A PROJECT REPORT ON

"IDENTIFICATION OF MEDICINAL PLANTS USING MACHINE LEARNING"

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE. IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE

BACHELOR OF ENGINEERING

IN

COMPUTER ENGINEERING

SUBMITTED BY

Ms. Gauri P. Girme	**Exam No. B400400301**
Ms. Rachana D. Kulthe	**Exam No. B400400330**
Ms. Rutugandh P. Gaikwad	**Exam No. B400400295**
Ms. Vaishnavi S. Deshmukh	**Exam No. B400400290**

UNDER THE GUIDANCE OF

Prof. Dr. P. D. Lambhate

DEPARTMENT OF COMPUTER ENGINEERING

JSPM's Jayawantrao Sawant College of Engineering, Pune

Hadapsar, Tal-Haveli, Pin- 411028, India

2024-2025

DEPARTMENT OF COMPUTER ENGINEERING

JSPM's Jayawantrao Sawant College of Engineering, Pune

Hadapsar, Tal-Haveli, Pin- 411028, India

CERTIFICATE

This is to certify that the Project report entitled

"IDENTIFICATION OF MEDICINAL PLANTS USING MACHINE LEARNING"

Submitted by

Ms. Gauri P. Girme	**Exam No. B400400301**
Ms. Rachana D. Kulthe	**Exam No. B400400330**
Ms. Rutugandh P. Gaikwad	**Exam No. B400400295**
Ms. Vaishnavi S. Deshmukh	**Exam No. B400400290**

This is a bonafide work carried out under the supervision of Prof. Dr. P. D. Lambhate and it is submitted towards the partial fulfillment of the requirement of Savitribai Phule Pune University, Pune for the award of the degree of Bachelor of Engineering in Computer Engineering, at JSPM'S Jayawantrao Sawant College of Engineering, Pune.

Dr. P. D. Lambhate
Internal Guide

External Examiner

Prof. N. R. Zinzurke
Project Coordinator

Dr. P. D. Lambhate
Head of Department

Dr. R. D. Kanphade
Principal

ACKNOWLEDGEMENT

As a wise person once said about the nobility of the teaching profession,” Catch a fish, and you feed a man for dinner, but teach a man how to catch a fish, and you feed him for life.”

We would like to express our gratitude to our Project guide, **Prof. Dr. P. D. Lambhate**, who has been an unwavering source of inspiration and assistance. She not only encourage us to undertake this project but also provided valuable guidance through- out its duration, offering support during challenging times.

We are also deeply grateful to **Prof. N. R. Zinzurke**, our project coordinator, for their exceptional support and guidance, which played a pivotal role in the success of this assignment.

We are also grateful to **Prof. Dr. P. D. Lambhate**, Head of Computer Engineering Department, JSCOE for her indispensable support and suggestions.

We are also grateful to **Dr. R. D. Kanphade**, Principal of JSCOE for his indispensable support and suggestions.

Furthermore, we extend our gratitude to all the faculty members of the Computer Engineering Department. Their assistance and willingness to go the extra mile were instrumental in our efficient completion of this project, and they consistently supported us during times of need.

Gauri Girme
Rachana Kulthe
Rutugandh Gaikwad
Vaishnavi Deshmukh

(B. E. Computer Engineering)
JSCOE, Pune

Skillset Used in Our Project and Its Outcome

Computer Vision & Machine Learning

- **What We Learned:** Implemented image classification using a hybrid CNN-SVM model, gaining experience in feature extraction, training neural networks, and combining traditional ML with deep learning techniques.
- **Outcome:** Developed an accurate image-based plant identification system that could distinguish between multiple medicinal plant species with high precision.

Flask (Backend Development)

- **What We Learned:** Built RESTful APIs to handle image uploads, trigger predictions, and manage communication between frontend and backend securely.
- **Outcome:** Established a reliable and modular backend service that connected the ML model and frontend, ensuring low-latency interactions and accurate output delivery.

Image Preprocessing & Handling

- **What We Learned:** Implemented image resizing, format checking, and noise handling to improve the quality of inputs passed to the model.
- **Outcome:** Increased prediction accuracy and reduced model errors by standardizing input images before inference.

Database Management (MongoDB)

- **What We Learned:** Utilized MongoDB for storing plant metadata, user queries, and prediction history using pymongo.
- **Outcome:** Enabled persistent storage and retrieval of relevant data, facilitating logging, future training, and user engagement.

Version Control & Collaboration

- **What We Learned**: Used Git and GitHub for collaborative development, issue tracking, and version management.
- **Outcome**: Improved team coordination, reduced merge conflicts, and maintained a well-documented development history.

Abstract

Plants play an important role in human life by providing oxygen, food, housing, medicine and energy. Plants are rich in medicinal esteem and contain dynamic elements for medicinal use because of the global warming populace, lack of government upholding research exercises and familiarity with medicinal plants. Numerous utility plants are getting pulverized. Manual identification of plants requires significant investment and the identification of plants with the help of experts. To address this issue, we need to acquire more prominence in robotized identification and classification of medicinal plants. Feature extraction and classification are important developments in identifying medicinal plants that affect system's overall accuracy for scientific organizations. The main features required to identify a medicinal plants is its leaf shape, color and texture. Classification algorithms then used these features to divide medicinal plants according to their morphological characteristics. Convolutional Neural Networks(CNN) and Support Vector Machine(SVM) are used for picture identification. These techniques can analyze big datasets and extract features to identify medicinal plants.

Keywords: Medicinal plants, CNN, image recognition, feature extraction, SVM, morphological features, plant identification.

INDEX

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1. Introduction

The identification of Medicinal Plants using Machine Learning Algorithms, lies at the intersection of multiple scientific disciplines, including botany, pharmacology, data science and artificial intelligence. This domain has gained attention due to the critical role medicinal plants play in healthcare systems, both traditional and modern. Medicinal plants are extensively used in treating a wide variety of health conditions globally, providing natural remedies for ailments ranging from minor infections to serious diseases. However, the high diversity of plant species, subtle morphological differences and the overlap in characteristics among species make accurate identification of challenge.

Misidentification of medicinal plants can have serious consequences, including toxicity and ineffective treatments, which highlights the importance of precision in plant identification. Traditionally, experts such as botanists perform this task by relying on in-depth knowledge of plant morphology, ecology and taxonomy. Yet, due to the vast number of species and the demand for faster identification methods, manual identification is time-consuming and not scalable. This is where machine learning(ML) and image processing techniques offer a solution, enabling automated, high-accuracy identification and classification of plants based on visual and biological attributes.

Machine learning algorithms, particularly those focused on image recognition and classification are adopt at processing large datasets and learning complex patterns within them. By leveraging ML techniques, plant characteristics such as leaf shape, color, vein structure, texture and even environmental context can be captured, analyzed and used to identify plant species accurately. In this approach, models are trained on datasets containing numerous plant images, with the algorithm "learning" distinguishing features of each species. This enables the model to recognize and classify plant species accurately when presented with new images.

Moreover, advancements in mobile computing and accessible ML frameworks allows these identification systems to be deployed on portable devices making plant identification feasible in real-time and on-site.

1.2. Significance of This Project

The project contributes to the preservation and documentation of medicinal plant species. Many medicinal plants are endangered or at risk due to deforestation and climate change. By developing an automated identification system, the project aids in creating a digital repository of medicinal plants, making it easier for researchers and conservationists to study, classify and protect these valuable species. The availability of such an advanced identification system could also support sustainable practices in herbal medicine by ensuring the correct plants are harvested and used.

The project advances the field of artificial intelligence in botanical research by integrating deep learning and machine learning techniques for plant identification. Traditional classification methods often struggle with complex, high-dimensional data but the hybrid CNN-SVM approach provides a robust and adaptable solution. The model can be further expanded to include more plant species and even be adapted for other environmental applications.

The project has significant implications in healthcare, research, conservation and technology. By providing a highly accurate, accessible and automated method for identifying medicinal plants, it not only supports scientific research but also promotes the safe and effective use of natural medicines.

The key significances of this project is its ability to enhance the efficiency and reliability of plant identification. By using CNN for feature extraction and SVM for classification, the model ensures better accuracy compared to traditional classification techniques. CNN extracts intricate visual features from plant images, while SVM provides robust classification capabilities, reducing the chances of misidentification. This hybrid approach leads to a more precise and scalable solution for identifying medicinal plants, benefitting researchers, herbalists and healthcare professionals.

Chapter 2

LITERATURE SURVEY

2.1. Reference Paper

In this field of Ayurveda, the exotic plant species are crucial to preserving a patient's life. Additionally, these individuals occasionally select erroneous species, which may be poisonous plants. This literature survey examines relevant research papers that align with our study and provide advancements in this domain.

The paper [1] presents a technique for automatic identification of medicinal plants using machine learning. Ayurvedic medicine relies heavily on the use of medicinal plants and accurate identification of these plants is crucial. The authors have implemented a random forest algorithm that uses color, texture and geometric features to classify medicinal plant species. The proposed approach aims to automate the identification process, which has traditionally been done manually. The authors review prior work on medicinal plant classification using image processing and machine learning techniques to demonstrate the effectiveness of their approach through experiments on a dataset of plant leaves.

- Ayurvedic medicine relies on medicinal plants and accurate identification is important.
- The authors have developed a machine learning based approach to automate the identification process.
- The technique uses random forest algorithm and extracts color, texture and geometric features from plant leaf images.
- Prior work on medicinal plant classification using image processing and machine learning is reviewed.

The paper [2] represents a novel approach to medicinal plant identification through computer vision and deep learning. It introduces a custom dataset called DeepHerb, with 2515 leaf images of 40 Indian medicinal herbs. The research compares various pre-trained neural networks for feature extraction and classification using ANN and SVM. The Xception-ANN model achieved 97.5% accuracy. The paper also describes "HerbSnap", a mobile app utilizing this model for real-time plant identification.

- The DeepHerb dataset was specifically designed for study, containing 2515 leaf images from 40 medicinal plant species, making it a valuable resource for future research.
- The paper uses Bayesian optimization to fine-tune the SVM hyper-parameters, improving the accuracy of the model.

- Integrated mobile application, HerbSnap can identify plants in under 1 second per image making it highly efficient for practical use.
- Unlike other plant recognition systems, DeepHerb targets medicinal plants emphasizing their importance for biodiversity and healthcare.
- The research leverages transfer learning to overcome the challenge of small datasets, utilizing pre-trained CNNs to achieve high classification an important for real-time applications.
- The HerbSnap app was developed using Flutter, making it accessible across Android and iOS platform.

The paper [3] focuses on the use of digital image processing techniques to identify plant species using features like shape, color and texture of leaves. The study applies methods like Gaussian filtering, K-means clustering and Principal Component Analysis (PCA) for feature extraction. Support Vector Machine (SVM) is used as the primary classifier, achieving an accuracy of 95.17%. The system effectively enhances plant identification using machine learning, but feature research could address challenges with smaller leaves.

- The study uses techniques like Gaussian filtering for noise reduction and K-means clustering for image segmentation.
- PCA (Principal Component Analysis) is used to extract key features like shape, color and texture from plant leaf images.
- SVM is the primary machine learning classifier, achieving 95.17% accuracy.
- The approach simplifies plant species identification using lead features, improving accuracy and efficiency over traditional methods.

The paper [4] describes an approach to automatically identify medicinal plants using a deep learning (DL) model. The research focuses on six plant species: Betel, Curry, Tulsi, Mint, Neem and Indian Beech, with 500 images per plant. Collected from kaggle. After applying pre-processing techniques like resizing and augmentation, the MobileNet DL model was used for classification. The model achieved 98.33% accuracy. It was deployed on Google Cloud and a mobile app was developed to allow real-time plant identification by capturing leaf images. The paper highlights the importance of using mobile-based deep learning systems to make plant identification more accessible to the general public. The mobile app allows users to scan or upload leaf images, sending them to a cloud-based model that returns the plant's name and its medicinal properties.The model has a 97%+ score in metrics like precision, recall, accuracy and the app provides easy access to medicinal plant information.

- Six medicinal plants(Betel, Curry, Tulsi, Mint, Neem, Indian Beech) were used from Kaggle dataset, with 500 images per species.
- MobileNet model was used for classification, achieving 98.33% accuracy.

Chapter 3

PROPOSED SYSTEM

3.1. Problem Statement

The project aims to create a system for identifying medicinal plants using their color, shape and texture features. The tool employs a hybrid implementation of Convolution Neural Networks (CNNs) and Support Vector Machine (SVM) to improve the model's performance, efficiency, and accuracy by extracting features and classifying them which will help in identifying the plants and display uses of that plant. The system's core functionality will include in creating a dataset of medicinal plants, preprocessing the dataset, feature extraction, classification, and implementing a hybrid implementation of CNN and SVM algorithms. Additionally, after identifying the plant the tool will also display the uses of that plant along with accuracy of identification.

3.2. Objectives

The primary objective of this study is to develop a system to identify different medicinal plant species based on their leaf images, by using image processing. To achieve this goal, the collected images need to be pre-processed and data augmentation techniques should be employed to increase the size and diversity of the dataset. The dataset should contain images of various medicinal plants with their corresponding labels.

1. Gathering various images of plant leaves, creating dataset and pre-processing the images to ensure consistency in size, resolution and quality of images for effective training and testing the model.

2. Implementing a Convolutional Neural Network architecture to extract features from images of plants and Support Vector Machine to classify the extracted features into their respective plant species. Optimizing the parameters to achieve high accuracy.

3. Automated plant identification can accelerate research in botany, herbal medicine, and pharmaceuticals. The objective is to provide a robust tool for researchers to explore and study the medicinal properties of plants, potentially leading to the discovery of new drugs and therapies.

4. The goal is to standardize the identification process globally, ensuring that medicinal plants are classified in a uniform manner. This can lead to better data sharing and collaboration in research and conservation efforts.

5. Combining the CNN and SVM models into a hybrid system where CNN will work as features extractor and SVM will serve as classifier and evaluating their performance against traditional CNN and SVM models to demonstrate effective performance.

6. Developing a user-friendly application which will allow users to upload plant images.

3.3. Project Scope

The project focuses on designing and implementing a machine learning based system to identify medicinal plants based on leaf characteristics. It encompasses various phases, including data collection, model development, and system deployment, aimed at creating a reliable tool for plant identification. Implement multiple machine learning techniques such as Convolutional Neural Networks (CNNs), Support Vector Machines (SVM), to classify plants based on leaf features. The project will focus on identifying medicinal plants using visual leaf characteristics only, and it may not extend to other plant parts (e.g., flowers, stems). This project will provide a tool for automated, accurate identification of medicinal plants, aiding in research, healthcare, and conservation efforts.

The system aims to automate the traditionally manual process, reducing reliance on human expertise, minimizing errors, and enhancing the efficiency and speed of plant identification. It is designed to handle large datasets of plant images, making it scalable for both research and field applications. The project will also focus on creating a user-friendly interface that can be used by a wide range of users, from botanists and researchers to farmers and herbalists, enabling accessibility in both urban and rural areas. Moreover, the project seeks to contribute to plant conservation efforts by cataloging endangered species and supporting scientific research in herbal medicine and drug discovery.

Chapter 4

SOFTWARE REQUIREMENT AND SPECIFICATION

4.1. Algorithm

4.1.1 Convolutional Neural Network (CNN) :

A Convolutional Neural Network (CNN) is a specialized type of machine learning algorithm widely employed in computer vision tasks. Specifically designed for processing grid-like data, such as images or videos, CNNs have revolutionized tasks like image classification, feature detection, and image recognition. At the core of a CNN are convolutional layers, where learnable filters scan the input image to detect distinctive features like edges, textures, or shapes. These filters slide across the image, computing dot products at each position and generating activation maps. Non-linear activation functions, like Rectified Linear Units (RLU), introduce non-linearity and allow the network to learn complex relationships. Pooling layers down sample the activation maps, reducing computational complexity and aiding in feature abstraction. Subsequently, fully connected layers process the extracted features, leading to a final output layer with neurons corresponding to the classes in the classification task.

