Plantify-Enhanced Medicinal Plant Identification Using Convolutional Neural Networks

Submitted for partial fulfillment of the requirements

for the award of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE ENGINEERING – ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

by

P. Naga Sai Venkata Dakshaini	**-**	**22BQ5A4214**
R. Chandu	**-**	**21BQ1A42F0**
Sk. Khadeer	**-**	**21BQ1A42G2**
Sk. Ushman Basha	**-**	**21BQ1A42G4**

Under the guidance of

Dr. T. Kameswara Rao, Professor

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING – ARTIFICIAL INTELLIGENCE & MACHINE LEARNING
VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY
Permanently Affiliated to JNTU Kakinada, Approved by AICTE
Accredited by NAAC with 'A' Grade, ISO 9001:2008 Certified
NAMBUR (V), PEDAKAKANI (M), GUNTUR – 522 508
Tel no: 0863-2118036, url: www.vvitguntur.com April 2024

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY
Permanently Affiliated to JNTUK, Kakinada, Approved by AICTE
Accredited by NAAC with ‘A’ Grade, ISO 9001:20008 Certified
Nambur, Pedakakani (M), Guntur (Gt) -522508

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING - ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

CERTIFICATE

This is to certify that this **Project Report** is the Bonafide work of **Ms. P. Naga Sai Venkata Dakshaini, Mr. R. Chandu, Mr. SK. Khadeer, Mr. SK. Ushman Basha**, bearing Reg. No. **22BQ5A4214, 21BQ1A42F0, 21BQ1A42G2, 21BQ1A42G4** respectively who had carried out the project entitled **“Plantify-Enhanced Medicinal Plant Identification Using Convolutional Neural Networks”** under our supervision.

Project Guide	**Head of the Department**
(Dr. T. Kameswara Rao)	(Dr. K. Suresh Babu)

Submitted for Viva voce Examination held on ______________________________

Internal Examiner	**External Examiner**

DECLARATION

We, Ms. P. Naga Sai Venkata Dakshaini, Mr. R. Chandu, Mr. Sk. Khadeer, and Mr. Sk. Ushman Basha hereby declare that the Project Report entitled **Plantify-Enhanced Medicinal Plant Identification Using Convolutional Neural Networks"** done by us under the guidance of Dr. T. Kameswara Rao, Professor, Computer Science Engineering - Artificial Intelligence & Machine Learning at Vasireddy Venkatadri Institute of Technology is submitted for partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science Engineering - Artificial Intelligence & Machine Learning. The results embodied in this report have not been submitted to any other University for the award of any degree.

DATE :

PLACE : Nambur

SIGNATURE OF THE CANDIDATES

P.N.S.V. Dakshaini

R. Chandu

SK. Khadeer

SK. Ushman Basha

ACKNOWLEDGEMENT

We take this opportunity to express my deepest gratitude and appreciation to all those people who made this project work easier with words of encouragement, motivation, discipline, and faith by offering different places to look to expand my ideas and helped me towards the successful completion of this project work.

First and foremost, we express my deep gratitude to **Mr. Vasireddy Vidya Sagar**, Chairman, Vasireddy Venkatadri Institute of Technology for providing necessary facilities throughout the B.Tech program.

We express my sincere thanks to **Dr. Y. Mallikarjuna Reddy**, Principal, Vasireddy Venkatadri Institute of Technology for his constant support and cooperation throughout the B.Tech program.

We express my sincere gratitude to **Dr. K. Suresh Babu**, Professor & HOD, Computer Science Engineering - Artificial Intelligence & Machine Learning, Vasireddy Venkatadri Institute of Technology for his constant encouragement, motivation and faith by offering different places to look to expand my ideas.

We would like to express my sincere gratefulness to our Guide **Dr. T. Kameswara Rao**, Professor, Computer Science Engineering - Artificial Intelligence & Machine Learning for his insightful advice, motivating suggestions, invaluable guidance, help and support in successful completion of this project.

We would like to express our sincere heartfelt thanks to our Project Coordinator **Mrs. K. Deepika**, Assistant Professor, Computer Science Engineering - Artificial Intelligence & Machine Learning for her valuable advices, motivating suggestions, moral support, help and coordination among us in successful completion of this project.

We would like to take this opportunity to express my thanks to the **Teaching and Non-Teaching** Staff in the Department of Computer Science Engineering - Artificial Intelligence & Machine Learning, VVIT for their invaluable help and support.

P. Naga Sai Venkata Dakshaini
R. Chandu
SK. Khadeer
SK. Ushman Basha

TABLE OF CONTENTS

CH No	Title	Page No
	Contents	1
	List of Figures	4
	Nomenclature	5
	Abstract	6
1	**INTRODUCTION**	
	1.1 Background of the project	7
	1.2 Problem Statement	8
	1.3 Objectives of the project	9
	1.4 Scope of the project	10
	1.5 Methodology overview	11
	1.6 Peculiarity of Project	12
2	**LITERATURE SURVEY**	
	2.1 Previous Research and Related work	13
	2.2 Existing Solutions and Their Limitations	14
	2.3 Gap Analys	15
	2.4 Relevance of the Project	16
3	**SYSTEM ANALYSIS**	
	3.1 Requirement Analysis	18
	3.1.1 Functional Requirements	19
	3.1.2 Non-functional Requirements	20
	3.1.3 System Requirements	22

3.2 Feasibility Study

3.2.1 Technical Feasibility 23

3.2.2 Economic Feasibility 23

3.2.3 Social Feasibility 23

3.2.4 Operational Feasibility 23

3.3 Proposed System overview 24

4 **SYSTEM DESIGN**

4.1 System Architecture 26

4.2 Input Design 27

4.3 Output Design 28

4.4 UML Diagrams 29

4.4.1 Use Case Diagram 29

4.4.2 Class Diagram 30

4.4.3 Sequence Diagram 31

4.4.4 Collaboration Diagram 33

4.4.5 Component Diagram 34

4.4.6 DFD Diagram 35

4.5 Database Design 36

5 IMPLEMENTATIONS

5.1 Programming Languages and Technologies used 39

5.2 Development Tools and Environments 40

5.3 Module-Wise Implementation Details 41

5.3.1 Medicinal Plant Identification by Image Upload 41

5.3.2 Search and Filter Functionality 41

5.3.3 Multi-Language Translation 42

5.3.4 Home remedies recommendation for common ailments 42

5.3.5 Plant Information Module 42

5.3.6 Contribute Plants 42

5.4 Algorithms used 43

5.4.1 EfficientNet 44

5.4.2 ResNet 45

6 RESULTS AND TESTING

6.1 Testing Methodologies

6.1.1 Unit Testing 48

6.1.2 Integration Testing 48

6.1.3 System Testing 48

6.2 Test case Reports 49

6.3 Performance Metrics Analysis 50

6.4 Output Screenshots 51

6.5 Deployment 56

6.5.1 Deployment Architecture 57

6.5.2 Tools Used 58

6.5.3 Challenges Faced 58

7 CONCLUSION AND FUTURE SCOPE 60

8 REFERENCES 62

APPENDIX

Publication Certificate 67

LIST OF FIGURES

Figure No	Figure Name	Page No
3.1	Functional Requirements	20
3.2	Non Functional Requirements	21
4.1	System Architecture	26
4.2	Use Case Diagram	30
4.3	Class Diagram	31
4.4	Sequence Diagram	32
4.5	Collaboration Diagram	33
4.6	Component Diagram	34
4.7	DFD Diagram	35
4.8	Database design	36
4.9	Medicinal_Plants	37
5.1	EfficientNet Architecture	44
5.2	ResNet Architecture	47
6.1	Testing Report	49
6.2	Classification Report	50
6.3	Home Page	51
6.4	Plantify Green Hub	51
6.5	Search and Retrieval	52
6.6	Multi Language	53
6.7	Image Recognition	53
6.7(a)	Output of image recognition	54
6.8	Remedies	54
6.8(a)	Remedies result	55
6.9	Contribute Plant	55
6.10	Deployment Architecture Diagram	57

NOMENCLATURE

Acronym	**Full Form**
AI	Artificial Intelligence
API	Application Programming Interface
CNN	Convolutional Neural Network
DL	Deep Learning
EDA	Exploratory Data Analysis
EfficientNet	Efficient Convolutional Network
GPU	Graphics Processing Unit
ML	Machine Learning
NLP	Natural Language Processing
ResNet	Residual Network
UI	User Interface

ABSTRACT

The identification of medicinal plants and raw materials plays a crucial role in various fields, including healthcare, pharmaceuticals, and environmental conservation. However, manual identification methods are often time-consuming and require expert botanical knowledge. In this project, we propose a novel approach to automate the identification process using image processing and machine learning algorithms.

We present a comprehensive study on the identification of different medicinal plants and raw materials through image processing techniques. Our methodology involves the collection of a diverse dataset containing images of various plant species and raw materials. We preprocess the dataset to standardize the images and remove noise, ensuring the robustness of our models.

We employ state-of-the-art Deep learning models, including Efficient Net, ResNet for plant identification. These models are trained on the dataset to learn features relevant to plant classification. We evaluate the performance of each model using standard metrics such as accuracy, precision, recall, and F1 score.

Our results demonstrate the effectiveness of our approach in accurately identifying medicinal plants and raw materials from images. We compare the performance of individual models and the ensemble model to identify the most effective approach. Additionally, we develop a user-friendly interface for seamless access to the plant identification system.

Overall, our project provides a valuable contribution to the automation of traditional botanical practices, enhancing accessibility to plant identification for users without expert botanical knowledge. Our system has broad applications in healthcare, pharmaceuticals, agriculture, and environmental conservation, paving the way for future advancements in plant recognition technology.

CHAPTER 1

INTRODUCTION

Medicinal plants have been used for centuries as sources of traditional medicine and natural remedies for various ailments. However, identifying these plants accurately can be challenging, especially for those without expert botanical knowledge. In recent years, advancements in image processing and machine learning algorithms have provided new opportunities for automating the identification process.

This project aims to leverage the power of image processing techniques and machine learning algorithms to develop a system capable of accurately identifying different medicinal plants and raw materials from images. By utilizing state-of-the-art models such as EfficientNet and ResNet the project seeks to achieve high accuracy and robustness in plant recognition.

1.1 Background of the project

Medicinal plants have been used for centuries as sources of traditional medicine and natural remedies for various ailments. However, the identification of these plants is often challenging, particularly for individuals without expert botanical knowledge. Traditional methods of plant identification rely heavily on manual observation and comparison, which can be time-consuming and prone to errors.

With the rapid advancement of technology, particularly in the fields of image processing and machine learning, there is a growing interest in automating the process of plant identification. By leveraging the power of computational techniques, it is possible to develop systems capable of accurately identifying medicinal plants and raw materials from images, thereby streamlining the identification process and making it more accessible to a wider audience.

The rationale for conducting this study lies in the potential benefits it offers to various stakeholders:

Accessibility: By developing an automated system for plant identification, we aim to make this essential task more accessible to a broader range of users, including researchers, healthcare professionals, herbalists, and enthusiasts. This accessibility democratizes botanical knowledge and promotes the sustainable use of medicinal plants.

Efficiency: Manual methods of plant identification are labour-intensive and time-consuming. By automating this process, we can significantly reduce the time and effort required for plant identification, enabling faster decision-making in various applications, such as healthcare, pharmaceuticals, and environmental conservation.

Accuracy: Machine learning algorithms have demonstrated impressive capabilities in image classification tasks. By training models on a diverse dataset of medicinal plants and raw materials, we aim to develop a system capable of accurately identifying plant species with high precision and recall, minimizing the risk of misidentification.

Conservation: The identification of medicinal plants is crucial for their conservation and sustainable use. By facilitating the identification process, our project contributes to efforts aimed at preserving biodiversity and promoting the responsible harvesting and cultivation of medicinal plants.

Innovation: The intersection of image processing, machine learning, and botany presents exciting opportunities for innovation. By pushing the boundaries of technology in plant identification, we pave the way for future advancements in fields such as precision agriculture, personalized medicine, and ecological monitoring.

1.2 Problem Statement

The identification of medicinal plants and raw materials is a critical task in various fields, including healthcare, pharmaceuticals, and environmental conservation. However, traditional methods of plant identification are often labor-intensive, time-consuming, and require expert botanical knowledge. Manual observation and comparison of plant features can be subjective and prone to errors, leading to potential misidentification.

Furthermore, the increasing demand for medicinal plants, coupled with habitat destruction and overexploitation, poses significant challenges to their conservation and sustainable use. Accurate identification of plant species is essential for ensuring their proper management and conservation.

1.3 Objectives of the project

The primary aim of this research project is to develop a robust system for the automated identification of medicinal plants and raw materials using image processing techniques and machine learning algorithms.

To build a comprehensive dataset containing images of various medicinal plants and raw materials, ensuring diversity in species, morphology, and environmental conditions.

To preprocess the dataset by standardizing images, removing noise, and augmenting data to enhance the robustness of the models.

To investigate and implement state-of-the-art machine learning models, including EfficientNet and ResNet, for plant identification.

To train the selected models on the dataset to learn features relevant to plant classification, optimizing hyperparameters and model architectures for improved performance.

To evaluate the performance of each model using standard metrics such as accuracy, precision, recall, and F1 score, comparing the effectiveness of individual models and the ensemble approach.

To develop a user-friendly interface for the deployment of the trained models, allowing users to upload images and receive real-time predictions on plant identification.

To assess the usability and effectiveness of the developed system through user testing and feedback, iteratively refining the interface and functionality based on user input.

To validate the accuracy and reliability of the system through field testing, comparing automated identifications with expert botanical assessments.

To explore potential applications of the developed system in various fields, including healthcare, pharmaceuticals, agriculture, and environmental conservation.

To contribute to the advancement of knowledge in the intersection of image processing, machine learning, and botany, fostering innovation and collaboration in interdisciplinary research.

1.4 Scope of the project

This research focuses on automating the identification process of medicinal plants and raw materials using advanced image processing techniques and machine learning algorithms. The study encompasses a diverse range of medicinal plant species and raw materials utilized in traditional medicine and natural remedies. Various image processing techniques will be employed to preprocess the dataset, including standardization, noise removal, and data augmentation, to ensure the quality and robustness of the images used for training machine learning models.

The research will investigate the effectiveness of state-of-the-art machine learning models such as EfficientNet and ResNet for accurately classifying medicinal plants and raw materials from images. It will involve rigorous training, evaluation, and comparison of these models to determine their performance and suitability for automated plant identification tasks.

Despite its scope and objectives, this research faces several limitations that may impact its outcomes. One such limitation is the size and diversity of the dataset used for model training and evaluation. While efforts will be made to collect a comprehensive dataset, it may not capture all possible variations in plant morphology, environmental conditions, and image quality, potentially limiting the generalization of the models.

1.5 Methodology Overview

The development process of the AI model for the automated identification of medicinal plants and raw materials involves several key steps, as outlined below:

- Data Collection and Preprocessing**:** Gather a diverse dataset containing images of medicinal plants and raw materials from various sources. This dataset should cover a wide range of plant species, morphologies, and environmental conditions. Preprocess the dataset to enhance its quality and suitability for model training. This includes tasks such as image standardization, noise removal, resizing, and data augmentation to increase dataset diversity.
- Model Selection and Architecture Design**:** Choose appropriate deep learning architectures for the task of image classification. Common choices include convolutional neural networks (CNNs) such as EfficientNet and ResNet, known for their effectiveness in image recognition tasks. Design the architecture of the AI model, specifying the number of layers, kernel sizes, activation functions, and other hyperparameters. Fine-tune the model architecture based on the characteristics of the dataset and the complexity of the classification task.
- Training and Validation**:** Split the pre-processed dataset into training, validation, and testing sets. The training set is used to update the model's parameters during training, while the validation set is used to tune hyperparameters and monitor model performance. Train the AI model using the training set and monitor its performance on the validation set. Adjust hyperparameters as needed to optimize model performance, avoiding overfitting or underfitting.
- Model Evaluation**:** Evaluate the trained AI model's performance using the testing set, which contains unseen data not used during training or validation. Measure metrics such as accuracy, precision, recall, and F1 score to assess the model's effectiveness in classifying medicinal plants and raw materials.

- Fine-Tuning and Optimization**:** Fine-tune the AI model based on the evaluation results and feedback from model performance. This may involve adjusting hyperparameters, refining preprocessing techniques, or incorporating additional data to improve model accuracy and generalization.
- Deployment and Integration**:** Deploy the trained AI model into the production environment, integrating it with the user interface or application developed for plant identification. Continuously monitor the model's performance in real-world scenarios, collecting feedback from users and updating the model as needed to ensure ongoing accuracy and effectiveness.

1.6 Peculiarity of the Project

The project focuses on automating the identification process of medicinal plants and raw materials through image processing and machine learning. It encompasses a wide range of medicinal plant species and raw materials typically used in traditional medicine and natural remedies. Various image processing techniques, such as standardization, noise removal, and data augmentation, will be employed to enhance the quality and robustness of the dataset used for training machine learning models.

The research will assess the effectiveness of advanced machine learning models like Efficient and ResNet\ in accurately classifying medicinal plants and raw materials from images. Rigorous training, evaluation, and comparison of these models will be conducted to determine their performance and suitability for automated plant identification tasks.

CHAPTER 2

LITERATURE SURVEY

2.1 Previous Research and Related work

Nowadays, an increasing number of people have shown interest in deep learning systems for solving obesity problems as well as managing diets using food recognition systems.

S. Prasad and P. P. Singh (2017) introduced the ‘‘Vision system for medicinal plant leaf acquisition and analysis,’’ [1] a web application designed for identification of plant species. Utilizing 2 models: l$\alpha\beta$-CNN model and l$\alpha\beta$-CNN + PCA model. The application identifies the plant and gives information for that plant. The study underscores the accuracy of image-based recognition systems resulting to an accuracy of 92 % and 91% respectively.

Similarly, Santhi Daggubati (2024) proposed a comprehensive system for plant image recognition [2]. By using deep learning models such as Inception v3, VGG16, VGG19, Ensemble Model, the system achieves notable accuracy. However, the study acknowledges the challenge of incorporating recipies that can be formed from these plants which can be useful for curing many diseases.

In another research, Sulthana Habiba (2018) proposed another approach for plant species identification. Their study, titled ‘‘Bangladeshi plant recognition using deep-learning based leaf classification’’ [3] explores the use of a VGG 16 function which scores an accuracy of 96%.

All these studies emphasize the deep learning techniques as beneficial in transforming some facets of identification and classification of automated images of medicinal plants for the better. Nevertheless, they all stressed the necessity of more detailed research to solve problems, such as not only identifying a single leaf, but rather several leaves fused or a branch together and working with more sophisticated representation and text classification techniques.

2.2 Existing System and their limitations

The existing systems for automated plant recognition often utilize a combination of image processing techniques and machine learning algorithms. While there are several systems available, some notable examples include:

Plantes Medicinales:

Plantes Medicinales is a mobile application that employs image recognition technology to identify plant diseases and nutrient deficiencies. Users can upload images of affected plants, and the app provides real-time diagnosis and recommendations for treatment. The system uses a combination of convolutional neural networks (CNNs) trained on large datasets of plant images to classify and diagnose plant diseases accurately. Plantees Medicinales also incorporates crowdsourced data and expert knowledge to continuously improve its accuracy and expand its database of plant diseases.

Flora Incognita:

Flora Incognita is a smartphone app developed for automated plant identification in the wild. Users can take pictures of plants, and the app identifies them based on leaf shape, flower colour, and other characteristics. The system utilizes deep learning algorithms trained on extensive datasets of plant images to achieve high accuracy in plant identification. Flora Incognita also includes features for species documentation and biodiversity monitoring, making it useful for both casual users and researchers.

MediScan:

MediScan is a web and mobile application that allows users to identify plants by uploading pictures of their leaves. The system employs image recognition algorithms trained on a small database of plant images contributed by users and experts. MediScan also provides additional information about identified plants, including their distribution, habitat, and ecological importance.

LeafSnap:

LeafSnap is a mobile app designed for identifying tree species from photographs of their leaves. Users can take pictures of leaves against a white background, and the app matches them to a database of tree species. The system uses computer vision algorithms to analyze leaf characteristics such as shape, margin, and venation patterns for accurate identification. LeafSnap also includes educational resources and information about tree species, making it a valuable tool for nature enthusiasts and educators.

Limitations

While existing systems for automated plant recognition offer valuable tools for plant identification and analysis, they also come with some disadvantages. Here are some common drawbacks associated with these systems:

1.Limited Accuracy: One of the main challenges faced by existing systems is limited accuracy, especially when dealing with complex plant species or variations in environmental conditions. Inaccurate identification results can undermine the reliability of these systems, leading to misdiagnosis or incorrect recommendations.

2.Dependency on Image Quality: Many existing systems rely heavily on the quality of input images for accurate identification. Poor lighting conditions, background clutter, or occlusions can negatively impact the system's performance, reducing its effectiveness in real-world scenarios where capturing high-quality images may be challenging.

3.Limited Coverage: Existing systems may have limited coverage of plant species or geographic regions. They may be biased towards commonly encountered species or regions with extensive training data, while lacking support for less common or region-specific plants. This limitation restricts the applicability of these systems in diverse botanical contexts.

4.Resource Intensive: Some automated plant recognition systems require significant computational resources for image processing and model training, making them inaccessible to users with limited computing capabilities or internet connectivity.

