Mini-Project Report On

PlantSense: Plant Identification And Medicine Recommendation

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science & Engineering

By

Ebin Shibu Nanthalathu(U2003074)
Firoz Yoosef(U2003080)
Hitha Mathew P(U2003096)
Karunya James(U2003116)

Under the guidance of
Ms. Tripti C

RSET
RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

Department of Computer Science & Engineering
Rajagiri School of Engineering and Technology (Autonomous)
Rajagiri Valley, Kakkanad, Kochi, 682039

July 2023

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
RAJAGIRI SCHOOL OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
RAJAGIRI VALLEY, KAKKANAD, KOCHI, 682039

CERTIFICATE

This is to certify that the mini-project report entitled ***"PlantSense: Plant Identification And Medicine Recommendation"*** *is a bonafide work done by* ***Mr. Ebin Shibu Nanthalathu (U2003074),Mr. Firoz Yoosef (U2003080), Ms. Hitha Mathew P (U2003096) and Ms. Karunya James (U2003116)****, submitted to the University in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in Computer Science and Engineering during the academic year 2022-2023.*

Dr. Preetha K. G.
Head of Department
Dept. of CSE
RSET

Mr. Uday Babu P.
Mini-Project Coordinator
Asst. Professor
Dept. of CSE
RSET

Ms. Tripti C
Mini-Project Guide
Asst. Professor
Dept. of CSE
RSET

ACKNOWLEDGEMENT

We wish to express our sincere gratitude towards **Dr. P. S. Sreejith**, Principal of RSET, and **Dr. Preetha K. G.**, Head of Department of Computer Science and Engineering for providing us with the opportunity to undertake our mini-project, "PlantSense: Plant Identification And Medicine Recommendation".

We are highly indebted to our mini-project coordinator **Mr. Uday Babu P**, Assistant Professor, Department of Computer Science and Engineering for his valuable support.

It is indeed our pleasure and a moment of satisfaction for us to express our sincere gratitude to our mini-project guide **Ms. Tripti C**, for her patience and all the priceless advice and wisdom she has shared with us.

Last but not the least, we would like to express our sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

Ebin Shibu Nanthalathu

Firoz Yoosef

Hitha Mathew P

Karunya James

ABSTRACT

PlantSense: Empowering Nature's Healing Touch - A Plant Identification and Medicine Recommendation Web Application

There are several medicinal plants in our locality; however, we don't know much about them to understand which are good for health. Due to our limited knowledge, we are not in a position to identify the plants. Therefore, an automated system is proposed here for Plant Identification and to suggest their Medicinal uses. With the rise in interest and demand for herbal remedies, PlantSense serves as a comprehensive platform for plant identification and personalized medicine recommendations. Using advanced image recognition technology, PlantSense allows users to upload images of plants, whether from their gardens, hiking trips, or anywhere else, and quickly obtain accurate identifications. The application leverages a vast plant database to deliver precise results, enabling users to access essential information about identified plants, such as scientific names, common names,and more.

PlantSense goes beyond identification by offering personalized medicine recommendations based on the identified plants' medicinal properties. Drawing upon a comprehensive database of medicinal plant knowledge, the application suggests natural remedies and their traditional uses to address various ailments and promote overall well-being. The user-friendly interface of PlantSense provides a seamless experience, allowing users to effortlessly navigate through the application's features. By combining plant identification and medicinal knowledge, PlantSense offers a valuable resource for individuals seeking natural remedies and a deeper connection with the plant kingdom. PlantSense is a gateway to unlocking the secrets of nature's pharmacy and harnessing the healing touch that lies within our surroundings.

CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

1.1 Background

1.1.1 Definition

Plants are eukaryotic organisms that mainly photosynthesize and form the kingdom Plantae. Most are multicellular. Historically, plants included algae and all non-animal creatures, including fungi. All ingredients now exclude fungi and some algae. According to one definition, plants form the lineage Viridiplantae (Latin for "green plants"), which consists of green algae and embryophytes, or land plants. The latter includes snapdragons, liverworts, mosses, pine trees, ferns, conifers and other gymnosperms, as well as flowering plants. The genome composition includes green plants as well as red and gray algae in the primary plant lineage.

Green plants derive most of their energy from sunlight using chloroplasts formed endosymbiotically with cyanobacteria. Chloroplasts perform photosynthesis using the pigment chlorophyll, which gives them their green color. Some plants are parasitic and have lost their ability to normally produce chlorophyll or perform photosynthesis. Plants are characterized by sexual reproduction and alternation of generations, but asexual reproduction is also common.

There are about 380,000 known plant species, and most (about 260,000) produce seeds. Green plants provide the majority of the world's molecular oxygen and form the basis of most of the world's ecosystems. Grains, fruits, and vegetables are staple foods for humans and have been domesticated for thousands of years. Plants have many cultural and other uses such as decoration, household items, material gathering, and of course they are also medicinal products. The branch of science that studies plants is called botany and is a branch of biology.

1.1.2 Diversity

There are about 382,000 recognized plant species, most of which produce (about 293,000) seeds. The table below shows some types of forecasts for different green plants. About 85-90 percent of plants are flowering plants. Many projects are currently trying to collect information on all types of plants in online databases.

Plants are the size of a single cell, as are most algae, including complex algae (over 10 microns in diameter) and microfauna (less than 3 microns in diameter) for trees such as conifer sequoia (up to 100 meters in diameter). 120 m))) and angiosperm regnans (up to 325 ft (99 m)).

1.2 Existing Systems

1.2.1 Plant Species Identification using a TensorFlow-Lite model within mobile device

Deep Learning models are huge and requires high computation for inferencing. Can we train Deep Learning models which require less computation power, are smaller in size and can be deployed on mobile phones? Well, the answer is 'yes'. With the integration of capability to train TensorFlow lite models with ArcGIS API for Python, we can now train DL models that can be deployed on mobile devices and are smaller in size.

Where can we use them? We can use them up to train multiple DL models to perform classification tasks specifically for mobile devices. One such integration we did is in the "Survey123" application which is a simple and intuitive form-centric data gathering solution being used by multiple surveyors while performing ground surveys, where we integrated a tf-lite model to classify different plant species while clicking it's picture in the app.

This notebook intends to showcase this capability to train a deep learning model that can be used in mobile applications for a real time inferencing using TensorFlow Lite framework. As an example, we will train the same plant species classification model which was discussed earlier but with a smaller dataset.

1.2.2 iNaturist

iNaturist helps you identify plants and animals in your environment while creating information for research and conservation. Connect with a community of millions of scientists and naturalists who can help you learn more about nature! Also, by writing and sharing your observations, you will create good research data for researchers working to better understand and prevent situations. So if you want to document your outdoor explorations or just enjoy learning about life

Vision: iNaturalist's vision is a world where everyone can understand and sustain biodiversity through the practice of observing wild organisms and sharing information about them.

Mission: iNaturalist's mission is to connect people to nature and advance biodiversity science and conservation.

Naturalist.org was launched in 2008 as a graduate project by Ken-ichi Ueda, Nate Agrin, and Jessica Kline of the UC Berkeley School of Information. After graduation, Nate and Ken-ichi continued to work on the site with the help of Sean McGregor. Ken-ichi started working with Scott Loarie when they founded iNaturalist, LLC in 2011 and began expanding the site through various partnerships. iNaturalist became an initiative of the California Academy of Sciences in 2014 and was launched in 2017 in conjunction with the National Geographic Society. In 2023, iNaturalist becomes an independent non-profit organization. Internationally, iNaturalist works with many different organizations through the iNaturalist Network, providing regional knowledge and greater reach and relevance.

1.2.3 MedLeaf

Introduction: MedLeaf does Medicinal Plant Identification Using Android Application Based on Leaf Image. Medicinal plants have been used for centuries in various traditional medicinal systems as sources of natural remedies for various ailments. However, with the advancement of technology, there is a growing interest in developing modern tools to aid in the identification of medicinal plants. In recent years, mobile applications that use image recognition algorithms have gained popularity, offering a convenient and accessible way to identify plants based on leaf images.

1. MedLeaf Project: The MedLeaf project aims to contribute to the field of medicinal plant identification by developing an Android application that leverages state-of-the-art image recognition algorithms. The app will allow users to snap pictures of plant leaves and quickly obtain information about the medicinal properties and potential uses of the identified plant. The success of such a project depends on the accuracy of the image recognition model and the comprehensive database of medicinal plants.

2. Database and Plant Information: A crucial aspect of the MedLeaf project is the database of medicinal plants. For accurate identification, the application must have a comprehensive collection of leaf images representing a wide range of medicinal plants. Additionally, the database should include detailed information about each plant's medicinal properties, active compounds, traditional uses, potential side effects, and precautions.

3. Challenges and Limitations: While plant identification applications have shown promise, they do face certain challenges and limitations. Some medicinal plants may have similar-looking leaves, making it challenging to distinguish between them accurately. Moreover, the accuracy of the image recognition model relies heavily on the quality of the leaf images captured by users, which may vary based on lighting conditions and camera quality.