4.1.2 Support Vector Machine (SVM) :

The Support Vector Machine (SVM) is a Supervised Machine Learning algorithm used for both classification and regression. Though we say regression problems as well it’s best suited for classification. The main objective of the SVM algorithm is to find the optimal hyperplane in an N-dimensional space that can separate the data points in different classes in the feature space. The hyperplane tries that the margin between the closest points of different classes should be as maximum as possible. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three. SVM algorithms are very effective as we try to find the maximum separating hyperplane between the different classes available in the target feature.

4.2. System Requirements

4.2.1 Software Requirements

- **Operating System:** Windows 10/11
- **Programming Language:** Python **3.8+**
- **Database:** MongoDB for storing medicinal plants and user data
- **IDE/Code Editor:** Google Colab, Spyder

4.2.2. Hardware Requirements

- **Processor**: Intel Core i5 (8th Generation)
- **RAM**: 8 GB (or higher)
- **GPU** (Recommended): NVIDIA GTX 1650 or higher for efficient ML model processing
- **Network**: Stable internet connection for database and model updates
- **Storage**: Hard Disk (500 GB), SSD (500 GB)

4.3. Libraries Used

1. NumPy

NumPy (Numerical Python) is a powerful open-source library used for numerical computing in Python. It provides support for multi-dimensional arrays and matrices, along with a collection of mathematical functions to perform operations efficiently. NumPy arrays, known as ndarrays, are faster and more memory-efficient than Python lists, making them ideal for handling large datasets. The library includes functions for array manipulation, linear algebra, statistical computations, and Fourier transforms. NumPy also supports broadcasting, which allows operations on arrays of different shapes without explicit loops. Its ability to handle complex mathematical computations with ease makes it a fundamental tool in artificial intelligence, data analysis, and engineering applications.

2. OpenCV

OpenCV (Open-Source Computer Vision Library) is an open-source computer vision and image processing library widely used for real-time applications. It provides various tools and functions for image manipulation, object detection, feature extraction, and video processing. OpenCV supports multiple programming languages, including Python, C++, and Java, making it highly flexible for developers. Common tasks performed using OpenCV include image resizing, filtering, edge detection, and object recognition. OpenCV is widely used in fields such as robotics, medical imaging, and augmented reality. With built-in support for cameras and video streams, it is ideal for real-time computer vision applications. The library is continuously updated with new features, making it a powerful tool for AI and image- processing projects.

3. TensorFlow

TensorFlow is an open-source machine learning framework developed by Google, widely used for building and deploying deep learning models. It provides a flexible ecosystem for numerical computation, primarily using dataflow graphs, where computations are represented as nodes and edges. TensorFlow supports both CPU and GPU acceleration, making it efficient for large-scale AI tasks. It includes Keras, a high-level API that simplifies deep learning model development. TensorFlow is commonly used for image recognition, natural language processing (NLP), and predictive analytics. It allows easy deployment on cloud platforms and edge devices, making it a popular choice for AI researchers and developers.

4. Scikit-learn

Scikit-learn is a popular open-source machine learning library in Python, built on NumPy, SciPy, and Matplotlib. It provides efficient tools for data preprocessing, feature selection, model training, and evaluation. The library includes a wide range of supervised and unsupervised learning algorithms such as Support Vector Machines, Decision Trees, Random Forest, K-Nearest Neighbors (KNN), and Clustering techniques. It also supports tasks like dimensionality reduction, model selection, and hyperparameter tuning using Grid Search CV and Randomized Search CV. Scikit-learn is widely used in real-world applications, including predictive analytics, image classification, and natural language processing.

5. Pandas

Pandas is a powerful open-source Python library used for data manipulation and analysis. It provides flexible and efficient data structures, primarily Series (one-dimensional) and Data Frame (two-dimensional), making it easy to handle structured data. Pandas is widely used in machine learning, data preprocessing, and exploratory data analysis (EDA). It allows users to load data from various sources like CSV, Excel, SQL, and JSON, and perform operations such as filtering, grouping, merging, and reshaping data. With built-in functions for handling missing values, statistical analysis, and time series manipulation, Pandas simplifies complex data tasks. Its user-friendly syntax enables quick data exploration, making it popular among data scientists and analysts.

6. Matplotlib

Create static, animated and interactive plots. It provides a variety of chart types, including line plots, bar charts, histograms, scatter plots and heat maps. The library is built on NumPy arrays, making it efficient for handling large datasets. The primary module pyplot offers a simple MATLAB-like interface for easy plotting. Users can customize plots with labels, titles, legends and different color schemes. Matplotlib supports multiple backends for rendering in various environments such as Jupyter Notebooks, GUIs and web applications.

4.4. Plant Dataset

Dataset Purpose : In the medicinal plant identification project, the dataset plays a crucial role in training and validating the hybrid CNN-SVM model for accurate plant classification. It consists of labelled images of medicinal plants, serving as the foundation for feature extraction and pattern recognition. The dataset enables the model to learn distinct characteristics such as leaf shape, color, texture, and vein structure, which are essential for accurate identification. Additionally, a well-structured dataset helps in evaluating model accuracy, making it a key component in developing a reliable, scalable, and efficient plant identification system for real- world applications..

Data Collection : Data collection involves gathering a diverse and high-quality dataset of medicinal plant images for training and testing the hybrid CNN-SVM model. This dataset can be sourced from public repositories such as Kaggle. The images should cover various species, capturing different angles, lighting conditions, backgrounds, and stages of plant growth to improve model generalization. Each image should be labelled accurately with the plant name and additional metadata such as habitat, medicinal properties, and scientific classification. Preprocessing steps like resizing, noise reduction, and augmentation (flipping, rotation, contrast adjustments) can further enhance data quality.

Annotation : Annotation refers to the process of labelling images in the dataset with relevant information to train the hybrid model(CNN & SVM). This involves manually or automatically marking key attributes such as plant species name, medicinal properties and specific features like shape, color or texture. Proper annotation ensures that the model learns meaningful patterns and improves classification accuracy.

Evaluation Metrics : To assess the performance of the hybrid CNN-SVM model in classifying plant species accurately. Key metrics include accuracy, which measures the overall correctness of predictions and precision which indicates the proportion of correctly identified plant species among all predicted instances.

Cost Estimation

The cost estimation is based on the Constructive Cost Model (COCOMO) and effort distribution:

Effort and Timeline:

- Total Effort: 31.64 Person-Months

- Project Duration: 2 Months
- Team Size: 4 Members
- Effort Distribution:
    - Requirement Analysis & Design: 15% (~4.75 PM)
    - Coding and Integration: 50% (~15.82 PM)
    - Testing and Deployment: 25% (~7.91 PM)
    - Documentation & Final Review: 10% (~3.16 PM)

4.5. Effort Estimation

The effort estimation follows the COCOMO model:

- Effort (Person-Months):
    - a (Semi-Detached): 3.0
    - b (Semi-Detached): 1.12
    - Development Time (Months):
    - c (Semi-Detached): 2.5
    - d (Semi-Detached): 0.35
- Total Effort: 31.64 Person-Months
- Individual Contribution: ~7.91 Person-Months per member

4.6. SDLC (Software Development Life Cycle)

1. Requirement Gathering and Analysis : In this initial phase, both functional and non-functional requirements are collected from stakeholders, including botanists, researchers, and potential users. The focus is on building a system that can accurately identify medicinal plant species based on leaf images. Requirements cover image input handling, classification accuracy, model interpretability, and user interface design for ease of use.

2. System Design: The system architecture is designed to include key components such as image preprocessing, CNN feature extraction, SVM classification, and a web-based front end. Design artifacts like data flow diagrams, system architecture diagrams, and database schemas (for storing plant data and user logs) are developed. Special attention is given to modularizing the pipeline for image input, model inference, and result display.

3.Implementation: This phase involves coding the core system. A Convolutional Neural Network (CNN) is implemented or fine-tuned to extract features from leaf images, and these features are passed to a Support Vector Machine (SVM) classifier for final prediction. The system is developed using frameworks like TensorFlow or PyTorch for the CNN and scikit-learn for the SVM. A Flask-based web application developed for users to upload images and view results.

4. Testing and Validation: The system undergoes various tests to ensure performance and reliability. Unit tests verify individual components, while integration tests ensure the CNN-SVM pipeline works as expected. Performance testing evaluates model accuracy using datasets, and usability testing gathers feedback from domain experts. Validation is performed using cross-validation and real-world test cases.

5. Maintenance and Updates: The system requires periodic updates including model retraining with new plant image data to improve accuracy. Bug fixes, performance improvements, and user-requested features are addressed in this phase. Automated monitoring tools track system health, and user feedback is incorporated for ongoing refinement.

4.8. System Features

1. **Image-based Plant Identification :** Users can upload images of plant leaves, which are processed and classified to identify the plant species.

2. **Hybrid Classification Model (CNN+SVM) :** A Convolutional Neural Network (CNN) is used to extract visual features from leaf images. These features are passed to a Support Vector Machine (SVM) for final classification, combining deep learning's feature extraction with SVM's robust classification.

3. **High Accuracy and Robustness :** The hybrid model improves overall classification performance, especially when working with limited training data or subtle differences between species.

4. **User Friendly Web Interface :** A simple web-based interface built using Flask allows users to upload images and instantly view the identification results, plant details, and possible medicinal uses.

5. **Medicinal Information Integration :** After identifying the plant, the system displays relevant medicinal uses, active compounds, and traditional healing information (based on integrated databases or stored information).

6. **Secure and Lightweight Backend :** Built with security and efficiency in mind, ensuring that image uploads are safely processed and storage is optimized.

Chapter 5

PROJECT PLAN & DESIGN

5.1. Timeline Flowchart :

A timeline flowchart is a visual tool that shows the sequence and timing of tasks or phases in a project. It combines flowchart logic with a chronological timeline, starting from project initiation through phases like planning, design, development, testing, deployment, and maintenance. Tasks are displayed in boxes along the timeline, connected by arrows to indicate dependencies. Time markers and status indicators (e.g., “in progress,” “completed”) help track progress. This flowchart helps stakeholders understand the project flow, manage deadlines, and monitor overall progress efficiently.

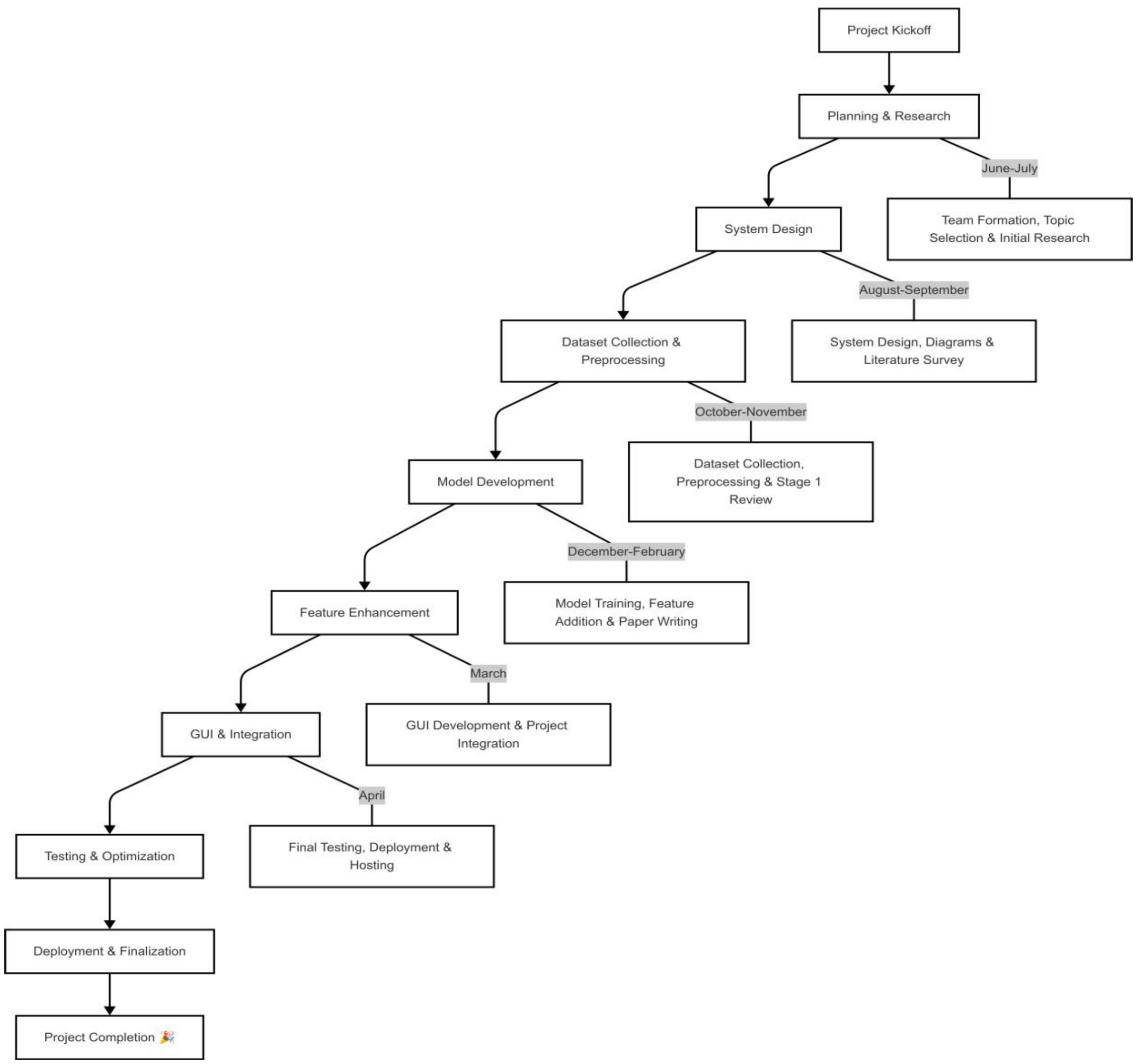


Fig 5.1 : Timeline Chart

Table 5.1: Project Plan

Project Plan		
Month	Week	Task
June	Week 3	Team Formation
	Week 4	Project Topic Selection
July	Week 1	Submission of Topic
	Week 3	Project Synopsis
August	Week 1	Project Related Research
	Week 3	Review 1
September	Week 1	Literature Survey
	Week 3	Review 2
October	Week 1	System Design Diagrams
	Week 2	Review 3
	Week 4	Review 4
November	Week 2	Project Stage 1 Review
	Week 3	Research Paper Initiated
	Week 4	Dataset Collected
December	Week 4	Dataset Preprocessing
January	Week 2	Review 1
	Week 4	Developing and Training Models
February	Week 1 2	Developing and Training Models
	Week 3	Adding more features to project
	Week 4	Review 2
	Week 4	Research Paper Publication
March	Week 2	GUI Development
	Week 3	Integration
	Week 4 5	Testing and Tuning
April	Week 2	Deployment and Hosting
	Week 3	Finalizing Details

5.2 Data Flow Diagram

5.2.1. DFD Level 0 Diagram: -

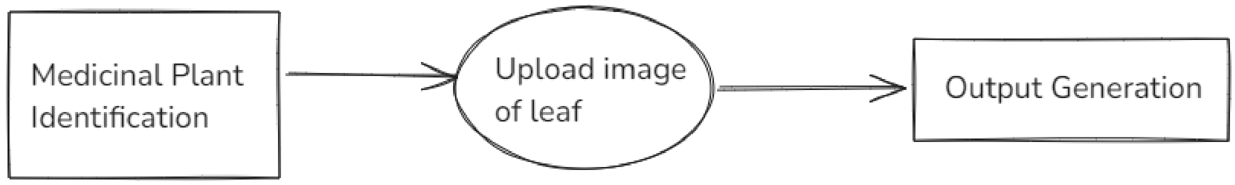


Fig 5.2.1 : DFD Level 0

Explanation :

A Level 0 DFD is also known as context diagram. It's designed to be an abstraction view, showing the system as a single process which its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by arrows.

- The system is named Medicinal Plant Identification.
- The user uploads a leaf image to the system.
- The main process is represented as "Upload image of leaf", indicating the core function.
- The input to the process comes from external entity(user/system environment).
- The system processes the image internally to identify the medicinal plant.
- The output of this process is sent to "Output Generation", which gives the identification result.
- This diagram shows only one process, without breaking it into sub-processes.

