes by CNN. Different Kernels are used to divide the classes. CNN is basically a binary classifier that determines hyper plane in dividing two classes. The boundary is maximized between the hyperplane and two classes. The samples that are nearest to the margin will be selected in determining the hyperplane is called support vectors.

8.5 CNN Algorithm

Convolutional Neural Network is one of the main categories to do image classification and image recognition in neural networks. Scene labeling, objects detections, and face recognition, etc., are some of the areas where convolutional neural networks are widely used. CNN takes an image as input, which is classified and process under a certain category such as dog, cat, lion, tiger, etc. The computer sees an image as an array of pixels and depends on the resolution of the image. Based on image resolution, it will see as **h*w*d**, where h= height w= width and d= dimension. For example, An RGB image is **6*6*3** array of the matrix, and the grayscale image is **4*4* 1** array of the matrix. In CNN, each input image will pass through a sequence of convolution layers along with pooling, fully connected layers, filters (Also known as kernels). After that, we will apply the Soft-max function to classify an object with probabilistic values 0 and 1.

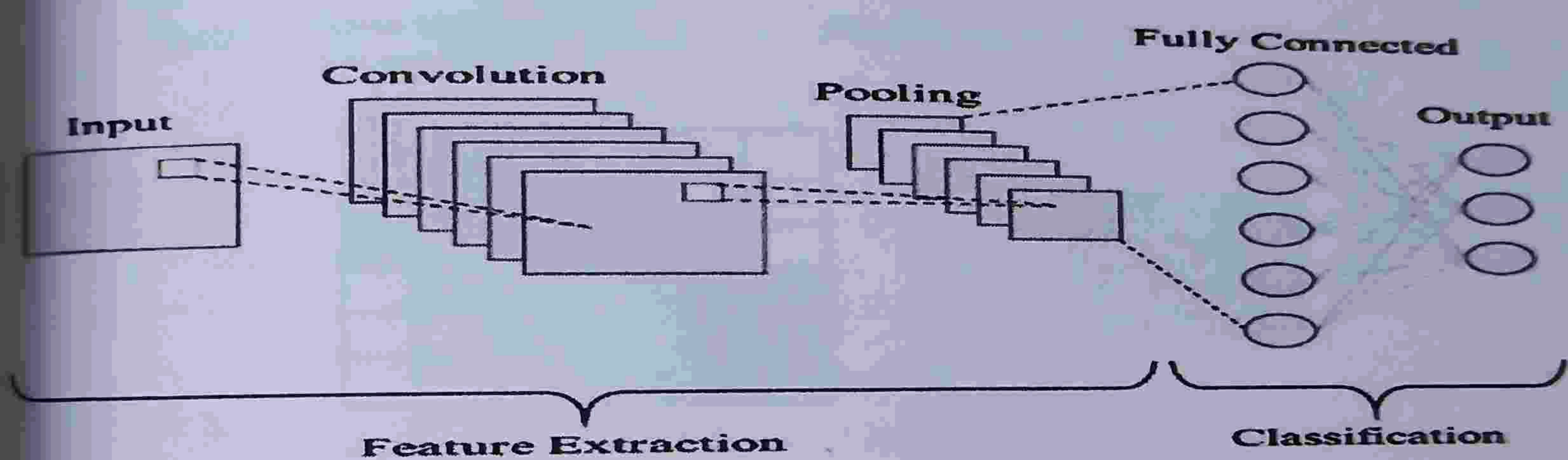

Fig8.5: Convolution Neural Network

8.5.1 Convolution Layer

Convolution layer is the first layer to extract features from an input image. By learning image features using a small square of input data, the convolutional layer preserves the relationship between pixels. It is a mathematical operation which takes two inputs such as image matrix and a kernel or filter.

- The dimension of the image matrix is **$h \times w \times d$**.
- The dimension of the filter is **$f_h \times f_w \times d$**.
- The dimension of the output is **$(h-f_h+1)\times(w-f_w+1)\times 1$**.

Let's start with consideration a 5*5 image whose pixel values are 0, 1, and filter matrix 3*3 as:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

5×5 — Image Matrix $\quad$ 3×3 — Filter Matrix

The convolution of 5*5 image matrix multiplies with 3*3 filter matrix is called "Features Map" and show as an output.

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 3 & 4 \\ 2 & 4 & 3 \\ 2 & 3 & 4 \end{bmatrix}$$

Convolved Feature

Convolution of an image with different filters can perform an operation such as blur, sharpen, and edge detection by applying filters.

8.5.2 Padding

Padding plays a crucial role in building the convolutional neural network. If the image will get shrink and if we will take a neural network with 100's of layers on it, it will give us a small image after filtered in the end. If we take a three by three filter on top of a grayscale image and do the convolving then what will happen?