Conclusion: The development of the MedLeaf Android application has the potential to revolutionize the way individuals access information about medicinal plants. By leveraging advanced image recognition algorithms and a robust medicinal plant database, this app can become a valuable tool for herbalists, researchers, and nature enthusiasts. Nevertheless, addressing the challenges and ensuring a comprehensive and accurate database will be critical to the success of the MedLeaf project.

1.3 Problem Statement

We aim to identify plants and find their medicinal uses. This is because we are surrounded by an abundant source of flora; but we are ignorant about them and their uses.

Our Web Application identifies a plant and suggest home remedies that uses the identified plant as an ingredient.

1.4 Objectives

Our model is to help identify the plants which may help the society to improvise the medical effects using these plants. This may help the medical laboratories to improve the medical contents in a medicine and its medicinal value.

- **Database** - To enter the login details into the database
- **Object recognition**- To recognize and identify a plant & its species
- **Image to text conversion** - the identified image to be converted into text
- **Medicine Recommendation**- To recommend medicines based on the extracted plants

1.5 Scope

- **Enhanced Accuracy** - Future progress could focus on improving the accuracy of plant identification algorithms. This will involve combining advanced machine learning techniques such as deep learning and neural networks to accurately and reliably identify facilities.
- **Mobile Integration** - Integration of plant identification with mobile devices enables real-time, mobile plant identification. Users just need to take a photo of the facility with their smartphones to get instant results.
- **Expanded Plant Database** - Keep expanding and updating facility information to improve app performance. The inclusion of a variety of plants, including rare and exotic species, makes the app more useful to a wider audience.
- **Environmental Monitoring**- Extending the app's functionality to include environmental monitoring allows users to provide information on plant distribution, weather, and habitat changes. This collection of information is essential for researchers and conservationists.

- **Augmented Reality (AR) and Virtual Reality (VR)**- Using AR or VR technology can provide users with immersive experiences. Through the interactive and interactive interface, they can see the plant in a realistic environment, explore the beautiful garden or access more information about the plant.

Chapter 2

Literature Review

2.1 PlantNet

PlantNet is a deep learning based smart phone app for large scale plant identification. Plant distinguishing proof and medication suggestion web applications have picked up critical consideration in later a long time due to their potential in giving important data for plant devotees, healthcare experts, and people inquisitive about home grown cures. This writing audit explores the existing investigate and advancements within the field, centering on plant distinguishing proof procedures, therapeutic plant databases, and the integration of plant distinguishing proof with medication suggestion frameworks.

- **Plant Identification Techniques** - Plant distinguishing proof and medication suggestion web applications have picked up critical consideration in later a long time due to their potential in giving important data for plant devotees, healthcare experts, and people inquisitive about home grown cures. This writing audit explores the existing investigate and advancements within the field, centering on plant distinguishing proof procedures, therapeutic plant databases, and the integration of plant distinguishing proof with medication suggestion frameworks.

  Image-based recognition systems using deep learning techniques such as convolutional neural networks (CNNs) have shown great results in accurately identifying plants based on leaf images or general plant morphology. For example, Mäder et al. (2017) developed PlantNet, a deep learning-based mobile application that provides high accuracy in plant identification. This automated process is efficient, scalable and usable by many users

Quattrocchi (2012) compiled the CRC Global Herbs and Herbs Dictionary, a comprehensive collection of scientific and medicinal uses and scientific and medicinal uses of plants. Jain and Jain (2013) developed a database of Indian herbs focusing on Indian herbs and their medicinal uses. This information can serve as important information for extracting information about herbs and their health benefits. Plant identification and drug approval integration: Integration of plant identification and drug approval processes has the potential to provide personalized and evidence-based medicine based on plants. These systems combine plant identification results with user data, health and research data to recommend appropriate herbs or treatments. Chen et al. (2019) developed a traditional Chinese medicine (TCM) herb identification and drug recommendation using case studies of psoriasis. The system integrates plant identification results, TCM information, and patient-specific information to recommend personalized herbs for psoriasis treatment. Oliveira et al. (2020) presented PhytoRecommend, a web application that combines plant identification, user preferences, and research data to provide plant-based drug recommendations. The integration of plant identification and drug recommendations can be further improved by integrating electronic health records (EHR) and clinical decision making (CDSS). By combining patient history and clinical criteria, the accuracy and reliability of drug recommendations can be improved.

2.2 Plants Leaves Recognition

Leaf recognition is a critical step in various applications, involving the capturing of leaf images, applying pre-processing techniques, extracting features, and finally classifying the leaves. The flowchart in Fig. 1 illustrates the major steps involved in the leaf recognition process.

A. Images Capturing

Researchers have used various devices for capturing leaf images in different studies. Cameras like Samsung DV300F, SONY W730, Microsoft Kinect 2.0, and MX808 have been utilized to capture images of different fruits and plants. Additionally, thermal cameras and laser scanners have been employed to collect 2D and 3D images for specific research purposes.

B. Images Pre-Processing Methods

The pre-processing phase plays a crucial role in the leaf recognition system. It involves several steps such as image re-orientation, cropping, grayscale and binary conversion, noise removal, contrast stretching, and threshold inversion. Machine learning methods have been used to develop efficient pre-processing techniques. Some studies have proposed dividing the leaf image into parts and extracting features like vein, color, and Fourier descriptors to achieve high accuracy. Other techniques, such as SLIC and Guided Active Contour (GAC), have been developed for segmentation and shape enhancement.

C. Feature Extraction Methods

Feature extraction is essential for leaf recognition, and characteristics such as color, size, shape, and texture are commonly used. Various statistical measures like Color Co-occurrence Matrix (CCM), Spatial Grey Level Dependence Matrix (SGLDM), Grey Level Co-occurrence Matrix (GLCM), and Local Binary Patterns (LBP) have been employed to extract textural features. Studies have also explored area labeling and Fourier descriptors for feature extraction, resulting in promising outcomes.

The use of feature extraction methods greatly impacts the accuracy and precision of the leaf recognition/classification systems built on machine learning mechanisms. The choice of feature extraction method affects the architecture of machine learning networks and, subsequently, the performance on different problems. Researchers have employed diverse feature extraction methods to cater to specific research objectives and achieve high recognition accuracy.

In summary, the literature review provides valuable insights into the different steps involved in leaf recognition, from image capturing to feature extraction. The studies demonstrate a wide range of approaches and methodologies, indicating the importance of tailoring the techniques to specific research requirements. The knowledge gained from these studies will inform the development of our own leaf recognition system, enabling us to contribute to this field of visual-based machine learning effectively.

2.3 Automated Real-Time Identification of Medicinal Plants Species in Natural Environment Using Deep Learning Models—A Case Study from Borneo Region

"Automated Real-Time Identification of Medicinal Plants Species in Natural Environment Using Deep Learning Models—A Case Study from Borneo Region" presents a significant contribution to the field of plant identification and medicinal applications. Here, we explore key studies related to plant species identification, deep learning models, and automated systems for medicinal plant recognition.

1. Plant Species Identification: Several studies have focused on plant species identification using various methods. Traditional approaches involve manual visual inspection or expert botanists' intervention, which can be time-consuming and subjective. To address this, researchers have explored computer vision techniques. For instance, "Plant Species Identification Using Convolutional Neural Networks" by Wei et al. (2016) demonstrated the use of CNNs for accurate and efficient plant classification based on leaf images. Similar studies by Majeed et al. (2018) and Chen et al. (2019) have also explored deep learning models for plant species recognition using leaf and flower images, respectively.

2. Deep Learning Models: Deep learning models, particularly Convolutional Neural Networks (CNNs), have shown remarkable success in various image recognition tasks. CNNs can automatically learn hierarchical features from images, enabling them to capture intricate patterns and distinguishing characteristics. Notable advancements in deep learning for image recognition have been showcased through models like AlexNet, VGG, ResNet, and Inception, which have achieved state-of-the-art results in large-scale image classification competitions like ImageNet.

3. Medicinal Plant Recognition: Automated recognition of medicinal plants is gaining importance due to the growing interest in traditional and alternative medicine. Studies such as "Identification of Medicinal Plants using Deep Convolutional Neural Networks" by Ataee et al. (2020) have demonstrated the potential of deep learning models in accurately identifying medicinal plants based on leaf images. Furthermore, "Computer Vision for Plant Recognition: A Survey" by Al-Tameemi et al. (2021) provides an extensive review of various computer vision techniques applied to plant recognition, including those for medicinal plants.

4. Real-Time and Natural Environment Applications: Efforts to deploy plant recognition systems in real-time and natural environments have been gaining traction. Researchers have explored deploying these systems on smartphones or portable devices, enabling users to identify plants on the spot. For instance, "Plant Identification in Natural Environment using MobileNet" by Hu et al. (2018) demonstrated a real-time plant recognition system on mobile devices using the MobileNet model. These real-time applications are crucial for field researchers, botanists, and individuals seeking immediate plant identification.