5.2.2 DFD Level 1 Diagram: -

DFD Level 1 provides a more detailed view of the system by breaking down the major processes identified in the level 0 DFD into sub-processes. Each sub-process is depicted as a separate process on the level 1 DFD. The data flows and data stores associated with each sub-process are also shown. In 1-level DFD, the context diagram is decomposed into multiple bubbles/processes.

- The following figure shows Level 1 of DFD, which expands the main process from level 0 into more detailed sub-processes.
- The system starts with Medicinal Plant Identification as the external entity providing the leaf image.
- The first process is "Upload leaf image", where the user inputs the image into the system.

- The image is sent to the next process, which is “Preprocessing of data”, to clean and prepare the image for analysis.
- The pre-processed data is then passed to the “Training Dataset”, which stores and organizes data for training.
- The training dataset is used to train a Machine Learning Model.
- The trained model is used to perform prediction, and the output is sent to “Output Generation”.
- Finally, the output generation provides the result back to the Medicinal Plant Identification system(user).

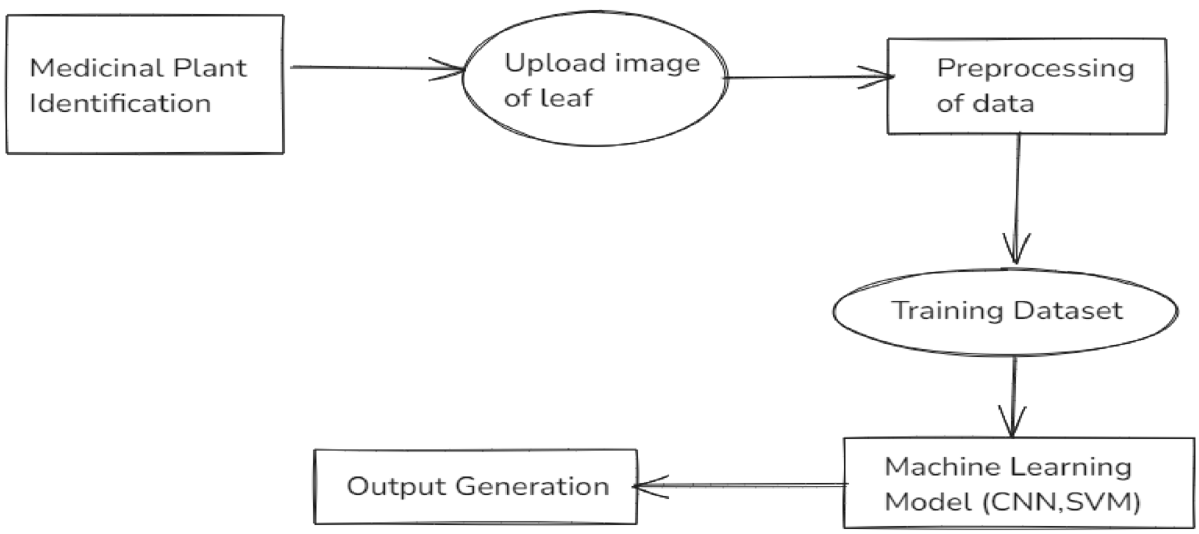


Fig 5.2.2 : DFD Level 1

5.2.3 DFD Level 2 Diagram: -

DFD Level 2 provides an even more detailed view of the system by breaking down the sub-processes identified in the level 1 DFD into further sub-processes. Each sub-process is depicted as a separate process on the level 2 DFD. The data flows and data stores associated with each sub-process are also shown.

- The level 2 of DFD provides an even more detailed breakdown of the processes shown in level 1.

- The process begins with the user from Medicinal Plant Identification uploading a leaf image.
- The uploaded image goes through preprocessing of data to enhance and clean it for model input.
- After preprocessing, the next step is to Validate the Model to ensure proper setup and readiness.
- The validated data is then used to Train the Model using CNN and SVM, which forms the learning phase.
- Once trained, the system tests the model using CNN and SVM for performance and accuracy.
- This plant data is sent to a Server for further handling and response generation.
- Finally, the output generation module represents the identified plant details to user.

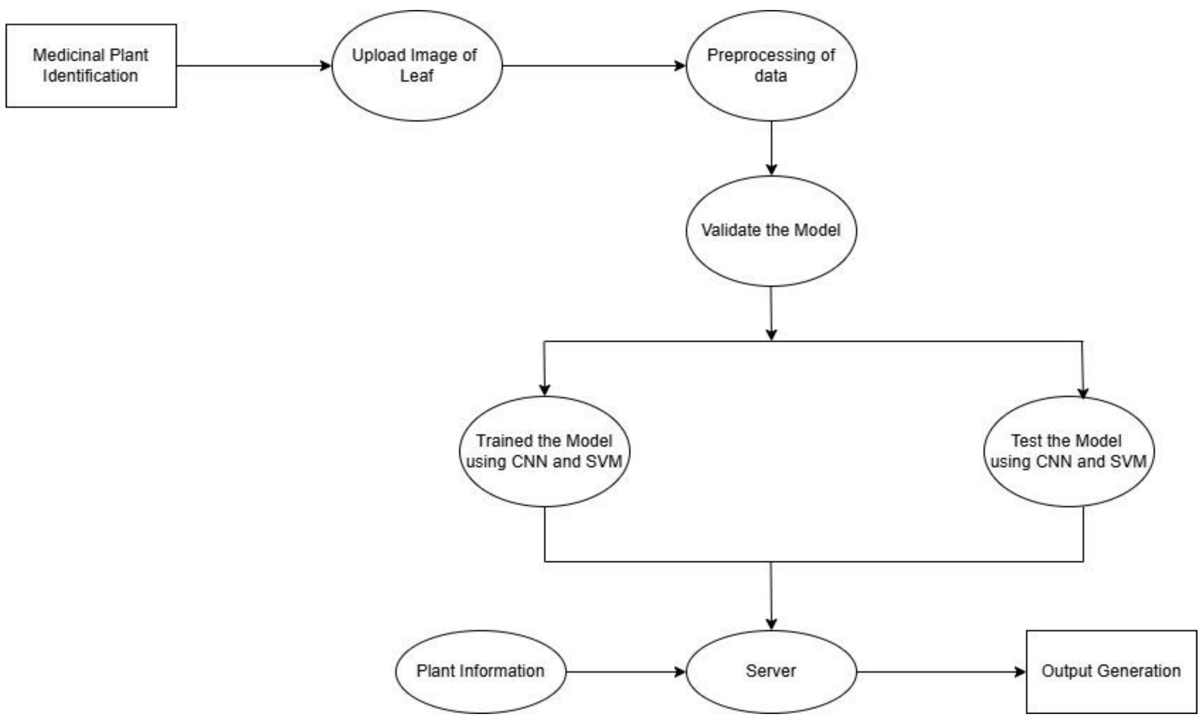


Fig 5.2.1.3 : DFD Level 2

5.3 UML Diagram

5.3.1 Class Diagram

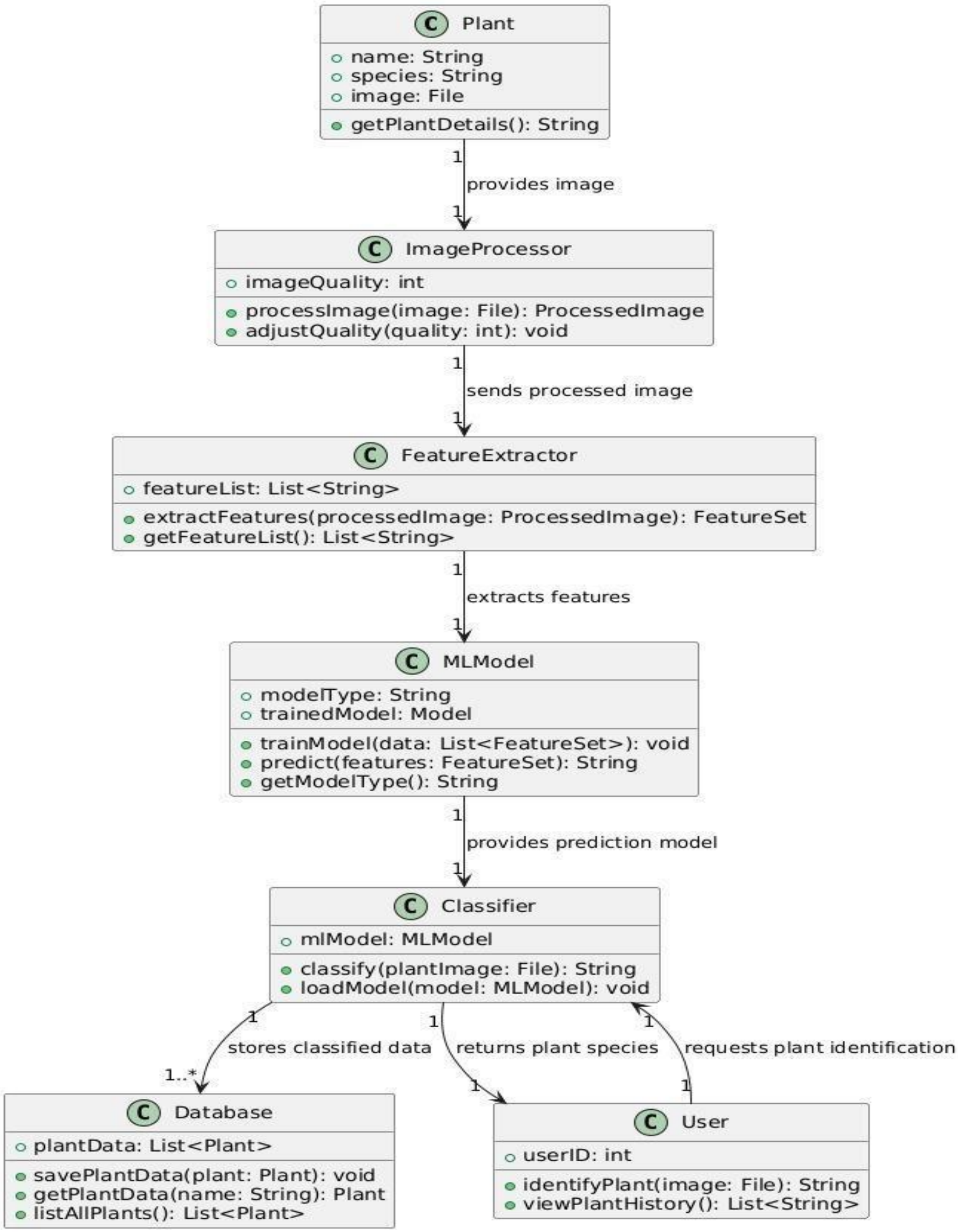


Fig 5.3.1 : Class Diagram

Explanation :

Class diagram are a type of UML (Unified Modelling Language) diagram used in software engineering to visually represent the structure and relationships of classes within a system i.e. used to construct and visualize object-oriented systems.

1. **Plant Class**

   Attributes :
   - name: String – Name of the medicinal plant.
   - image: Image – Image of the plant/leaf.
   - plantDetails : String – Unique identifier for plant details.

   Methods:
   - getImage() – Returns the plant image
   - getPlantDetailsID() – Retrieves the plant detail ID.
   - getDetails() – Fetches additional details about the plant.

2. **ImageProcessor Class**

   Attributes:
   - imageQuality : String – Defines the quality level of the uploaded image.
   - processedImage : Image – Holds the processed image data.

   Methods :
   - processImage() – Performs preprocessing steps like resizing, normalization, etc.
   - getProcessedImage() – returns the cleaned and processed image.

3. **FeatureExtractor Class**

Attributes :

- featuresList : List<String> - List of features such as shape, color and texture.
- processedImage : Image – Image used for feature extraction.

Methods :

- extractFeatures() – Extracts relevant features from the processed image.
- getExtractedFeatures() – Returns the list of extracted features.

4. MLModel Class

Attributes :

- modelType : String – Type of model used(e.g., CNN or SVM).
- trainedModel : Object – Trained machine learning model

Methods :

- trainModel(featuresList) – Trains the model using extracted features.
- getModelType() – Returns the type of ML model.

5. Classifier Class

Attributes :

- classifiedData : List – Contains classification results after prediction.

Methods :

- classify() – Uses the trained model to classify the plant from uploaded image.

6. Database Class

Attributes :

- plantData : List<Plant> - Stores details of known medicinal plants.

Methods :

- savaPlantData() – Stores new plant information into the database.

- setPlantData() – Retrieves plant data for display or training.

7. **User Class**

   Attributes :

   - userID : Int – Unique user identifier.
   - name : String – Name of the user

   Methods :

   - viewPlantDetails() – Allows users to view identified plant details.
   - requestPlantIdentification() – User upload an image for identification.

5.3.2 Use Case Diagram

1. **Start –** The entry point where the user begins the plant identification process.
2. **Select Image –** User selects a leaf image for identification.
3. **Upload Image –** The selected image is uploaded to the system.
4. **Pre Processing –** The system preprocesses the uploaded image (e.g., resizing, noise removal).
5. **Feature Extraction –** System extracts relevant features from the image required for classification.
6. **Model –** Represents the trained machine learning model (CNN/SVM) used for prediction.
7. **Train Model –** The system trains the model using a dataset.
8. **Predict Output –** The system uses trained model to predict medicinal plant based on uploaded image.
9. **Get Result –** User receives result – the name or information about the identified plant.

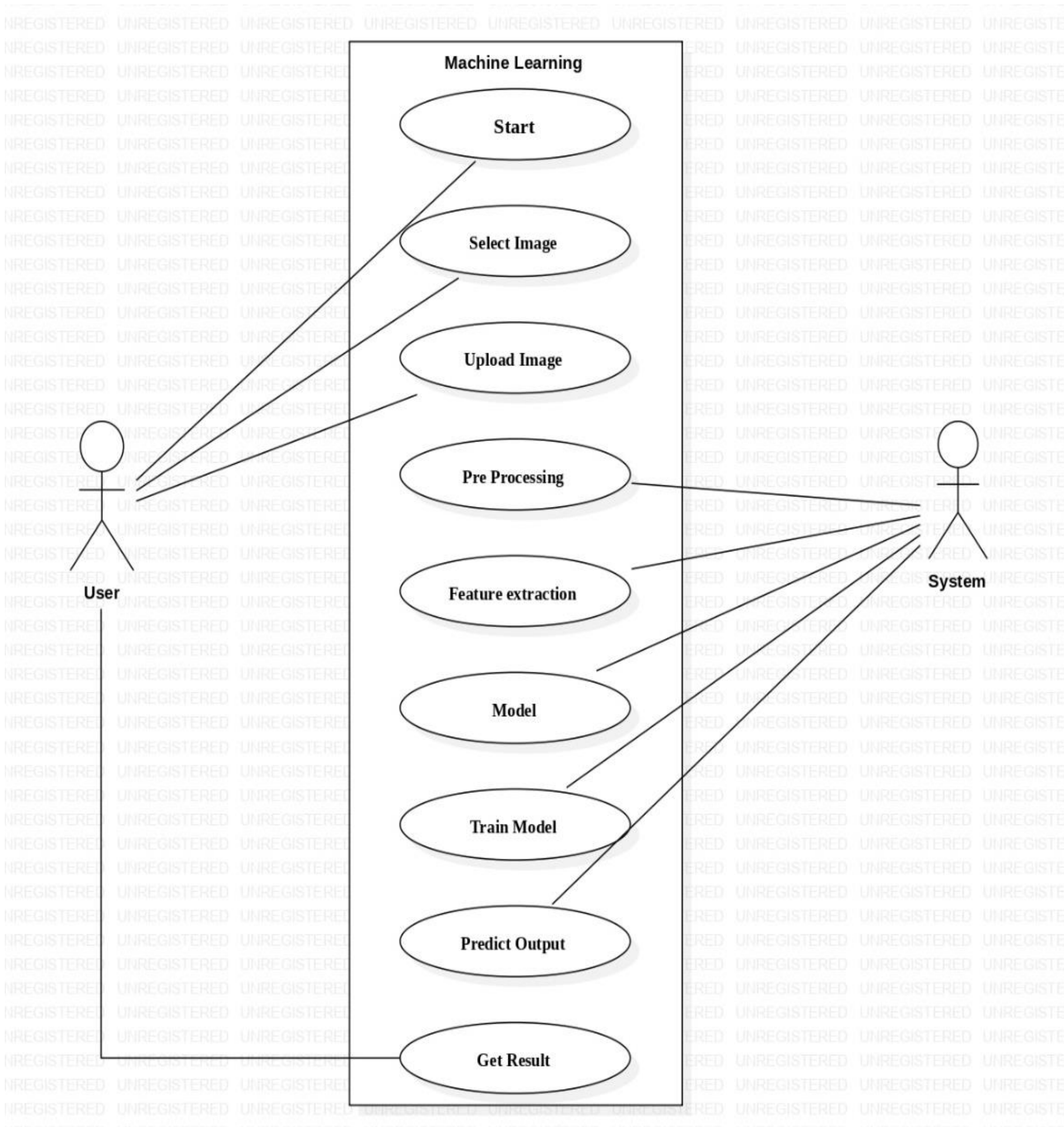


Fig 5.3.2 : Use Case Diagram

5.3.3 Activity Diagram

Activity diagrams show the steps involved in how a system works, helping us understand the flow of control. They display the order in which activities happen and whether they occur one after the other (sequential) or at the same time (concurrent). These diagrams help explain what triggers certain actions or events in a system.