2.3 GAP analysis

The separation of therapeutic plants from other non-edible plants is crucial in the realms of botany and the food industry. However, conventional techniques for identifying medicinal plants are difficult, time-consuming, and require skilled specialists. An autonomous real-time vision-based system has been presented to identify commonly used medicinal herbs with similar leaves in order to solve this problem. This system makes use of a convolutional and classifier block-based upgraded convolutional neural network (CNN) network. Global Average Pooling (GAP), dense, dropout, and softmax layers are all present in the classifier. This technique improves the model's speed and accuracy while reducing the number of parameters compared to earlier studies. With overall accuracy rates of 99.66%, 99.32%, and 99.45%, respectively, the proposed CNN model (Fig 2.2.1 shows images related to the first activationlayer of CNN model) can recognize medicinal plant photos at three different levels of image definition, 64 64, 128 128, and 256 256 pixels. As a result, combining image processing with the suggested CNN algorithm is a productive replacement for conventional approaches.

2.4 Relevance of the project

The image recognition based system Plantify has immense relevance today because the ease of identification needed is significant in agriculture, environmental science, education, and herbal science. The increase in global interest regarding plant harvest diversity, sustainable agriculture, and green technology makes efficient plant identification critical for researchers, farmers, botanists, and nature enthusiasts. Yet, the traditional identification processes involve expert level knowledge and extensive field experience which doesn’t serve the general public or students.

This project tackles these issues by employing modern AI technologies like EfficientNet and ResNet to achieve high accuracy in image classification of plants. To enable friendly user experience, a Flask based web interface has been built. In addition, to encourage wider use Plantify offers a searchable plant catalouge for plant information retrieval, multi language support, and a knowledge database containing diverse plants and their features, medicinal uses, and ecological importance which enhance Plantify's interactivity.

CHAPTER 3

SYSTEM ANALYSIS

3.1 Requirement Analysis

Requirement analysis is the process of identifying, gathering, and defining user and stakeholder needs for a system. It ensures the system meets business goals and functions as intended. Techniques like interviews, surveys, and document analysis help collect functional (system behavior) and non-functional (performance, security) requirements. Proper documentation, such as SRS, ensures clear communication, reducing misunderstandings and streamlining development**.**

3.1.1 Functional Requirements

Functional Requirements define what a system should do. They specify system behavior, features, and functionalities

1.Medicinal Plant Identification

The system must be able to identify medicinal plants accurately based on input images.

2.Extended CNN-Based Classification

The model should implement an Extended Convolutional Neural Network (CNN) to classify medicinal plants effectively.

3.Handling Image Variability

The system should account for variations in lighting, angles, and background clutter to improve recognition accuracy.

4.Training Data Utilization

The model must leverage diverse and extensive training datasets to minimize biases and enhance classification performance.

1.Integration of Specialized Architectures

The system should incorporate advanced architectures such as ResNet and DenseNet to improve feature extraction and classification accuracy.

2.Training Data Augmentation

The model should include data augmentation techniques to enhance the diversity of training samples and improve robustness.

3.Performance Evaluation

The system must compare the accuracy of the Extended CNN model with traditional techniques to validate its effectiveness.

4.User Input and Image Processing

The system should accept images of medicinal plants as input, preprocess them, and classify them accordingly.

5.Output Generation

The system must provide accurate classification results, including plant names and confidence scores, for practical applications.

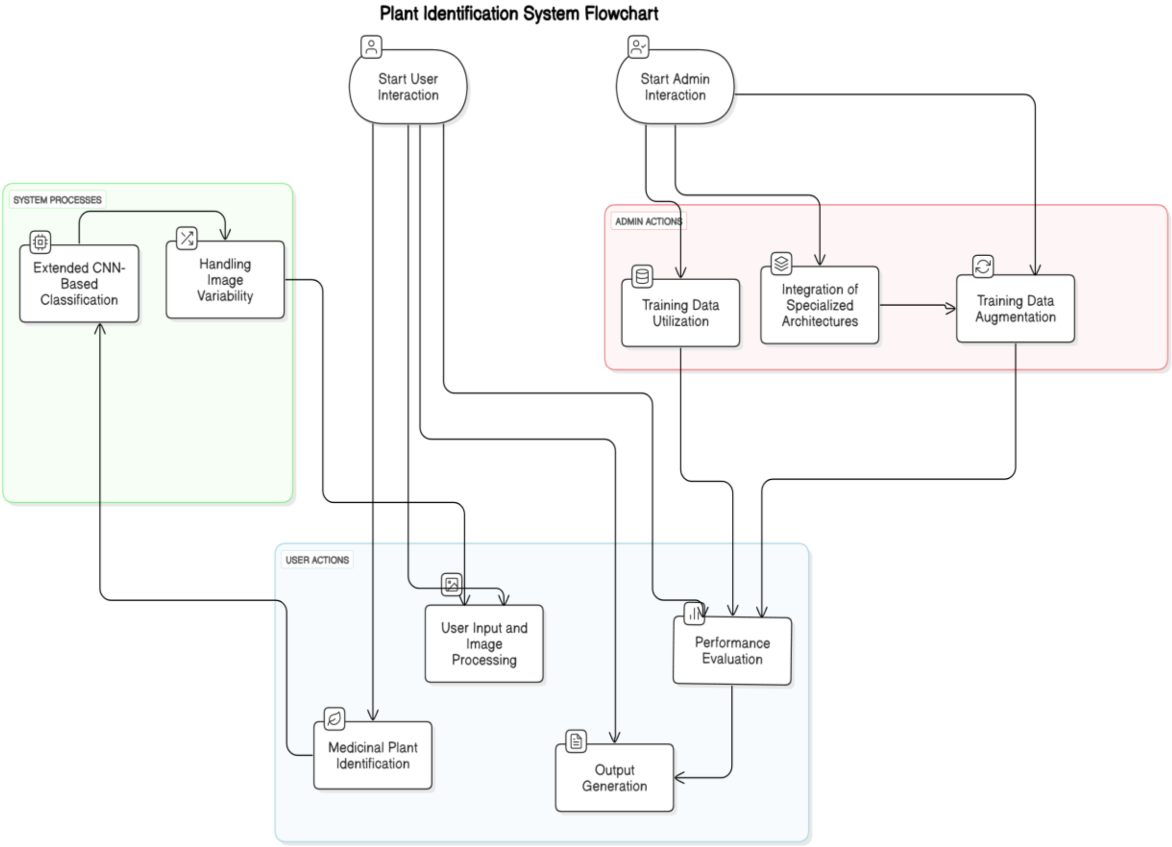

Plant Identification System Flowchart
Start User Interaction
Start Admin Interaction
SYSTEM PROCESSES
Extended CNN-Based Classification
Handling Image Variability
ADMIN ACTIONS
Training Data Utilization
Integration of Specialized Architectures
Training Data Augmentation
USER ACTIONS
User Input and Image Processing
Performance Evaluation
Medicinal Plant Identification
Output Generation

Fig. 3.1 Functional requirements

3.1.2 Non-Functional Requirements

Non-Functional Requirements define how a system should perform. They focus on system quality, performance, and constraints. These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements.

- **Performance:**
  The system should ensure high detection speed and accuracy, minimizing false positives and false negatives while processing large datasets efficiently.

- **Scalability:**

  The system must be able to scale and handle increased traffic, ensuring real-time phishing detection for large volumes of websites without performance degradation.

- **Security:**

  The system must ensure secure handling of sensitive user data and prevent tampering with the ML models or detection process to maintain the integrity of the results.

- **Reliability:**

  The system should ensure consistent uptime and availability, especially during high-demand periods, to reliably detect phishing websites at all times.

- **Usability:**

  The system should have an intuitive interface for end-users, providing clear, understandable alerts and explanations for detected phishing sites without requiring deep technical knowledge.

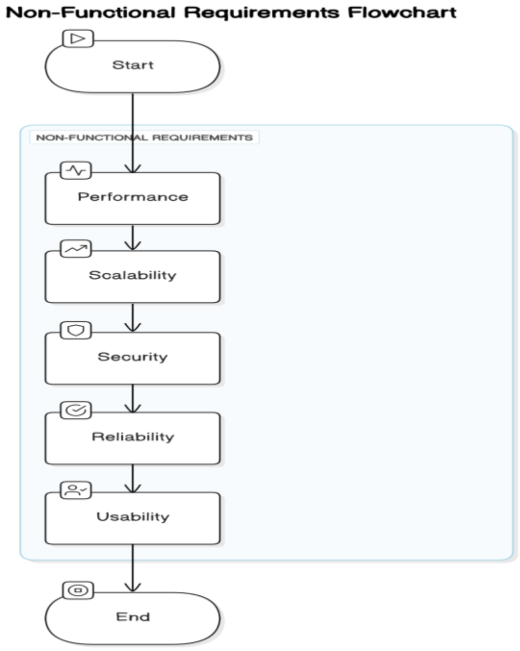

Non-Functional Requirements Flowchart
Start
NON-FUNCTIONAL REQUIREMENTS
Performance
Scalability
Security
Reliability
Usability
End

Fig. 3.2 Non functional requirements

3.1.3 System Requirements

Software Requirements

Software’s	: Python 3.6 or high version
IDE	: Visual Studio Code
Framework	: Deep Learning Frame works (TensorFlow)

Hardware Requirements

Operating system	: Windows 7 or 7+
RAM	: 8 GB or more
Hard disc or SSD	: More than 500 GB
Processor	: Intel 3rd generation or high or Ryzen with 8 GB Ram

3.2 Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

3.2.1 Economical Feasibility

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

3.2.2 Technical Feasibility

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

3.2.3 Social Feasibility

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

3.2.4 Operational Feasibility

Operational feasibility focuses on whether a system can function effectively within an organization's existing workflows and processes. It evaluates if the system meets business needs, integrates with current operations, and enhances productivity. For instance, an inventory management system in a retail store should streamline stock tracking without disrupting daily operations. If a system is operationally feasible, it ensures smooth implementation and efficient performance.

3.3 Proposed System Overview

The proposed system for automating the identification process of medicinal plants and raw materials through image processing and Deep learning aims to address the limitations of existing systems while leveraging cutting-edge technologies to achieve accurate and efficient plant recognition.

The system will begin with the compilation of a comprehensive dataset containing images of various medicinal plant species and raw materials, ensuring diversity in morphology, habitat, and geographic location. Preprocessing techniques such as standardization, noise removal, and data augmentation will be applied to enhance the quality and robustness of the dataset. Machine learning models, including state-of-the-art architectures like EfficinetNet and ResNet, will be trained on the pre-processed dataset to learn features relevant to plant classification.

Model selection and hyperparameter optimization will be conducted to maximize performance and generalization capabilities. The developed system will include a user-friendly interface that allows users to upload images and receive real-time predictions on plant identification. Continuous monitoring and updates will ensure the system's accuracy, reliability, and adaptability to changes in plant taxonomy, environmental conditions, and user feedback.

By integrating advanced image processing techniques with machine learning algorithms, the proposed system aims to provide a valuable tool for researchers, healthcare professionals, and conservationists in accurately identifying medicinal plants and raw materials for various applications, including drug discovery, biodiversity monitoring, and sustainable resource management.

Advantages

1.Enhanced Accuracy: By leveraging state-of-the-art machine learning algorithms and preprocessing techniques, the proposed system aims to achieve higher accuracy in plant identification. Advanced models like EfficientNet and ResNet have demonstrated superior performance in image classification tasks, leading to more reliable identification results.

2.Improved Coverage: The comprehensive dataset compiled for training the machine learning models ensures a wide coverage of medicinal plant species and raw materials. This diversity enhances the system's ability to accurately identify plants across different morphologies, habitats, and geographic regions, addressing the limitation of limited coverage in existing systems.

1.User-Friendly Interface: The development of a user-friendly interface facilitates seamless interaction with the automated identification system. Users, including researchers, healthcare professionals, and enthusiasts, can easily upload images and receive real-time predictions on plant identification without requiring specialized botanical knowledge or technical expertise.

1.Continuous Updates and Maintenance: The proposed system incorporates mechanisms for continuous updates and maintenance, ensuring that the system remains up-to-date with changes in plant taxonomy, environmental conditions, and user feedback. Regular updates to the dataset and model retraining help to improve accuracy and reliability over time, addressing the challenge of outdated information in existing systems.

1.Scalability and Adaptability: With its scalable architecture and adaptable design, the proposed system can accommodate a growing volume of data and users. It can be easily extended to support additional plant species, features, and functionalities, making it suitable for various applications and research domains.

2.Potential for Integration: The proposed system has the potential to be integrated with existing platforms and databases, further enhancing its utility and interoperability. Integration with other systems, such as biodiversity databases or mobile applications, can broaden its reach and impact in botanical research, conservation, and practice.

CHAPTER 4

SYSTEM DESIGN

4.1 System Architecture

System architecture is a conceptual model that describes the structure and behavior of multiple components and subsystems.

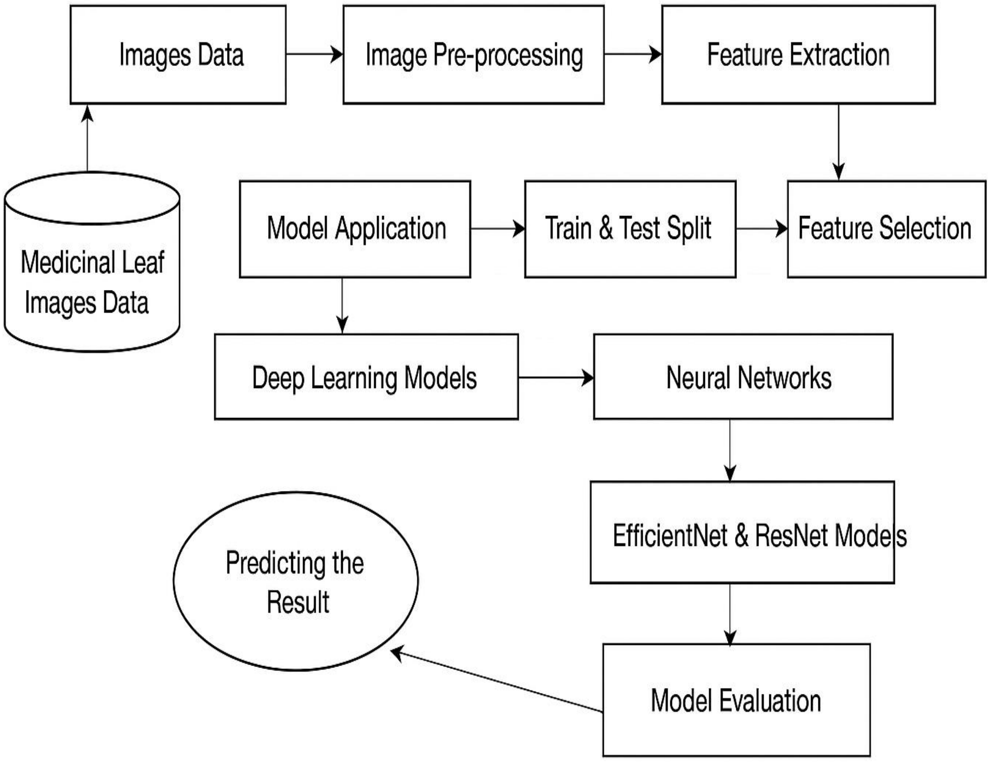


Fig. 4.1 System Architecture

4.2 Input Design

The design encompasses various aspects related to user inputs and their objectives, as outlined below:

User-Friendly Interface: The primary objective of the system input design is to create a user-friendly interface that enables users to interact with the system effortlessly. This involves designing intuitive user interfaces with clear navigation, well-organized layouts, and user-friendly controls for uploading images and initiating plant identification tasks.

Image Upload: The system allows users to upload images of medicinal plants and raw materials for identification. The input design should support various file formats (e.g., JPEG, PNG) and provide mechanisms for uploading single or multiple images conveniently.

Real-Time Feedback: The system provides real-time feedback to users during the image upload process, indicating the progress of the identification task and any errors or issues encountered. This objective ensures that users are informed about the status of their requests and can take appropriate actions if necessary**.**

Error Handling: Effective error handling mechanisms are integrated into the input design to detect and address common errors or invalid inputs from users. Clear error messages and prompts are displayed to guide users in resolving issues and resubmitting their requests.

Metadata Collection: In addition to image data, the system may collect metadata associated with uploaded images, such as timestamps, geographic location, and user-provided annotations. The input design includes fields or prompts for capturing relevant metadata to enrich the dataset and improve the accuracy of plant identification**.**

Accessibility: The input design considers accessibility requirements to ensure that users with diverse needs and abilities can interact with the system effectively. This may involve providing alternative input methods, such as voice commands or keyboard shortcuts, and adhering to accessibility standards for web-based interfaces.

Security Measures: Security measures are implemented in the input design to protect user data and prevent unauthorized access. This includes mechanisms for securely handling and transmitting images, encrypting sensitive information, and implementing user authentication and authorization controls.

Scalability: The input design is scalable to accommodate varying levels of user demand and system load. This involves optimizing input processing workflows, leveraging asynchronous processing techniques, and provisioning resources dynamically to handle spikes in user activity.

4.3 Output Design

The design encompasses various aspects related to presenting output information and achieving specific objectives, as outlined below:

Identification Results Display: The primary objective of the output system design is to display the identification results to users in a visually appealing and comprehensible format. This involves presenting the identified plant species, confidence scores or probabilities, and any additional information or metadata associated with the identification.

Confidence Levels and Uncertainty: The output design may include confidence levels or uncertainty estimates associated with the identification results to convey the reliability or uncertainty of the classification. This objective helps users interpret the identification results and make informed decisions based on the level of confidence.

User Interaction: The output system design enables user interaction with the identification results, allowing users to explore additional information, provide feedback, or take further actions based on the results. This objective enhances user engagement and satisfaction with the system.

4.4 UML Diagrams

UML stands for Unified Modelling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed and was created by, the Object Management Group.

UML is a very important part of developing object-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS: The Primary goals in the design of the UML are as follows:

1. Provide users with a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extensibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of the OO tools market.
6. Support higher-level development concepts such as collaborations, frameworks, patterns, and components.
7. Integrate best practices.

4.4.1 Use Case Diagram

A use-case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis.

Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases (Fig. 4.2).

The main purpose of a use case diagram is to show what system functions are performed for which actor. The roles of the actors in the system can be depicted system can be depicted.

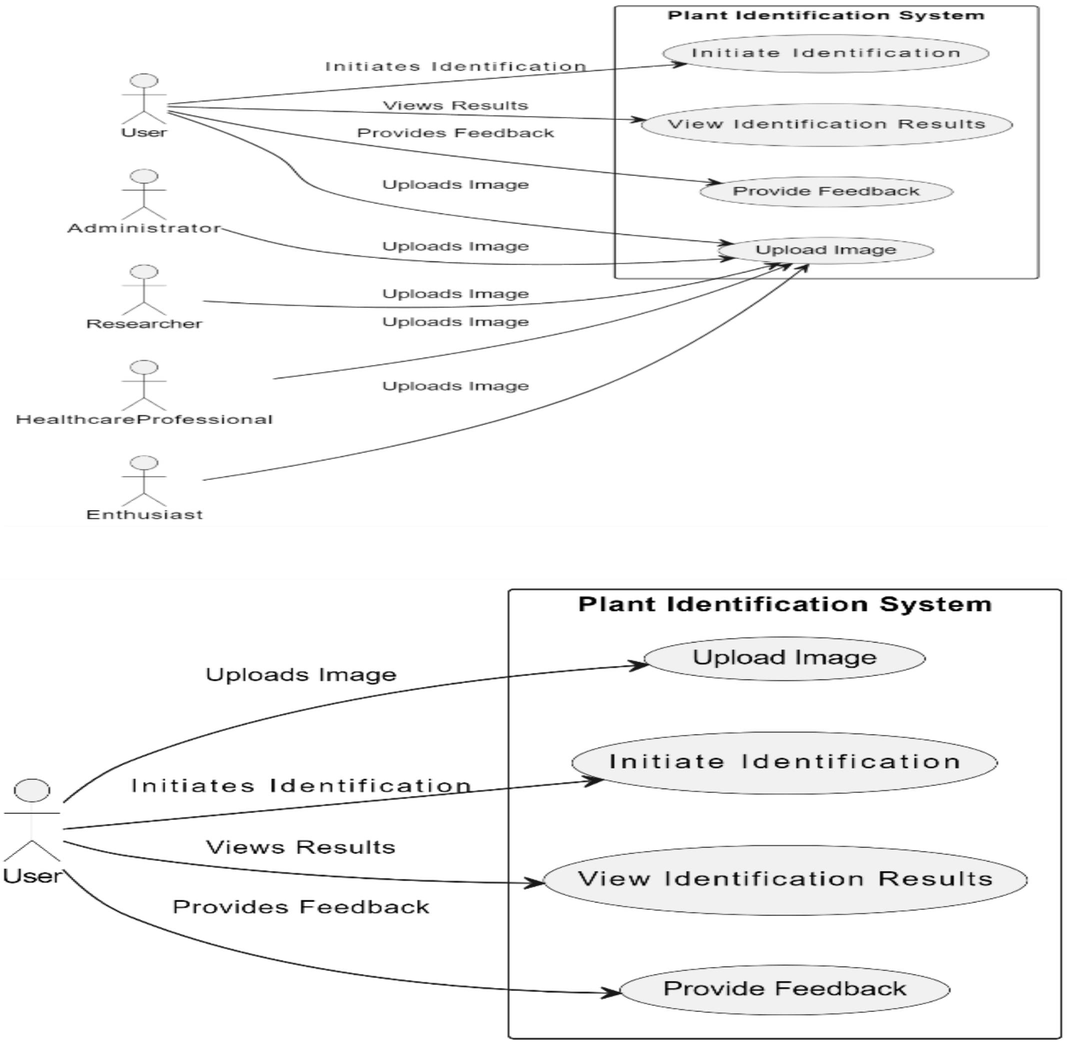


Fig. 4.2 Use Case Diagram

4.4.2 Class Diagram

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, attributes, operations (or methods), and the relationships among the classes. It explains which class contains information (Fig4.3).