5. Borneo Region and Ethnobotanical Significance: The Borneo region, known for its rich biodiversity, hosts numerous medicinal plant species used by indigenous communities for centuries. Studies like "Ethnobotany of medicinal plants in the highlands of northern Borneo" by Chin et al. (2020) highlight the ethnobotanical significance of the region. The automated identification system proposed in the paper holds great promise for facilitating the discovery and documentation of medicinal plants in this ecologically diverse area.

In conclusion, the paper "Automated Real-Time Identification of Medicinal Plants Species in Natural Environment Using Deep Learning Models—A Case Study from Borneo Region" contributes to the growing body of research in automated plant species identification and its application in the context of medicinal plants. By leveraging deep learning models and exploring real-time deployment, this study opens new avenues for plant recognition in natural environments, with specific relevance to the ethnobotanical knowledge of the Borneo region.

2.4 Conclusion

The literature review highlights advances in identification technology, the development of plant information, and the integration of plant identification and drug recommendations. Automated plant information using computer vision and deep learning algorithms has shown great potential. Extensive herbal databases provide important information for drug recommendations. The integration of plant identification and suggested medicinal use is promising for the personalization of herbs and improving health.

Future research may focus on improving plant identification accuracy, expanding plant information, and conducting user studies to evaluate efficacy and usability. Additionally, integration with EHRs and CDSSs could increase the applicability of these systems in real clinical settings. With the continued development of technology and the interest in herbal medicine, plant identification and prescriptions on the web hold great promise in promotinzg evidence-based healthcare and facilitating the dissemination of plant knowledge.

Chapter 3

System Analysis

3.1 Expected System Requirements

The system of user which is a smart phone is expected to have the following features:

- Requirement of Internet connection for plant identification.
- Multi-core CPU, minimum 4GB RAM, and sufficient disk space for hosting the application code and assets.
- Powerful CPU and RAM, with enough disk space to store plant data, user information, and other relevant data.
- Suitable backend programming language and framework for robust web application development
- APIs and External Services to ensure the server can handle API requests and smoothly integrate external plant data and medicine recommendations services.
- User Authentication and Authorization to implement a system to control access to sensitive features and data.
- Plan for scalability to handle increased user traffic.

3.2 Feasibility Analysis

3.2.1 Technical Feasibility

- The project is technically feasible since a large majority of the population own smart devies, which meet the minimum requirements to run the web application.

- Smart Devices today come equipped with sufficient processing power, RAM, and storage capacity to handle the web application's frontend and backend operations.

3.2.2 Data Availability

- There is a wealth of plant and medicine-related data available from various online sources and databases, making it feasible to access comprehensive and up-to-date information for the application.
- Many APIs and services provide access to plant databases and medicine recommendations, facilitating smooth integration into the web application.

3.2.3 User Acceptance and Market Analysis

- The project is likely to have significant user acceptance, as it caters to individuals interested in plant identification and personalized medicine recommendations.
- Conducting market analysis and user surveys can further validate the potential demand for such an application.

3.2.4 Development Time and Resources

The development time for the web application should be manageable, given the availability of skilled developers and readily accessible technologies. The minimum hardware requirements for smart Devices ensure that the web application can be efficiently implemented without significant resource constraints.

3.2.5 Scalability

- The project's architecture should be designed with scalability in mind to accommodate potential growth in user traffic and data load.
- Cloud-based services and scalable databases can be employed to handle increased demand.

3.2.6 Integration with External Services

The availability of APIs and external services for plant data and medicine recommendations makes integration feasible and straightforward. Proper documentation and support from external providers will aid in the seamless integration process.

3.2.7 Economic Feasibility

The app can reduce the overhead of expense incurred by people in order to treat their wounds/diseases. The development of the web application is also zero budget as it was built using free resources.

3.3 Hardware Requirements

The following are the system requirements to develop the Unify App.

- Processor: Intel Core i3/i5/i7
- Hard Disk: Minimum 100GB
- RAM: Minimum 8GB RAM or higher

3.4 Software Requirements

The following are the softwares used in the development of the app.

Operating System: Windows

3.4.1 Development Tools and Languages

- Backend: Choose a programming language and framework suitable for building the backend of the web application.
- Frontend: Use HTML, CSS, and JavaScript for the frontend development.
- Database: Select a database management system like MySQL, or SQLite, depending on the data requirements and scalability needs

3.4.2 APIs and External Services

- Identify and integrate with external APIs or services that provide plant data and medicine recommendations. Ensure compatibility and smooth integration with your chosen backend and frontend technologies.APIs used are Plant.id API and chat.openai.com

Plant.id API

Plant.id is an API and service that offers plant identification functionality using machine learning algorithms. It allows developers to integrate plant identification capabilities into their applications or websites by sending an image of a plant and receiving back information about the plant species.

The API works by leveraging machine learning models to analyze the provided image and compare it to a large database of plant species to identify the closest matches. It can return information such as the common name, scientific name, family, and other relevant details about the plant.

chat.openai.com

chat.openai.com refers to the platform provided by OpenAI to access and interact with their language models, including GPT-3.5, which powers this conversation. OpenAI provides various API services that allow developers to integrate natural language processing and generation capabilities into their applications, products, and services.

The "chat.openai.com" platform enables developers to experiment and interact with OpenAI's language models through an interface that allows for text-based input and output. By using this platform, developers can explore the capabilities of the language models and understand how to use them effectively.

3.4.3 Development Environment

- Set up a development environment with code editors, version control systems (e.g., Git, Visual Studio Code), and local servers for testing and debugging

Visual Studio Code

Visual Studio Code (VS Code) is a popular, free, and open-source code editor developed by Microsoft. It is designed to be lightweight, highly customizable, and versatile, making it a preferred choice for many developers across various programming languages and platforms. Here are some key features of Visual Studio Code:

- Code Editing: VS Code offers a rich set of code editing features, including syntax highlighting, auto-completion, bracket matching, and code formatting for various programming languages.
- Extensibility: One of the standout features of VS Code is its vast extension ecosystem. Developers can install a wide range of extensions from the Visual Studio Code Marketplace to enhance functionality, add language support, integrate with version control systems, and more.
- Live Share: The "Live Share" feature allows developers to collaborate in real-time with others, enabling them to edit and debug code together in shared sessions.
- IntelliSense: VS Code offers intelligent code completion and suggestions (IntelliSense) based on the context of the code, enhancing productivity and reducing typing errors.
- Debugging: The editor supports built-in debugging for many programming languages, enabling developers to set breakpoints, inspect variables, and step through code for easy troubleshooting.
- Integrated Terminal: VS Code comes with an integrated terminal, allowing developers to run shell commands and scripts directly within the editor, eliminating the need to switch to a separate terminal window.

Chapter 4

Methodology

Block diagram + Explain each block.

4.1 Proposed Method

- Develop a web application that can recognize plants and suggest home remedies using the identified plant.
- We built model using two existing APIs: Plant.id and chat.openai API and then implemented the model by web development using HTML, CSS and Javascript that gives a better interface to our application.
- User gives an uploaded or captured photo as input and application gives the name of recognized plant as output.
- Application also contains features like View medicines, Get Plant etc. that helps our clients know more about the identified plant.

4.1.1 Plant Identification using Plant.id API

Introduction:

We are using Plant.id API as an integral component for Plant Identification. Plant.id is a plant identification service powered by deep convolutional network model. Plant.id has the capacity to recognize over 12,000 different types of plants, encompassing flowers, trees, bushes, fungi, and lichens from various regions around the globe. In addition to providing the Latin name, it offers details such as common names, a concise description, and the plant's taxonomy.