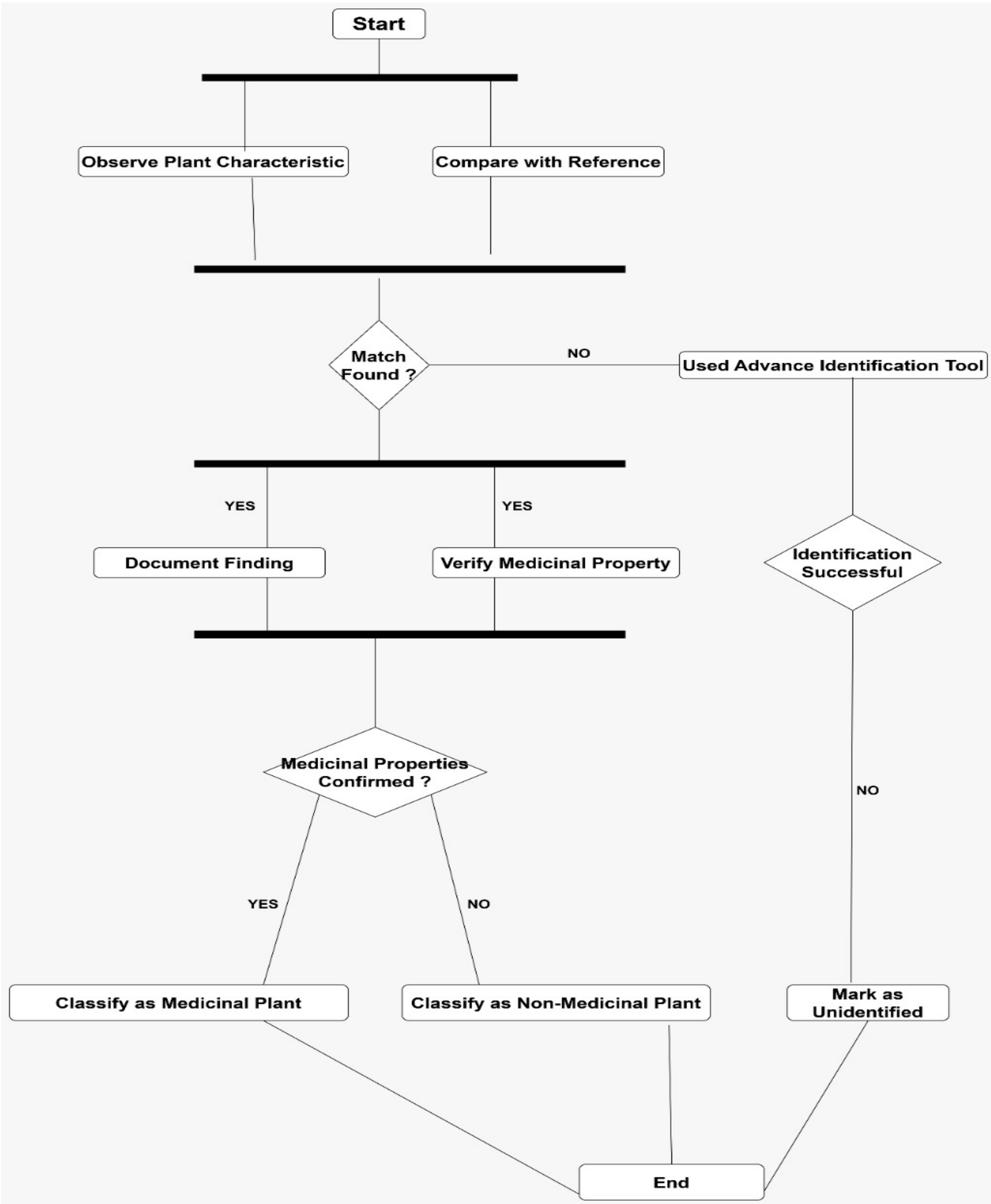


Fig 5.3.3 : Activity Diagram

1. **User → System: Initiate plant identification**
   The process begins when the user initiates identification (e.g., by uploading an image).

2. **System → Dataset: Preprocess dataset**
   The system sends the uploaded image to be pre-processed (like resizing, enhancing, etc.).

3. **System → FeatureExtractor: Extract features**
   After preprocessing, features like color, texture, or shape are extracted from the image.

4. **FeatureExtractor → System: Return extracted features.**
   The extracted features are sent back to the system.

5. **System → Classifier: Classify features using CNN and SVM**
   The system uses machine learning classifiers (CNN, SVM) to identify the plant.

6. **Classifier → System: Return classification result**
   The predicted plant ID or label is returned to the system.

7. **System → PlantDatabase: Fetch plant information**
   Based on the predicted ID, the system queries the database for detailed plant info.

8. **PlantDatabase → System: Return plant information**
   The system receives detailed data about the plant (uses, description, etc.).

9. **System → User: Display identification result and uses**
   Final output is shown to the user including plant name and medicinal uses.

5.3.4 Sequence Diagram

Sequence diagrams are a type of UML (Unified Modeling Language) diagram that visually represent the interactions between objects or components in a system over time. They focus on the order and timing of messages or events exchanged between different system elements. The diagram captures how objects communicate with each other through a series of messages, providing a clear view of the sequence of operations or processes.

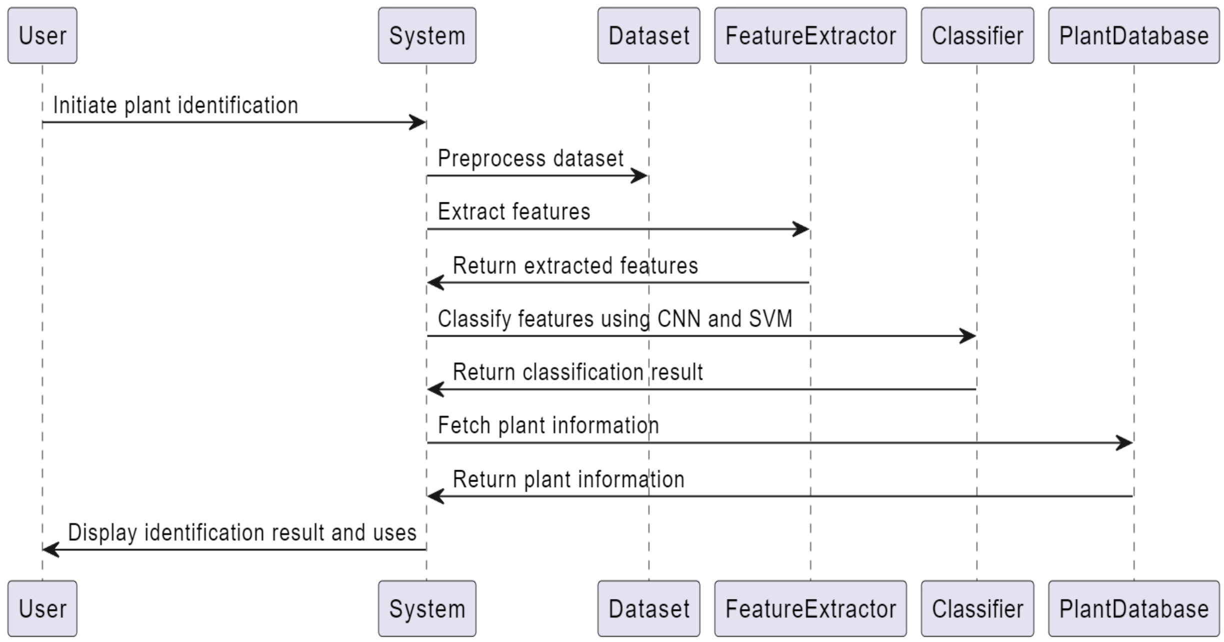


Fig 5.3.4 : Sequence Diagram

1. **User and System Interaction**
    - The user initiates the plant identification process through the system interface.
    - The system receives the request and begins processing with the available dataset and components.

2. **Preprocessing Stage**
    - The system preprocesses the dataset related to plant images and data to ensure consistent formatting.
    - This may involve resizing images, normalization, or cleaning unnecessary noise.

3. **Feature Extraction**
    - The FeatureExtractor module is invoked to extract relevant features from the dataset (e.g., leaf shape, texture, color patterns).

- These extracted features are returned to the system for further processing.

4. **CNN-SVM Classification Analysis**
   - The system uses the Classifier module, which applies a hybrid CNN (Convolutional Neural Network) and SVM (Support Vector Machine) approach.
   - This module classifies the plant based on the extracted features and returns the classification result to the system.

5. **Database Lookup and Prediction**
   - Once classified, the system fetches plant information (name, uses, medicinal properties, etc.) from the PlantDatabase**.**
   - The plant information is returned to the system to prepare the final output.

6. **Result Delivery**
   - The system displays the identification result and the plant's uses to the user.
   - This concludes the plant identification process.

5.4 Entity Relationship Diagram (ER)

Entity Relationship Diagram (ERD) is a graphical representation that shows how entities, such as people, objects, or concepts, relate to each other within a system. ERDs are commonly used in database design to visualize the relationships between entities and their attributes.

They help in understanding the logical structure of databases by showing entities connections and relationships using symbols like rectangles, diamonds, ovals, and connecting lines.

1. **User and Identification Interaction**

   a) The **User** entity contains:
   - User_id (Primary Key)

- User Name
- Email

b) A user can perform multiple identifications:

- This is represented by the **"Performs"** relationship (many-to-many: M to M).

2. Identification Process

a) The **Identification** entity records each attempt to identify a plant:

- Identification_id (Primary Key)
- User_id (Foreign Key)
- Plant_id (Foreign Key)
- Identification Date

b) It has a relationship called **"Identifies"** with the Plant entity, showing that an identification is linked to a specific plant.

3. Plant Information

a) The **Plant** entity includes:

- Plant_id (Primary Key)
- Common Name
- Scientific Name

b) A plant is associated with:

- Images through the **"Has"** relationship (Many-to-Many)
- Medicinal Uses through another **"Has"** relationship (Many-to-Many)

4. Medicinal Use Mapping

a) The **Medicinal Use** entity contains :

- Use_id (Primary Key)
- Description
- Type
- Characteristics

b) Each plant can have multiple medicinal uses, captured via the **"Has"** relationship.

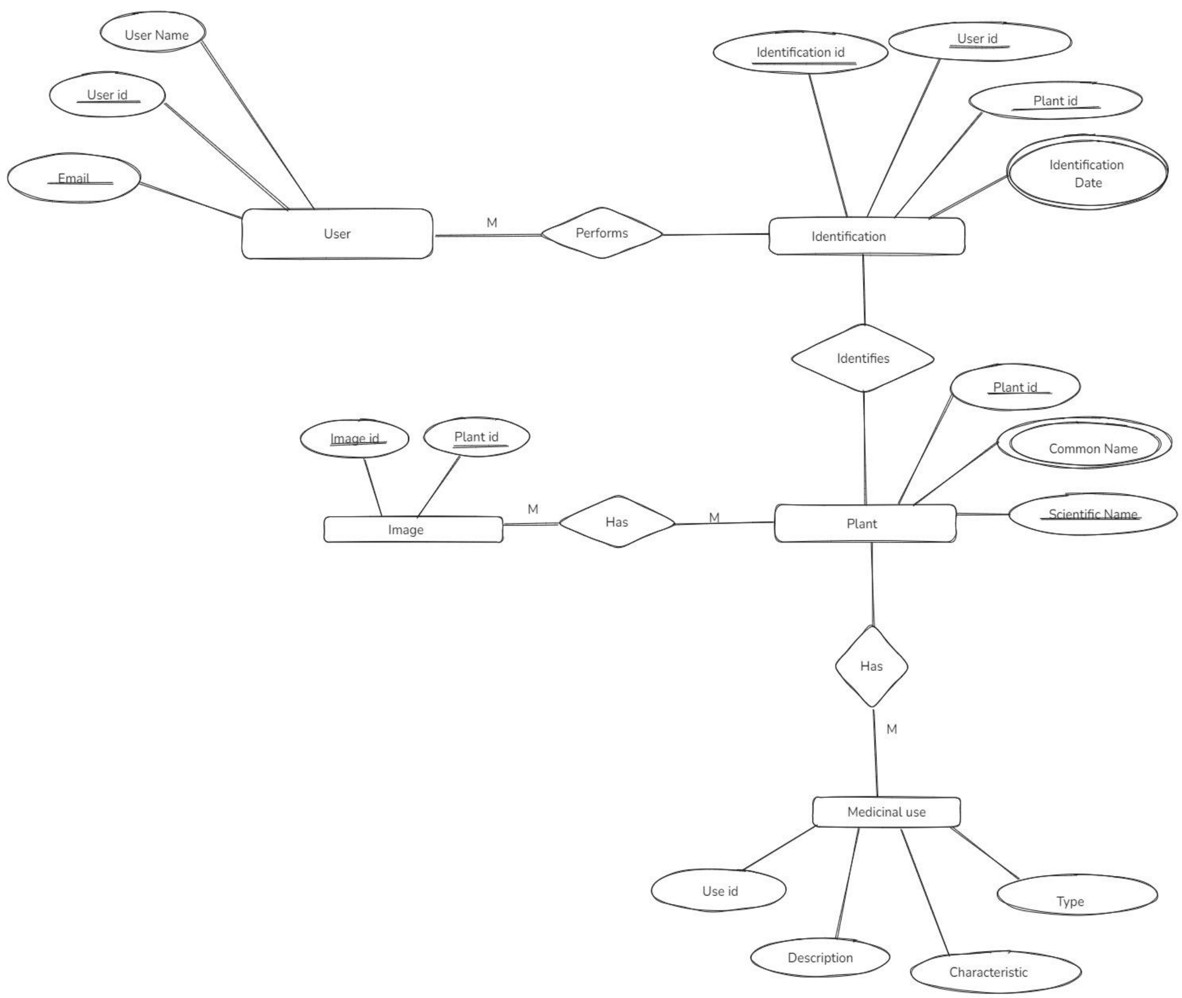


Fig 5.4 : ER Diagram

Chapter 6

SYSTEM ARCHITECTURE

6.1. System Architecture

- The identification of medicinal plants begins when the user uploads an image of a medicinal leaf through the Graphical User Interface (GUI) of the application. This could be from a web or mobile app.
- The input image is received by the system and is then passed on to the Preprocessing module. This module is responsible for preparing the image for further analysis.
- Preprocessing involves operations like resizing the image, noise removal, enhancing contrast, and converting the image to a suitable format (e.g., grayscale or normalized format). This step ensures uniformity and improves the accuracy of feature extraction and classification.
- Once preprocessing is complete, the image is sent to the Classification module, which is the core part of the system. This module is divided into two main phases: Training phase and Testing phase.
- In the Training phase, a large dataset of labelled medicinal leaf images is used. Both CNN (Convolutional Neural Network) and SVM (Support Vector Machine) algorithms are trained on this dataset. CNN automatically learns visual features like texture, shape, and patterns, while SVM is used to classify the images based on these features.
- After training, the model is ready for the Testing phase Here, the pre-processed input image is analyzed using the trained CNN and SVM models. The system extracts feature from the image and classifies it to predict the plant species.
- The Prediction module takes the classification result and generates the name or ID of the predicted plant.
- The Model Deployment interacts with a Server to query the relevant plant information. This server is connected to a Plant Information Database that contains medicinal properties, scientific names, uses, etc., for each plant.
- The Server fetches the plant details corresponding to the predicted plant ID and sends the information back to the model deployment module.
- The system then prepares this data for user presentation and sends it to the Output module connected to the GUI.
- Finally, the user sees the output on the GUI. This output includes the identified plant

name, its properties, and relevant medicinal information.