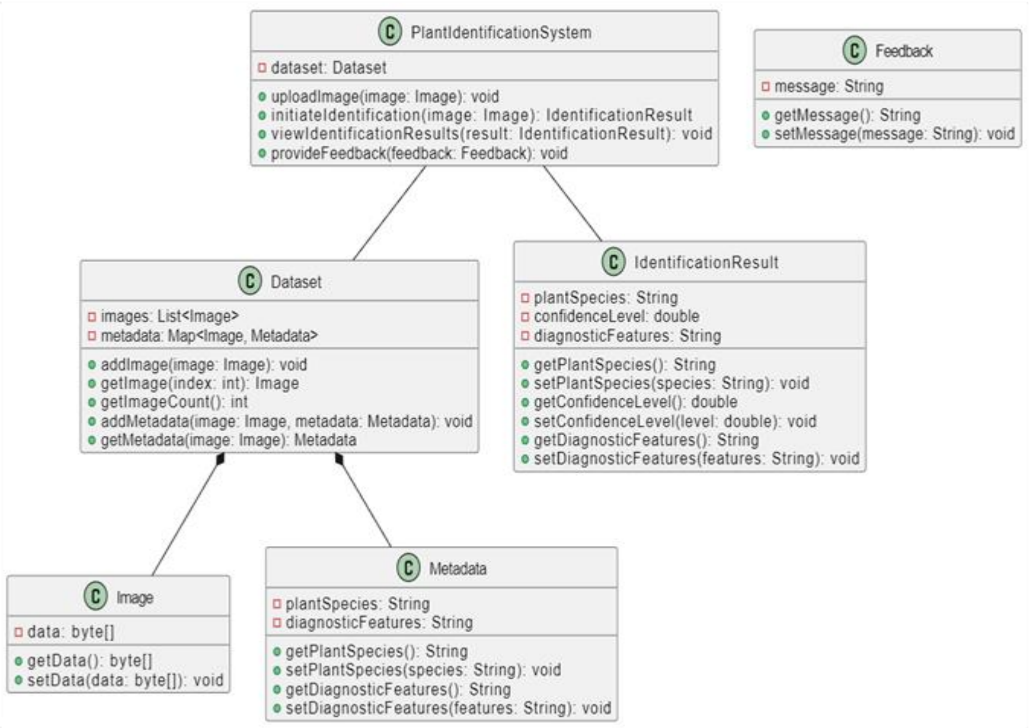

C PlantIdentificationSystem
dataset: Dataset
uploadImage(image: Image): void
initiateIdentification(image: Image): IdentificationResult
viewIdentificationResults(result: IdentificationResult): void
provideFeedback(feedback: Feedback): void
C Feedback
message: String
getMessage(): String
setMessage(message: String): void
C Dataset
images: List<Image>
metadata: Map<Image, Metadata>
addImage(image: Image): void
getImage(index: int): Image
getImageCount(): int
addMetadata(image: Image, metadata: Metadata): void
getMetadata(image: Image): Metadata
C IdentificationResult
plantSpecies: String
confidenceLevel: double
diagnosticFeatures: String
getPlantSpecies(): String
setPlantSpecies(species: String): void
getConfidenceLevel(): double
setConfidenceLevel(level: double): void
getDiagnosticFeatures(): String
setDiagnosticFeatures(features: String): void
C Image
data: byte[]
getData(): byte[]
setData(data: byte[]): void
C Metadata
plantSpecies: String
diagnosticFeatures: String
getPlantSpecies(): String
setPlantSpecies(species: String): void
getDiagnosticFeatures(): String
setDiagnosticFeatures(features: String): void

Fig. 4.3 Class diagram

4.4.3 Sequence Diagram

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams (Fig. 4.4).

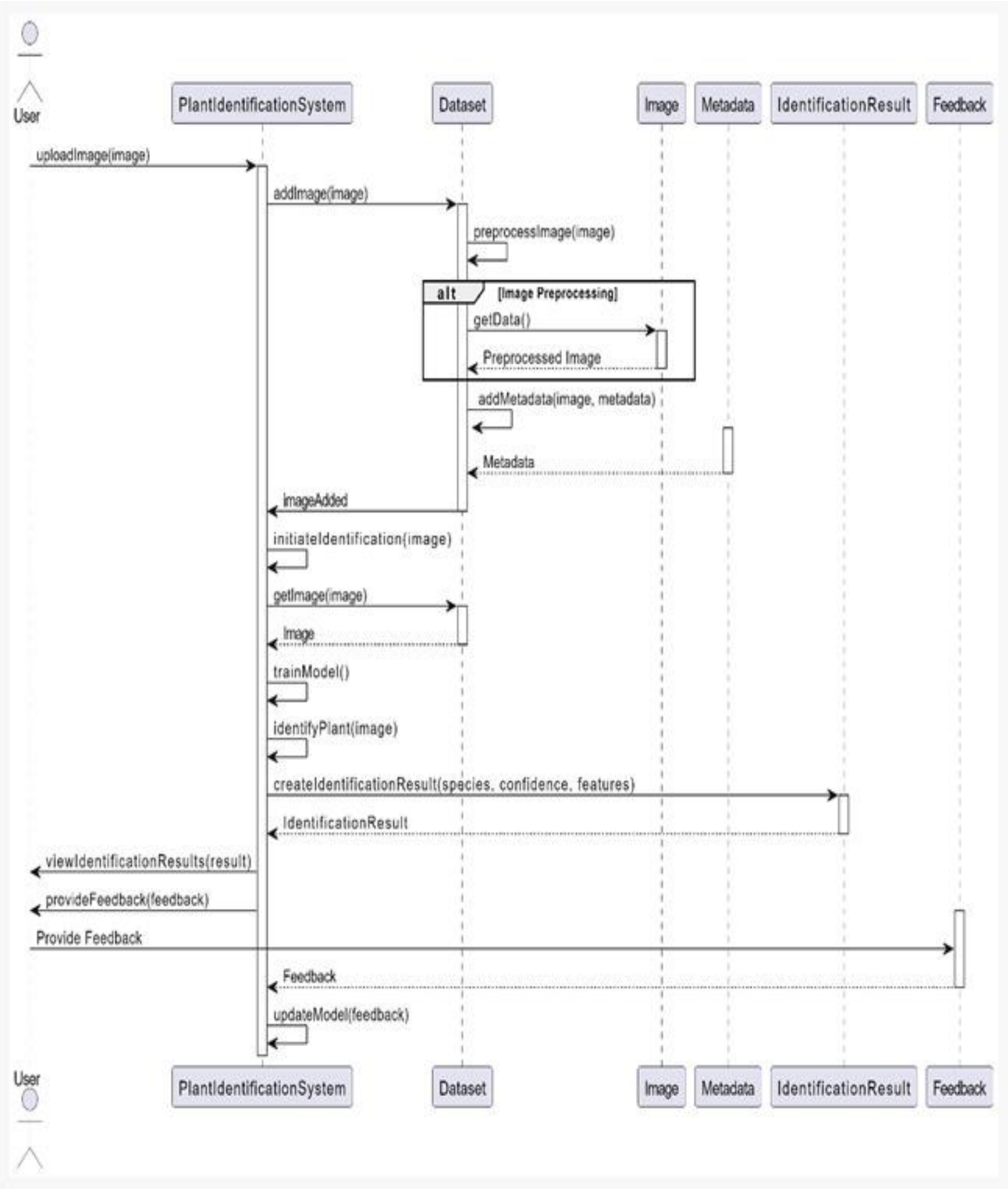


Fig. 4.4 Sequence Diagram

4.4.4 Collaboration Diagram

In the collaboration dsiagram, the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram(fig.4.5). The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.

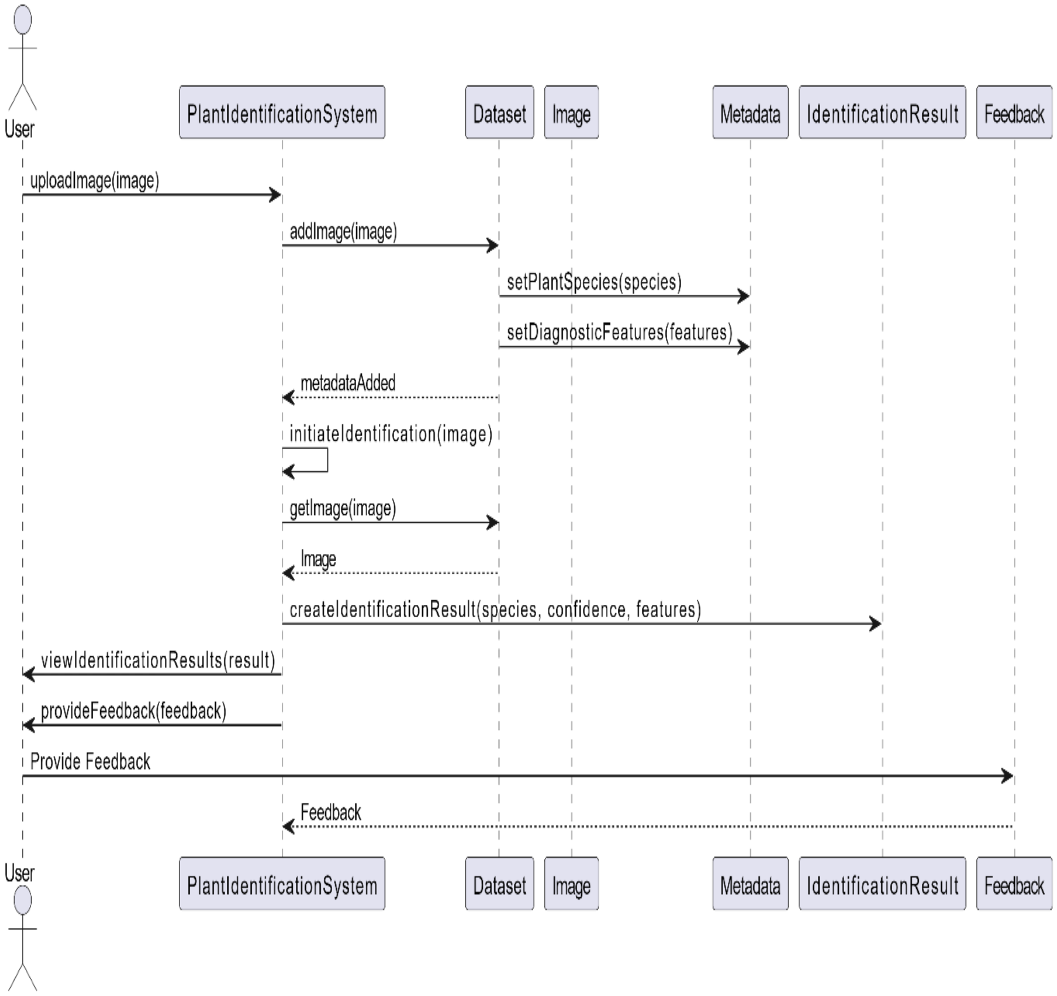

User
PlantIdentificationSystem
Dataset
Image
Metadata
IdentificationResult
Feedback
uploadImage(image)
addImage(image)
setPlantSpecies(species)
setDiagnosticFeatures(features)
metadataAdded
initiateIdentification(image)
getImage(image)
Image
createIdentificationResult(species, confidence, features)
viewIdentificationResults(result)
provideFeedback(feedback)
Provide Feedback
Feedback
User
PlantIdentificationSystem
Dataset
Image
Metadata
IdentificationResult
Feedback

Fig. 4.5 Collaboration diagram

4.4.5 Component Diagram

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical **c**omponents in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required function is covered by planned development (Fig. 4.6).

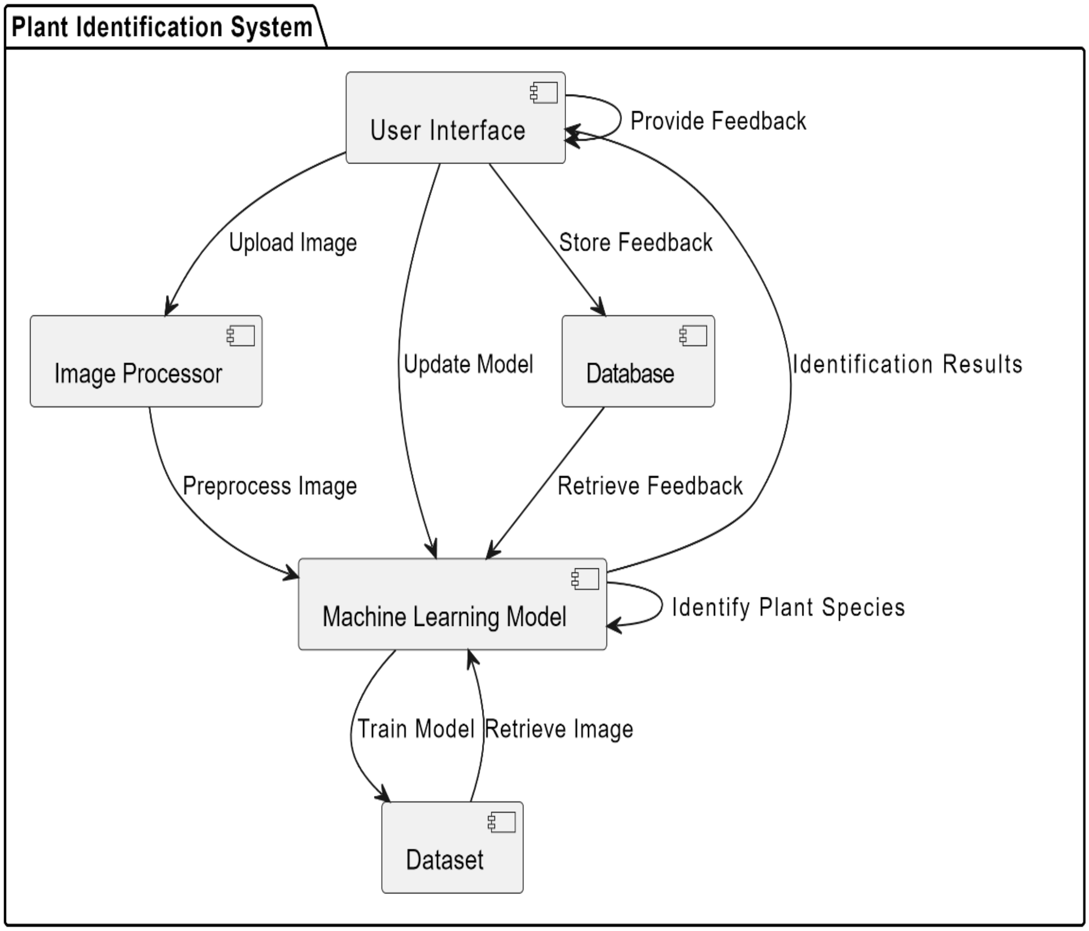


Fig. 4.6 Component Diagram

4.4.6 DFD Diagram

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole (Fig. 4.7).

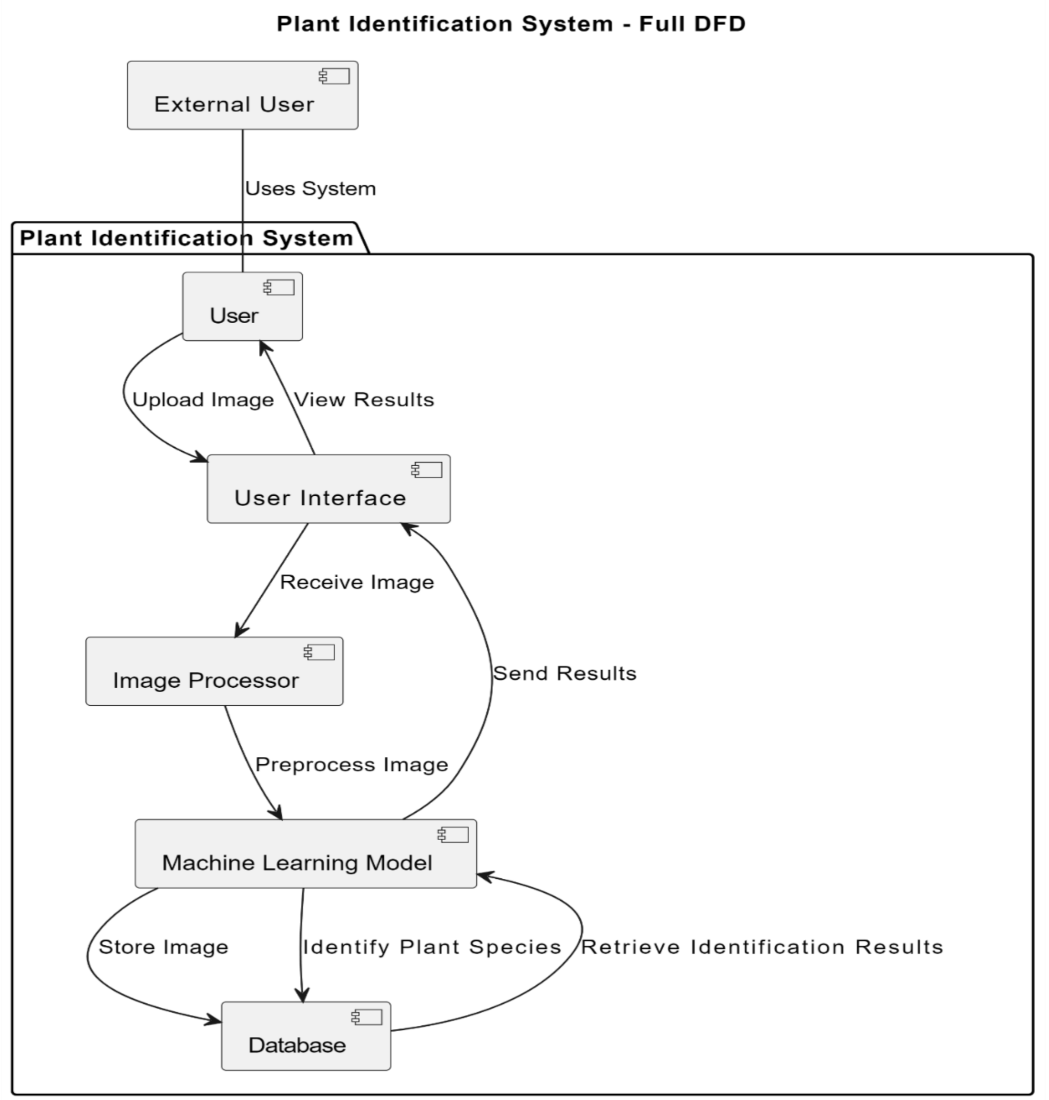


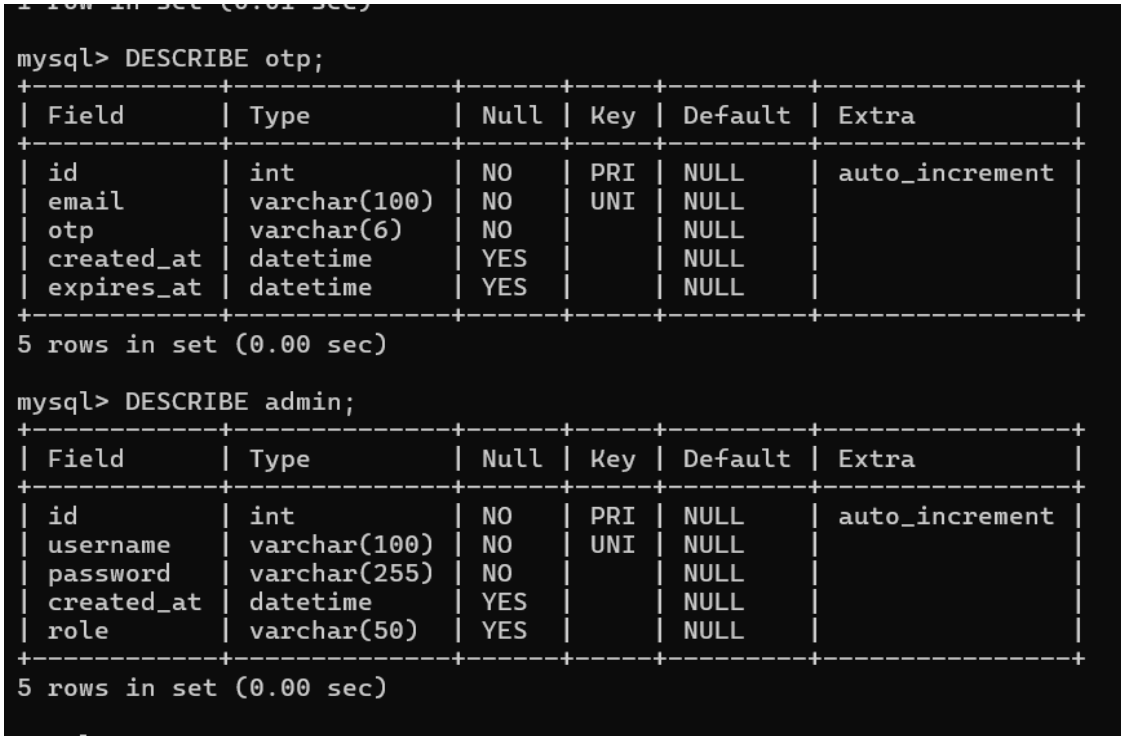
Fig. 4.7 DFD

4.5 Database Design

Database design is a crucial step in the development of any software system, ensuring efficient data storage, retrieval, and management. It involves structuring the database to optimize performance, maintain integrity, and support scalability.