The Plant.id API is an application programming interface that employs machine learning for identifying plants. By submitting images of your plants, we can receive poten-

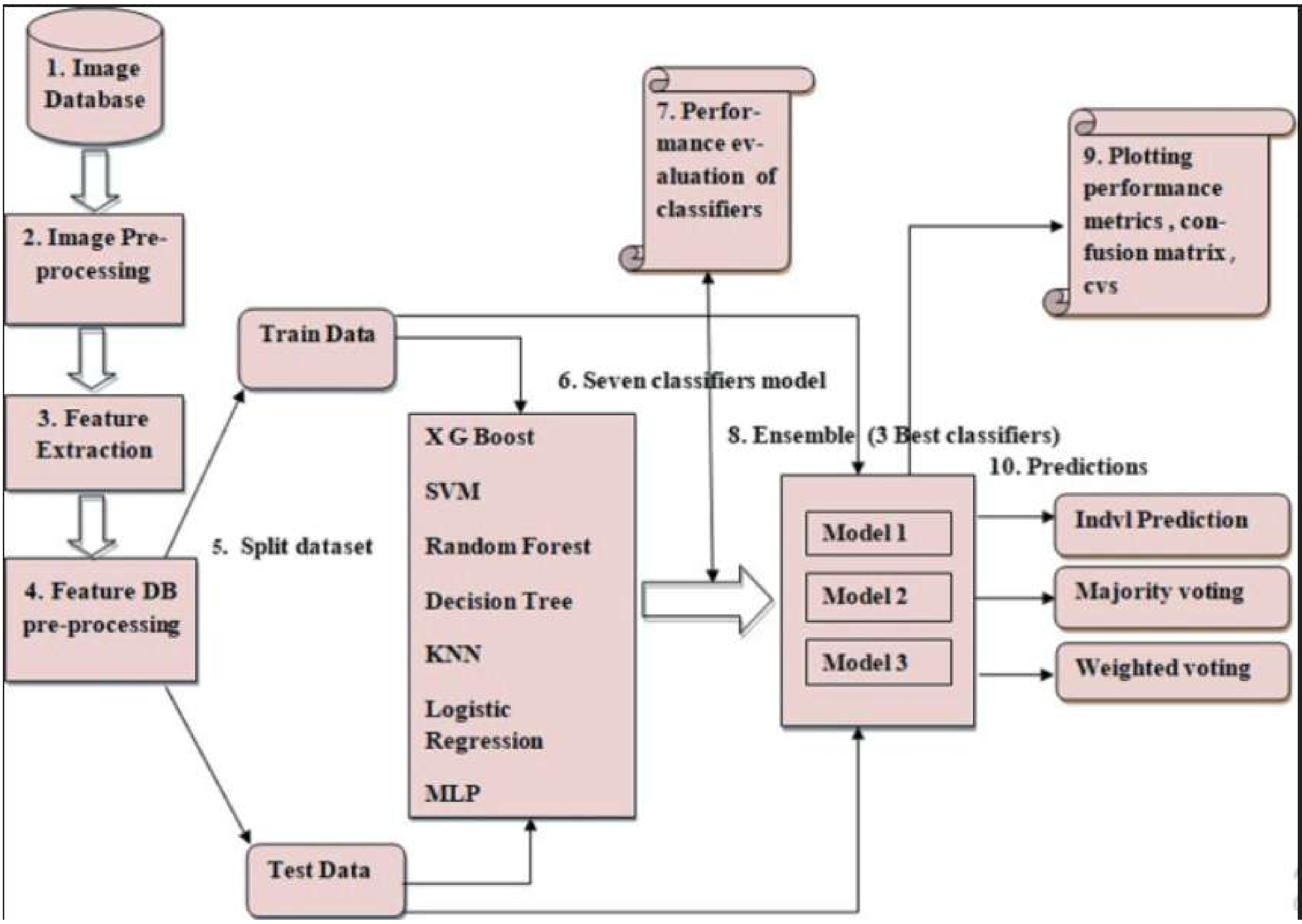

Figure 4.1: model structure - Plant.id API

tial species matches along with relevant details, including representative images of those species. This identification process is powered by advanced machine learning algorithms. The API has 90 percent accuracy for the top 3 plants. It is the probability that the correct plant is listed in the first 3 suggestions. The TOP1 accuracy is 79 percent and the TOP10 accuracy is 95 percent.

Features of Plant.ID API include: Unlimited scalability, Easy integration, Extensive documentation, Complex plant info, and Adaptive representative images

Data Collection:

To develop and evaluate the model, a diverse dataset of plant images will be collected. This dataset will comprise images of various plant species, capturing different parts of the plants, such as leaves, flowers, and stems. The dataset will be curated to ensure a comprehensive representation of plant species, considering variations in appearance, growth stages, and lighting conditions. Additionally, any necessary data preprocessing, such as resizing or normalization, will be performed to enhance model performance.

Integration of Plant.id API:

The core component of the proposed model is the integration of the Plant.id API. By making appropriate API calls, the model will send plant images to the API for identification. The API's advanced machine learning algorithms will process the images and generate potential matches for the plant species present in the images. The API will also provide accompanying information, such as common names, scientific names, and other relevant details about the identified species.

Model Architecture:

The model architecture will primarily focus on facilitating seamless communication with the Plant.id API and handling the responses effectively. The model will be designed to preprocess the user-submitted images, format them according to the API's requirements, and process the returned identification results. Additionally, if desired, the model may incorporate other machine learning techniques, such as feature extraction or image classification, to complement the Plant.id API's output and enhance overall accuracy.

Inference Process:

The inference process will involve the following steps:

User submits an image of a plant to the application. The proposed model receives the image and preprocesses it, if required. The model sends the preprocessed image to the Plant.id API through API calls. The API processes the image and returns potential matches along with relevant information. The model presents the identification results to the user, displaying the species suggestions and associated details.

Integration in our Application:

The proposed model is integrated into our web application to make it easily accessible to users. The application will allow users to upload images of plants through a user-friendly interface and promptly receive identification results. Integration with other gardening or plant-related apps could also be explored to expand the model's potential use cases.

Conclusion:

The proposed model harnesses the powerful Plant.id API for accurate and efficient plant identification. By integrating this API into the model, users can effortlessly identify various plant species and gain valuable insights into the plants they encounter. The model's successful integration and evaluation will lead to a valuable tool for both plant enthusiasts and professionals in botany, horticulture, and ecological research.

4.1.2 Medicine Recommendation using OpenAI API

Objective:

The primary objective of the proposed model is to develop an intelligent home remedies system that provides safe and effective remedies based on input plant names. By leveraging the OpenAI API, the model will take the name of a specific plant as input and generate informative responses that suggest home remedies utilizing the medicinal properties of the identified plant. The aim is to offer users a reliable and accessible resource for natural remedies that can address various health concerns.

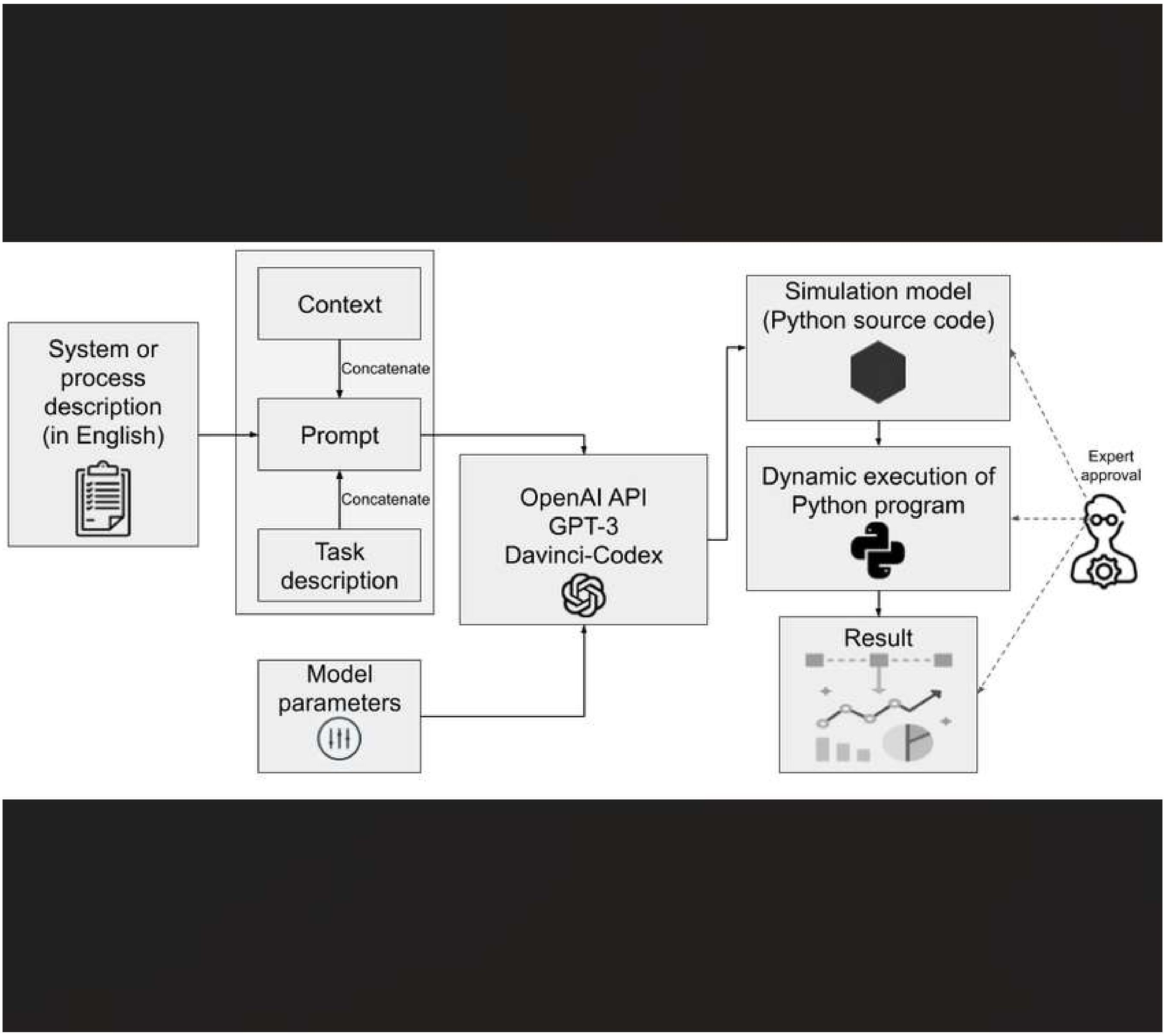

Figure 4.2: model structure - OpenAI API

Data Collection:

To ensure the model's effectiveness, an extensive dataset will be compiled, consisting of diverse plant names and their corresponding medicinal properties. The data collection process will involve sourcing information from reputable botanical resources, herbal knowledge databases, and scientific literature. Additionally, crowd-sourced data from herbalists and traditional medicine practitioners may be included to enrich the dataset with practical and traditional knowledge.