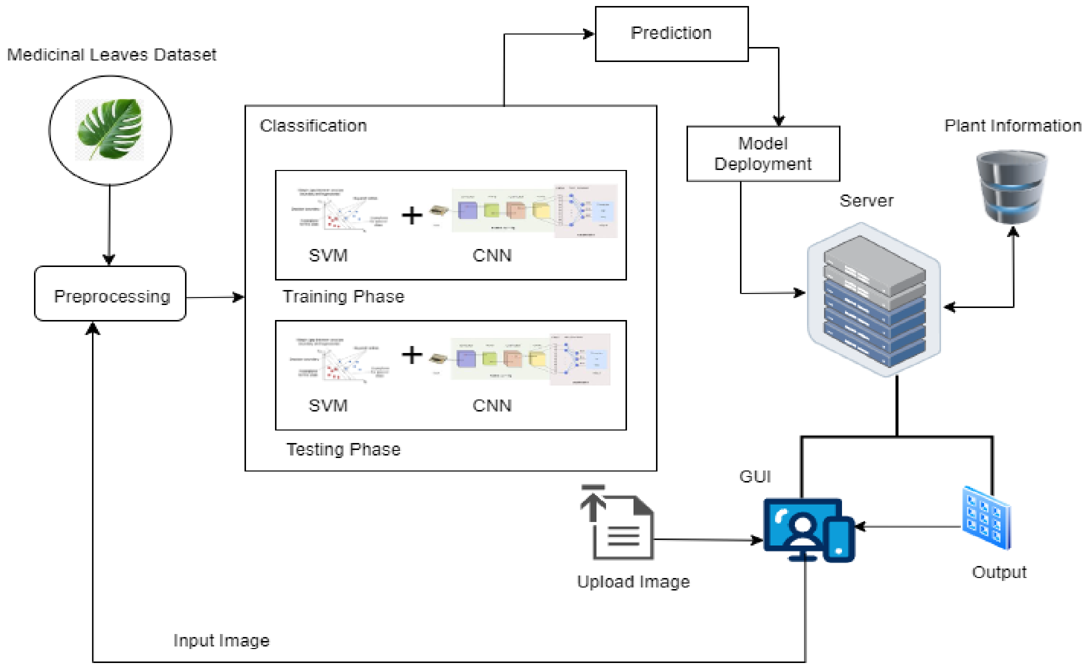


Figure 6.1 System Architecture

6.2. Methodology

- The project planning involves setting the direction of the project by identifying goals, choosing appropriate technologies, defining scope and preparing a structured timeline for execution. It ensures the project stays on track and meets its objectives efficiently.
- Data Collection involves gathering a high-quality dataset of medicinal plant leaves, which is essential for training and testing the model. The data should be diverse, well-labelled and suitable for image-based classification.
- Deep learning and machine learning models such as CNN and SVM are designed, trained, and tested. The model is built to classify medicinal plants based on their leaf images by learning key features during training.
- A user-friendly interface is developed for users to interact with the system. This could be a web or mobile application that allows users to upload plant images, trigger the model, and receive classification results in real-time.

- The developed system is rigorously tested to ensure it functions correctly and provides accurate predictions. Performance metrics like accuracy, precision, and recall are evaluated to validate the model's effectiveness.

- All stages of the project are thoroughly documented, including system design, model architecture, code implementation, and testing results. A final report and presentation are prepared to communicate the outcomes and findings of the project.

6.3 Mathematical Model

$$\boxed{y = \text{sign}(W^T \, CNN(x) + b)}$$

Step 1 : x- Input Leaf Image

- This is a photo of a medicinal plant leaf.
- For example, a leaf from Neem, Tulsi or Aloe Vera

Step 2 : CNN(x) — Feature Extraction

- The CNN takes the leaf image and analyzes its shape, texture, veins, edges, and color patterns.
- It converts this visual info into a numerical feature vector f.
- This vector captures the unique fingerprint of the leaf.

Step 3 : W^T.f +b – SVM classification

- The SVM takes the feature vector and applies a learned linear function to it.
- W and b are learned during training based on your labelled dataset (e.g., you tell the model "this is a Neem leaf", "this is a Tulsi leaf", etc.)
- This gives a score that tells how similar the features are to each known class.

Step 4 : sign(·) — Output Prediction

- The sign function determines which class (plant species) the leaf belongs to.

- In multi-class SVM, a slightly extended version is and the plant species with the highest score is chosen.

6.4 Algorithm

6.4.1 Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) is a class of deep learning algorithms highly effective in tasks involving image and visual data analuysis. CNNs are designed to automatically and adaptively learn spatial hierarchies of features from input images through the use of multiple building blocks such as convolutional layers, pooling layers and fully connected layers.

Unlike traditional machine learning algorithms that rely on handcrafted features, CNNs extract features directly from raw image pixels. This capability makes them particularly well-suited for complex image-related tasks such as image classification, object detection, face recognition, and medicinal plant identification, which is the focus of our project.

6.4.1.1 Convolutional Layer

At the heart of CNNs are convolutional layers, where a set of learnable filters (also called kernels) slide over the input image. Each filter performs a mathematical operation called convolution, computing dot products between the filter and portions of the input image. These operations produce activation maps (feature maps) that highlight the presence of specific features such as edges, corners, textures, or patterns.

The filters are trained during the learning process and become specialized in detecting different aspects of the image. For example, early layers may capture low-level features like lines and curves, while deeper layers can capture more abstract features like leaf patterns, textures, or overall shapes, which are especially important in distinguishing between different medicinal plants.

6.4.1.2 Activation Function

After convolution, the output is passed through a non-linear activation function, typically the Rectified Linear Unit (ReLU). The ReLU function introduces non-linearity into the model, enabling it to learn complex patterns and relationships in the data. Without such non-linear functions, the CNN would behave like a linear model, limiting its learning capacity.

6.4.1.3 Pooling Layer

To further process the data, CNNs use pooling layers (such as max pooling or average pooling) to reduce the spatial dimensions of the feature maps. Pooling helps in down-sampling the data, which decreases computational load and makes the network more robust to small translations or distortions in the image. This abstraction allows the network to retain the most important features while discarding unnecessary details.

6.4.1.4 Fully Connected Layers

After several convolution and pooling layers, the high-level feature maps are flattened into a vector and passed to one or more fully connected (dense) layers. These layers function like traditional neural networks and help in decision-making by combining the extracted features to predict the correct class label. For our application, the final fully connected layer contains neurons equal to the number of medicinal plant categories, and uses a SoftMax activation function to output the classification probabilities.

6.4.1.5 Training Process

During training, the CNN uses a large dataset of labelled images to learn the optimal filter weights through backpropagation and gradient descent. The model's performance is evaluated using a loss function (such as categorical cross-entropy), and its accuracy improves over multiple training iterations (epochs).

6.4.2 Support Vector Machine (SVM)

The Support Vector Machine (SVM) is a powerful and versatile supervised machine learning algorithm, primarily used for classification tasks, though it can also handle regression and outlier detection problems. In the context of our project, SVM is utilized as a classification tool to categorize medicinal plants based on extracted features from images.

The fundamental objective of SVM is to find the optimal hyperplane that best separates data points belonging to different classes in a multi-dimensional feature space. This hyperplane acts as a decision boundary: any new data point falling on one side of the hyperplane is classified into one category, while points on the other side belong to a different category.

The optimal hyperplane is the one that maximizes the margin the distance between the hyperplane and the nearest data points from each class. These closest data points are known as support vectors, and they are critical to defining the position and orientation of the hyperplane. A wider margin is desirable as it reduces the model’s generalization error and increases robustness.

In our implementation, after image pre-processing and feature extraction (possibly using CNN or other methods), the resulting feature vectors are fed into the SVM classifier. The SVM then learns to differentiate between various medicinal plant species based on these features. Once trained, it can classify new plant images with high accuracy.

Chapter 7

RESULT & ANALYSIS

7.1. Implementation

7.1.1. Graphical User Interface (GUI)

Home Page

This is the homepage of our project PhytoAI. At the top, there is a navigation bar with options like **Home**, **About**, and a prominent button labelled **"Identify the Plant"** this serves as the main call-to-action, inviting users to try the AI identification feature. The central banner warmly welcomes users with the message **"Welcome to PhytoAI"**, followed by a short description that explains the core idea of the project using revolutionary AI to unlock the secrets of medicinal plants. There is also a **"Read More"** button, which likely leads to additional information about the project, its functionality, or the science behind it. The homepage also includes navigation dots and arrows, indicating the presence of a slideshow or carousel feature for more content. Overall, the interface is user-friendly, visually appealing, and effectively communicates the project's purpose.

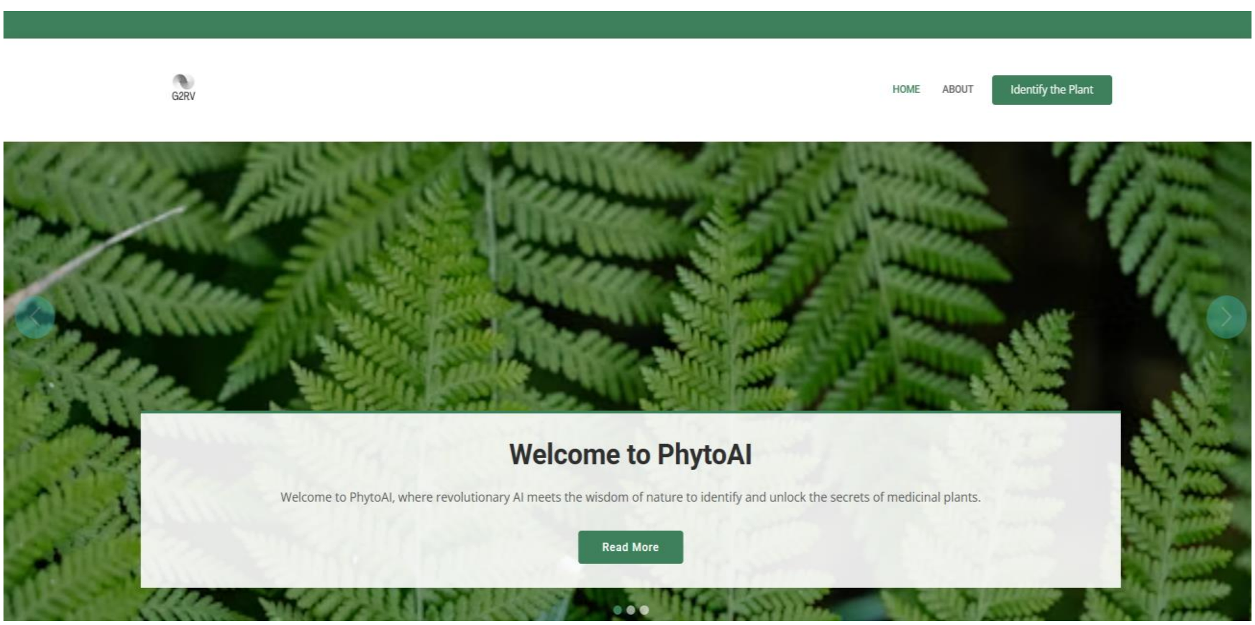


Fig 7.1.1: Home Page of the System

About Us Page

This is the About Us section of the PhytoAI homepage, introducing the purpose of the project. PhytoAI is an AI-powered platform designed to identify medicinal plants from images with accuracy and ease. Users can upload a plant image to learn about its species, medicinal uses, and traditional benefits. The platform is user-friendly and ideal for researchers, students, and nature enthusiasts. It encourages users to explore the healing power of plants with just a single click.

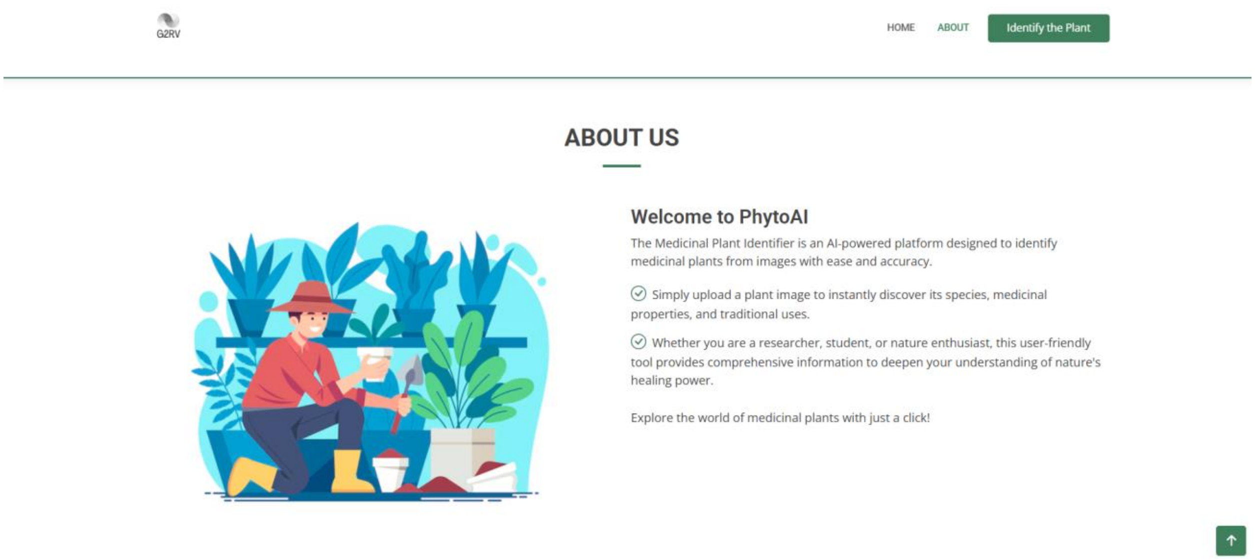


Figure 7.1.2: About Us Page

Prediction Page

This is the Prediction (Identification) page of the PhytoAI project, where users can identify medicinal plants using AI. At the top, users can upload an image of a plant by clicking the "Choose File" button and then selecting "Click here to see result" for analysis. Below that, the "How It Works" section explains the process in three simple steps:

1. Take a photo of the plant or affected area.
2. Upload Image to the platform for AI-powered analysis.
3. Get Results showing the plant’s identity and suggested medicinal uses or treatment options.

This interface makes plant identification easy, efficient, and accessible to all users.

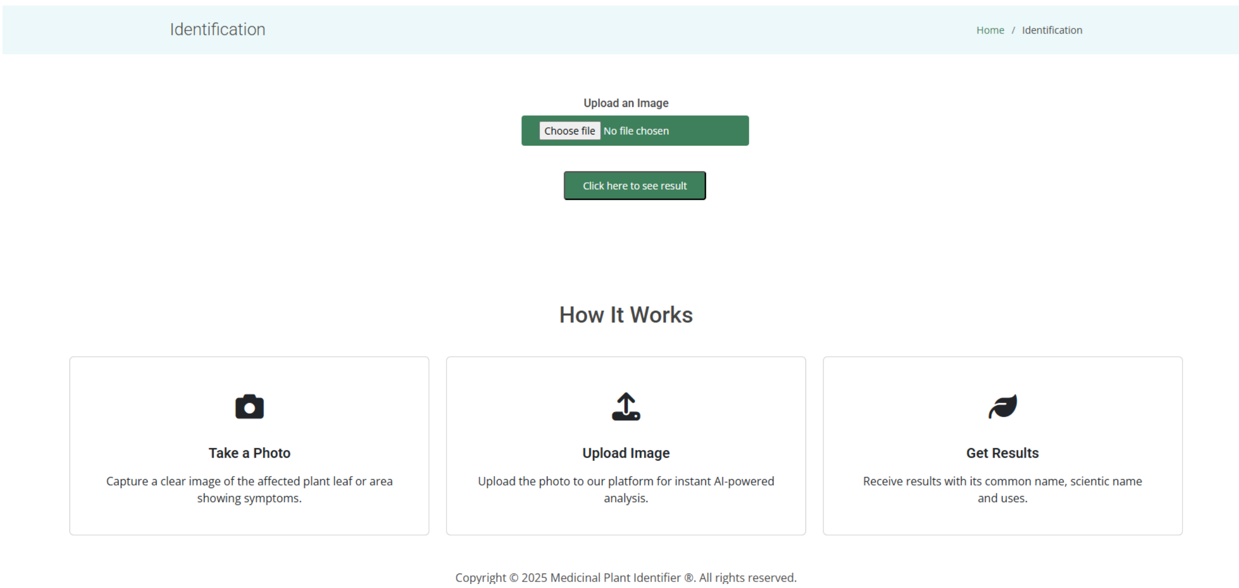


Figure 7.1.3: Prediction Page

Final Output

This is the final output page of the PhytoAI project, showcasing the result after an image is uploaded for plant identification. In this example, the uploaded image of a leaf has been successfully recognized as Ocimum Tenuiflorum (Tulsi), also known as Holy Basil. The result displays both the common name and scientific name of the plant. Additionally, it provides a brief description of its medicinal uses. This output makes the platform highly informative and practical for users seeking quick and accurate plant insights.