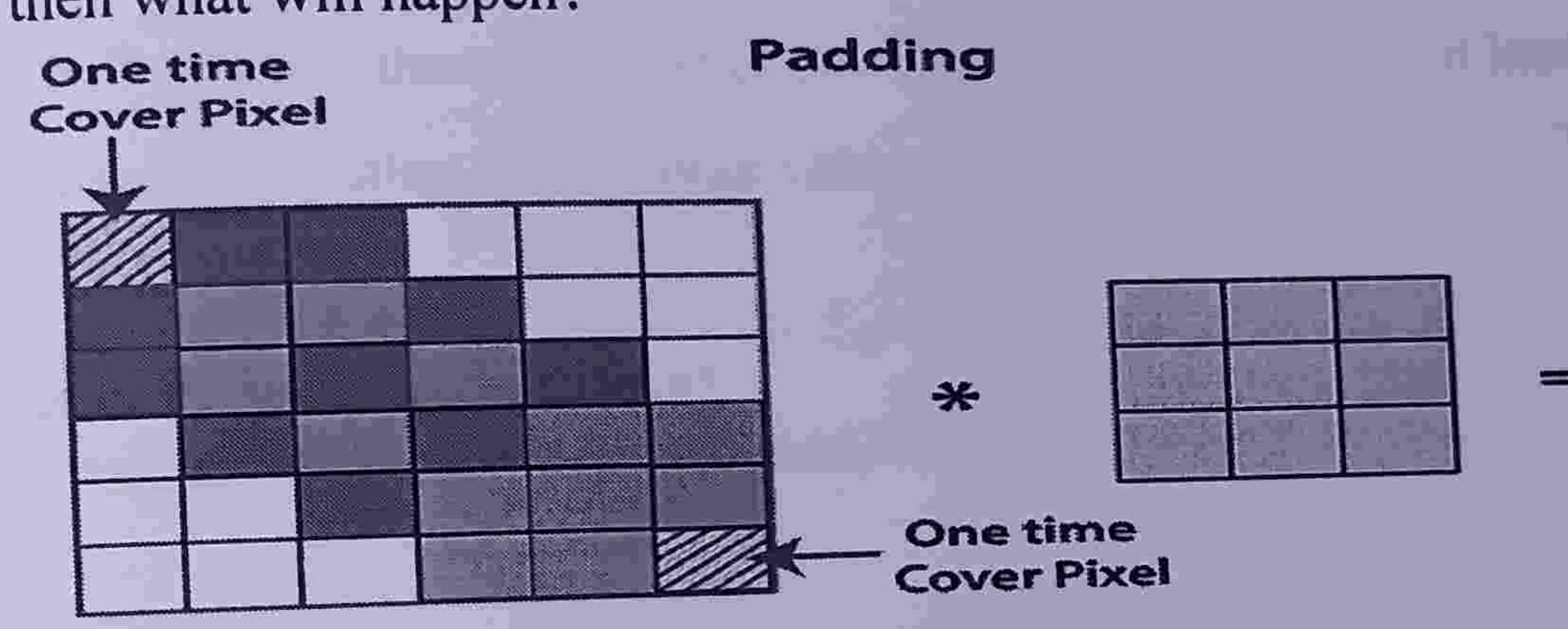

Fig 8.5.2: Padding

It is clear from the above picture that the pixel in the corner will only get covers one time, but the middle pixel will get covered more than once. It means that we have more information on that middle pixel, so there are two downsides:

- Shrinking outputs
- Losing information on the corner of the image.

To overcome this, we have introduced padding to an image. "Padding is an additional layer which can add to the border of an image."

8.5.3 Pooling Layer

Pooling layer plays an important role in pre-processing of an image. Pooling layer reduces the number of parameters when the images are too large. Pooling is "downscaling" of the image obtained from the previous layers. It can be compared to shrinking an image to reduce its pixel density. Spatial pooling is also called down sampling or subsampling, which reduces the dimensionality of each map but retains the important information.

There are the following types of spatial pooling:

Max Pooling

Max pooling is a "sample-based discretization process". Its main objective is to downscale an input representation, reducing its dimensionality and allowing for the assumption to be made about features contained in the sub-region binned. Max pooling is done by applying a max filter to non-overlapping sub-regions of the initial representation.

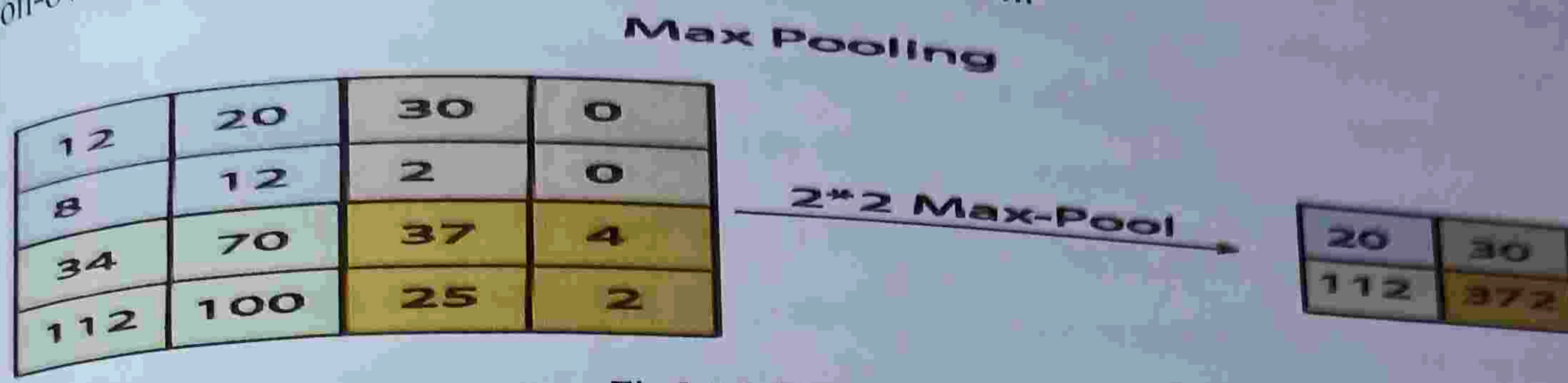

Fig 8.5.3: Max Pooling

Average Pooling

Down-scaling will perform through average pooling by dividing the input into rectangular pooling regions and computing the average values of each region.