Preprocessing and Feature Extraction:

Before feeding the plant names to the model, preprocessing steps will be implemented to standardize the input format, remove any irrelevant characters, and handle misspellings or variations in plant names. Feature extraction will involve retrieving relevant information about the plant's medicinal benefits, historical uses, potential side effects, and any known contraindications to ensure accurate and informative responses.

Model Architecture:

The proposed model will utilize a state-of-the-art neural network architecture, such as GPT-3.5 or similar models, available through the OpenAI API. These models have demonstrated exceptional performance in natural language processing tasks, making them well-suited for generating coherent and contextually appropriate home remedies.

Training Data:

The training dataset will consist of a diverse range of plant names and their associated home remedies or medicinal information. This data will be carefully curated to ensure its accuracy and reliability. Human experts and domain specialists will review and validate the dataset to maintain the highest quality of information.

Training Process:

The model will undergo a rigorous training process that may involve fine-tuning it on specific home remedy-related tasks. Transfer learning techniques may be applied to adapt the model to the unique context of home remedies generation. The training process will focus on optimizing the model to provide safe and effective remedies while avoiding irrelevant or potentially harmful suggestions.

Output Generation:

When a user submits a plant name, the model will process the input and generate home remedies based on the plant's medicinal properties and historical uses. The responses will be structured to provide step-by-step instructions, dosage recommendations, and safety precautions for each suggested remedy. The model will also consider the user's specific health conditions or allergies, tailoring the remedies accordingly.

Testing and Validation:

The model's performance will be thoroughly tested and validated using various test cases and user feedback. This validation process will ensure that the model consistently produces accurate and helpful responses for a wide range of plant inputs. User feedback will be continuously gathered and analyzed to improve the model's accuracy and responsiveness.

API Integration:

The model will be integrated with the OpenAI API to handle input and output interactions. Users will be able to submit plant names through a user-friendly interface, and the model's responses will be seamlessly displayed back to the users through the API integration.

User Interface (UI):

To enhance user experience, a user-friendly interface will be designed, allowing users to easily input plant names and view the generated home remedies. The UI will be intuitive, responsive, and visually appealing, making it accessible to users of all levels of technological proficiency.

By implementing these components in the proposed model, we aim to create a comprehensive and trustworthy home remedies system from the identified plant.

Chapter 5

System Design

Draw usecase diagrams, activity diagrams, sequence diagrams etc. Add a brief explanation for each UML diagram.