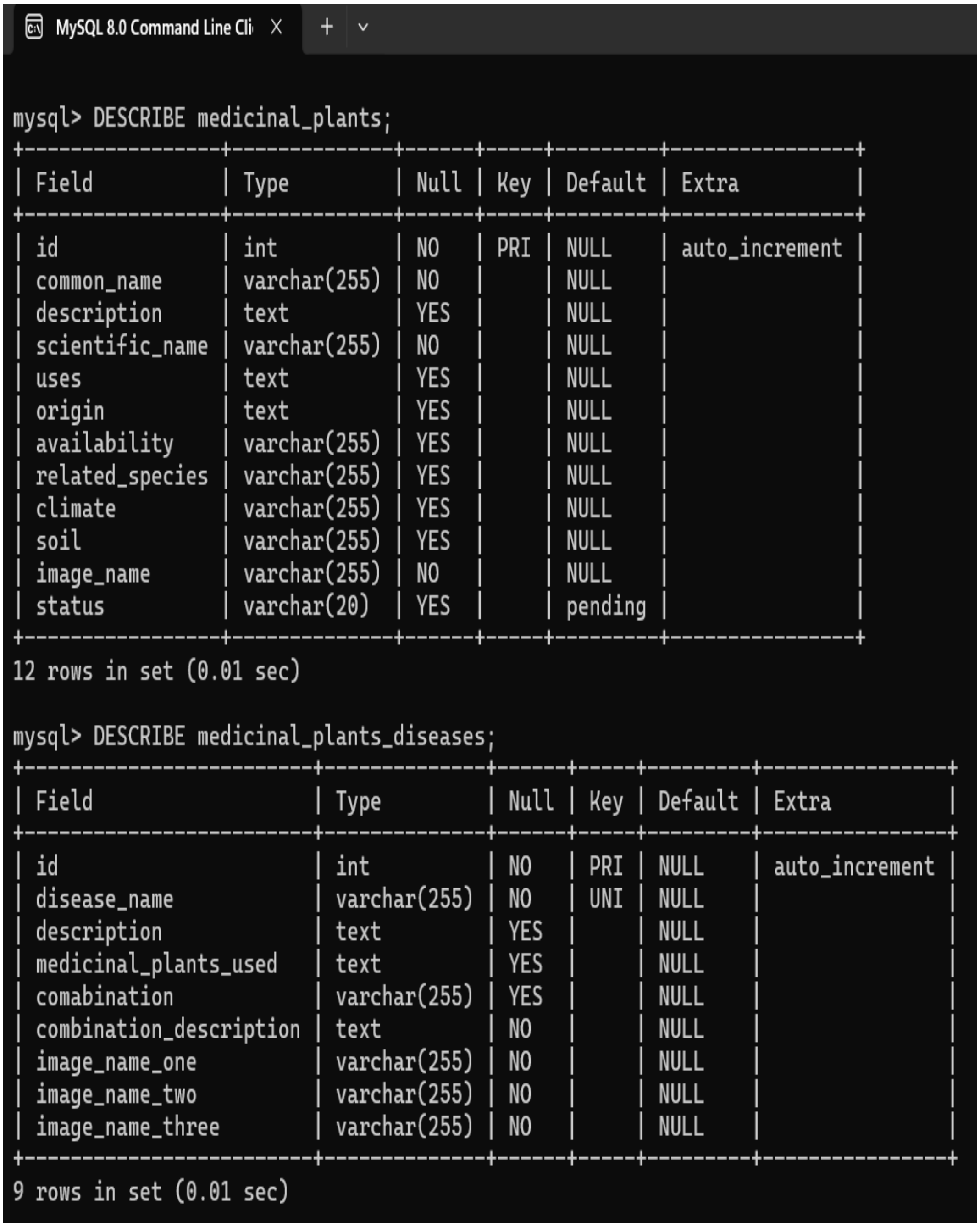
4.5.1 Tables Created for Plantify Project :

- User
- Admin
- OTP
- Prediction
- Medicinal_plants
- Medicinal_plants_diseases



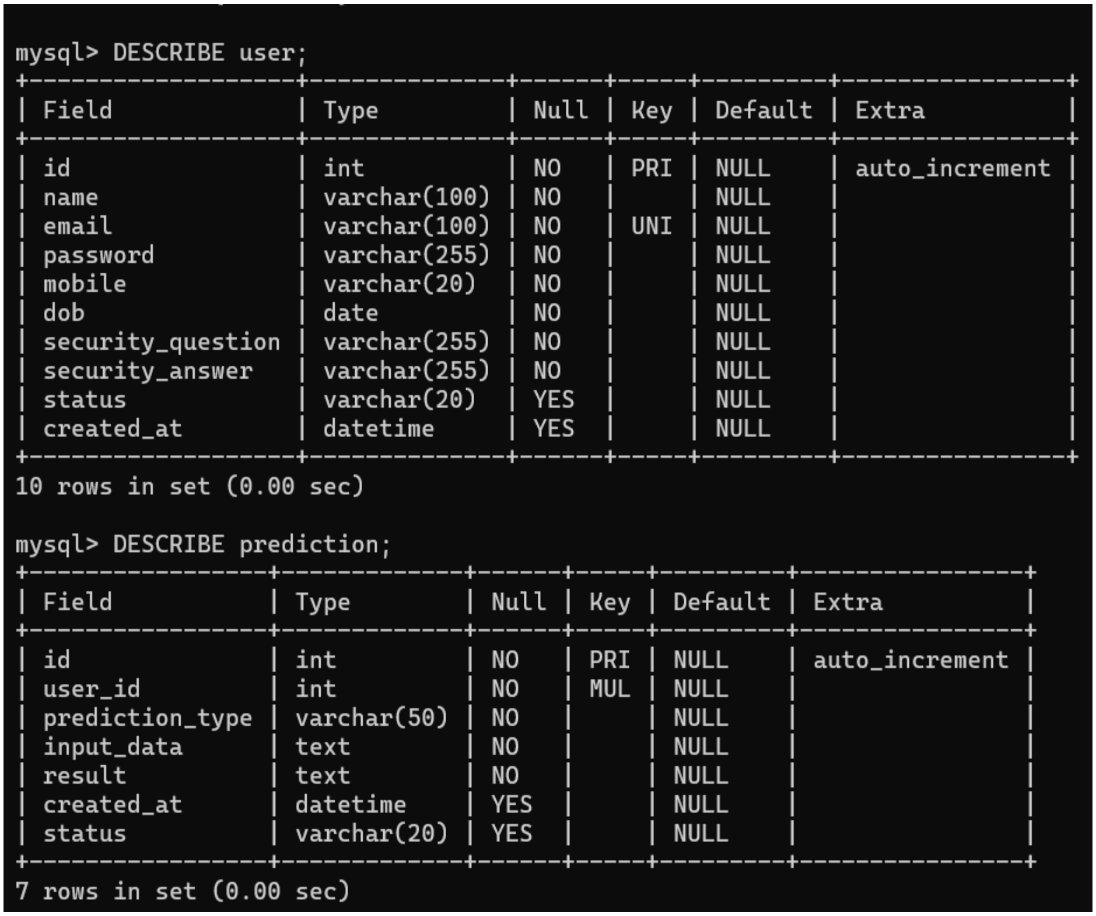
```
1 row in set (0.01 sec)

mysql> DESCRIBE otp;
+------------+--------------+------+-----+---------+----------------+
| Field      | Type         | Null | Key | Default | Extra          |
+------------+--------------+------+-----+---------+----------------+
| id         | int          | NO   | PRI | NULL    | auto_increment |
| email      | varchar(100) | NO   | UNI | NULL    |                |
| otp        | varchar(6)   | NO   |     | NULL    |                |
| created_at | datetime     | YES  |     | NULL    |                |
| expires_at | datetime     | YES  |     | NULL    |                |
+------------+--------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)

mysql> DESCRIBE admin;
+------------+--------------+------+-----+---------+----------------+
| Field      | Type         | Null | Key | Default | Extra          |
+------------+--------------+------+-----+---------+----------------+
| id         | int          | NO   | PRI | NULL    | auto_increment |
| username   | varchar(100) | NO   | UNI | NULL    |                |
| password   | varchar(255) | NO   |     | NULL    |                |
| created_at | datetime     | YES  |     | NULL    |                |
| role       | varchar(50)  | YES  |     | NULL    |                |
+------------+--------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

Fig. 4.8 Admin



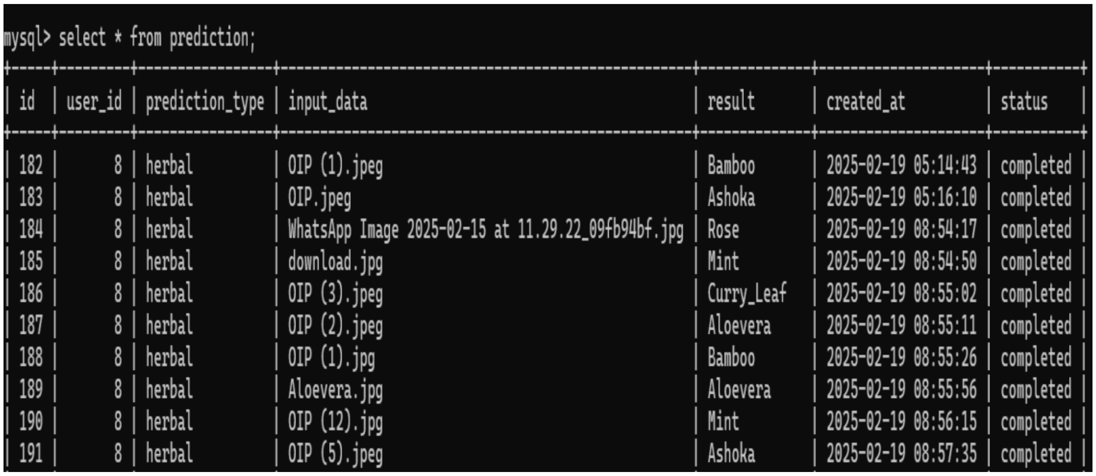
```
mysql> DESCRIBE medicinal_plants;
+-----------------+--------------+------+-----+---------+----------------+
| Field           | Type         | Null | Key | Default | Extra          |
+-----------------+--------------+------+-----+---------+----------------+
| id              | int          | NO   | PRI | NULL    | auto_increment |
| common_name     | varchar(255) | NO   |     | NULL    |                |
| description     | text         | YES  |     | NULL    |                |
| scientific_name | varchar(255) | NO   |     | NULL    |                |
| uses            | text         | YES  |     | NULL    |                |
| origin          | text         | YES  |     | NULL    |                |
| availability    | varchar(255) | YES  |     | NULL    |                |
| related_species | varchar(255) | YES  |     | NULL    |                |
| climate         | varchar(255) | YES  |     | NULL    |                |
| soil            | varchar(255) | YES  |     | NULL    |                |
| image_name      | varchar(255) | NO   |     | NULL    |                |
| status          | varchar(20)  | YES  |     | pending |                |
+-----------------+--------------+------+-----+---------+----------------+
12 rows in set (0.01 sec)

mysql> DESCRIBE medicinal_plants_diseases;
+-------------------------+--------------+------+-----+---------+----------------+
| Field                   | Type         | Null | Key | Default | Extra          |
+-------------------------+--------------+------+-----+---------+----------------+
| id                      | int          | NO   | PRI | NULL    | auto_increment |
| disease_name            | varchar(255) | NO   | UNI | NULL    |                |
| description             | text         | YES  |     | NULL    |                |
| medicinal_plants_used   | text         | YES  |     | NULL    |                |
| comabination            | varchar(255) | YES  |     | NULL    |                |
| combination_description | text         | NO   |     | NULL    |                |
| image_name_one          | varchar(255) | NO   |     | NULL    |                |
| image_name_two          | varchar(255) | NO   |     | NULL    |                |
| image_name_three        | varchar(255) | NO   |     | NULL    |                |
+-------------------------+--------------+------+-----+---------+----------------+
9 rows in set (0.01 sec)
```

Fig. 4.9 Medicinal_Plants



```
mysql> DESCRIBE user;
+-------------------+--------------+------+-----+---------+----------------+
| Field             | Type         | Null | Key | Default | Extra          |
+-------------------+--------------+------+-----+---------+----------------+
| id                | int          | NO   | PRI | NULL    | auto_increment |
| name              | varchar(100) | NO   |     | NULL    |                |
| email             | varchar(100) | NO   | UNI | NULL    |                |
| password          | varchar(255) | NO   |     | NULL    |                |
| mobile            | varchar(20)  | NO   |     | NULL    |                |
| dob               | date         | NO   |     | NULL    |                |
| security_question | varchar(255) | NO   |     | NULL    |                |
| security_answer   | varchar(255) | NO   |     | NULL    |                |
| status            | varchar(20)  | YES  |     | NULL    |                |
| created_at        | datetime     | YES  |     | NULL    |                |
+-------------------+--------------+------+-----+---------+----------------+
10 rows in set (0.00 sec)

mysql> DESCRIBE prediction;
+-----------------+-------------+------+-----+---------+----------------+
| Field           | Type        | Null | Key | Default | Extra          |
+-----------------+-------------+------+-----+---------+----------------+
| id              | int         | NO   | PRI | NULL    | auto_increment |
| user_id         | int         | NO   | MUL | NULL    |                |
| prediction_type | varchar(50) | NO   |     | NULL    |                |
| input_data      | text        | NO   |     | NULL    |                |
| result          | text        | NO   |     | NULL    |                |
| created_at      | datetime    | YES  |     | NULL    |                |
| status          | varchar(20) | YES  |     | NULL    |                |
+-----------------+-------------+------+-----+---------+----------------+
7 rows in set (0.00 sec)
```

Fig. 4.9(a) Medicinal_Describe