Syntax

layer = averagePooling2dLayer(poolSize)
layer = averagePooling2dLayer(poolSize,Name,Value)

Sum Pooling

The sub-region for sum pooling or mean pooling are set exactly the same as for max-pooling but instead of using the max function we use sum or mean.

8.5.4 Fully Connected Layer

The fully connected layer is a layer in which the input from the other layers will be flattened into a vector and sent. It will transform the output into the desired number of classes by the network.

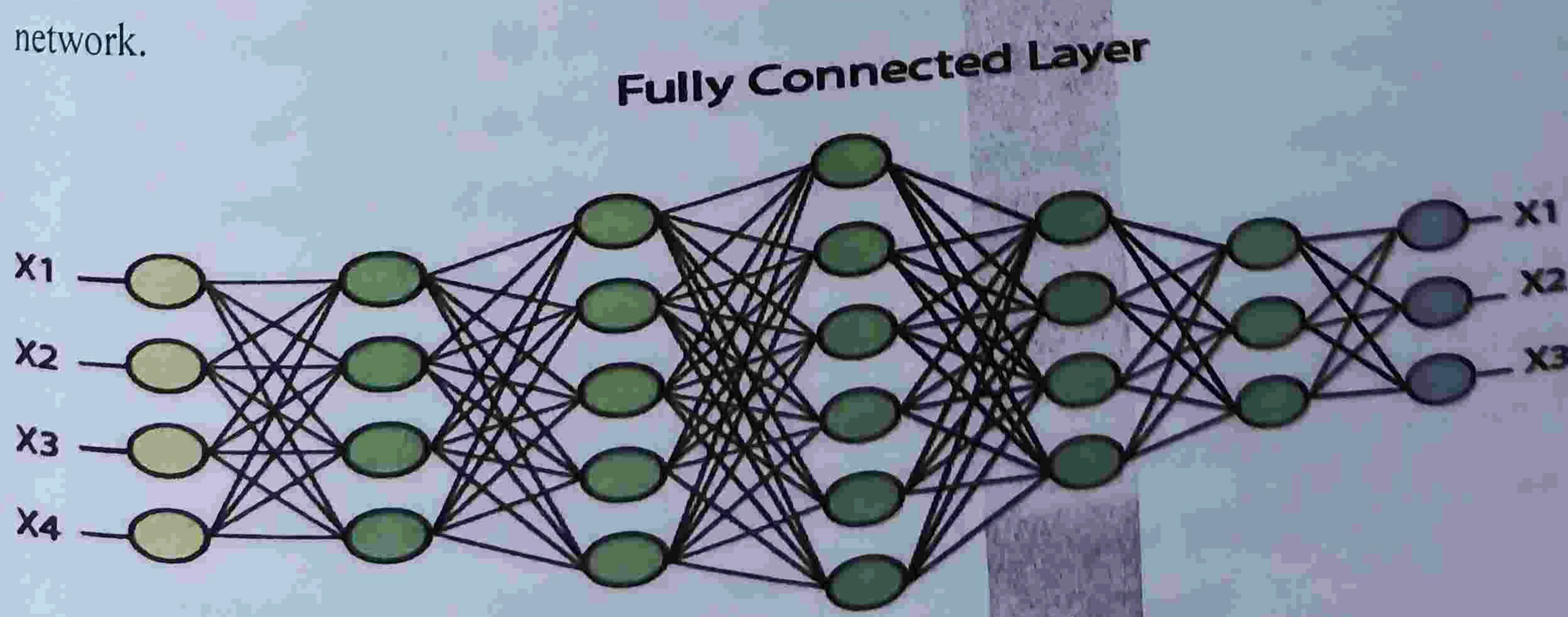

Fig 8.5.4: Fully Connected Layer

CHAPTER 9

TESTING

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. Testing is executing a system to identify any gaps, errors, or missing requirements in contrary to the actual requirements. System testing of a software or hardware is a testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing fails within the scope of black-box testing, and such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, of all the _integrated' software components that have passed integration testing and the software system itself integrated with any applicable software systems. The purpose of integration testing is to detect any inconsistencies betweenthe software units that are integrated together. System testing is a more limited type of testing. It seeks to detect defects both within the inter-assemblages and within the system. System testing is performed on the entire system in the context of a Functional Requirement Specification (FRS) and / or a System Requirement Specification (SRS). System testing testsnot only the design, but also the behavior and even the believed expectation of the customer. It is also intended to test up to and beyond the bounds defined in the software / hardware requirement specification. Before applying methods to design effective test cases, a software engineer must understandthe basic principle that guides software testing. All the tests should be traceable to customer requirements.

9.1 Types of testing

Software testing methods and traditionally divided into two: white-box and black-box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

a) **White-box testing** (also known as clear box testing, glass box testing, transparent box testing and structural testing, by seeing the source code) tests internal structures or workings of a program, as opposed to the functionality exposed to the end-user. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. While white-box testing can be applied at the unit, integration, and system levels of the software testing process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between sub systems duringa system-level test. Though this method of test design can uncover many errors or problems, it might not detect unimplemented parts of the specification or missing requirements.

b) **Black box testing**: The technique of testing without having any knowledge of the interior workings of the application is called black-box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black-box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon. This project has been tested under different circumstances, which includes different types such as Unit testing, Integration testing and System testing that are described below.

9.2 Levels of Testing:

There are different levels during the process of testing. Levels of testing include different methodologies that can be used while conducting software testing. The main levels of software testing are:

➢ **Functional Testing**: This is a type of black-box testing that is based on the specifications of the software thatis to be tested. The application is tested by providing input and then the results are examined that need to conform to the functionality it was intended for. Functional testingof software is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. There are five steps that are involved while testing an application for functionality.