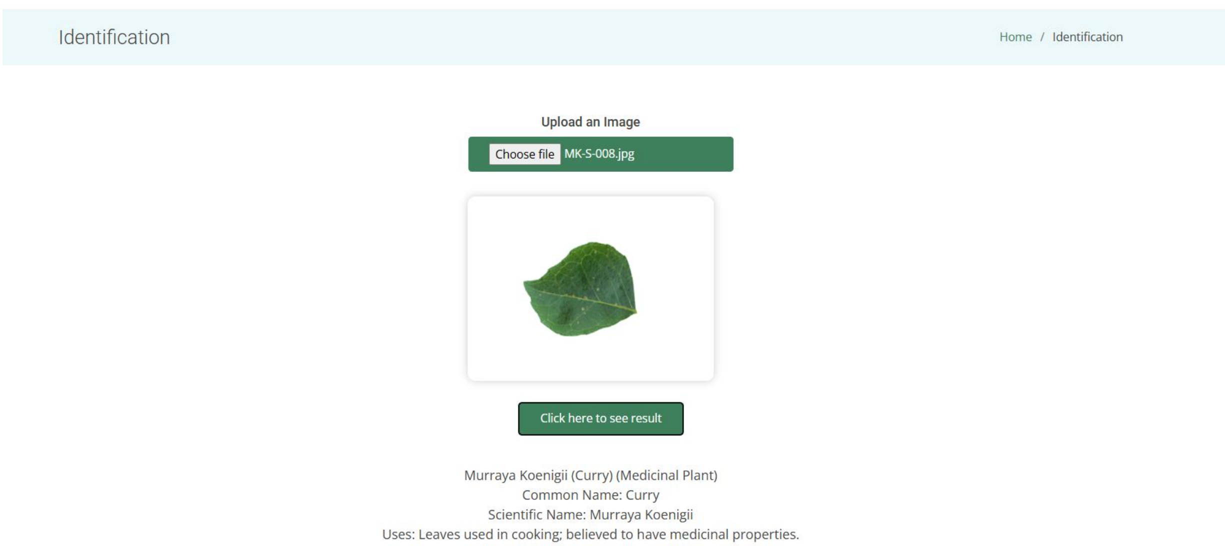


Figure 7.1.4: Final Output

7.2. Unit Testing

7.2.1. Test Case : LeafFeatureExtraction (CNN)

Test Case 1: Summarizing a Short Legal Document

Description: This test case validates the accuracy of the convolutional neural network (CNN) in extracting relevant features (e.g., texture, vein patterns, color) from the leaf image.

Inputs: A set of leaf images (pre-processed or raw).

Expected Output: A feature vector representing the leaf's visual characteristics.

Test Steps:

1. Input the leaf image into the CNN feature extractor.
2. Extract the feature vector from the CNN's output layer.
3. Compare the extracted feature vector to the expected output
4. If the output matches within an acceptable range of variation, the test case passes; otherwise, it fails.

7.2.2. Test Case : Morphological Feature Extraction (SVM)

Description: This test ensures that the Support Vector Machine (SVM) model extracts morphological features such as leaf shape, size, and edge properties accurately.

Inputs: A set of predefined morphological descriptors (e.g., length, width, serration count).

Expected Output: A numerical vector representing the morphological features.

Test Steps:

1. Input the leaf image into the SVM feature extraction pipeline.
2. Extract the morphological features (e.g., aspect ratio, perimeter, convexity).
3. Compare the extracted features to a predefined set of expected values.
4. If the extracted and expected values are within the acceptable error threshold, the test case passes; otherwise, it fails.

7.2.3. Test Case : Hybrid CNN-SVM Classification

Description: This test case evaluates the combined SVM and CNN hybrid model's ability to classify medicinal plants accurately.

Inputs: Feature vectors generated from both CNN and SVM (previous test outputs).

Expected Output: Correct classification of the plant species (e.g.,"Neem," "Tulsi," "Aloe Vera").

Test Steps:

1. Pass the CNN and SVM feature vectors to the hybrid classifier.
2. Generate a classification prediction using the hybrid model.
3. Compare the predicted plant species to the expected species.
4. If the predicted species matches the expected output, the test case passes; otherwise it fails.

7.3. Analysis

7.3.1. Model Accuracy Over Epoch

This indicates the graph is tracking how the model's accuracy changes over time as it trains, although for SVM, training doesn't happen in multiple epochs like deep learning models.

X-Axis (Epochs):

It represents the number of training iterations, but since SVM does not train over epochs, the x-axis shows just a single point or very limited values.

Y-Axis (Accuracy):

This shows the accuracy score of the model, ranging from 0 to 1 (0% to 100%), and here it's around 0.85, meaning the model correctly classifies 85% of the data.

Red Dashed Line (SVM Accuracy):

It shows a flat accuracy line at 0.85, because SVM gives a fixed accuracy after training once, unlike models that improve over time.

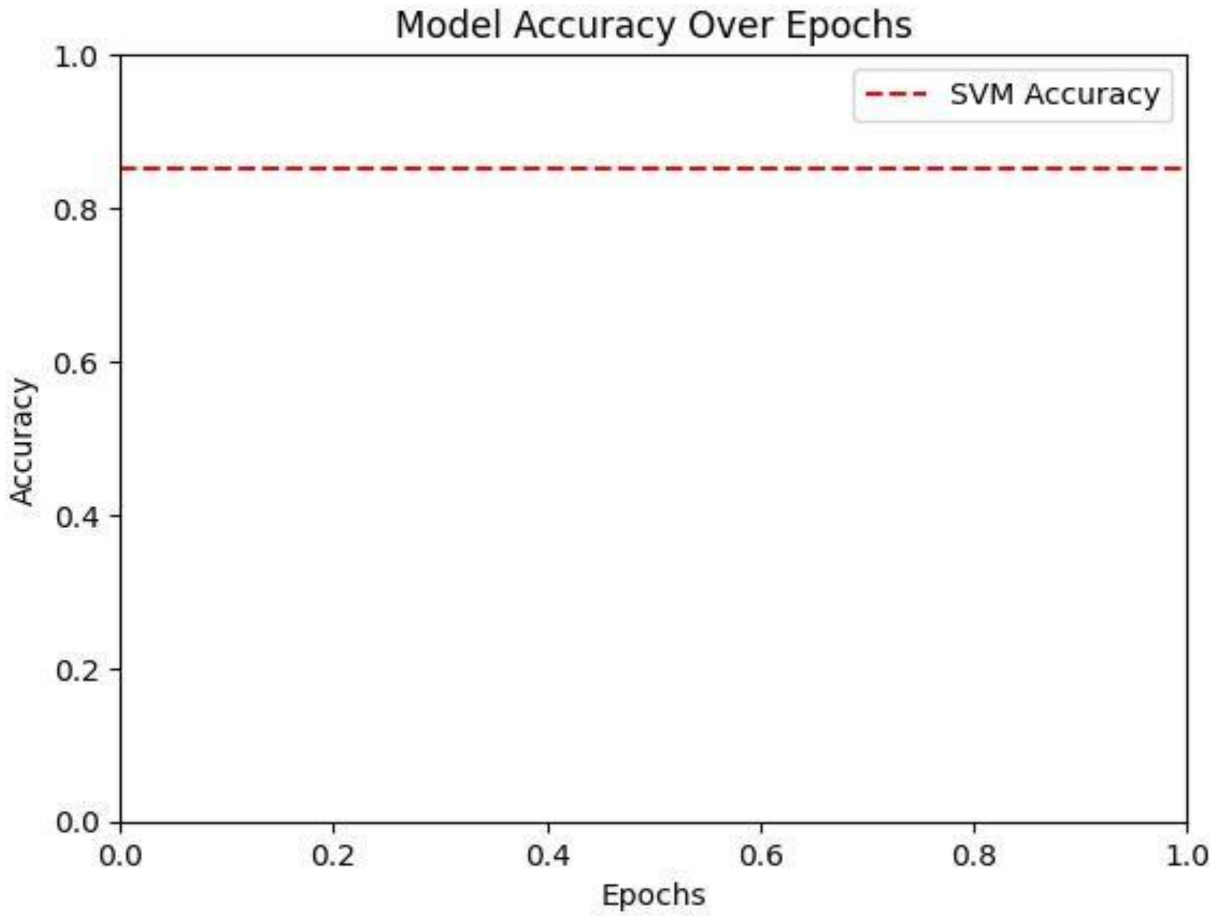


Fig. 7.3.1 : Model Accuracy Over Epochs

7.3.2. CNN-SVM Training and Validation Accuracy

The following graph shows the training and validation accuracy of the CNN model over 80 epochs. The red line (training accuracy) increases steadily and reaches nearly 100%, showing the model learns well on training data.

The blue line (validation accuracy) fluctuates heavily, indicating instability and overfitting. Early in training, both accuracies rise together, but later the gap widens. High variance in validation accuracy suggests the model struggles to generalize. This could be due to small or imbalanced dataset. Regularization or early stopping might help improve validation performance.

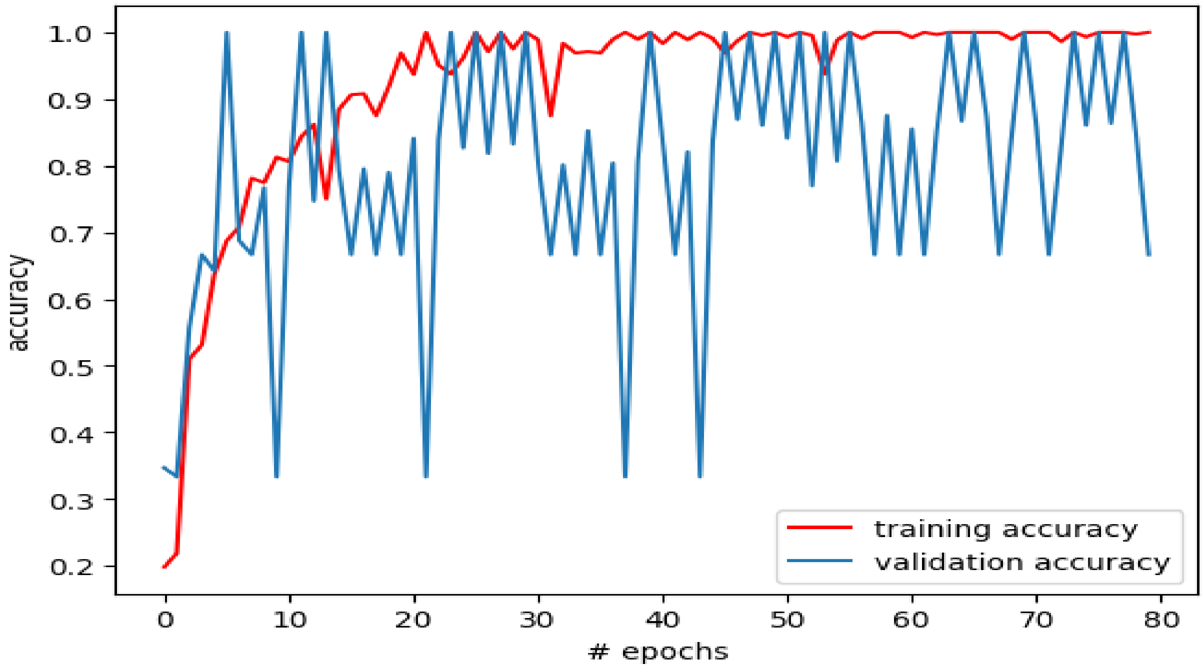


Fig 7.3.2 CNN-SVM Training and Validation Accuracy

Chapter 8

TECHNICAL SPECIFICATION

8.1. Advantages

- **Accuracy and Efficiency :** Machine learning models can process vast amounts of data and identify patterns more quickly and accurately than traditional methods. This leads to a higher success rate in correctly identifying medicinal plants based on images, leaf shapes, chemical compositions, etc.

- **Cost-Effectiveness :** Automating the identification process reduces the need for extensive fieldwork and manual labour, cutting down on time and cost, especially in large-scale or commercial applications.

- **Scalability :** Once a machine learning system is developed, it can be easily scaled up to cover a wide variety of plants across different regions, making it highly adaptable and useful in global biodiversity projects.

- **Consistent Learning and Improvement :** With continuous data input, machine learning models can improve over time. This means the system can adapt to new discoveries or data about medicinal plants, enhancing its performance continuously.

- **Cross-Disciplinary Applications :** These algorithms can be integrated with other fields like chemistry or pharmacology to not only identify the plants but also understand their active compunds and potential medicinal applications.

8.2. Disadvantages

- **Limited Data Availability :** Machine learning models rely heavily on large and high quality datasets. However, there is often limited data available for some medicinal plants, especially rare species. This can lead to inaccurate identification or poor model performance.

- **Overfitting :** Machine learning models may overfit the data if not properly designed or trained. This means the model might perform well on training data but fail when exposed to new, unseen plant data.

8.3. Applications

- **Drug Discovery :** Machine learning models can help identify plants with medicinal properties, aiding pharmaceutical companies in discovering new drugs based on plant compounds.

- **Biodiversity Studies :** Machine learning can automate the process of identifying and cataloguing medicinal plants, contributing to the study and preservation of biodiversity in various ecosystems.

- **Precision Agriculture :** Farmers can use plant identification systems to recognize medicinal plants growing in their fields and optimize their cultivation. This can be especially useful for farmers who cultivate herbal plants for pharmaceutical industries.

- **Pest and Disease Management :** Recognizing the medicinal properties of certain plants can lead to natural pest and disease management solutions, promoting sustainable farming practices.

- **Herbal Medicine Studies :** Students and professionals in the fields of herbal medicine and ethnobotany can use machine learning systems for educational purposes, allowing them to quickly learn about the characteristics and uses of various medicinal plants.

Chapter 9

CONCLUSION

9.1. Conclusion

The automated identification and classification of medicinal plants using machine learning represent a significant advancements in both botanical science and herbal machine. By leveraging the power of technology, this approach enhances the accuracy and efficiency of plant identification, overcoming the limitations associated with traditional methods that often rely on expert knowledge and are susceptible to human error. Machine learning not only facilitates rapid and reliable identification of medicinal plants but also democratizes access to this knowledge, empowering non-experts to engage with plant identification through user-friendly applications.

The applications of this technology span various fields, including drug discovery, conservation, agriculture, education, and healthcare, underscoring its potential to make a meaningful impact on biodiversity preservation and the sustainable use of plant resources. Moreover, the integration of machine learning with traditional knowledge offers a pathway to preserve valuable cultural heritage while advancing scientific understanding.

However, it is essential to acknowledge the challenges and limitations inherent in this approach, such as data dependency, the need for continuous model updates, and the black-box nature of some machine learning algorithms. Addressing these challenges will require ongoing collaboration between botanists, data scientists, and technologists to develop robust, interpretable, and scalable systems.

BIBLIOGRAPHY

[1] Anu Paulson, S. Ravishankar.AI Based Indigenous Medicinal Plant Identification, 2020 Advanced Computing and Communication Technologies for High- Performance Applications (ACCTHPA), July 202.