5.1 Architecture Diagram

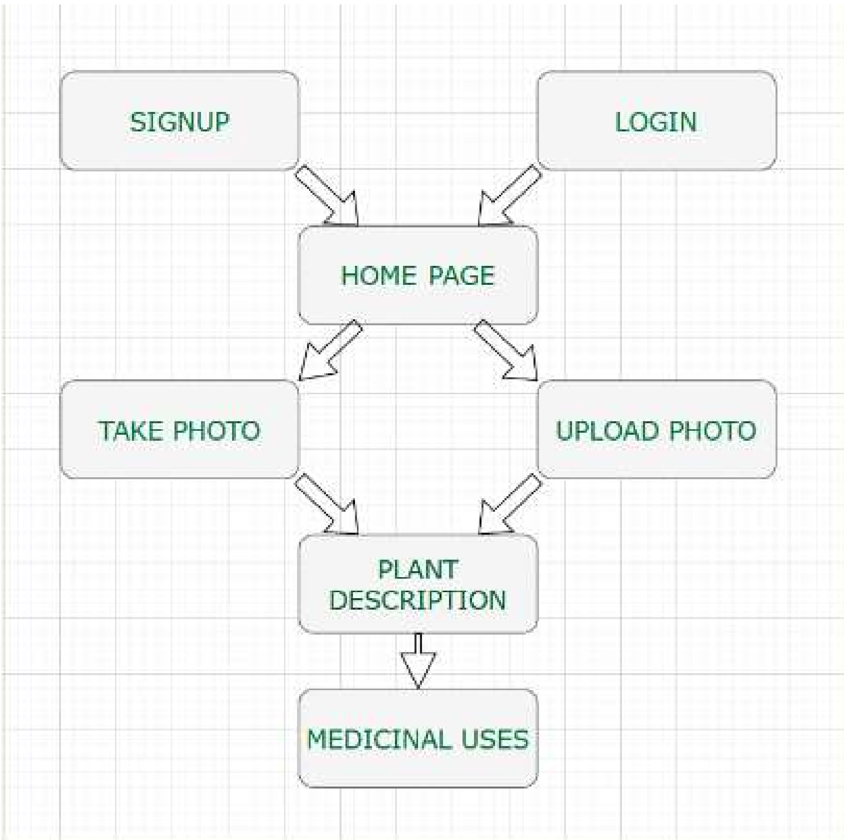

Figure 5.1: Architecture diagram

5.2 Sequence diagram

5.2.1 Login

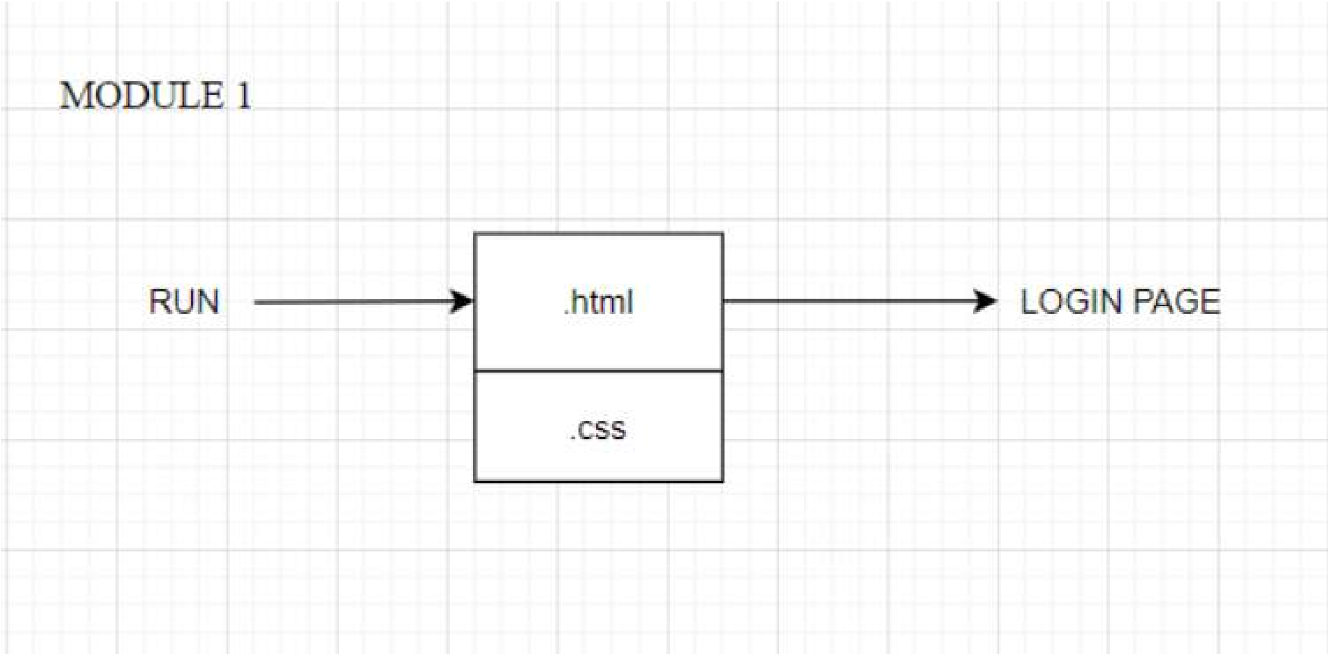

Figure 5.2: Login

5.2.2 Database Connection

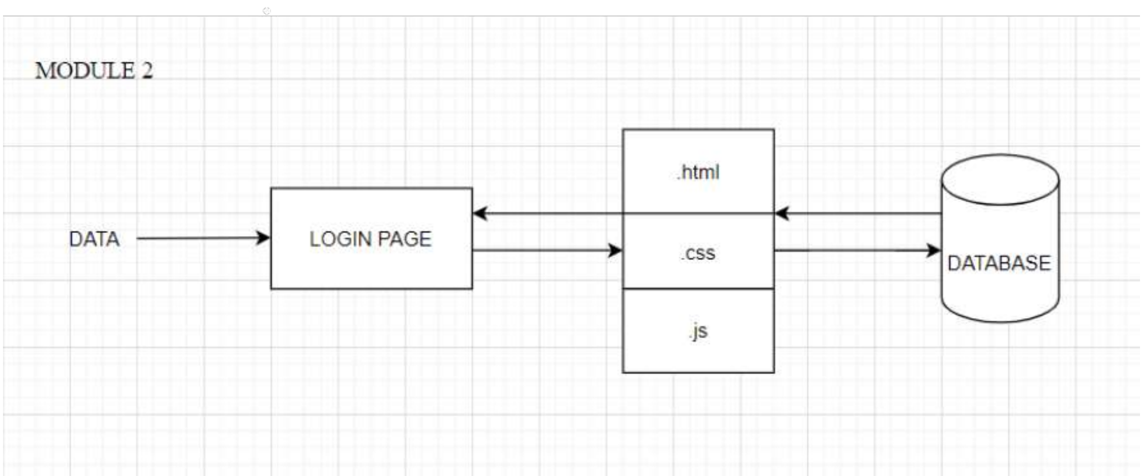

Figure 5.3: Database Connection

5.2.3 Plant Identification

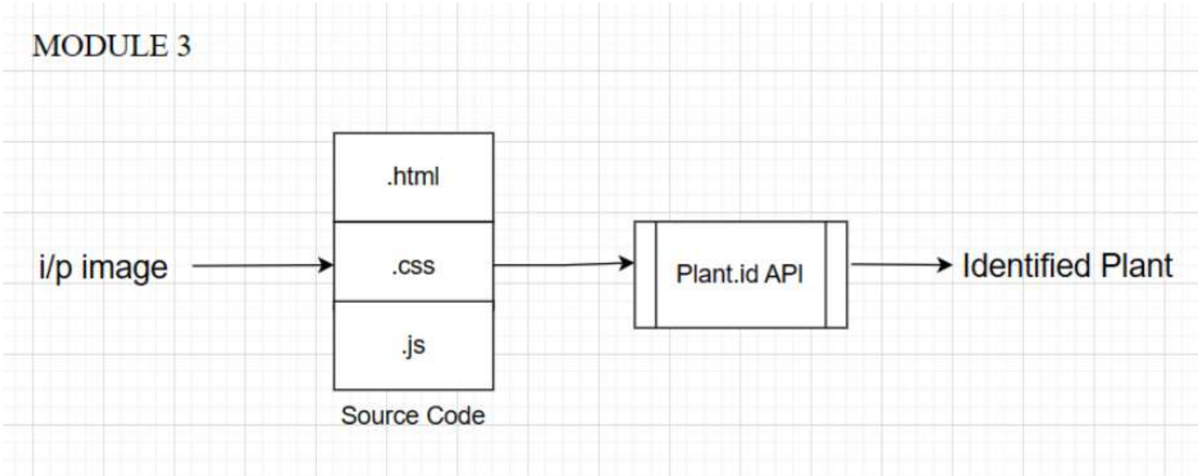

Figure 5.4: Plant Identification

5.2.4 Medicine Recommendation

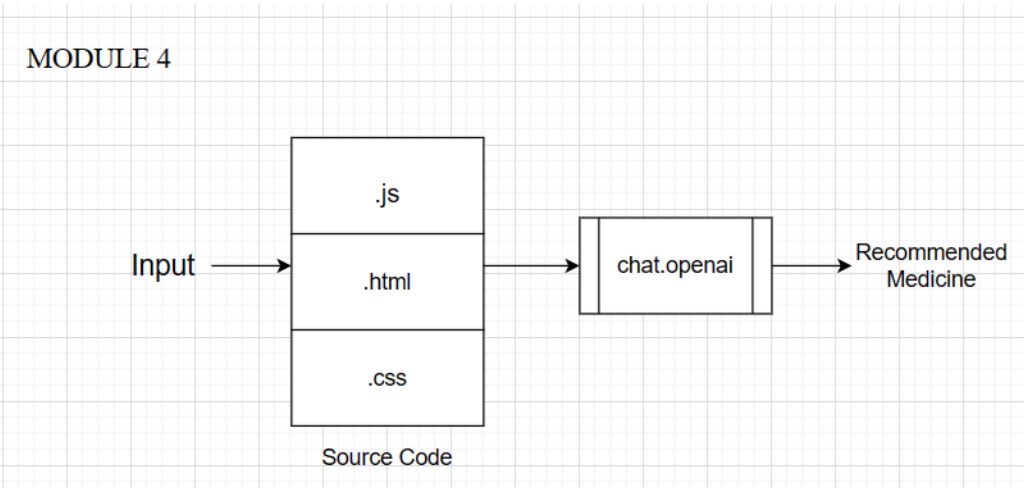

Figure 5.5: Medicine Recommendation

Chapter 6

System Implementation

Explain the various modules in the system in detail + Code.

6.1 Model Architecture

Model architecture for Plant Identification and Medicine Recommendation Web Application using API:

Plant Identification Model: - Utilize an external plant identification API that provides plant recognition capabilities based on machine learning algorithms.

- Input: Images of plants uploaded/taken live by users through the web application.

- Processing: The web application sends the user-uploaded images to the external API for analysis and identification.

- Output: The API returns the identified plant species and related information to the web application.

Medicine Recommendation Model: - Leverage an external medicine recommendation API that offers medical information and personalized medication suggestions.

- Input: System-provided name of the plant obtained at the identification level.

- Processing: The web application sends inputs to the external API for analysis and medicine recommendations.

- Output: The API provides relevant medicine recommendations based on the input data.

Web Application Integration: - User Interface: Design an intuitive and user-friendly interface where users can upload plant images and input their health-related information.

- Frontend Development: Utilize HTML, CSS, and JavaScript to create the user interface and capture user inputs.

- Backend Development: Implement server-side logic to handle user requests, interact with the external APIs, and process responses.

- API Integration: Integrate the external plant identification and medicine recommendation APIs into the backend of the web application.

- Data Flow: The plant images are sent to the plant identification API, and the health-related information is sent to the medicine recommendation API.

- Data Presentation: Display the identified plant species and related information along with the personalized medicine recommendations to the user.

By leveraging external APIs for plant identification and medicine recommendation, this model architecture simplifies the development process, allowing developers to focus on building the user interface, backend logic, and user experience. It also enables the web application to leverage existing machine learning and medical knowledge, providing accurate and reliable results to users.

6.2 Login

The user initiates the login process by accessing the web application's login page and providing their credentials i.e., email/username and password The frontend of the web application performs basic validation of the user's input, ensuring that required fields are not empty and that the input formats are valid.After the user submits the login form, the web application sends an authentication request to the backend server.The request includes the user's credentials (i.e., email and password) securely encrypted. The backend server receives the authentication request and verifies the user's credentials.The server checks the provided email/username and password against the stored user data in the database.If the credentials are valid, the user is authenticated.Upon successful login, the web application may redirect the user to the homepage

6.2.1 Database Connection

A database connection is established for the working of the login page which consist of a perfect graphical user interface i.e. the sign up and sign in.

Sign up The entered details are stored into the firebase database which makes it more easier for the admin to manage the access between the users.

Sign in The details which are stored in the firebase database are retrieved so that the user can login using those details to get into the home page

6.3 Plant Identification

User uploads/takes photo of the plant when take photo or upload photo button is clicked which then gets redirected to the "plant.id" service when identify plant button is pressed through a web application or API.The input image may undergo preprocessing to standardize its size, resolution and orientation, ensuring uniformity in the analysis.The service uses a trained machine learning model, often based on convolutional neural networks (CNNs), to extract features from the input image. CNNs excel at identifying patterns and features in images, allowing the model to capture unique characteristics of plant species. The extracted features are passed through the trained model, which makes predictions based on the learned patterns.The model outputs the probabilities of the input image belonging to different plant species or categories.The service may return the top-K predictions with their associated probabilities, where K represents the number of potential plant matches.The higher the probability, the more confident the model is about the prediction.The service may apply post processing techniques to refine the results and ensure accuracy. For example, it may use a confidence threshold to filter out low-confidence predictions. A confidence threshold helps avoid ambiguous or uncertain predictions, providing users with more reliable results.The service returns the results to the user, displaying the predicted plant species along with relevant information such as the common name, scientific name, and other characteristics.