- The determination of the functionality that the intended application is meant to perform.
- The creation of test data based on the specifications of the application.
- The creation of test data based on the specifications of the application.
- The output based on the test data and the specifications of the application.
- The writing of test scenarios and the execution of testcases.
- The comparison of actual and expected results based on the executed testcases.

➢ **Non-functional Testing**: This section is based upon testing an application from its non-functional attributes. Nonfunctional testing involves testing software from the requirements which are non- functional in nature but important such as performance, security, user interface, etc.

9.3 Unit Testing

Unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures and operating procedures are tested to determine if they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. During the development process itself all the syntax errors etc. got rooted out. For this developed test case that resultin executing every instruction in the program or module i.e. every path through program wastested.

9.4 Integration Testing:

Integration testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing. Integration testing is defined as the testing of combined parts of an application to determine if they function correctly. It occurs after unit testing and before validation testing. Integration testing can be done in two ways: Bottom-up integration testing and Top-down integration testing.

1. **Bottom-up Integration**: This testing begins with unit testing, followed by tests of progressively higher-level combinations of units called modules or builds.
2. **Top-down Integration**: In this testing, the highest-level modules are tested first and progressively, lower-level modules are tested thereafter. In a comprehensive software development environment, bottom-up testing is usually done first, followed by top-down testing. The process concludes with multiple tests of the complete application, preferably in scenarios designed to mimic actual situations.

9.5 System testing:

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic. System testing is important because of the following reasons:

- System testing is the first step in the Software Development Life Cycle, where the application is tested.
- The application is tested thoroughly to verify that it meets the functional and technical specifications.
- The application is tested in an environment that is very close to the production environment where the application will be deployed.
- System testing enables us to test, verify, and validate both the business requirements as well as the application architecture.