[2] Luciano D.S. Pacifico; Larissa F.S. Britto; Emilia G. Oliveira; Teresa B. Ludermir. Automatic Classification of Medicinal Plant Species Based on Color and Texture Features, IEEE 2019

[3] C. Sivaranjani; Lekshmi Kalinathan; R. Amutha; Ruba Soundar Kathavarayan. Real-Time Identification of Medicinal Plants using Machine Learning Techniques. IEEE 2019

[4] Perez, A.J., Lopez, F., Benlloch, J.V. and Christensen, S., 2000. Colour and shape analysis techniques for weed detection in cereal fields. Computers and e electronics in agriculture, 25(3), pp.197-212

[5] Vijayashree, T. and Gopal, A., 2015. Classification of Tulsi Leaves Based on Texture Analysis

[6] A. and Sharma, S., 2013. An automatic leaf recognition system for plant identification using machine vision technology. International Journal of Engineering Science and Technology, 5(4), p.874

[7] Im, C., Nishida, H. and Kunii, T.L., 1998, November. A Hierarchical Method of Recognizing Plant Species by Leaf Shapes. In MVA (pp. 158- 1Im, C., Nishida, H. and Kunii, T.L., 1998, November. A Hierarchical Method of Recognizing Plant Species by Leaf Shapes. In MVA (pp. 158-161)

[8] Sarish R. Chavan Dr. Poonam D. Lambhate, Vaishnavi P. Gayke, Amos N. Chavan, Srushti H. Moghe. Detection of Arrhythmia using Single-Lead ECG and Deep Neural Network. International Journal of Advanced Research in Science, Communication and Technology 2023/5

Annexure I

- **Copyright Details**

भारत सरकार / GOVERNMENT OF INDIA
कॉपीराइट कार्यालय / Copyright Office
बौद्धिक संपदा भवन,प्लॉट संख्या 32, सेक्टर 14, द्वारका, नई दिल्ली-110078 फोन: 011-28032496
Intellectual Property Bhawan, Plot No. 32, Sector 14, Dwarka, New Delhi-110078 Phone: 011-28032496
रसीद / Receipt

Copyright Office

PAGE No : 1

To,

POOONAM -IAMBHATE
JASMINIUM, H-204, MAGARPATTA CITY, HADAPSAR, PUNE--411028

(M)-9975563740 : E-mail- jscoeit@gmail.com

RECEIPT NO : 143307
FILING DATE : 04/10/2024
BRANCH : Delhi
USER : jscoeit

S.No	Form	Diary No.	Request No	Title	Amount (Rupees)
1	Form-XIV	30996/2024-CO/L	158376	Identification of Medicinal Plants Using Hybrid Model	500
Amount in Words			Rupees Five Hundreds		500

PAYMENT MODE	Transaction Id	CIN
Online	C-0000177166	0410240008191

(Administrative Officer)

*This is a computer genereated receipt, hence no signature required.

*Please provide your email id with every form or document submitted to the Copyright office so that you may also receive acknowledgements and other documents by email.

Annexure II

- **Published Research Paper**

Journal of University of Shanghai for Science and Technology ISSN: 1007-6735

Identification of Medicinal Plants using Hybrid Model

Dr. Poonam Lambhate
Department of Computer Engineering
JSPM's JSCOE
jscoeit@gmail.com

Ms. Gauri Girme
Department of Computer Engineering
JSPM's JSCOE
girmegauri18@gmail.com

Ms. Kulthe Rachana
Department of Computer Engineering
JSPM's JSCOE
rachana.kulthe04@gmail.com

Ms. Rutugandh Gaikwad
Department of Computer Engineering,
JSPM's JSCOE
rutugandhgaikwad@gmail.com

Ms. Vaishnavi Deshmukh
Department of Computer Engineering
JSPM's JSCOE
vaishud150604@gmail.com

Abstract - Plants play a vital role in our lives by providing various facilities such as food, oxygen, medicines, etc. They are valuable for their medicinal properties and contain substances that are used in medicines. However, many such plants are being destroyed due to many environmental challenges such as global warming, pollution, and Limited government support for the research on medicinal plants. Manual identification of plants requires manual observation and expertise. This issue must be addressed by creating an automated identification system for medicinal plants. The hybrid model of CNN and SVM is to be implemented in the system. This hybrid model of CNN and SVM will extract features and classify them based on various features such as the color, shape, and texture of the leaf of a medicinal plant. This system will help identify the medicinal plants effectively and accurately and display information about them if they are identified as medicinal plants.

Keywords – Plants, Medicinal plants, CNN, Hybrid model, SVM, Accuracy, Identification

I. INTRODUCTION

(Identification of Medicinal Plants using Hybrid Model)

Ayurveda is an ancient practice of medicine in India and has its roots in Vedic times approximately 5000 years ago. The main constituents of Ayurvedic medicines are plant leaves and other parts of plants like roots, bark, etc. More than 8000 plants of Indian origin are of medicinal value. Over 80% of plants used for ayurvedic formulations are collected from forests and wastelands whereas the remaining are cultivated in agricultural lands. With the extensive use of medicinal plants, identifying them is a challenging task as they require manual observation which can be time-consuming and may lead to errors in the identification of plants. This is uncertain for those with limited knowledge of the botanical field which may lead to false identification of plants and outcomes. We aim to train the Machine Learning model on a dataset of known medicinal plants and use them to classify the plants. This recognition based on leaves depends on finding exact descriptors and extracting the feature vectors from them. Then the feature vectors of the training samples are compared with the feature vectors of the test sample to find the degree of similarity using an appropriate classifier.

Overview: The system has been developed to automate the identification process of medicinal plants. The system first takes an image of plants as input and identifies it using the model trained by training and testing datasets. After successful identification, whether the plant is medicinal or not the system fetches information such as name, uses, and applications of medicinal plant from the database of plant information and displays it on GUI.

II. LITERATURE SURVEY

The paper[1] presents a technique for automatic identification of medicinal plants using machine learning. The authors have implemented a random forest algorithm that uses color, texture, and geometric features to classify medicinal plant species. The proposed approach aims to automate the identification process, which has traditionally been done manually. The authors review prior work on medicinal plant classification using image processing and machine learning techniques, and demonstrate the effectiveness of their approach through experiments

Journal of University of Shanghai for Science and Technology ISSN: 1007-6735

on a dataset of plant leaves.
- The technique uses random forest algorithm and extracts color, texture, and geometric features from plant leaf images
- Prior work on medicinal plant classification using image processing and machine learning is reviewed
- Experiments show the effectiveness of the proposed approach in accurately classifying medicinal plant species undefined.

The paper[2] presents a novel approach to medicinal plant identification through computer vision and deep learning. It introduces a custom dataset called DeepHerb, with 2,515 leaf images of 40 Indian medicinal herbs. The research compares various pre-trained neural networks (VGG16, VGG19, InceptionV3, Xception) for feature extraction and classification using Artificial Neural Network (ANN) and Support Vector Machine (SVM). The Xception-ANN model achieved 97.5% accuracy. The paper also describes "HerbSnap," a mobile app utilizing this model for real-time plant identification.
- The DeepHerb dataset was specifically designed for the study, containing 2,515 leaf images from 40 medicinal plant species, making it a valuable resource for future research.
- The paper uses Bayesian optimization to fine-tune the SVM hyperparameters, improving the model's accuracy.
- The research leverages transfer learning to overcome the challenge of small datasets, utilizing pre-trained CNNs to achieve high classification accuracy without needing massive data.

The paper[3] focuses on using digital image processing techniques to identify plant species using features like shape, color, and texture of leaves. The study applies methods like Gaussian filtering, K-means clustering, and Principal Components Analysis (PCA) for feature extraction. Support Vector Machine (SVM) is used as the primary classifier, achieving an accuracy of 95.17%. The system effectively enhances plant identification using machine learning, but future research could address challenges with smaller leaves.
- The study uses techniques like Gaussian filtering for noise reduction and K-means clustering for image segmentation.
- PCA (Principal Components Analysis) is used to extract key features like shape, color, and texture from plant leaf images
- SVM is the primary machine learning classifier, achieving 95.17% accuracy.

Future work could improve the detection of small and immature leaves.

The paper[4] describes an approach to automatically identify medicinal plants using a deep learning (DL) model. The research focuses on six plant species: Betel, Curry, Tulsi, Mint, Neem, and Indian Beech, with 500 images per plant collected from Kaggle. After applying pre-processing techniques like resizing and augmentation, the MobileNet DL model was used for classification. The model achieved 98.33% accuracy. The model has a 97%+ score in metrics like precision, recall, and accuracy, and the app provides easy access to medicinal plant information.
- Six medicinal plants (Betel, Curry, Tulsi, Mint, Neem, Indian Beech) were used from a Kaggle dataset, with 500 images per speciesMobileNet model was used for classification, achieving 98.33% accuracy
- Data augmentation techniques like flipping, rotation, and color manipulation were applied to increase training samples

III. METHODOLOGY

The system focuses on designing and implementing a machine learning-based system to identify medicinal plants based on leaf characteristics. It encompasses various phases, including data collection, model development, and system deployment, aimed at creating a reliable tool for plant identification. Implement a hybrid machine learning model combining Convolutional Neural Networks (CNNs) and Support Vector Machines (SVM), to classify plants based on leaf features. The system focuses on identifying medicinal plants using visual leaf characteristics only, and it may not extend to other plant parts (e.g., flowers, stems). This system will provide a tool for automated, accurate identification of medicinal plants, aiding in research, healthcare, and conservation efforts.

A. ARCHITECTURE MODEL

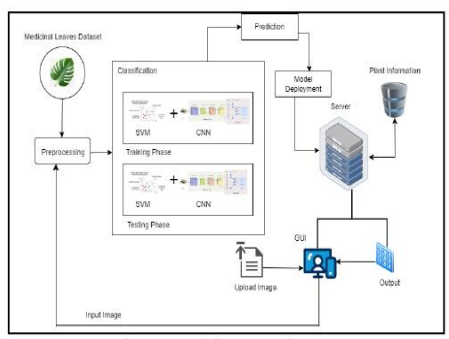


Fig. 3.1 Architecture Diagram

Volume 27, Issue 1, January - 2025 111

Journal of University of Shanghai for Science and Technology ISSN: 1007-6735

1) Data Acquisition and Preprocessing: The system collects images of medicinal leaves to create training and testing datasets. These images are preprocessed to prepare them for analysis. The preprocessing includes steps like resizing, normalization, and feature extraction.

2) Model Training: The system employs the combination of two classification algorithms:

a) Support Vector Machine(SVM): This is a classification algorithm that finds an optimal hyperplane to separate data point classes.

b) Convolutional Neural Network(CNN): This algorithm is for image classification tasks. They learn hierarchical features from input images.

The preprocessed images are used to train both SVM and CNN models. This process involves adjusting parameters to minimize errors between predicted output and actual labels.

The use of both SVM and CNN models is a hybrid approach where SVM handles smaller datasets and lower dimensional feature spaces and CNN handles large datasets and complex image features.

The combination of CNN and SVM helps to improve accuracy as CNN can extract features and SVM can classify them.

3) Model Testing: Once the models are trained, they are evaluated using a separate set of images i.e. training dataset. It helps assess the model's accuracy and generalization ability.

4) Model Deployment: The model is deployed on the server. This allows the system to be accessible remotely.

5) User Interface(GUI): Users can upload an image of a leaf through GUI. The uploaded image is preprocessed and fed into a hybrid model. The system predicts the plant species based on the output of the model. The system displays predicted plant species and relevant plant information, such as medicinal properties and use.

B. MATHEMATICAL MODEL

$$y = \text{Softmax}\,(W_{FC}\,\text{Pooling}\,(\text{ReLU}\,(W_{Conv} * X + b_{Conv})) + b_{FC})$$

where,
X: Input data
W_{conv}: Weights of the convolutional layer
b_{Conv}: Bias of the convolutional layer
ReLU: Activation function
Pooling: Poling operation
W_{FC}: Weights of fully connected layers

1) Convolution Operation ($W_{Conv} * X + b_{Conv}$)
   - X is the input to the convolutional layer, typically an image or a feature map.
   - W_{Conv} represents the weights of the convolutional filters, and b_{Conv} is the bias term.
   - The convolution operation (denoted by "*") is applied to the input, where the filters (W_{Conv}) scan the input and detect local features (edges, textures, patterns, etc.).
2) ReLU Activation (ReLU(...))
   - After the convolution, a Rectified Linear Unit (ReLU) activation function is applied. This introduces non-linearity into the model by setting all negative values to zero while leaving positive values unchanged.
   - ReLU helps the network learn complex patterns by allowing the model to focus on positive activations and ignore irrelevant information.
3) Pooling Operation (Pooling(...))
   - After the ReLU activation, a Pooling operation is applied to reduce the spatial dimensions of the feature maps, which reduces computational complexity and helps the model focus on the most important features.
4) Fully Connected Layer (W_{FC} * ... + b_{FC})
   - The output of the pooling operation is flattened into a vector and passed to a Fully Connected (FC) layer.
   - W_{FC} represents the weights of the fully connected layer, and b_{FC} is the bias term.
5) Softmax Output
   - Finally, the output of the fully connected layer is passed through a Softmax function, which converts the raw scores into probabilities.

C. ALGORITHM

A Convolutional Neural Network (CNN) is a specialized machine learning algorithm widely employed in computer vision tasks. Specifically designed for processing grid-like data, such as images or videos, CNNs have revolutionized tasks like image classification, feature detection, and image recognition. At the core of a CNN are convolutional layers, where learnable filters scan the input image to detect distinctive features like edges, textures, or shapes. For a given input image, I, and a filter (or kernel) K, the convolution operation at a particular position (i, j) is defined as:

$$S(i, j) = \sum m \sum n\, I(i+m, j+n) \cdot K(m, n)$$

where:

- S (i, j) is the value of the resulting feature map at position (i, j).
- I (i+m, j+n) represents the pixel value of the input image at position (i+m, j+n).
- K (m, n) is the weight of the kernel at position (m, n).
- m and n are the dimensions of the filter.

This equation essentially describes how the filter slides across the image, multiplying and summing the overlapping values of the filter and the image to create an activation map. This process helps detect specific features like edges or textures in the image.

Volume 27, Issue 1, January - 2025 112

Journal of University of Shanghai for Science and Technology ISSN: 1007-6735

IV. RESULTS AND ANALYSIS

A. RESULT

1) Images before training

The Fig. 4.1 shows the collection of various leaf images in the dataset. Each image is labeled with its corresponding plant species.

Fig 4.1. Images before training

2) Images after training

Fig. 4 .2 shows the results of the CNN model trained on a dataset of leaf images. Here, the model classified the images based on their visual features, which it had learned from the training data.

Fig. 4.2. Images after training

3) Accuracy graph

Fig. 4.3. shows the performance of the model likely a neural network, during its training process. The x-axis represents the number of training epochs, while the y-axis indicates the accuracy of the model.

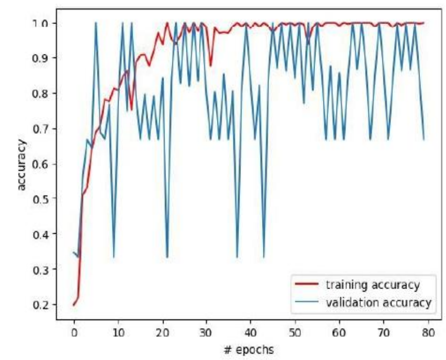


Fig 4.3. CNN accuracy graph

The two curves on the graph represent:

i) Training Accuracy: This curve shows how accurately the model predicts the current class of training data. As the model is trained, it typically improves its accuracy on training data.

ii) Validation Accuracy: This curve shows how accurately the model predicts the correct class of validation dataset, which is a separate set of data not used for training.

B. OUTPUT

Fig. 4. 4 shows the output of the medicinal plants identification system where users can upload an image of a leaf and then process it by a trained Convolutional Neural Network(CNN) model.

Fig. 4.4. Output

This model predicts the plant species and displays the result, including its common and scientific names and medicinal properties.