```
mysql> select * from prediction;
+-----+---------+-----------------+----------------------------------------------------+------------+---------------------+-----------+
| id  | user_id | prediction_type | input_data                                         | result     | created_at          | status    |
+-----+---------+-----------------+----------------------------------------------------+------------+---------------------+-----------+
| 182 |       8 | herbal          | OIP (1).jpeg                                       | Bamboo     | 2025-02-19 05:14:43 | completed |
| 183 |       8 | herbal          | OIP.jpeg                                           | Ashoka     | 2025-02-19 05:16:10 | completed |
| 184 |       8 | herbal          | WhatsApp Image 2025-02-15 at 11.29.22_09fb94bf.jpg | Rose       | 2025-02-19 08:54:17 | completed |
| 185 |       8 | herbal          | download.jpg                                       | Mint       | 2025-02-19 08:54:50 | completed |
| 186 |       8 | herbal          | OIP (3).jpeg                                       | Curry_Leaf | 2025-02-19 08:55:02 | completed |
| 187 |       8 | herbal          | OIP (2).jpeg                                       | Aloevera   | 2025-02-19 08:55:11 | completed |
| 188 |       8 | herbal          | OIP (1).jpg                                        | Bamboo     | 2025-02-19 08:55:26 | completed |
| 189 |       8 | herbal          | Aloevera.jpg                                       | Aloevera   | 2025-02-19 08:55:56 | completed |
| 190 |       8 | herbal          | OIP (12).jpg                                       | Mint       | 2025-02-19 08:56:15 | completed |
| 191 |       8 | herbal          | OIP (5).jpeg                                       | Ashoka     | 2025-02-19 08:57:35 | completed |
```

Fig 4.9(b) Prediction

CHAPTER 5

IMPLEMENTATION

5.1 Programming Languages and Technologies used

Python Flask

It is a lightweight and versatile web framework that simplifies the process of building web applications in Python. With its minimalist design and easy-to-use syntax, Flask offers developers the flexibility to create web services and APIs quickly and efficiently. Flask's modular structure allows for easy integration with other Python libraries and frameworks, making it suitable for a wide range of web development tasks. By leveraging Flask, Plantify can implement robust server-side functionality, handle HTTP requests, and serve dynamic web content with ease, providing a seamless user experience.

HTML, CSS, and JavaScript

1. Description: HTML (HyperText Markup Language) provides the structure, CSS (Cascading Style Sheets) defines the design, and JavaScript enables dynamic behavior in web pages.

2. Usage: HTML is used for structuring web pages, CSS for styling the user interface, and JavaScript for interactive features like dynamic plant searches, filtering.

5.2 Development tools and Environments

Flask

1. Description: Flask is a lightweight Python web framework that is used for building web applications. It provides essential features such as routing, request handling, and templating without the overhead of larger frameworks.

2. Usage: Manages the core application logic, handles different routes for plant information, herbal teas, plant identification, and user feedback. It integrates with MySQL via SQLAlchemy and uses Jinja2 for rendering dynamic web pages.

MySQL

1. Description: MySQL is an open-source relational database management system.

2. Usage: Stores data such as plant information, medicinal remedies, yoga poses, herbal tea recipes, user feedback, and more. It is connected to Flask using SQLAlchemy for data retrieval and storage.

TensorFlow & Keras

1. Description: TensorFlow is an open-source machine learning framework developed by Google, and Keras is a high-level neural network API. These tools are used for deep learning tasks.

2. Usage: Used for Medicinal Plant Identification through image uploads. The system uses machine learning models (like ResNet50 and EfficientNetB0) to classify plants based on leaf images.

Scikit-learn

1. Description: Scikit-learn is a Python library for machine learning, used for data mining and data analysis.

2. Usage: Handles tasks like splitting datasets into training and testing sets, as well as evaluating machine learning models using metrics like accuracy, precision, and recall.

Pillow

1. Description: Pillow is an image manipulation library for Python.

2. Usage: Pre-processes plant images for identification, including resizing, normalizing pixel values, and augmenting the dataset (e.g., image rotation, flipping).

Visual Studio Code

(VS Code) is a widely used integrated development environment (IDE) known for its lightweight yet powerful features. It provides a user-friendly interface with extensive support for various programming languages, including Python. With features such as syntax highlighting, IntelliSense code completion, and integrated Git control, VS Code streamlines the development process and enhances productivity. Its vast ecosystem of extensions allows developers to customize their workflow according to their preferences and project requirements, making it an ideal choice for developing Python applications.

5.3 Module wise implementation

5.3.1 Medicinal Plant Identification by Image Upload

Users can upload images of plants to identify them using the system's machine learning-based image recognition feature. The platform accurately identifies the plant species and displays relevant information on its medicinal properties and traditional uses, making plant identification accessible to all users. 6,900 leaf images for 80 species are used.Pillow resizes images to 224x224, normalizes, and augments (rotation, flipping).TensorFlow and Keras fine-tune ResNet50 and EfficientNetB0 for 20 and 5 epochs.Scikit-learn splits data into training and validation data set.Matplotlib/Seaborn visualize metrics (accuracy, precision).Flask backend processes user-uploaded images for identification.

5.3.2 Search and Filter Functionality

Users can efficiently find medicinal plants by using the search and filter options. They can filter plants based on criteria such as name, health benefits, or Ayurvedic classification. This feature ensures quick access to relevant information, improving the overall user experience. Flask queries MySQL for plants by name. Flask enables filtering and updates results.

5.3.3 Multi-Language Translation

To cater to a diverse audience, the platform supports multiple languages, including English, Telugu, and Hindi. Users can switch between languages to read plant information in their preferred language, ensuring a more inclusive and accessible experience.It is implemented using the Google Translate API.

5.3.4 Home remedies recommendation for common ailgnments

Based on the user’s input regarding their health concerns, the system recommends suitable home remedies derived from Ayurvedic classical texts. This feature promotes natural healing by providing personalized solutions aligned with traditional medicine practices.Plant-remedy mappings stored in MySQL.Flask queries remedies based on identified plants using SQLAlchemy.

5.3.5 Plant Information Module

This module offers comprehensive details about medicinal plants, including their names, scientific classifications, properties, benefits, and traditional uses. Users can visually explore the plants and gain insights into their significance in the Plantify system, enriching their knowledge of natural remedies.Flask retrieves plant data from MySQL using SQLAlchemy.

5.3.6 Contribute Plants

This module offers user to contribute to the research by providing details of the plants which arer not present in the research. The user should provide comprehensive details about medicinal plants, including their names, scientific classifications, properties, benefits, and traditional uses. Users can then visually explore the plants that they provided after the approval of the admin.

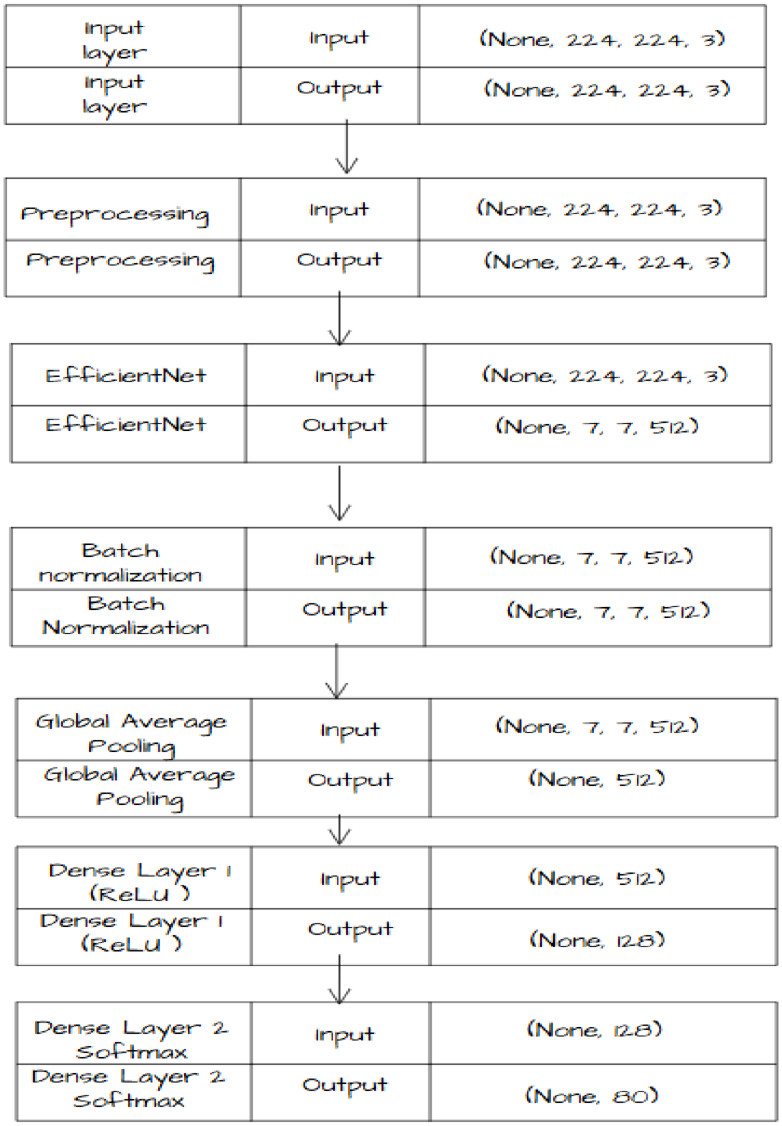
5.4 Algorithms and logic used

5.4.1 EfficientNet

EfficientNet is a picture classification demonstrate that can be depicted as shocking due to the culminate agreement it keeps up between exactness and effectiveness. Its most special include is innovative compound scaling procedure, which EfficientNet's maker Google Brain created.

Here are some key characteristics and components of the EfficientNet model:

Architecture: EfficientNet ideally combines demonstrate profundity (number of layers), width (number of channels per layer), and determination (input picture size). As restricted to old-fashioned models, EfficientNet at the same time scales all three with a shrewdly methodology.



Input layer	Input	(None, 224, 224, 3)
Input layer	Output	(None, 224, 224, 3)
Preprocessing	Input	(None, 224, 224, 3)
Preprocessing	Output	(None, 224, 224, 3)
EfficientNet	Input	(None, 224, 224, 3)
EfficientNet	Output	(None, 7, 7, 512)
Batch normalization	Input	(None, 7, 7, 512)
Batch Normalization	Output	(None, 7, 7, 512)
Global Average Pooling	Input	(None, 7, 7, 512)
Global Average Pooling	Output	(None, 512)
Dense Layer 1 (ReLu)	Input	(None, 512)
Dense Layer 1 (ReLu)	Output	(None, 128)
Dense Layer 2 Softmax	Input	(None, 128)
Dense Layer 2 Softmax	Output	(None, 80)

Fig. 5.1 Architecture of EfficientNet

Compound Scaling: EfficientNet uses a compound scaling method to balance width, depth, and resolution systematically. This allows it to achieve higher accuracy with fewer parameters compared to traditional CNN architectures.

Depth-wise Separable Convolutions: EfficientNet employs depth-wise separable convolutions, reducing the number of parameters and improving efficiency while maintaining feature extraction capability.

Squeeze-and-Excitation (SE) Blocks: EfficientNet integrates SE blocks to enhance feature learning by adaptively recalibrating channel-wise feature responses. This improves model performance without significantly increasing computational cost.

Pretrained Weights: EfficientNet models are pretrained on large datasets like ImageNet, making them highly useful for transfer learning. Fine-tuning with pretrained weights leads to faster convergence and improved accuracy on smaller datasets.

Activation Functions EfficientNet often uses the Swish activation function (shown in Fig. 5.1), which helps in smooth gradient flow and better representation learning the network and facilitate the learning of complex patterns and representations in the input data.

Overall, EfficientNet is a powerful and versatile convolutional neural network architecture that has been widely used for various computer vision tasks, including image classification, object detection, and image segmentation. Its modular design and efficient utilization of computational resources make it well-suited for real-world applications requiring high-performance image recognition capabilities.

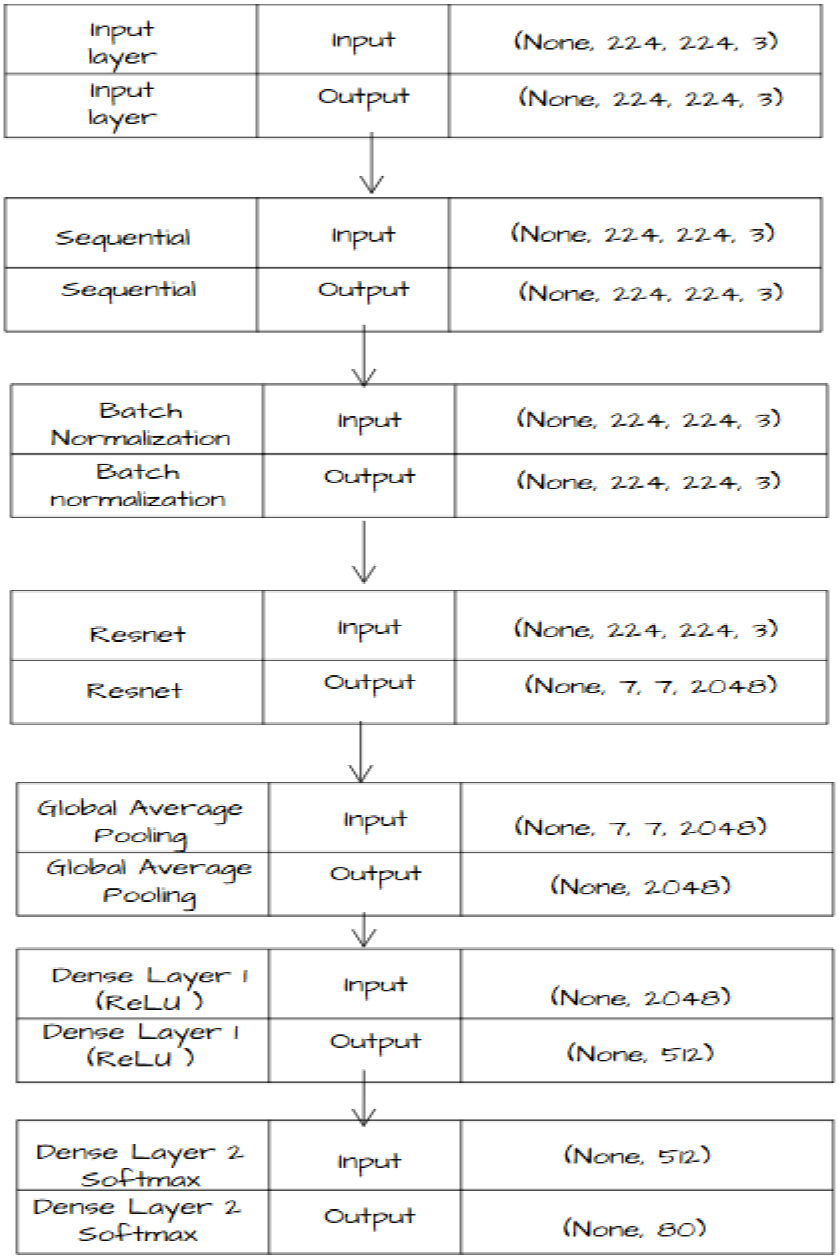
5.4.2 ResNet

Created by Microsoft in 2015, the ResNet (Leftover Arrange) engineering utilizes profound learning to fathom the vanishing angle issue frequently found in profound neural systems. With conventional profound systems, the more layers are included, the more prominent the execution decay gets to be which complicates the method of preparing.

Each remaining piece contains a set of multilayer convolution squares with a bypass personality alternate route that permits one or more layers to be skipped. Much obliged to the skip association, the show learns the leftover or contrast between the input and yield as restricted to coordinate mapping which incredibly streamlines the preparing handle for profound systems.

Variations of ResNet incorporate ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152, the number speaking to the full layers within the demonstrate. As anticipated, ResNet-50 has 50 layers counting convolutional, bunch normalization, ReLU actuation, and totally associated layers.

Diverse layers in a ResNet design have diverse part sizes. The exceptionally to begin with convolution layer utilizes a 7x7 part to capture tall level highlights with the beginning step, afterward detail extraction is done utilizing 3x3 bits within the last mentioned steps of convolutions (Fig. 5.2). Highlights are down examined through max pooling and stride-2 convolutional layers which capture vital information whereas diminishing the generally estimate.



Input layer	Input	(None, 224, 224, 3)
Input layer	Output	(None, 224, 224, 3)
Sequential	Input	(None, 224, 224, 3)
Sequential	Output	(None, 224, 224, 3)
Batch Normalization	Input	(None, 224, 224, 3)
Batch normalization	Output	(None, 224, 224, 3)
Resnet	Input	(None, 224, 224, 3)
Resnet	Output	(None, 7, 7, 2048)
Global Average Pooling	Input	(None, 7, 7, 2048)
Global Average Pooling	Output	(None, 2048)
Dense Layer 1 (ReLu)	Input	(None, 2048)
Dense Layer 1 (ReLu)	Output	(None, 512)
Dense Layer 2 Softmax	Input	(None, 512)
Dense Layer 2 Softmax	Output	(None, 80)

Fig. 5.2 ResNet Architecture

CHAPTER 6

TESTING AND RESULTS

6.1 Testing Methodologies

Software testing methodologies define structured approaches to ensure the quality, reliability, and security of an application**.**

6.1.1 Unit Testing

Unit testing focuses on testing individual components or functions of the application in isolation. It ensures that each module works correctly before integration**.**

Example:

1**.**Testing Flask API endpoints (e.g., /upload_plant, /get_plant_details).

2.Validating the image processing function for plant identification.

6.1.2 Integration Testing

Integration testing checks if different modules of the system work together correctly. It ensures seamless interaction between backend, frontend, and database.

Example:

1.Testing interactions between the frontend and Flask API.

2.Verifying image upload, processing, and storage.

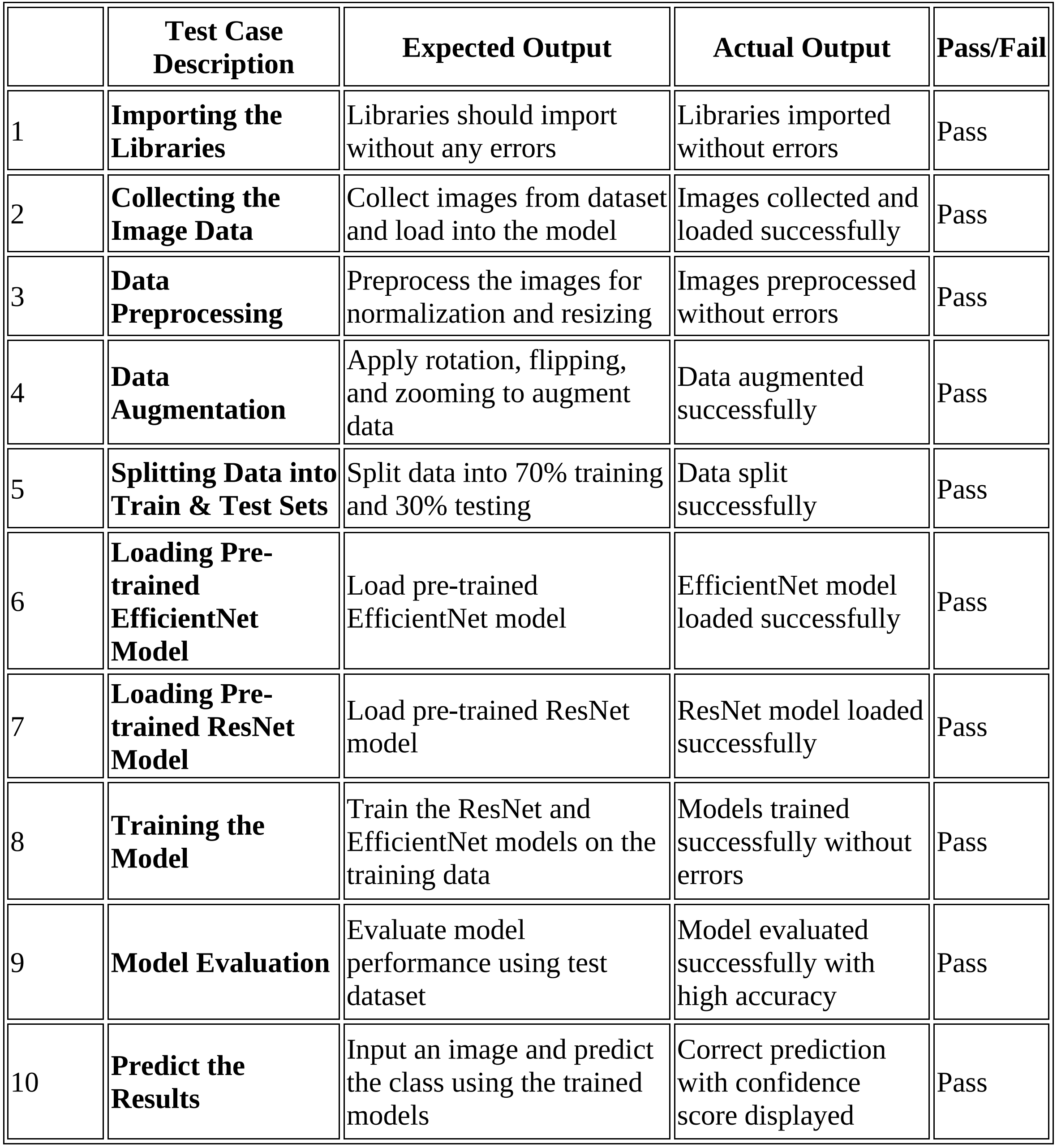
6.1.3 System Testing

System testing evaluates the entire application to verify that it meets all functional and non-functional requirements. It tests the system as a whole before deployment.

Example:

1.Ensuring plant identification works correctly for different image inputs

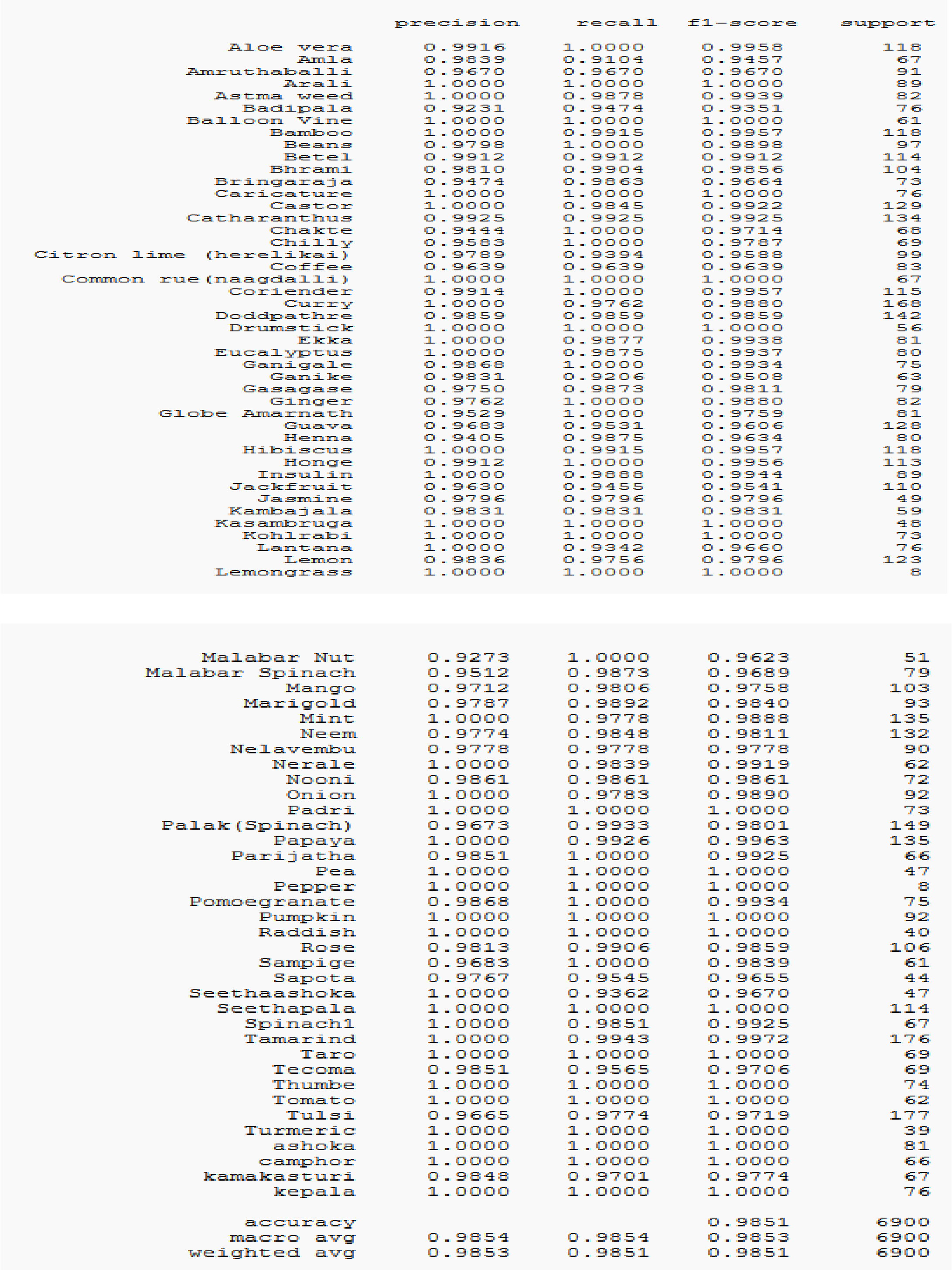
6.2 Testing Results



	Test Case Description	Expected Output	Actual Output	Pass/Fail
1	**Importing the Libraries**	Libraries should import without any errors	Libraries imported without errors	Pass
2	**Collecting the Image Data**	Collect images from dataset and load into the model	Images collected and loaded successfully	Pass
3	**Data Preprocessing**	Preprocess the images for normalization and resizing	Images preprocessed without errors	Pass
4	**Data Augmentation**	Apply rotation, flipping, and zooming to augment data	Data augmented successfully	Pass
5	**Splitting Data into Train & Test Sets**	Split data into 70% training and 30% testing	Data split successfully	Pass
6	**Loading Pre-trained EfficientNet Model**	Load pre-trained EfficientNet model	EfficientNet model loaded successfully	Pass
7	**Loading Pre-trained ResNet Model**	Load pre-trained ResNet model	ResNet model loaded successfully	Pass
8	**Training the Model**	Train the ResNet and EfficientNet models on the training data	Models trained successfully without errors	Pass
9	**Model Evaluation**	Evaluate model performance using test dataset	Model evaluated successfully with high accuracy	Pass
10	**Predict the Results**	Input an image and predict the class using the trained models	Correct prediction with confidence score displayed	Pass

Fig. 6.1 Testing Report

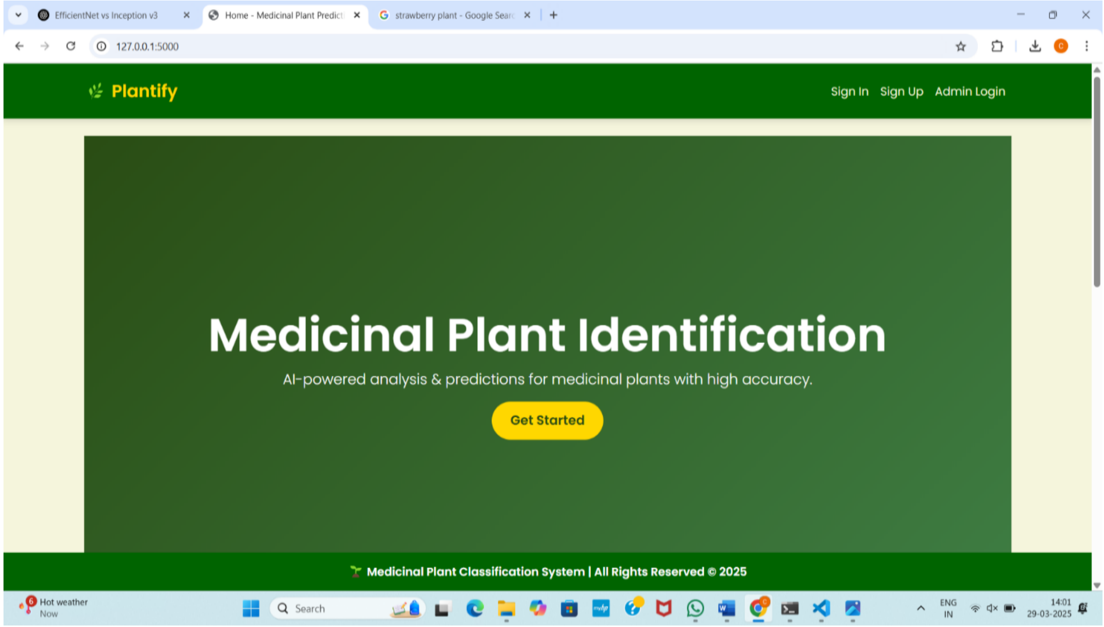
6.3 Performance Metrics