CHAPTER 10

RESULTS AND SNAPSHOTS

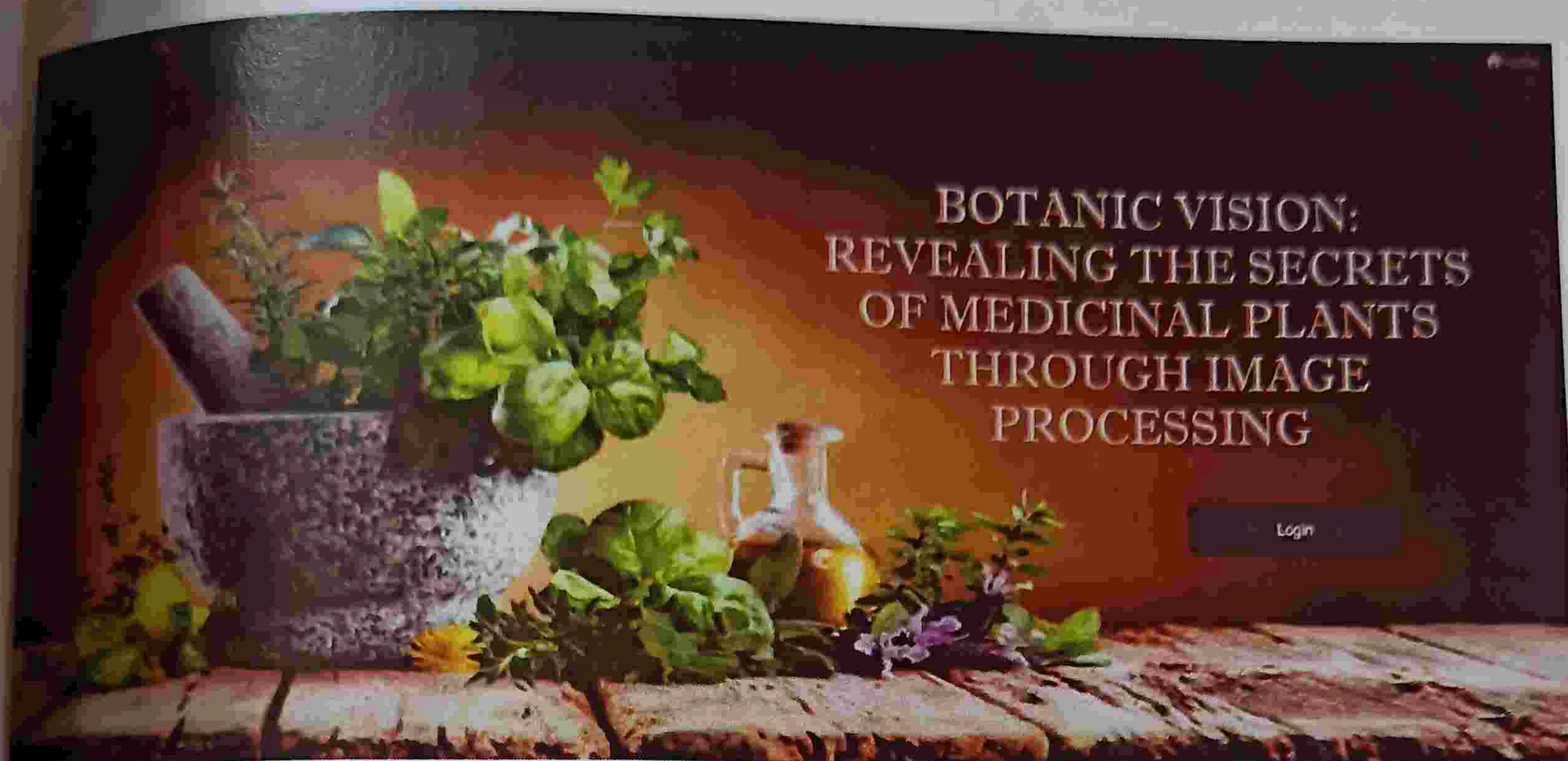

Fig 10.1: Website Interface

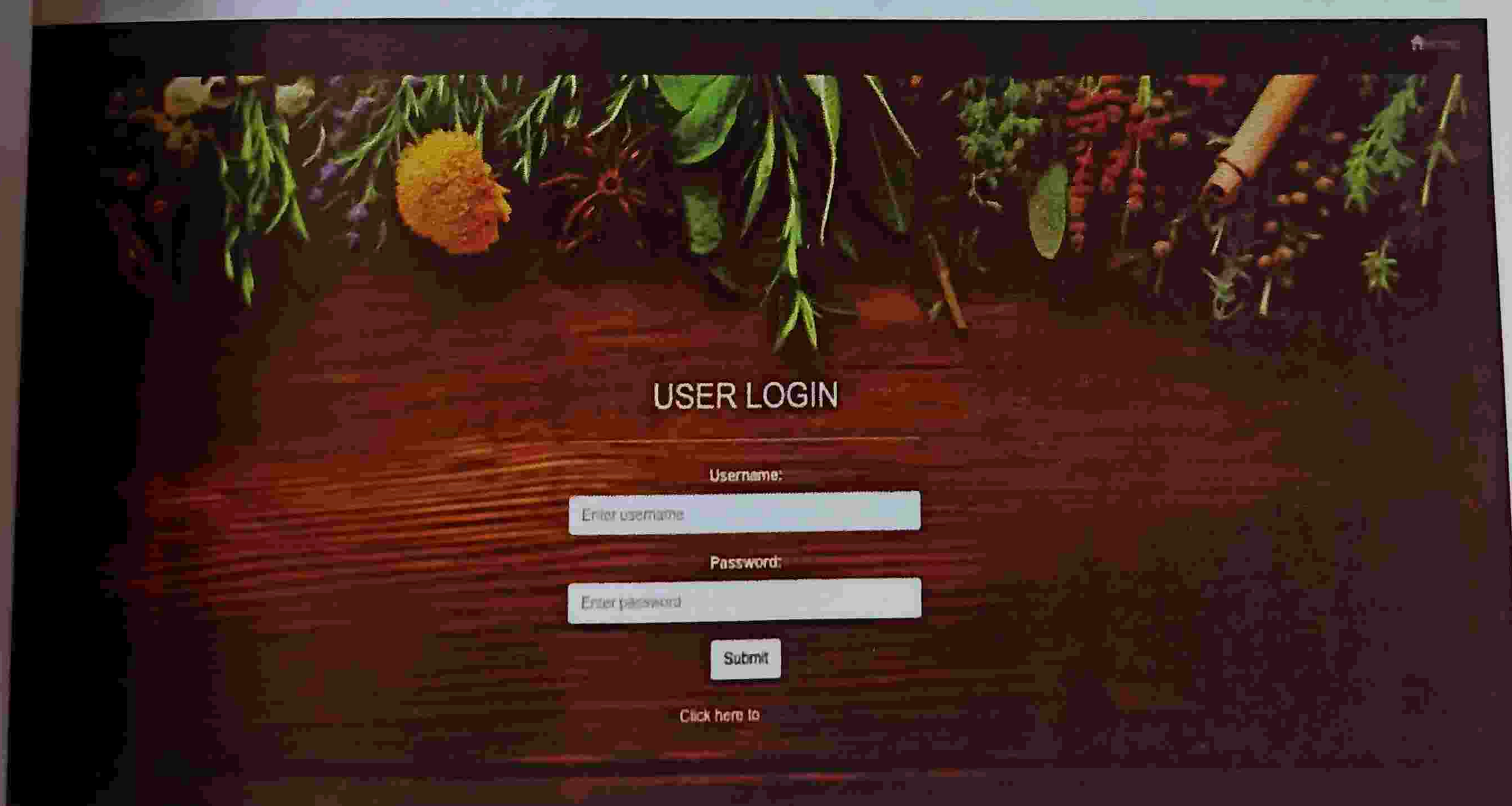

Fig 10.2: User Login

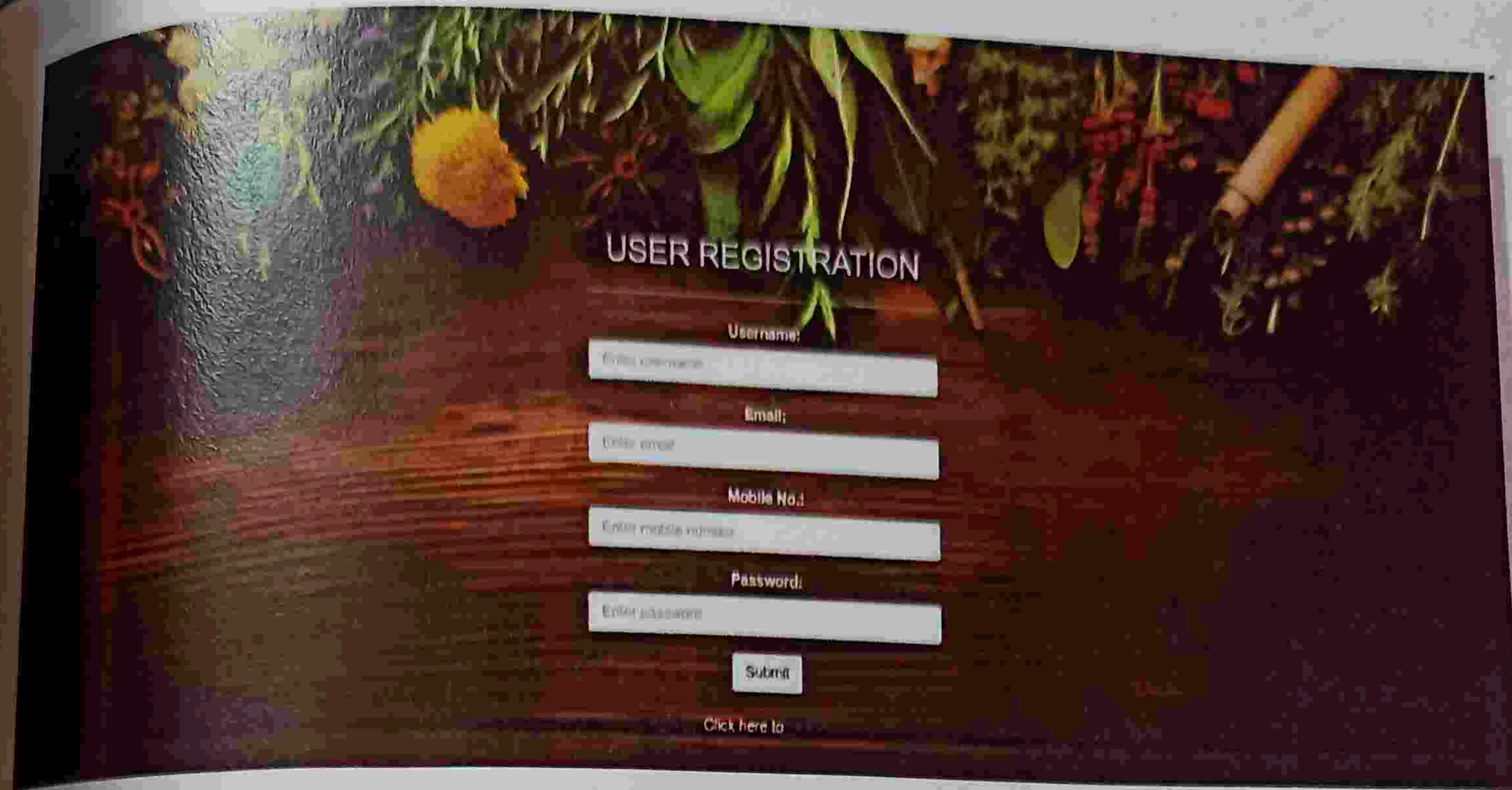

Fig 10.3: User Registration

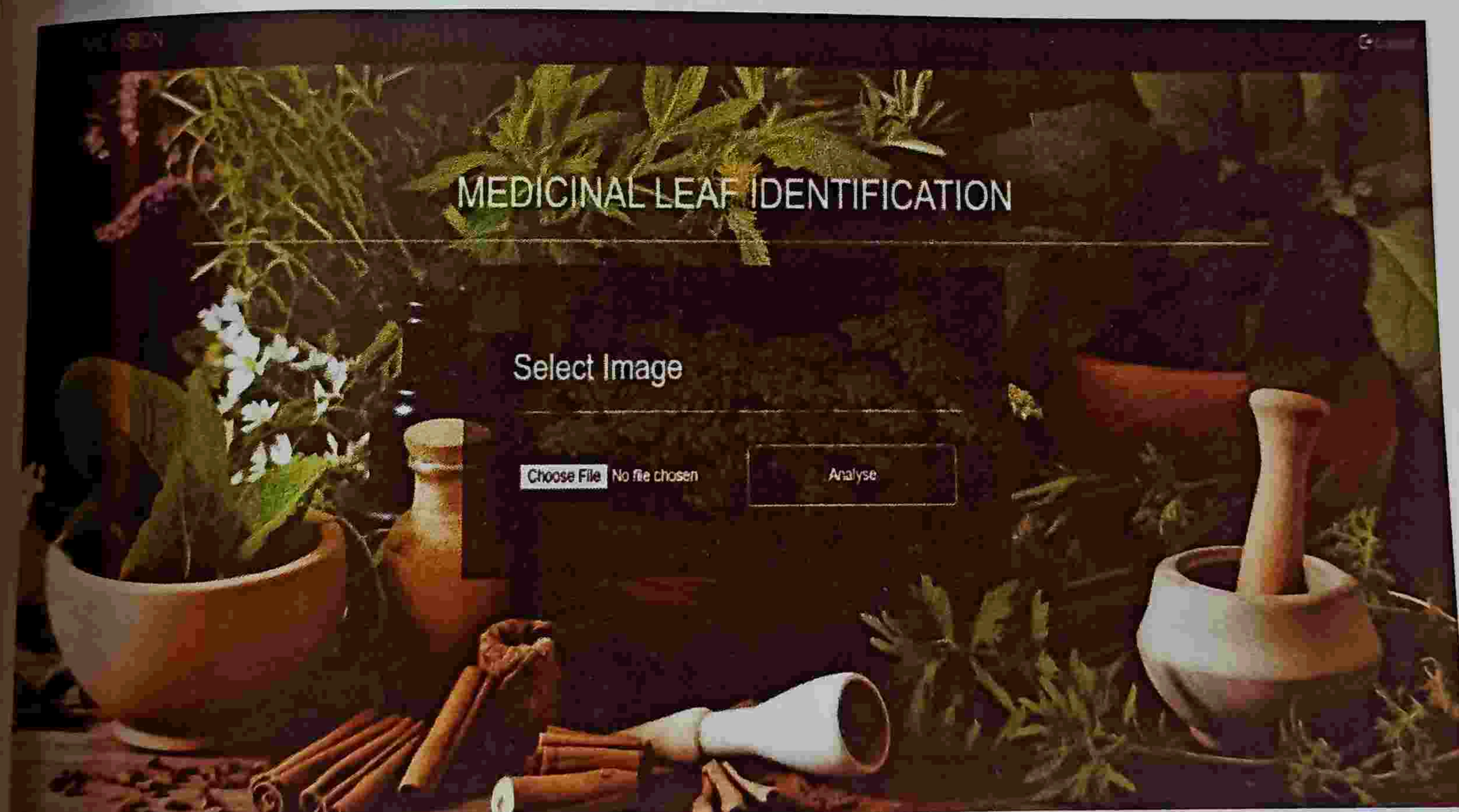

Fig 10.4: Upload Image To Be Predicted

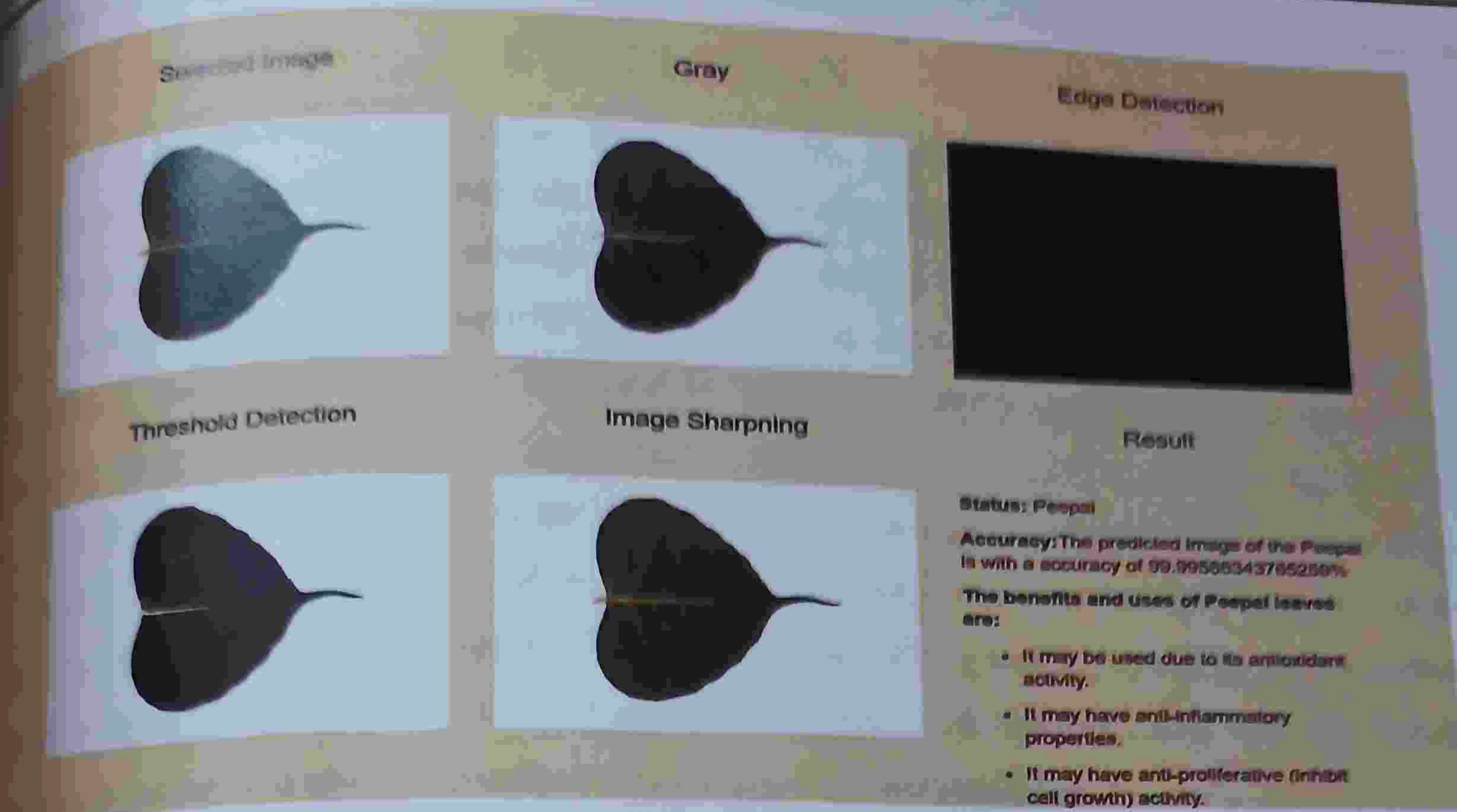

Fig 10.5: Resulted Image

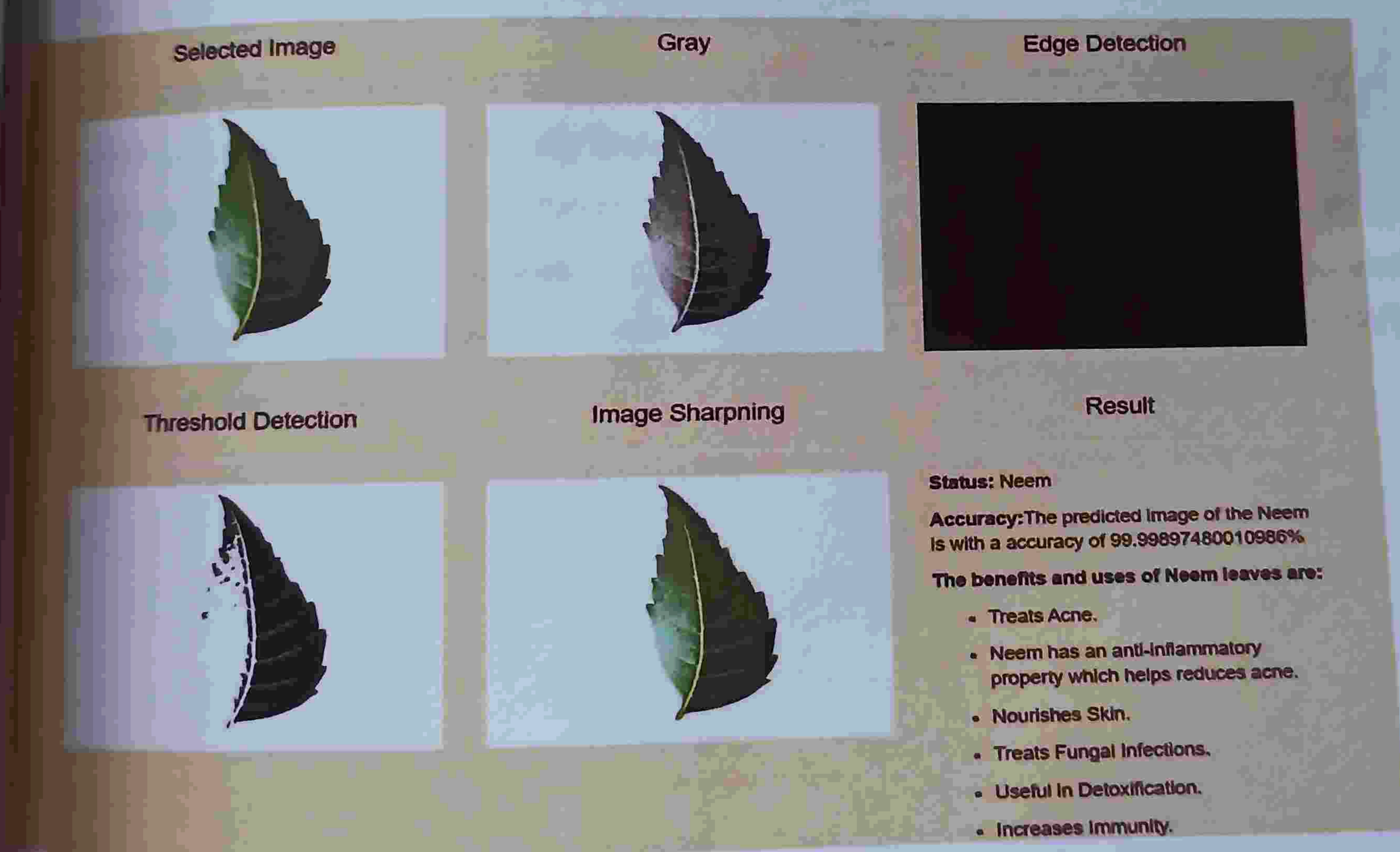

Fig 10.6: Resulted Image

CONCLUSION

Plants are necessary for human survival. Herbs, particularly, are employed by indigenous populations as folk medicines from old period. Herbs are typically recognized by clinicians based on decades of intimate sensory or olfactory experience. Recent improvements in analytical technology have made it much easier