Volume 27, Issue 1, January - 2025 113

Journal of University of Shanghai for Science and Technology ISSN: 1007-6735

V. CONCLUSION

The automated identification and classification of medicinal plants using machine learning represent a significant advancement in both botanical science and herbal medicine. By leveraging the power of technology, this approach enhances the accuracy and efficiency of plant identification, overcoming the limitations associated with traditional methods that often rely on expert knowledge and are susceptible to human error. Machine learning not only facilitates rapid and reliable identification of medicinal plants but also democratizes access to this knowledge, empowering non-experts to engage with plant identification through user-friendly applications.

VI. ACKNOWLEDGMENT

We would like to express our special thanks of gratitude to our guide and head of the Computer Department Dr. Poonam Lambhate for their able guidance and support. We would also like to extend our gratitude to the Principal Mr. R. D. Kanphade for providing us with all the facilities that was required. We would also like to acknowledge with much appreciation the role of our staff. We sincerely thank our parents and our friends who have been always helping and encouraging us.

VII. REFERENCES

[1] Anu Paulson, S. Ravishankar.AI Based Indigenous Medicinal Plant Identification, 2020 Advanced Computing and Communication Technologies for High-Performance Applications (ACCTHPA), July 202.

[2] Luciano D.S. Pacifico; Larissa F.S. Britto; Emilia G. Oliveira; Teresa B. Ludermir. Automatic Classification of Medicinal Plant Species Based on Color and Texture Features, IEEE 2019

[3] C. Sivaranjani; Lekshmi Kalinathan; R. Amutha; Ruba Soundar Kathavarayan. Real-Time Identification of Medicinal Plants using Machine Learning Techniques. IEEE 2019

[4] Perez, A.J., Lopez, F., Benlloch, J.V. and Christensen, S., 2000. Color and shape analysis techniques for weed detection in cereal fields. Computers and electronics in agriculture, 25(3), pp.197-212

[5] Vijayashree, T. and Gopal, A., 2015. Classification of Tulsi Leaves Based on Texture Analysis

[6] A. and Sharma, S., 2013. An automatic leaf recognition system for plant identification using machine vision technology. International Journal of Engineering Science and Technology, 5(4), p.874

[7] Im, C., Nishida, H. and Kunii, T.L., 1998, November. A Hierarchical Method of Recognizing Plant Species by Leaf Shapes. In MVA (pp. 158- 1Im, C., Nishida, H. and Kunii, T.L., 1998, November. A Hierarchical Method of Recognizing Plant Species by Leaf Shapes. In MVA (pp. 158- 161)

[8] Dr. Sandip D. Satav* Dr. Shailesh M Hambarde1 Dr. Aparna S Hambarde2 Dr. Poonam D Lambhate3 Mrs. Aarti S Satav3. Utilizing A Hybrid Gru Intrusion Detection System To Improve The Security Of IoT-Based Smart Farming. Journal For Basic Sciences. 2024/10

[9] Dr. Sandip D. Satav* Dr. Poonam D Lambhate1 Dr. Shailesh M Hambarde2 Dr. Aparna S Hambarde3 Mrs. Aarti S Satav. Using Random Forests And Internet Of Things Sensors To Optimize Predictive Maintenance In Smart Factories. Computer Research and Development. 2024/10

[10] Dr. Sandip D. Satav1 Dr. Poonam D Lambhate2 Dr. Shailesh M Hambarde3 Mrs. Aarti S Satav4. WSN Technology-Based IoT-Based Smart Agriculture System For Insect And Pest Identification, Crop Status Prevention. Journal of Technology. 2024/7

[11] Dr. Poonam D Lambhate1 Dr. Shailesh M Hambarde2 Mrs. Aarti S Satav3. Smart Cities Can Generate Efficient Power From Garbage Through The Integration Of IoT And LSTM-MLP. Journal of Informetrics 2024

[12] Dr. Sandip D. Satav* Dr. Shailesh M Hambarde1 Dr. Poonam D Lambhate2 Mrs. Aarti S Satav3. An Effective Allocation Scheme for Resources in Edge Computing enabled Networks with Uncertain Network Status. Dastavej Research Journal 2024

[13] Sarish R. Chavan Dr. Poonam D. Lambhate, Vaishnavi P. Gayke, Amos N. Chavan, Srushti H. Moghe. Detection of Arrhythmia using Single-Lead ECG and Deep Neural Network. International Journal of Advanced Research in Science, Communication and Technology 2023/5

- **Research Paper Acceptance Certificate**

JUSST
Multidisciplinary Journal
多学科杂志

Journal of University of Shanghai for Science and Technology
ISSN: 1007-6735
UGC-CARE(Group-II)Approved & SCOPUS ACTIVE Journal

ACCEPTANCE LETTER TO AUTHOR

Dear Author,

We are pleased to inform you that your paper entitled **"Identification of Medicinal Plants using Hybrid Model"** has been accepted for publication in the Journal of University of Shanghai for Science and Technology (JUSST), Volume 27, Issue 01, January 2025.

You're Paper ID: JUSST/25/01-35

Please note that to maintain the article online and to obtain a soft copy of the E-Certificate, we charge a processing fee of ₹2500/ (50 USD for international authors). Additionally, an extra fee of ₹200 or $5 is required to obtain the DOI. We would like to emphasize that this fee is nominal and covers only the online maintenance and processing expenses.

Article Processing Charges Includes
Online Publication & E-certificates
Online maintenance and processing charge
No limitation of number of pages
Editorial fee
Taxes

Note:

- The paper will be published online within 48-72 hours after we receive the fee.
- If you have any queries, please do not hesitate to contact us at editor@jusst.org. We appreciate an early reply.
- Please note that the fee paid for the publication of the paper is non-refundable under any circumstances.

Date
1/20/2025

Best regards,
Dr. Xuhui Feng
jusst.org

Dr. Xuhui Feng
Editor-in-Chief

ज्ञान-विज्ञान विमुक्तये
UGC-CARE
Approved Group-II Journal

Member of Crossref

Research Paper Acceptance Certificate

- **Research Paper Publication Certificate**

JUST
Multidisciplinary Journal
多学科杂志

ISSN:1007-6735
Journal of University of Shanghai for Science and Technology

Certificate of Publication

This is to certify that

Ms. Gauri Girme

Department of Computer Engineering, JSPM's Jayawantrao Sawant College of Engineering, Hadapsar, Pune, India.

Published a paper entitled

"Identification of Medicinal Plants using Hybrid Model"

Volume 27 - Issue 1 - 2025

Paper ID: JUSST/25/01-35

DOI: 10.51201/JUSST/25/0135

UGC APPROVED JOURNAL GROUP II

Scopus Google Scholar DOAJ DIRECTORY OF OPEN ACCESS JOURNALS Crossref

Dr. Xuhui Feng
Editor in chief

Publication Certificate – Gauri Girme

JUSST
Multidisciplinary Journal
多学科杂志

ISSN:1007-6735
Journal of University of Shanghai for Science and Technology

Certificate of Publication

This is to certify that

Ms. Kulthe Rachana

Department of Computer Engineering, JSPM's Jayawantrao Sawant College of Engineering, Hadapsar, Pune, India.

Published a paper entitled

"Identification of Medicinal Plants using Hybrid Model"

Volume 27 - Issue 1 - 2025

Paper ID: JUSST/25/01-35

DOI: 10.51201/JUSST/25/0135

UGC APPROVED JOURNAL GROUP II

Scopus Google Scholar DOAJ DIRECTORY OF OPEN ACCESS JOURNALS Crossref

Dr. Xuhui Feng
Editor in chief

Publication Certificate – Rachana Kulthe

ISSN:1007-6735
Journal of University of Shanghai for Science and Technology

JUSST
Multidisciplinary Journal
多学科杂志

Certificate of Publication

This is to certify that

Ms. Rutugandh Gaikwad

Department of Computer Engineering, JSPM's Jayawantrao Sawant College of Engineering, Hadapsar, Pune, India.

Published a paper entitled

"Identification of Medicinal Plants using Hybrid Model"

Volume 27 - Issue 1 - 2025

Paper ID: JUSST/25/01-35

DOI: 10.51201/JUSST/25/0135

UGC APPROVED JOURNAL GROUP II

Scopus Google Scholar DOAJ DIRECTORY OF OPEN ACCESS JOURNALS Crossref

Dr. Xuhui Feng

Editor in chief

Publication Certificate – Rutugandh Gaikwad


JUST
Multidisciplinary Journal
多学科杂志

ISSN:1007-6735
Journal of University of Shanghai for Science and Technology

Certificate of Publication

This is to certify that

Ms. Vaishnavi Deshmukh

Department of Computer Engineering, JSPM's Jayawantrao Sawant College of Engineering, Hadapsar, Pune, India.

Published a paper entitled

"Identification of Medicinal Plants using Hybrid Model"

Volume 27 - Issue 1 - 2025

Paper ID: JUSST/25/01-35

DOI: 10.51201/JUSST/25/0135

UGC APPROVED JOURNAL GROUP II

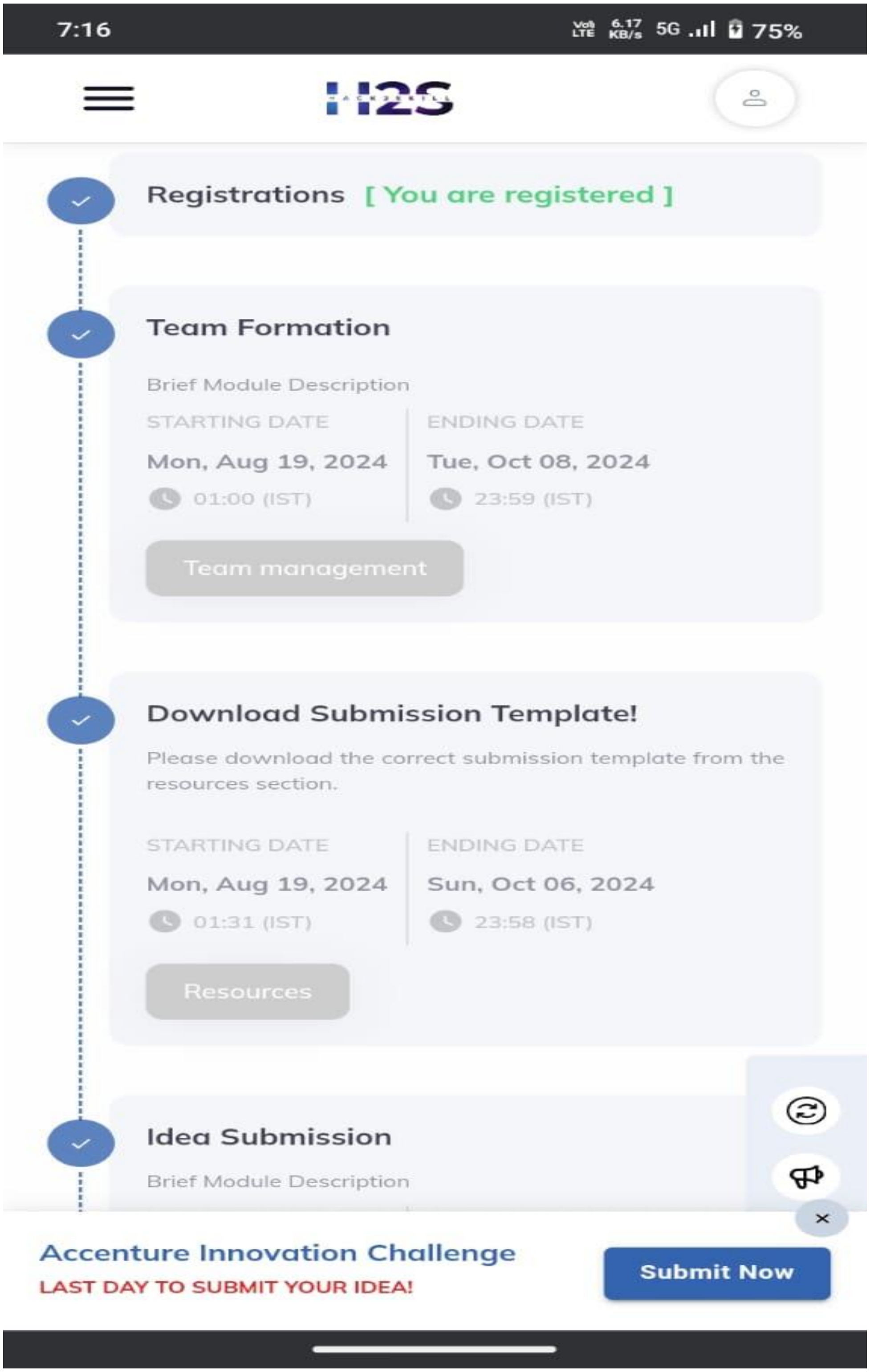
Scopus Google Scholar DOAJ DIRECTORY OF OPEN ACCESS JOURNALS Crossref

Dr. Xuhui Feng
Editor in chief

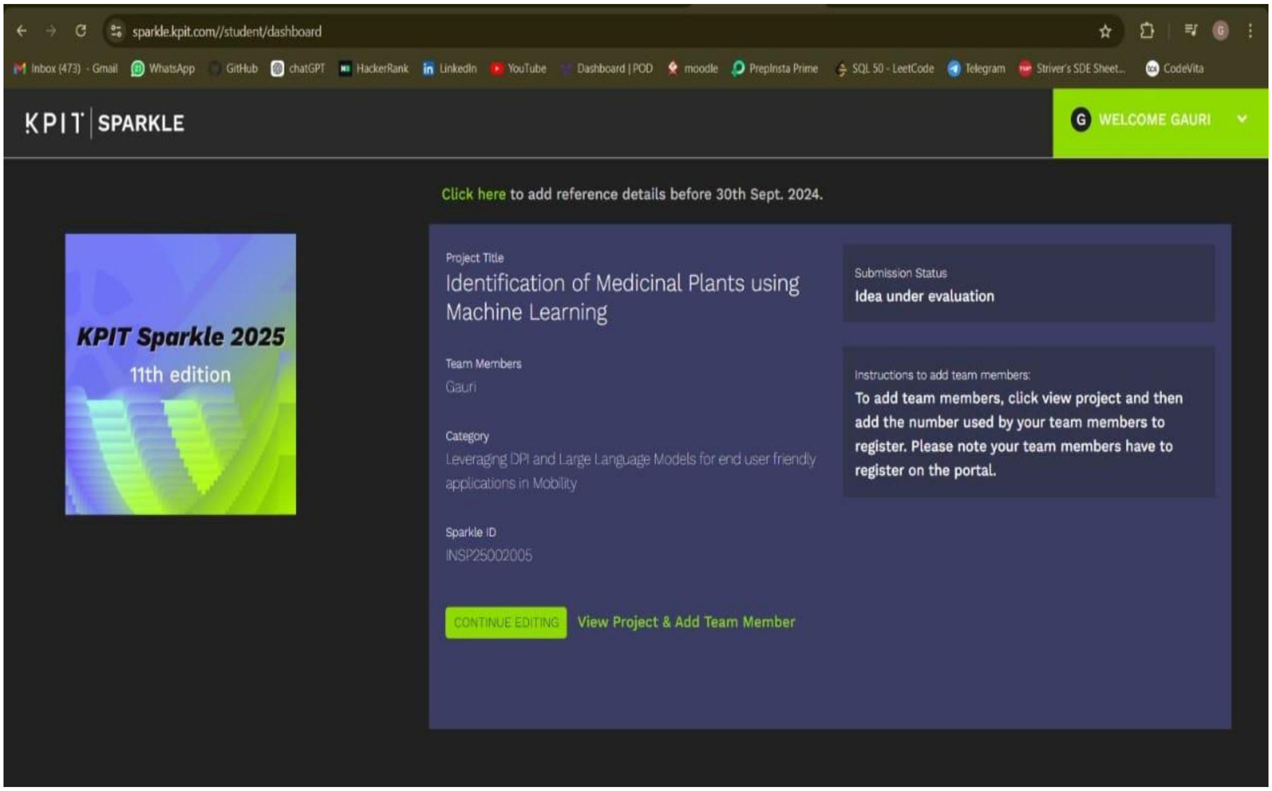
Publication Certificate – Vaishnavi Deshmukh

- Competition Participation

Techmanthan Registration



Accenture Innovation Challenge



KPIT Sparkle Registration