6.4 Medicine Recommendation

The API recommends uses of plants and the process involves several steps to generate helpful and relevant responses.The overview of what happens when a user submits a query or request related to plant uses which API receives as the input text.Leverages its language understanding capabilities to interpret the user's query and identify the intent behind the request for plant uses.chat.openai has been trained on a vast dataset that includes information about various plants and their uses. This training enables the model to generate informative and contextually appropriate responses. Based on the user's query and the knowledge it has acquired during training, Chat.openai generates a response that suggests potential uses of the specified plant.Before delivering the response to the user, the system applies safety and policy filters to ensure that the output adheres to guidelines, community standards, and ethical considerations.chat.openai provides the generated response, which includes recommendations for the uses of the requested plant or plants. .

Chapter 7

Testing

Different types of testing we have done (Unit Testing, Integration Testing, System Testing, Acceptance Testing, Security Testing, User Interface Testing). We have summarized the results.

7.1 Functional Testing

Ensured that each function of the web application is working as expected. It involves testing features like plant identification, medicine recommendation, user registration, login, and other functionalities.

7.2 Usability Testing

Evaluated the user interface (UI) and user experience (UX) of the application. Checked that the interface is intuitive, easy to navigate, and visually appealing.

7.3 Compatibility Testing

Tested the application on different web browsers (e.g., Chrome, Firefox) to ensure it functions correctly across various platforms.

7.4 Performance Testing

Evaluated the performance of the application to determine its responsiveness and loading speed. Tested the application under different traffic conditions to check for any performance bottlenecks.

7.5 Security Testing

Assessed the application's security measures to protect user data and prevent unauthorized access.

7.6 Accuracy of Plant Identification

Tested the accuracy of the plant identification by providing known plant images and checking if the application correctly identifies them.

7.7 Medicine Recommendation Testing

Verified the medicine recommendation functionality by inputting various plant identifications and ensuring the system suggests appropriate medicines.

7.8 Database Testing

Checked the integrity of the database, including data insertion, retrieval, and updating. Ensured that the data is stored correctly and consistently.

7.9 Error Handling

Tested the application to see how it handles various types of errors and exceptions. Made sure users receive meaningful error messages when something goes wrong.

7.10 Stress Testing

Subjected the application to a high load to assess its performance under stress.

7.11 Regression Testing

After fixing bugs or making updates, performed regression testing to ensure that the changes have not adversely affected existing functionalities.

7.12 User Acceptance Testing (UAT)

involved real users to perform UAT. Gathered feedback from them to understand their overall satisfaction with the application and identify any additional improvements.