to identify herbs depending on scientific evidence. This helps a lot of individuals, particularly those are not used to recognising herbs. additionally for time-consuming methods, laboratory-based analysis necessitates expertise in sample healing and data explanation. As a result, a simple and reliable method for identifying herbs is required. Herbal identification anticipated to benefit from the combination of computation and statistical examination. This non-destructive technique will be the preferred approach for quickly identifying herbs, especially for individuals who cannot able to use expensive analytical equipment. This work reviews about different methods for plants recognition and also reviews their advantages and disadvantages.

FUTURE SCOPE

The proposed methods are not suitable for tiny leaves or plants without a proper leaf. Efforts may be made to develop methods to identify these types of plants. The algorithms may be implemented on a standalone single board computer connected to a scanner. A portable system may be developed for field use.In future research in the area of plants identification, improved machine learning classifier with some pre-processing and feature selection models will be used to solve the accuracy related issues and enhance the performance.

REFERENCE

[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 ComputationalIntelligence 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, MayJune 2017.

[3] Adams Begue, VenithaKowlessur, FawziMahomoodally, 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] Marco Seeland, Michael Rzanny, NedalAlaqraa, Jana Wa ¨ldchen, Patrick Ma ¨der, "Plant species classification using flower images—A comparative study of local feature representations", PLOS ONE | DOI:10.1371/journal.pone.0170629 February 24, 2017.

[7] Pradeepkumar Choudhary, Rahul Khandekar, AakashBorkar, and PunitChotaliya, "Image processing algorithm for fruit identification", International Research Journal of Engineering and Technology (IRJET), Vol 4 Issue 3, e-ISSN: 2395 -0056, p-ISSN: 2395- 0072, Mar -2017. [8] D Venkataraman and Mangayarkarasi N, "Computer Vision Based Feature Extraction of Leaves for Identification of Medicinal Values of Plants", IEEE International Conference on ComputationalIntelligence and Computing Research, 978-1-5090-0612-0/16/$31.00©2016 IEE