```
                          precision    recall  f1-score   support

               Aloe vera     0.9916    1.0000    0.9958       118
                    Amla     0.9839    0.9104    0.9457        67
            Amruthaballi     0.9670    0.9670    0.9670        91
                   Arali     1.0000    1.0000    1.0000        89
              Astma weed     1.0000    0.9878    0.9939        82
                Badipala     0.9231    0.9474    0.9351        76
            Balloon Vine     1.0000    1.0000    1.0000        61
                  Bamboo     1.0000    0.9915    0.9957       118
                   Beans     0.9798    1.0000    0.9898        97
                   Betel     0.9912    0.9912    0.9912       114
                  Bhrami     0.9810    0.9904    0.9856       104
              Bringaraja     0.9474    0.9863    0.9664        73
              Caricature     1.0000    1.0000    1.0000        76
                  Castor     1.0000    0.9845    0.9922       129
           Catharanthus     0.9925    0.9925    0.9925       134
                  Chakte     0.9444    1.0000    0.9714        68
                  Chilly     0.9583    1.0000    0.9787        69
 Citron lime (herelikai)     0.9789    0.9394    0.9588        99
                  Coffee     0.9639    0.9639    0.9639        83
   Common rue(naagdalli)     1.0000    1.0000    1.0000        67
               Coriender     0.9914    1.0000    0.9957       115
                   Curry     1.0000    0.9762    0.9880       168
              Doddpathre     0.9859    0.9859    0.9859       142
               Drumstick     1.0000    1.0000    1.0000        56
                    Ekka     1.0000    0.9877    0.9938        81
              Eucalyptus     1.0000    0.9875    0.9937        80
                Ganigale     0.9868    1.0000    0.9934        75
                  Ganike     0.9831    0.9206    0.9508        63
                Gasagase     0.9750    0.9873    0.9811        79
                  Ginger     0.9762    1.0000    0.9880        82
          Globe Amarnath     0.9529    1.0000    0.9759        81
                   Guava     0.9683    0.9531    0.9606       128
                   Henna     0.9405    0.9875    0.9634        80
                Hibiscus     1.0000    0.9915    0.9957       118
                   Honge     0.9912    1.0000    0.9956       113
                 Insulin     1.0000    0.9888    0.9944        89
               Jackfruit     0.9630    0.9455    0.9541       110
                 Jasmine     0.9796    0.9796    0.9796        49
               Kambajala     0.9831    0.9831    0.9831        59
              Kasambruga     1.0000    1.0000    1.0000        48
                Kohlrabi     1.0000    1.0000    1.0000        73
                 Lantana     1.0000    0.9342    0.9660        76
                   Lemon     0.9836    0.9756    0.9796       123
              Lemongrass     1.0000    1.0000    1.0000         8


             Malabar Nut     0.9273    1.0000    0.9623        51
         Malabar Spinach     0.9512    0.9873    0.9689        79
                   Mango     0.9712    0.9806    0.9758       103
                Marigold     0.9787    0.9892    0.9840        93
                    Mint     1.0000    0.9778    0.9888       135
                    Neem     0.9774    0.9848    0.9811       132
               Nelavembu     0.9778    0.9778    0.9778        90
                  Nerale     1.0000    0.9839    0.9919        62
                   Nooni     0.9861    0.9861    0.9861        72
                   Onion     1.0000    0.9783    0.9890        92
                   Padri     1.0000    1.0000    1.0000        73
          Palak(Spinach)     0.9673    0.9933    0.9801       149
                  Papaya     1.0000    0.9926    0.9963       135
               Parijatha     0.9851    1.0000    0.9925        66
                     Pea     1.0000    1.0000    1.0000        47
                  Pepper     1.0000    1.0000    1.0000         8
            Pomoegranate     0.9868    1.0000    0.9934        75
                 Pumpkin     1.0000    1.0000    1.0000        92
                 Raddish     1.0000    1.0000    1.0000        40
                    Rose     0.9813    0.9906    0.9859       106
                 Sampige     0.9683    1.0000    0.9839        61
                  Sapota     0.9767    0.9545    0.9655        44
            Seethaashoka     1.0000    0.9362    0.9670        47
              Seethapala     1.0000    1.0000    1.0000       114
                Spinach1     1.0000    0.9851    0.9925        67
                Tamarind     1.0000    0.9943    0.9972       176
                    Taro     1.0000    1.0000    1.0000        69
                  Tecoma     0.9851    0.9565    0.9706        69
                  Thumbe     1.0000    1.0000    1.0000        74
                  Tomato     1.0000    1.0000    1.0000        62
                   Tulsi     0.9665    0.9774    0.9719       177
                Turmeric     1.0000    1.0000    1.0000        39
                  ashoka     1.0000    1.0000    1.0000        81
                 camphor     1.0000    1.0000    1.0000        66
             kamakasturi     0.9848    0.9701    0.9774        67
                  kepala     1.0000    1.0000    1.0000        76

                accuracy                         0.9851      6900
               macro avg     0.9854    0.9854    0.9853      6900
            weighted avg     0.9853    0.9851    0.9851      6900
```

Fig 6.2 Classifcation Report

6.4 OUTPUT SCREENSHOTS



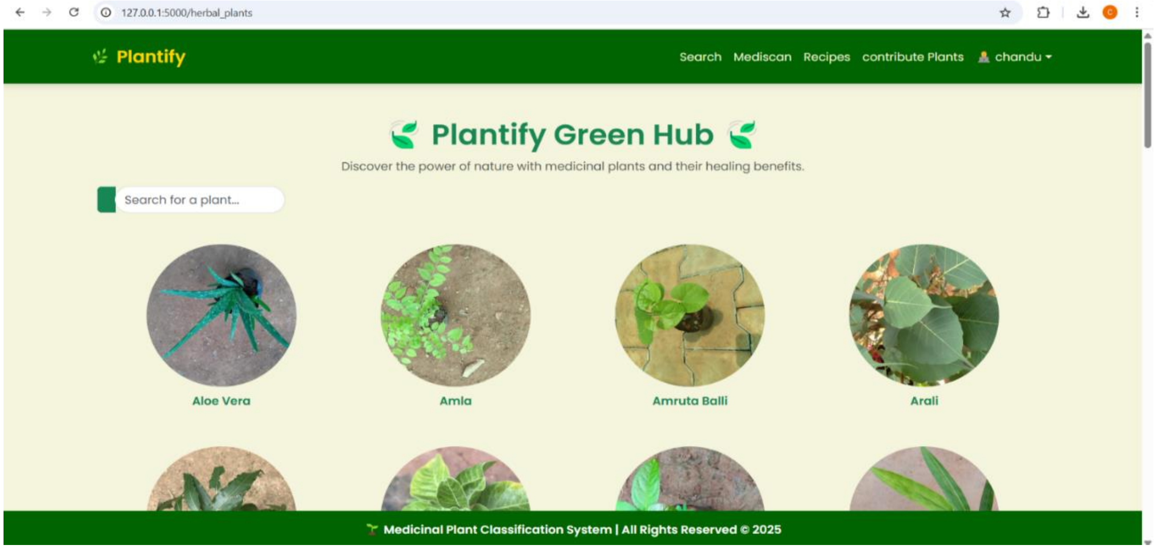
Medicinal Plant Identification

AI-powered analysis & predictions for medicinal plants with high accuracy.

Medicinal Plant Classification System | All Rights Reserved © 2025

Fig. 6.3 Homepage

The above fig. 6.3 Home Page serves as the application's starting point, offering welcoming interface with an introduction to the platform’s purpose. Users are presented with clear options to either log in or sign up, ensuring easy navigation.



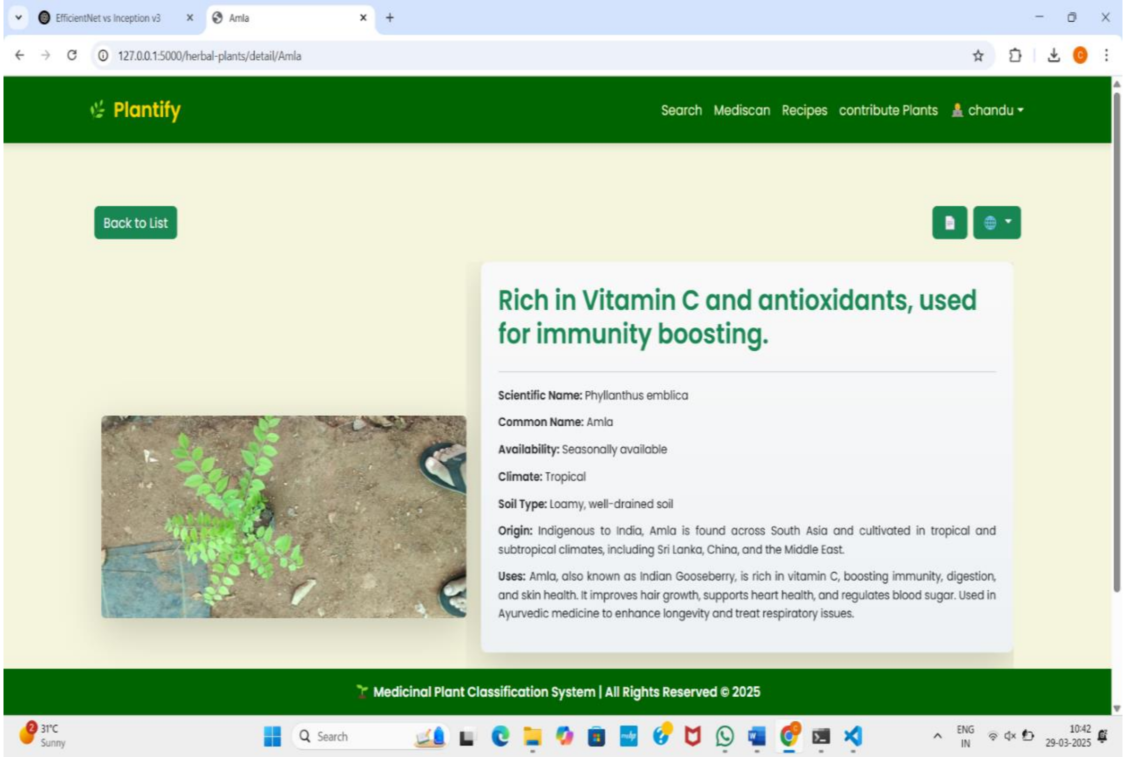
Plantify Green Hub

Discover the power of nature with medicinal plants and their healing benefits.

Medicinal Plant Classification System | All Rights Reserved © 2025

Fig. 6.4 Plantify green hub

The above fig. 6.4 Plantify green hub page serves as the applications image gallery, where they can scroll and find about the plants in the module. They can open it and access the information provided.


EfficientNet vs Inception v3
Amla
127.0.0.1:5000/herbal-plants/detail/Amla
Plantify
Search
Mediscan
Recipes
contribute Plants
chandu
Back to List

Rich in Vitamin C and antioxidants, used for immunity boosting.

Scientific Name: Phyllanthus emblica

Common Name: Amla

Availability: Seasonally available

Climate: Tropical

Soil Type: Loamy, well-drained soil

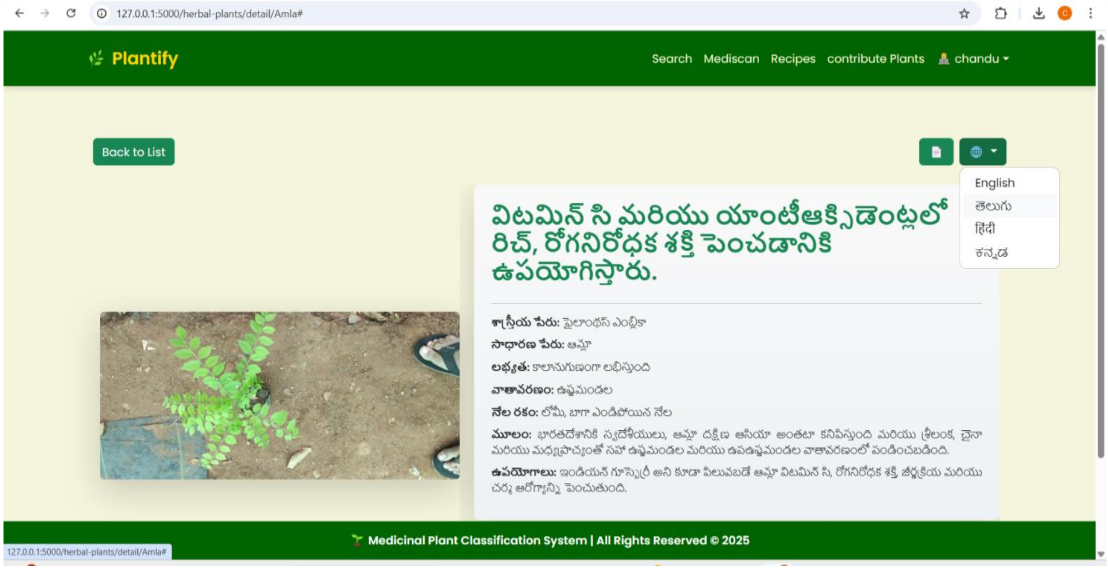
Origin: Indigenous to India, Amla is found across South Asia and cultivated in tropical and subtropical climates, including Sri Lanka, China, and the Middle East.

Uses: Amla, also known as Indian Gooseberry, is rich in vitamin C, boosting immunity, digestion, and skin health. It improves hair growth, supports heart health, and regulates blood sugar. Used in Ayurvedic medicine to enhance longevity and treat respiratory issues.

Medicinal Plant Classification System | All Rights Reserved © 2025
31°C
Sunny
Search
ENG
IN
10:42
29-03-2025

Fig. 6.5 Search and retreival

The above fig. 6.5 Search and retreival page serves as the applications search page, where user can search for a plant and find about the plants present in it. They can open it and access the information provided about the plants like Scientific name, Availability, Climate, Uses etc.


127.0.0.1:5000/herbal-plants/detail/Amla#
Plantify
Search
Mediscan
Recipes
contribute Plants
chandu
Back to List
English
తెలుగు
हिंदी
ಕನ್ನಡ
Medicinal Plant Classification System | All Rights Reserved © 2025
127.0.0.1:5000/herbal-plants/detail/Amla#

విటమిన్ సి మరియు యాంటీఆక్సిడెంట్లలో రిచ్, రోగనిరోధక శక్తి పెంచడానికి ఉపయోగిస్తారు.

శాస్త్రీయ పేరు: ఫైలాంథస్ ఎంబ్లికా

సాధారణ పేరు: ఆమ్లా

లభ్యత: కాలానుగుణంగా లభిస్తుంది

వాతావరణం: ఉష్ణమండల

నేల రకం: లోమీ, బాగా ఎండిపోయిన నేల

మూలం: భారతదేశానికి స్వదేశీయులు, ఆమ్లా దక్షిణ ఆసియా అంతటా కనిపిస్తుంది మరియు శ్రీలంక, చైనా మరియు మధ్యప్రాచ్యంతో సహా ఉష్ణమండల మరియు ఉపఉష్ణమండల వాతావరణంలో పండించబడింది.

ఉపయోగాలు: ఇండియన్ గూస్బెర్రీ అని కూడా పిలువబడే ఆమ్లా విటమిన్ సి, రోగనిరోధక శక్తి, జీర్ణక్రియ మరియు చర్మ ఆరోగ్యాన్ని పెంచుతుంది.

Fig. 6.6 Multi-Language

In this page Fig. 6.6 the multi-language option is available where the user can select any local language as per their convenience to read the information.

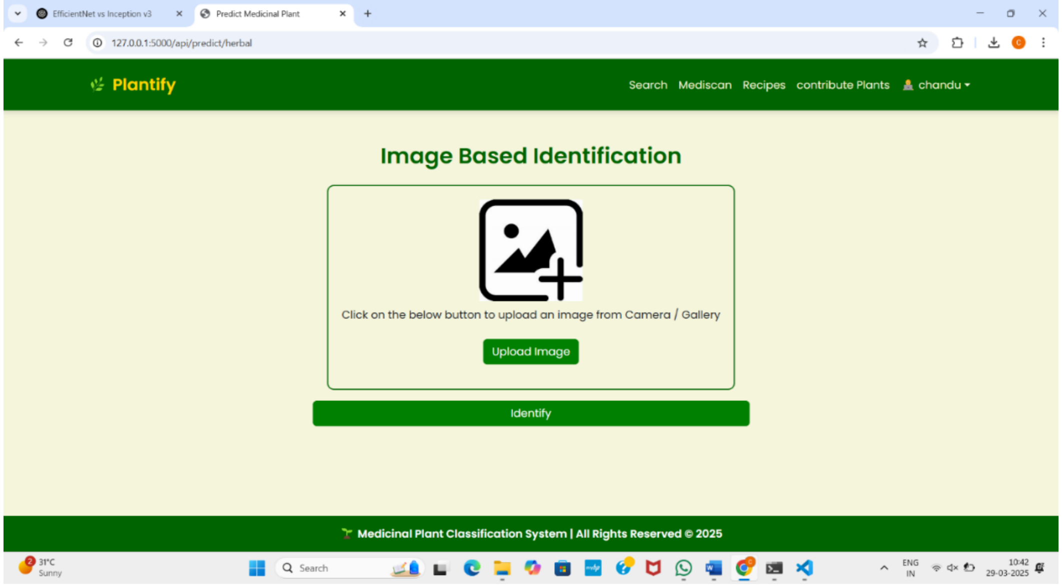
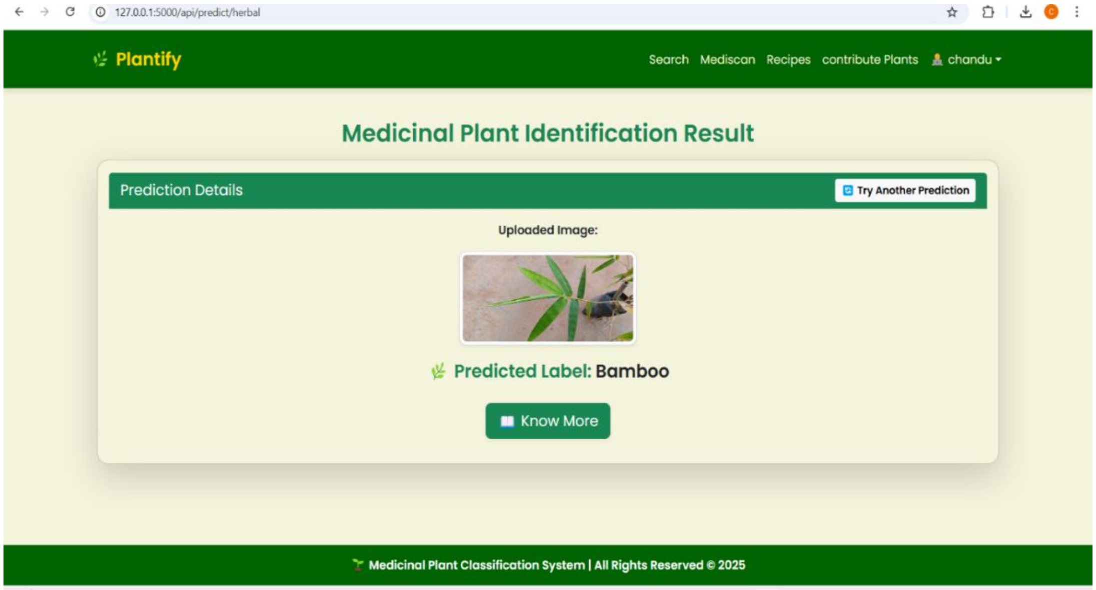

EfficientNet vs Inception v3
Predict Medicinal Plant
127.0.0.1:5000/api/predict/herbal
Plantify
Search
Mediscan
Recipes
contribute Plants
chandu
Image Based Identification
Click on the below button to upload an image from Camera / Gallery
Upload Image
Identify
Medicinal Plant Classification System | All Rights Reserved © 2025
31°C
Sunny
Search
ENG
IN
10:42
29-03-2025

Fig. 6.7 Image Based Identification



Medicinal Plant Identification Result

Prediction Details

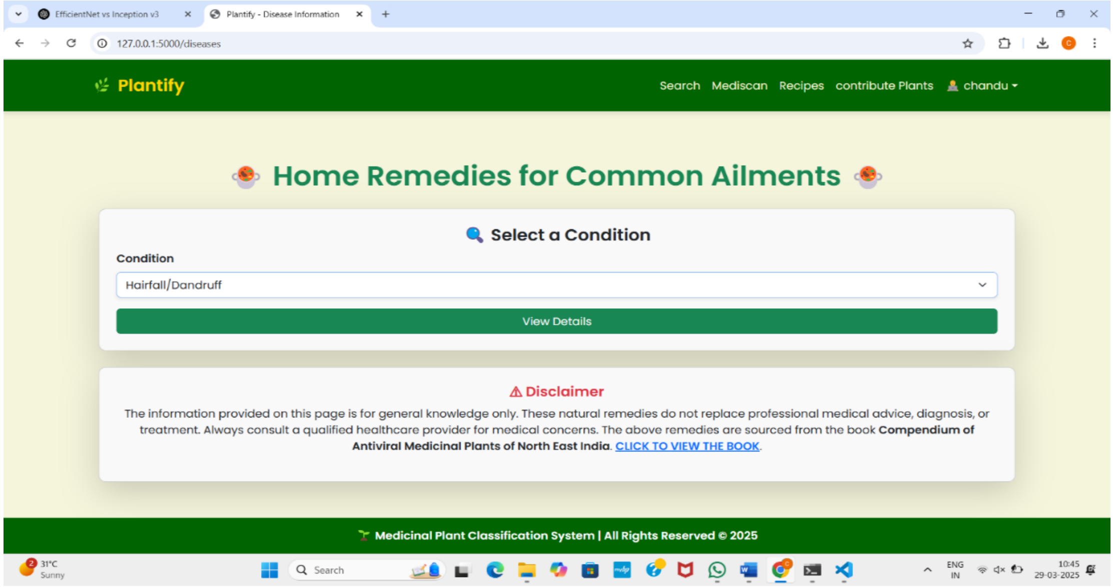
Uploaded Image:

Predicted Label: Bamboo

Medicinal Plant Classification System | All Rights Reserved © 2025

Fig 6.7(a) Output of the prediction

In this page Fig. 6.7 image based identification where the user can select any plant image as per their choice and the user gets the predicted output as shown in Fig. 6.7(a).



Home Remedies for Common Ailments

Select a Condition

Condition

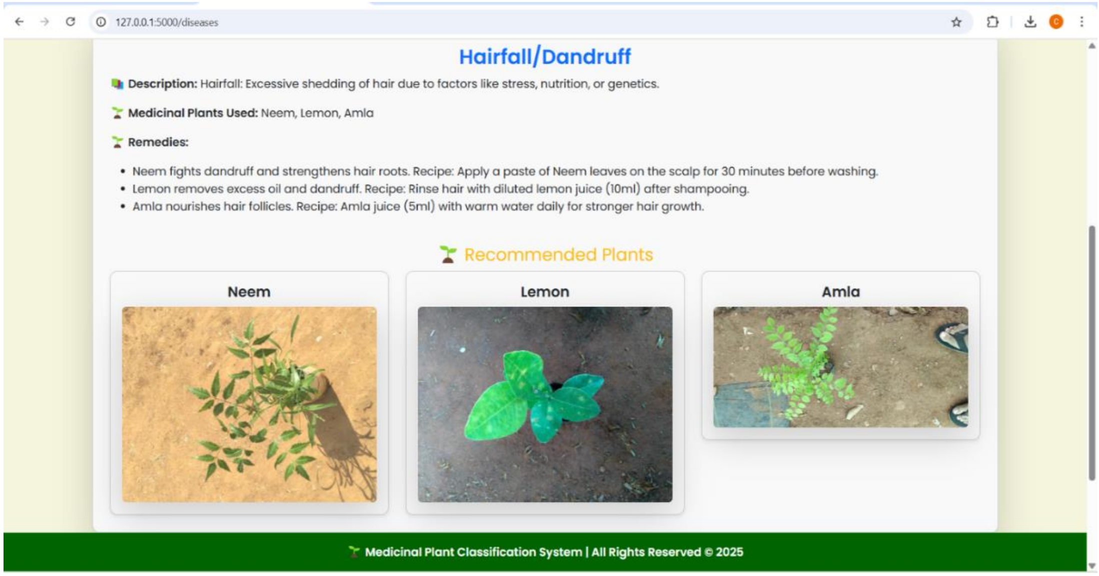
Hairfall/Dandruff

⚠ Disclaimer

The information provided on this page is for general knowledge only. These natural remedies do not replace professional medical advice, diagnosis, or treatment. Always consult a qualified healthcare provider for medical concerns. The above remedies are sourced from the book **Compendium of Antiviral Medicinal Plants of North East India**. CLICK TO VIEW THE BOOK.

Medicinal Plant Classification System | All Rights Reserved © 2025

Fig. 6.8 Remedies



Hairfall/Dandruff

Description: Hairfall: Excessive shedding of hair due to factors like stress, nutrition, or genetics.

Medicinal Plants Used: Neem, Lemon, Amla

Remedies:

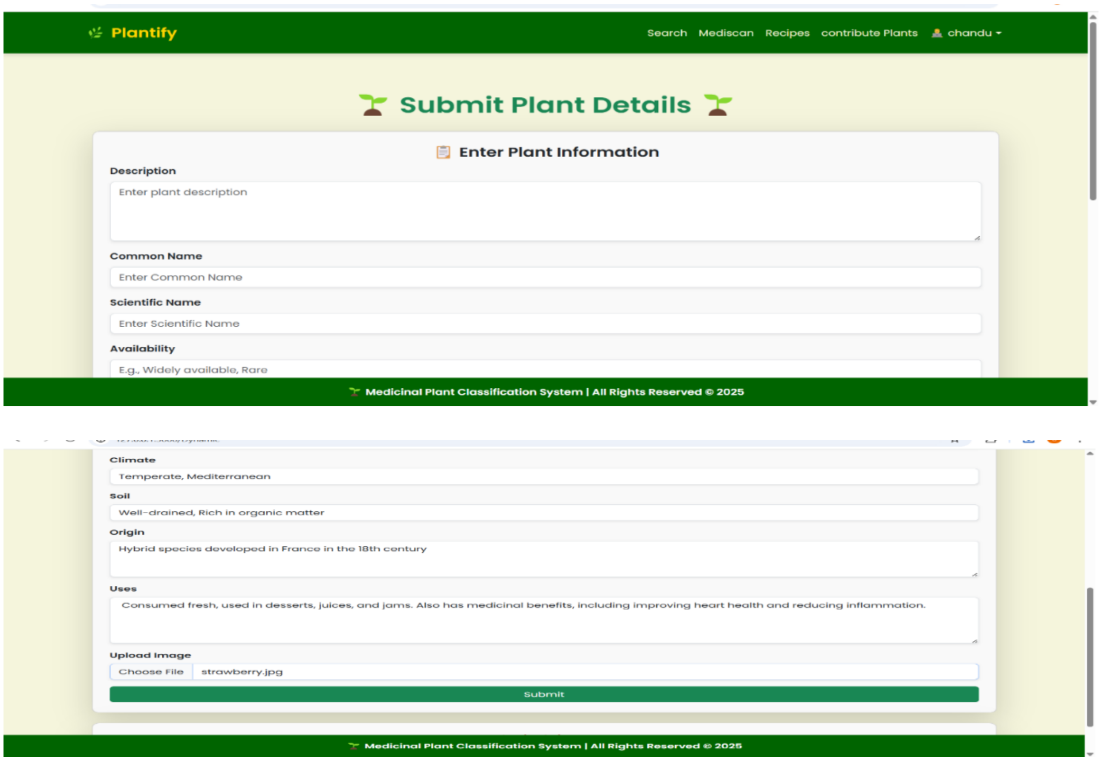
- Neem fights dandruff and strengthens hair roots. Recipe: Apply a paste of Neem leaves on the scalp for 30 minutes before washing.
- Lemon removes excess oil and dandruff. Recipe: Rinse hair with diluted lemon juice (10ml) after shampooing.
- Amla nourishes hair follicles. Recipe: Amla juice (5ml) with warm water daily for stronger hair growth.

Recommended Plants

Medicinal Plant Classification System | All Rights Reserved © 2025

Fig. 6.8(a) Remedies result

In this page Fig.6.8 Remedies where the user can select any disease name as per their choice and the user gets the remedies of that disease as shown in Fig.6.8(a).



Submit Plant Details

Enter Plant Information

Description

Enter plant description

Common Name

Enter Common Name

Scientific Name

Enter Scientific Name

Availability

E.g., Widely available, Rare

Medicinal Plant Classification System | All Rights Reserved © 2025

Climate

Temperate, Mediterranean

Soil

Well-drained, Rich in organic matter

Origin

Hybrid species developed in France in the 18th century

Uses

Consumed fresh, used in desserts, juices, and jams. Also has medicinal benefits, including improving heart health and reducing inflammation.

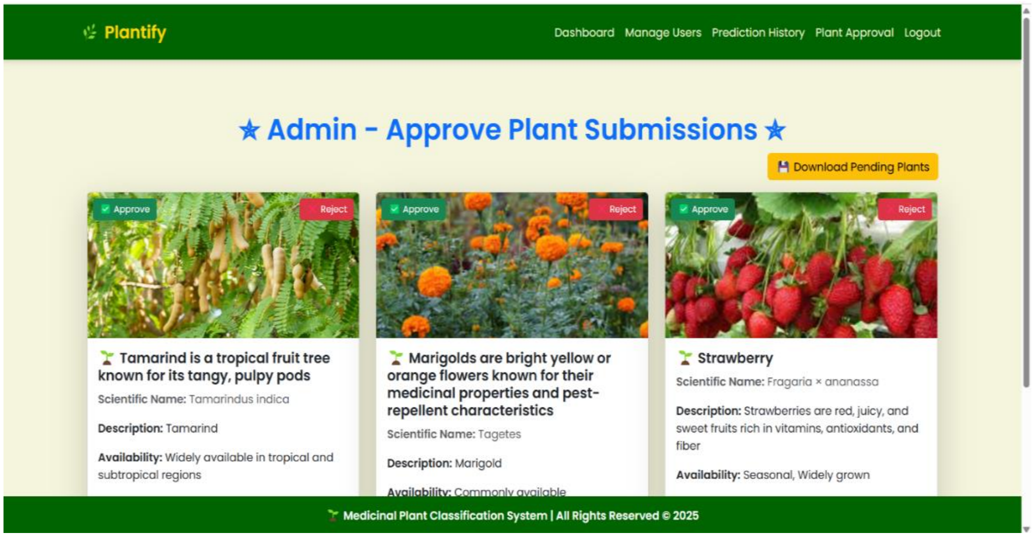
Upload Image

Choose File strawberry.jpg

Submit

Medicinal Plant Classification System | All Rights Reserved © 2025

Fig. 6.9 Contribute Plant



★ Admin - Approve Plant Submissions ★

Tamarind is a tropical fruit tree known for its tangy, pulpy pods

Scientific Name: Tamarindus indica

Description: Tamarind

Availability: Widely available in tropical and subtropical regions

Marigolds are bright yellow or orange flowers known for their medicinal properties and pest-repellent characteristics

Scientific Name: Tagetes

Description: Marigold

Availability: Commonly available

Strawberry

Scientific Name: Fragaria × ananassa

Description: Strawberries are red, juicy, and sweet fruits rich in vitamins, antioxidants, and fiber

Availability: Seasonal, Widely grown

Medicinal Plant Classification System | All Rights Reserved © 2025

In this page Fig. 6.9 Contribute Plant where the user can contribute to the research where he can provide the information about a plant which is not in our list and the admin will view the details and add either approve or reject based on the information accuarate rate.

6.5 Deployment

The deployment phase is crucial for making the Plantify-Enhanced Medicinal Plant Identification Using Convolutional Neural Networks system accessible to users. It involves setting up the backend, ensuring responsiveness, and maintaining real-time data processing capabilities. This section details the deployment architecture, tools used, and challenges encountered during the deployment process.

6.5.1 Deployment Architecture

The Plantify-Enhanced Medicinal Plant Identification Using Convolutional Neural Networks system is deployed using a cloud-based architecture for scalability, availability, and performance. The system is hosted on AWS EC2 (Amazon Elastic Compute Cloud), which provides flexible and scalable cloud computing resources.

- **Frontend:**

  Built with HTML, CSS, JavaScript, and Bootstrap, the frontend is hosted on a web server and optimized for responsiveness. It provides the interface for users to take surveys, view predictions, and access suggestions across devices (smartphones, tablets, desktops).

- **Backend:**

  The Django framework handles server-side logic, including user authentication, survey processing, model prediction, and data storage. It also facilitates communication with Google Dialogflow API for chatbot interactions.

- **Database:**

  User data, including survey responses and mental health predictions, is stored in a relational database like MySQL or PostgreSQL, hosted on AWS RDS (Relational Database Service). This ensures high availability, security, and scalability.

- **Cloud-Infrastructure:**

  The system runs on AWS EC2 instances, providing the necessary compute power for backend operations and machine learning models. AWS S3 stores static files like images and documents for efficient management.

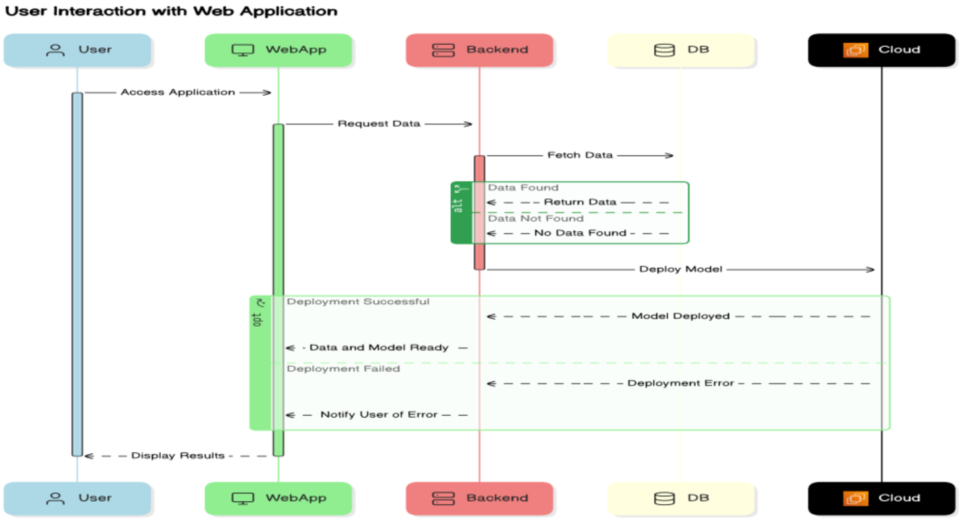


Fig. 6.10 Deployment Architecture Diagram

6.5.2 Tools Used

Various tools and technologies were employed to ensure the stability, security, and efficiency of the deployment process:

- **AWS EC2:** Cloud platform used for hosting the system, providing scalable compute power to handle varying loads.
- **MySQL:** Relational databases used for storing user data and mental health predictions, hosted on AWS RDS for scalability.
- **AWS S3:** Storage for static files such as images, reports, and documents.
- **Git:** Used for version control throughout development and deployment, hosted on platforms like GitHub.
- **AWS RDS:** Managed relational database service providing automated backups, scaling, and high availability.

6.5.3 Challenges Faced

Several challenges were encountered during deployment, ranging from optimizing model performance to handling real-time interactions and ensuring data security. Here are the key challenges and solutions:

1. **Model Performance Optimization:**

Challenge: Ensuring the machine learning models performed efficiently in real-time with varying loads.

Solution: Models were serialized using Pickle to enable faster loading and prediction. Model caching techniques were also used to avoid reloading models for every user interaction.

2. **Data Security and Privacy:**

Challenge: Ensuring the security of sensitive mental health data.

Solution: HTTPS was implemented to secure data transmission, and data encryption was used for database storage. The system was built to comply with data privacy regulations like GDPR to ensure responsible data handling.

3. **Scaling the System for Multiple Users:**

   Challenge: Ensuring the system could handle a large number of concurrent users without performance degradation.

   Solution: Auto-scaling was implemented on EC2 instances, allowing the system to dynamically adjust resources based on user demand, ensuring optimal performance during peak traffic.

4. **User Interface and Experience:**

   Challenge: Creating a user-friendly and intuitive interface for interacting with surveys, predictions, and suggestions.

   Solution: User acceptance testing (UAT) was conducted to gather feedback and refine the user interface. Adjustments were made to simplify navigation and improve the overall user experience.

Tools

- **Testing**: PyTest, Selenium
- **Performance**: TensorBoard
- **Deployment**: AWS EC2, NoHub

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

CONCLUSION

Concluding the project on "Plantify-Enhanced Medicinal Plant Identification Using Convolutional Neural Networks" presents a significant milestone in harnessing technology for botanical and medicinal applications. Throughout this endeavour, we have successfully leveraged machine learning algorithms and image processing techniques to automate the identification process, offering an efficient and accurate solution for classifying various medicinal plants and raw materials.

Our exploration began with the selection and preprocessing of a diverse dataset comprising images of medicinal plants and raw materials. We meticulously curated and standardized the dataset to ensure robustness and reliability in model training. Subsequently, we delved into the implementation of state-of-the-art machine learning models, including EfficientNet and ResNet, to extract meaningful features from the input images.

Through rigorous experimentation and evaluation, we observed the effectiveness of these models in accurately classifying the botanical specimens. The ensemble approach, combining the strengths of Efficient and ResNet architectures, exhibited promising results, showcasing improved classification accuracy and robustness. This highlights the significance of leveraging ensemble techniques to harness the complementary capabilities of multiple models.

Furthermore, the deployment of the trained models into a practical application demonstrates the real-world utility of our solution. Users can now conveniently identify various medicinal plants and raw materials by simply uploading images, thereby streamlining the identification process and promoting accessibility to botanical knowledge.

FUTURE SCOPE

The project on " Plantify -Enhanced Medicinal Plant Identification Using Convolutional Neural Networks " presents a promising avenue for future advancements and extensions. As we conclude this endeavour, it's essential to highlight the potential future scopes that can further elevate the project's impact and relevance in the field of botanical science and healthcare.

Firstly, one of the primary areas of focus for future enhancements lies in the refinement of the machine learning models utilized in the project. By continuously refining model architectures, optimizing hyperparameters, and exploring novel regularization techniques, we can enhance the performance of the classification models in terms of accuracy, speed, and generalization capabilities.

Expanding the dataset to encompass a broader spectrum of medicinal plants and raw materials from various geographical regions and ecosystems represents another crucial aspect of future development. By incorporating more annotated images and comprehensive metadata, we can enrich the dataset, enabling the models to adapt and generalize better to diverse botanical specimens.

Furthermore, there is immense potential in extending the project's scope beyond binary classification to more fine-grained classification tasks. This could involve species-level identification, classification based on specific medicinal properties or chemical constituents, or even identifying plants at different stages of growth or maturity.

Developing a user-friendly mobile application based on the trained models opens up new avenues for practical applications and widespread dissemination of botanical knowledge. Integration with smartphone cameras, cloud-based inference engines, and crowdsourced data collection can empower users to identify medicinal plants and raw materials conveniently and access relevant information on-the-go.

CHAPTER 8

REFERENCES

[1] S. Prasad and P. P. Singh, ‘‘Vision system for medicinal plant leaf acquisition and analysis,’’ in Applications of Cognitive Computing Systems and IBM Watson. Singapore: Springer, 2017, pp. 37–45.

[2] Santhi Daggubati (2024) “Mediscan: Real-Time Identification of Medicinal Plants Species Using Deep Learning Models.”

[3] S. U. Habiba, M. K. Islam, and S. M. M. Ahsan, ‘‘Bangladeshi plant recognition using deep-learning based leaf classification,’’ in Proc. Int. Conf. Comput., Commun., Chem., Mater. Electron. Eng. (IC ME 4), Jul. 2019, pp. 1–4.

[4] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). "MobileNets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861.

[5] Lekshmi Kalinathan, Ruba Soundar Kathavarayan & Jagadish Kumar K J, “Real-Time Identification of Medicinal Plants using Machine Learning Techniques”.

[6] Rincon, F.; Mayer, S.A. Intracerebral Hemorrhage: Clinical Overview and Pathophysiologic Concepts. Transl. Stroke Res. 2012, 3, 10–24.

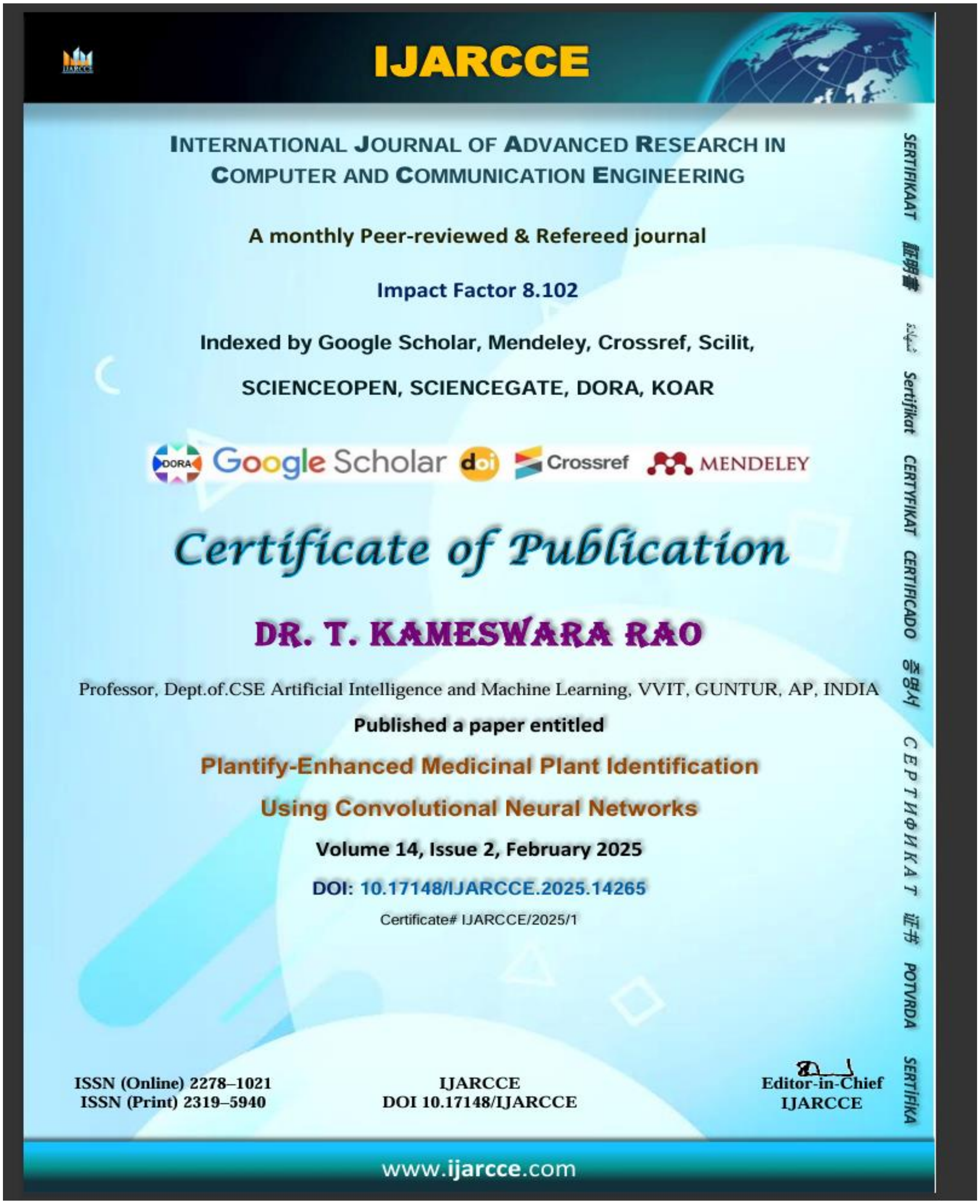
[7] Madan,S ; Ahmad, S.; Singh, G.N.; Kohli, K.; Kumar, Y.; Singh, R.; Garg, M. Stevia Rebaudiana (Bert.) Bertoni—A review. Indian J. Nat. Prod. Resour.2010, 1, 267–286.

[8] Herro, E.; Jacob, S.E. Mentha (Peppermint). Dermatitis 2010, 21, 327–329.

[9] Mahendran, G.; Rahman, L.U. Ethnomedicinal, phytochemical and pharmacological updates on Peppermint (Mentha piperita L.)—A review. Phytother. Res. 2020, 34, 2088–2139.

[10] Bhardwaj, R.L.; Nandal, U. Nutritional and therapeutic potential of bael (Aegle marmelos Corr.) fruit juice: A review. Nutr. Food Sci. 2015, 45, 895–919

APPENDIX - Publication Certificate



IJARCCE

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER AND COMMUNICATION ENGINEERING

A monthly Peer-reviewed & Refereed journal

Impact Factor 8.102

Indexed by Google Scholar, Mendeley, Crossref, Scilit,

SCIENCEOPEN, SCIENCEGATE, DORA, KOAR

Certificate of Publication

DR. T. KAMESWARA RAO

Professor, Dept.of.CSE Artificial Intelligence and Machine Learning, VVIT, GUNTUR, AP, INDIA

Published a paper entitled

Plantify-Enhanced Medicinal Plant Identification Using Convolutional Neural Networks

Volume 14, Issue 2, February 2025

DOI: 10.17148/IJARCCE.2025.14265

Certificate# IJARCCE/2025/1

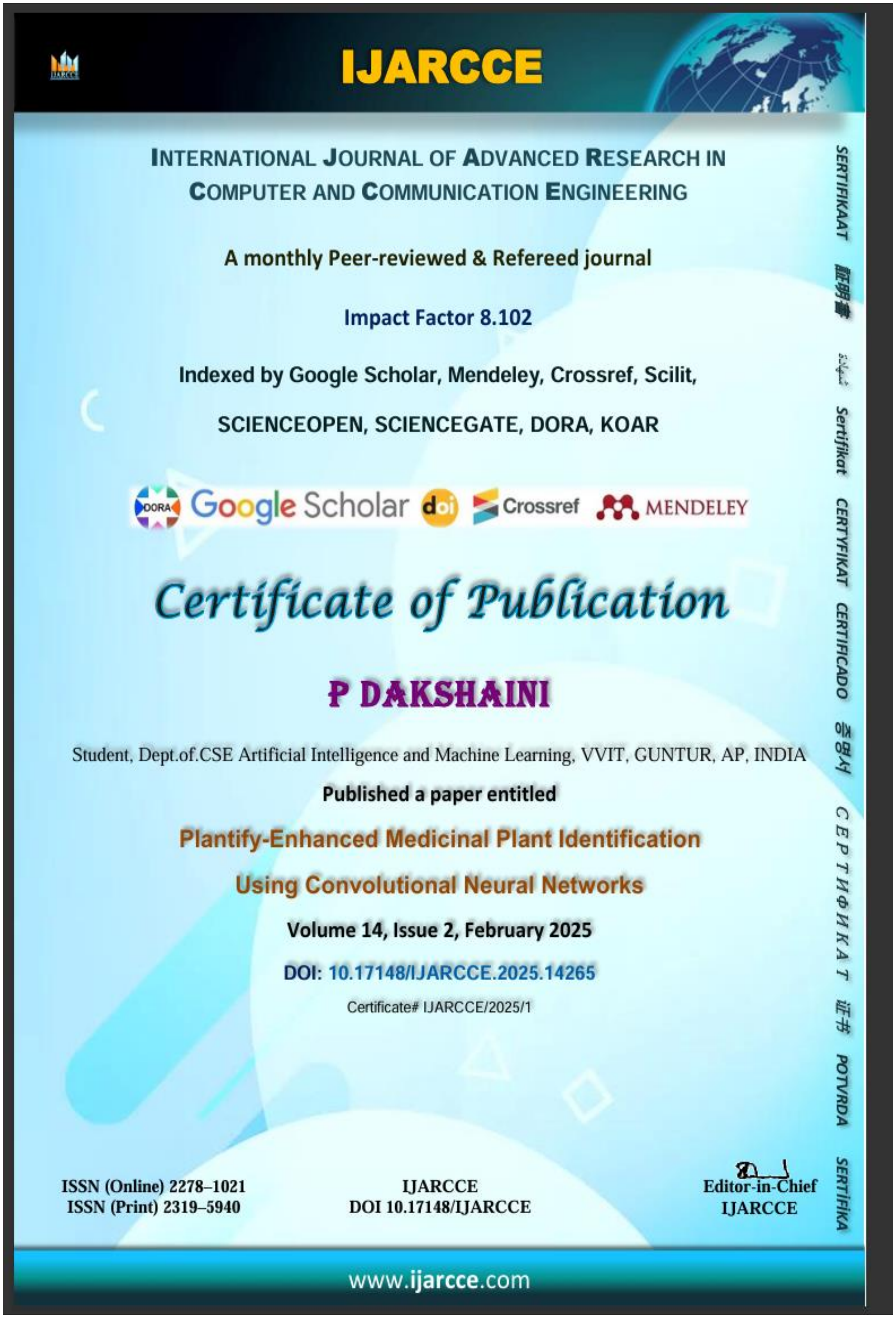
ISSN (Online) 2278–1021
ISSN (Print) 2319–5940

IJARCCE
DOI 10.17148/IJARCCE

Editor-in-Chief
IJARCCE

www.ijarcce.com

SERTIFIKAAT 証明書 شهادة Sertifikat CERTYFIKAT CERTIFICADO 증명서 СЕРТИФИКАТ 证书 POTVRDA SERTIFIKA



SERTIFIKAAT 証明書 شهادة Sertifikat CERTYFIKAT CERTIFICADO 증명서 СЕРТИФИКАТ 证书 POTVRDA SERTIFIKA

IJARCCE

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN
COMPUTER AND COMMUNICATION ENGINEERING

A monthly Peer-reviewed & Refereed journal

Impact Factor 8.102

Indexed by Google Scholar, Mendeley, Crossref, Scilit,

SCIENCEOPEN, SCIENCEGATE, DORA, KOAR

DORA Google Scholar doi Crossref MENDELEY

Certificate of Publication

P DAKSHAINI

Student, Dept.of.CSE Artificial Intelligence and Machine Learning, VVIT, GUNTUR, AP, INDIA

Published a paper entitled

Plantify-Enhanced Medicinal Plant Identification
Using Convolutional Neural Networks

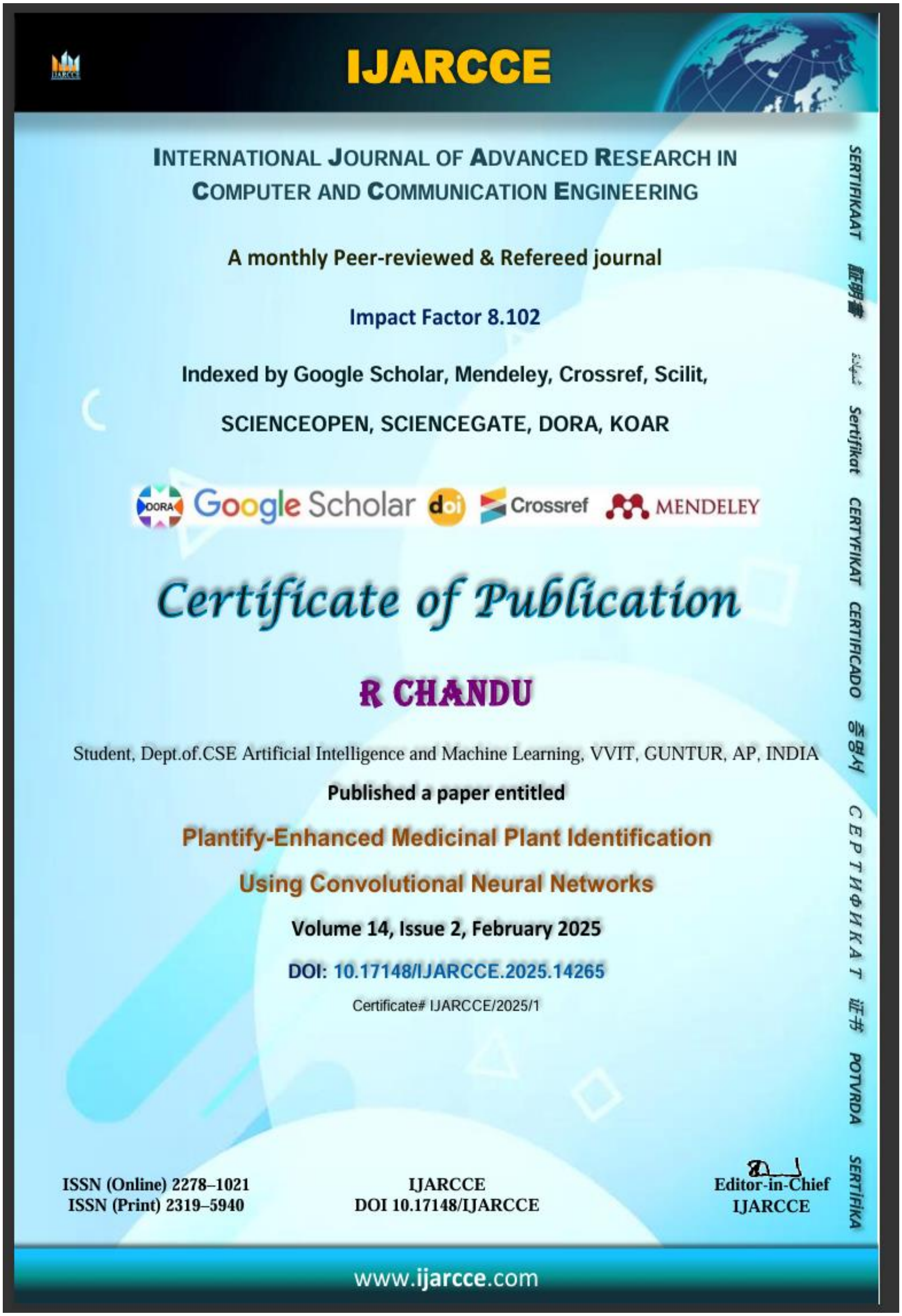
Volume 14, Issue 2, February 2025

DOI: 10.17148/IJARCCE.2025.14265

Certificate# IJARCCE/2025/1

ISSN (Online) 2278–1021 ISSN (Print) 2319–5940	IJARCCE DOI 10.17148/IJARCCE	Editor-in-Chief IJARCCE

www.ijarcce.com



IJARCCE

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER AND COMMUNICATION ENGINEERING

A monthly Peer-reviewed & Refereed journal

Impact Factor 8.102

Indexed by Google Scholar, Mendeley, Crossref, Scilit,

SCIENCEOPEN, SCIENCEGATE, DORA, KOAR

Certificate of Publication

R CHANDU

Student, Dept.of.CSE Artificial Intelligence and Machine Learning, VVIT, GUNTUR, AP, INDIA

Published a paper entitled

Plantify-Enhanced Medicinal Plant Identification Using Convolutional Neural Networks

Volume 14, Issue 2, February 2025

DOI: 10.17148/IJARCCE.2025.14265

Certificate# IJARCCE/2025/1

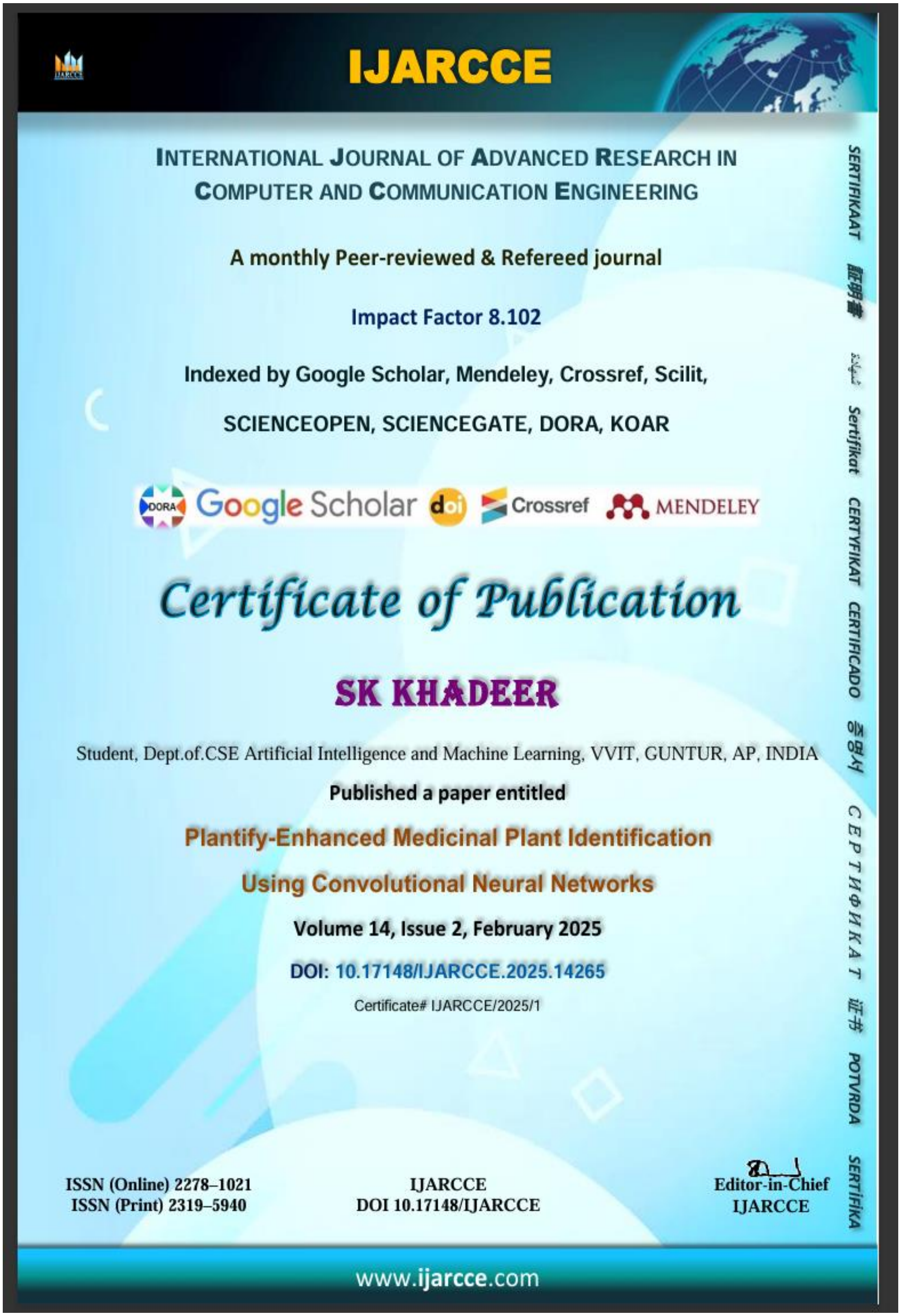
ISSN (Online) 2278–1021
ISSN (Print) 2319–5940

IJARCCE
DOI 10.17148/IJARCCE

Editor-in-Chief
IJARCCE

SERTIFIKAAT 証明書 شهادة Sertifikat CERTYFIKAT CERTIFICADO 증명서 СЕРТИФИКАТ 证书 POTVRDA SERTIFIKA

www.ijarcce.com



SERTIFIKAAT
証明書
شهادة
Sertifikat
CERTYFIKAT
CERTIFICADO
증명서
СЕРТИФИКАТ
证书
POTVRDA
SERTIFIKA

IJARCCE

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER AND COMMUNICATION ENGINEERING

A monthly Peer-reviewed & Refereed journal

Impact Factor 8.102

Indexed by Google Scholar, Mendeley, Crossref, Scilit,

SCIENCEOPEN, SCIENCEGATE, DORA, KOAR

DORA Google Scholar doi Crossref MENDELEY

Certificate of Publication

SK KHADEER

Student, Dept.of.CSE Artificial Intelligence and Machine Learning, VVIT, GUNTUR, AP, INDIA

Published a paper entitled

Plantify-Enhanced Medicinal Plant Identification

Using Convolutional Neural Networks

Volume 14, Issue 2, February 2025

DOI: 10.17148/IJARCCE.2025.14265

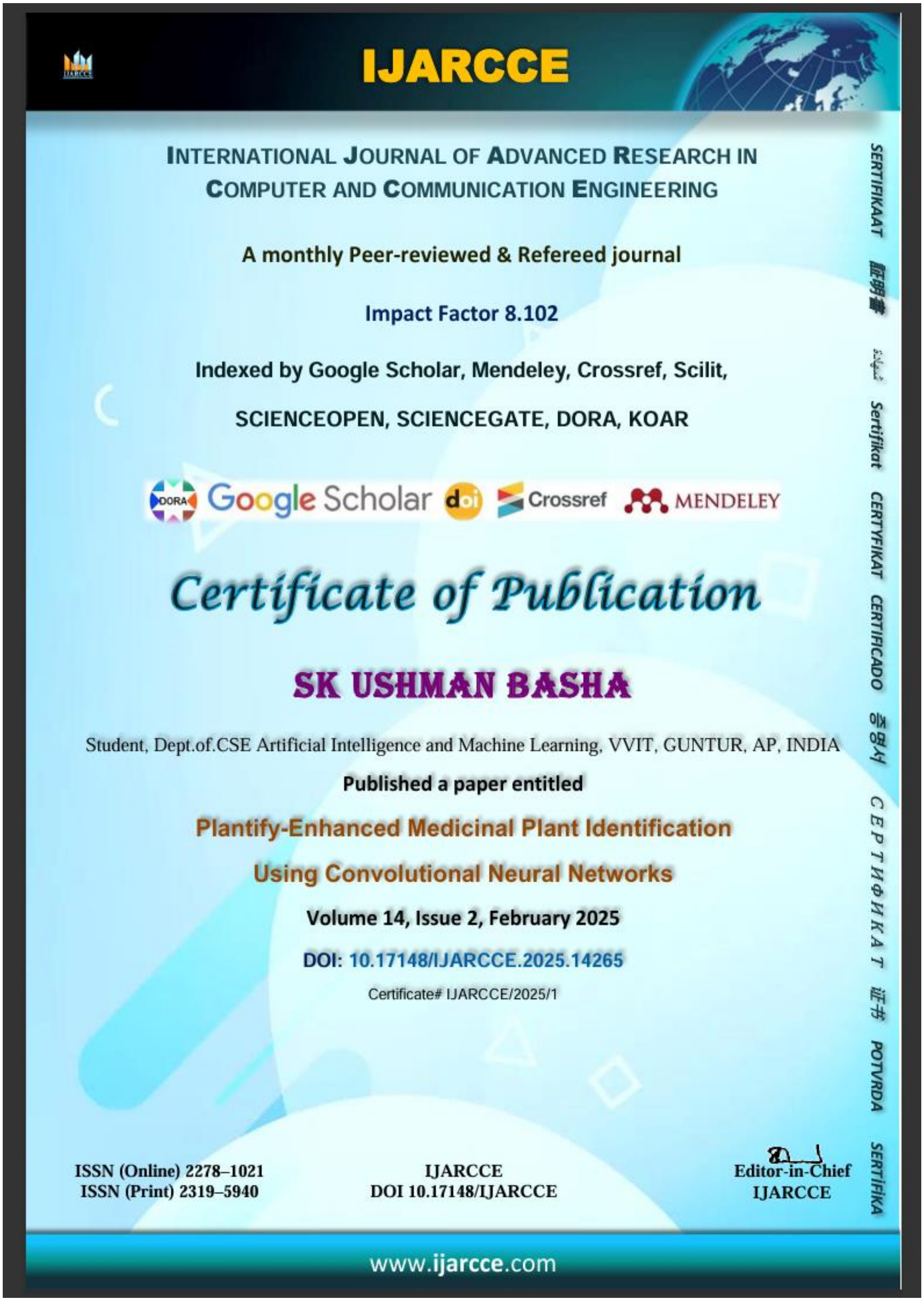
Certificate# IJARCCE/2025/1

ISSN (Online) 2278–1021
ISSN (Print) 2319–5940

IJARCCE
DOI 10.17148/IJARCCE

Editor-in-Chief
IJARCCE

www.ijarcce.com



IJARCCE

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER AND COMMUNICATION ENGINEERING

A monthly Peer-reviewed & Refereed journal

Impact Factor 8.102

Indexed by Google Scholar, Mendeley, Crossref, Scilit,

SCIENCEOPEN, SCIENCEGATE, DORA, KOAR

DORA Google Scholar doi Crossref MENDELEY

Certificate of Publication

SK USHMAN BASHA

Student, Dept.of.CSE Artificial Intelligence and Machine Learning, VVIT, GUNTUR, AP, INDIA

Published a paper entitled

Plantify-Enhanced Medicinal Plant Identification Using Convolutional Neural Networks

Volume 14, Issue 2, February 2025

DOI: 10.17148/IJARCCE.2025.14265

Certificate# IJARCCE/2025/1

ISSN (Online) 2278–1021
ISSN (Print) 2319–5940

IJARCCE
DOI 10.17148/IJARCCE

Editor-in-Chief
IJARCCE

www.ijarcce.com

SERTIFIKAAT 証明書 شهادة Sertifikat CERTYFIKAT CERTIFICADO 증명서 СЕРТИФИКАТ 证书 POTVRDA SERTIFIKA