Chapter 8

Results

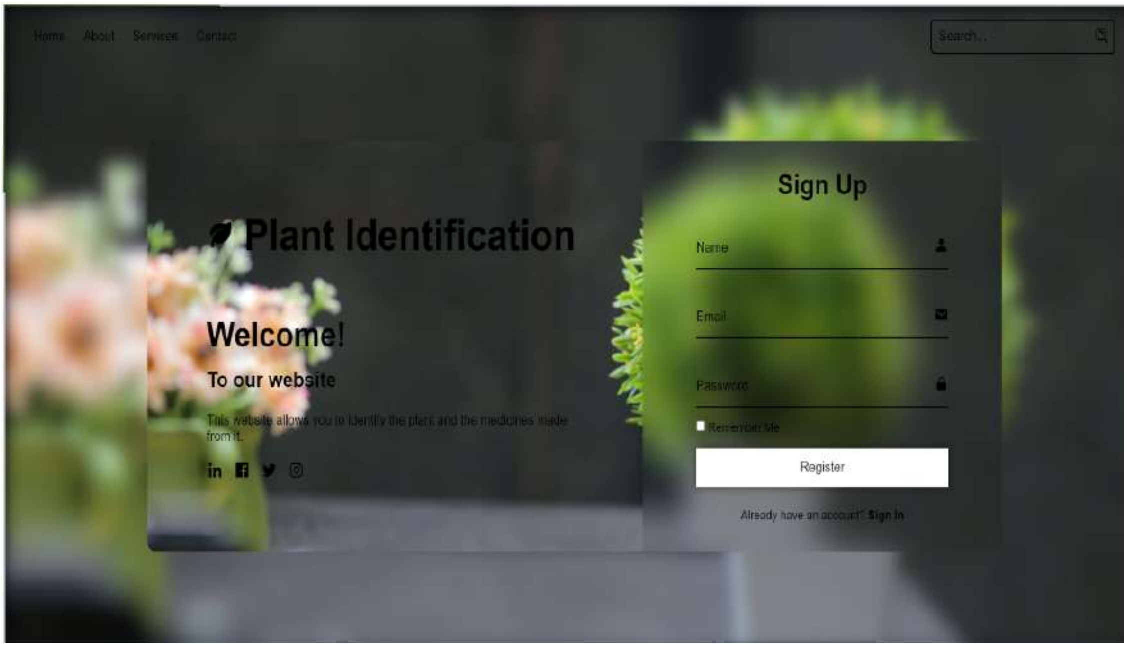

Figure 8.1: UI for Login

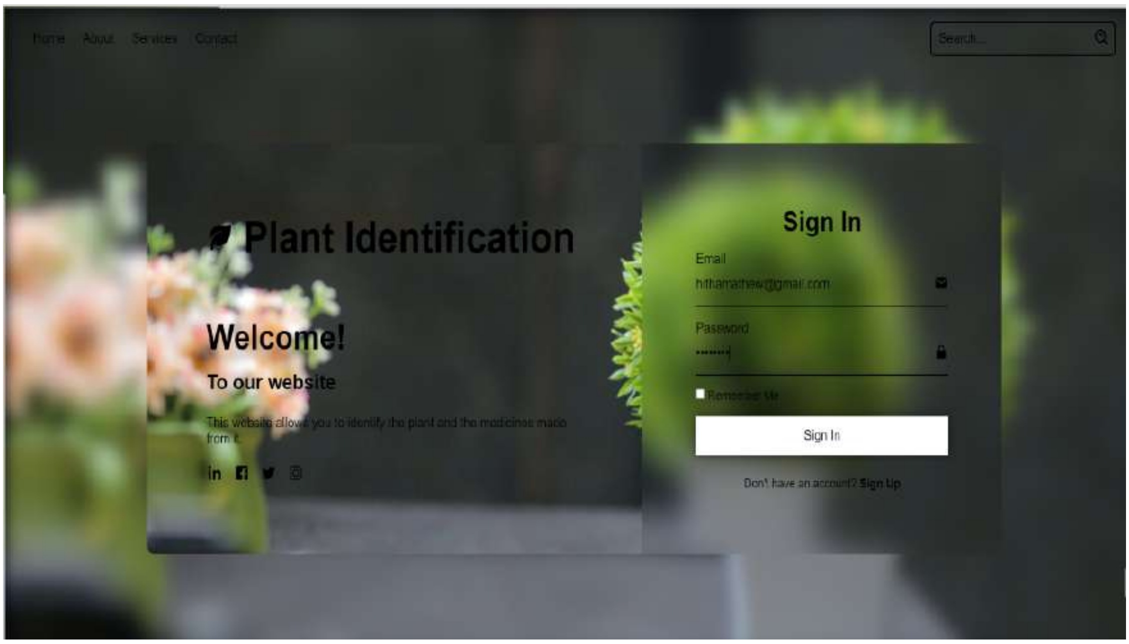

Figure 8.2: Login Details

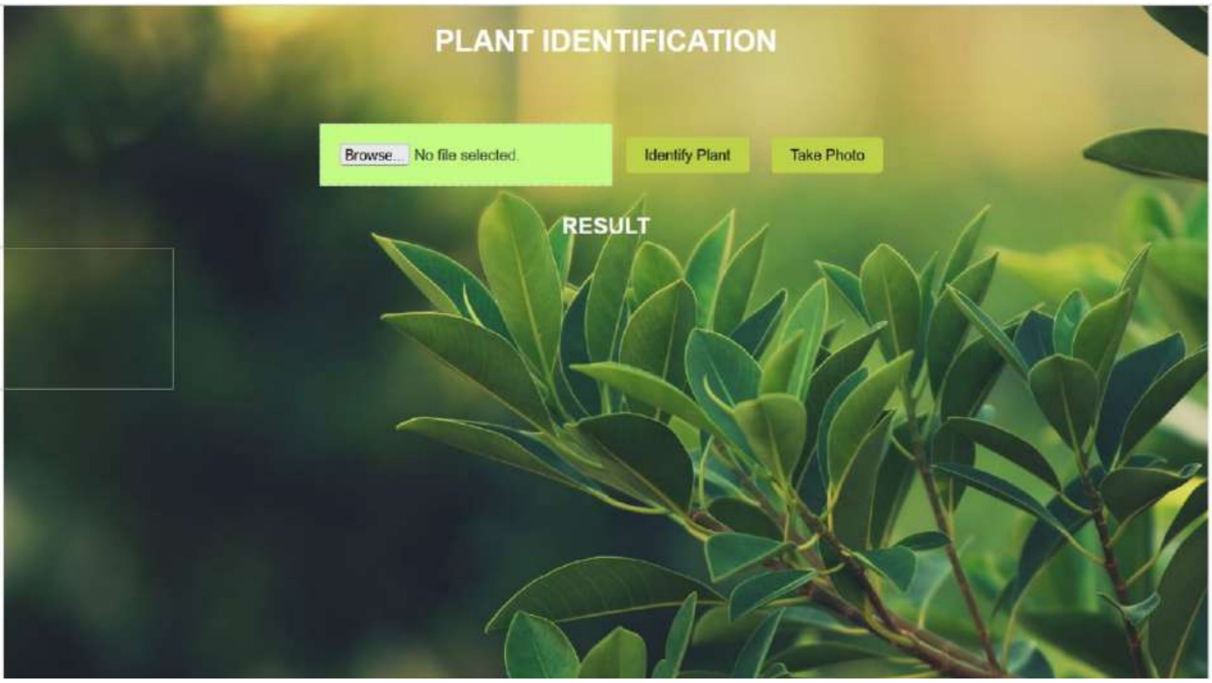

Figure 8.3: UI for Homepage

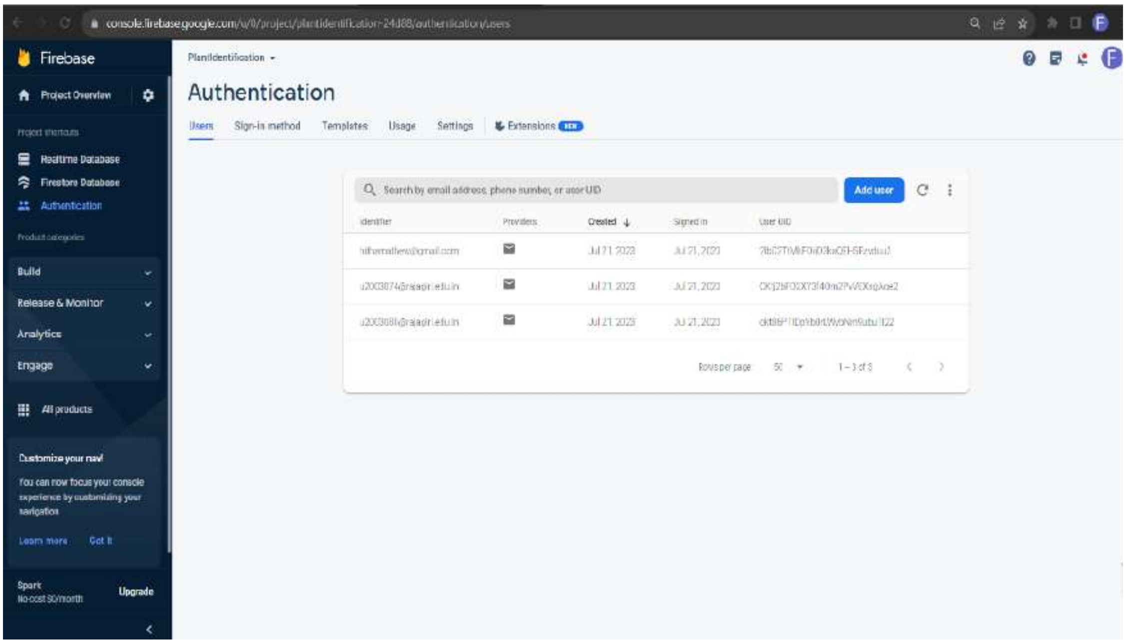

Figure 8.4: Database Connection

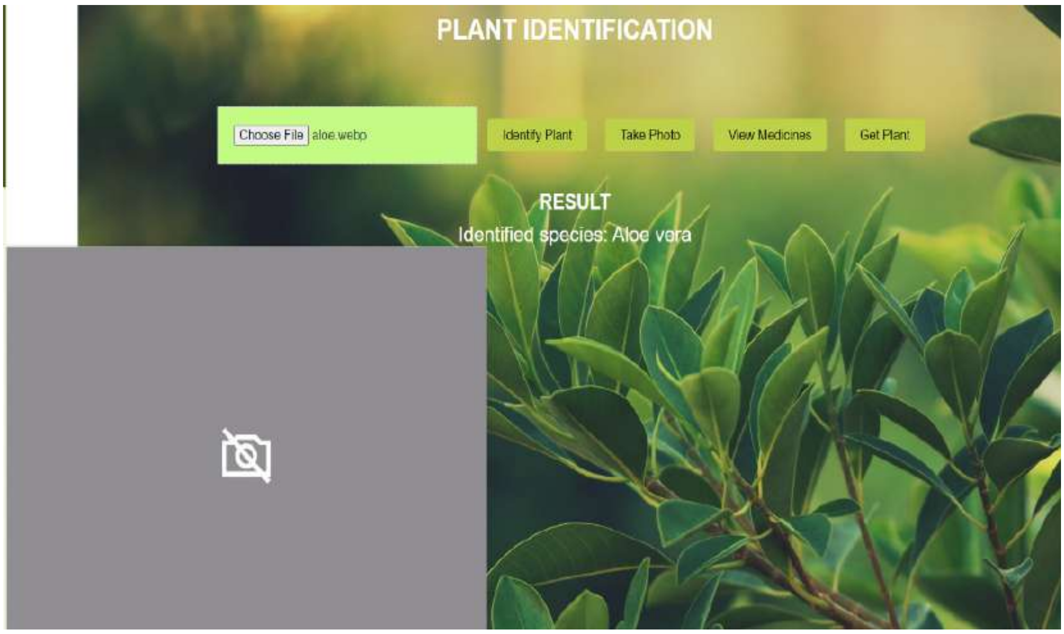

Figure 8.5: Output Of Upload Photo

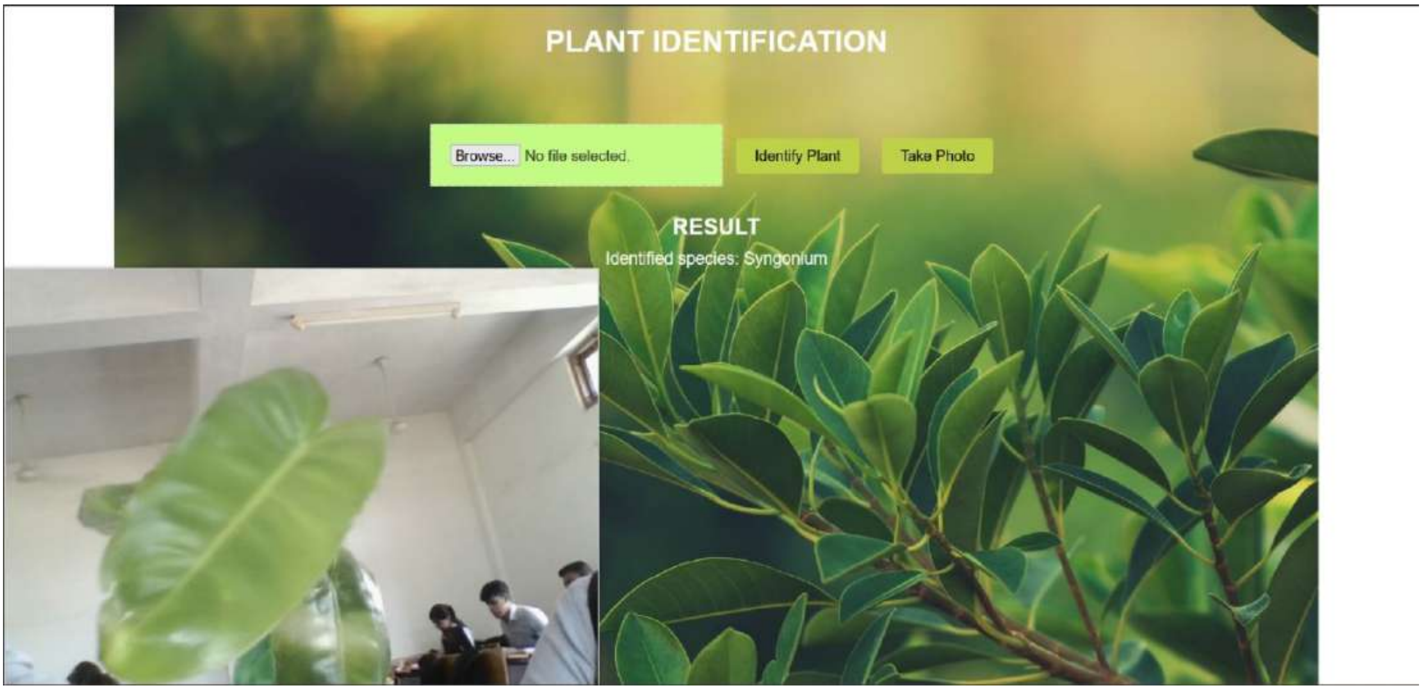

Figure 8.6: Output of Take Photo Option

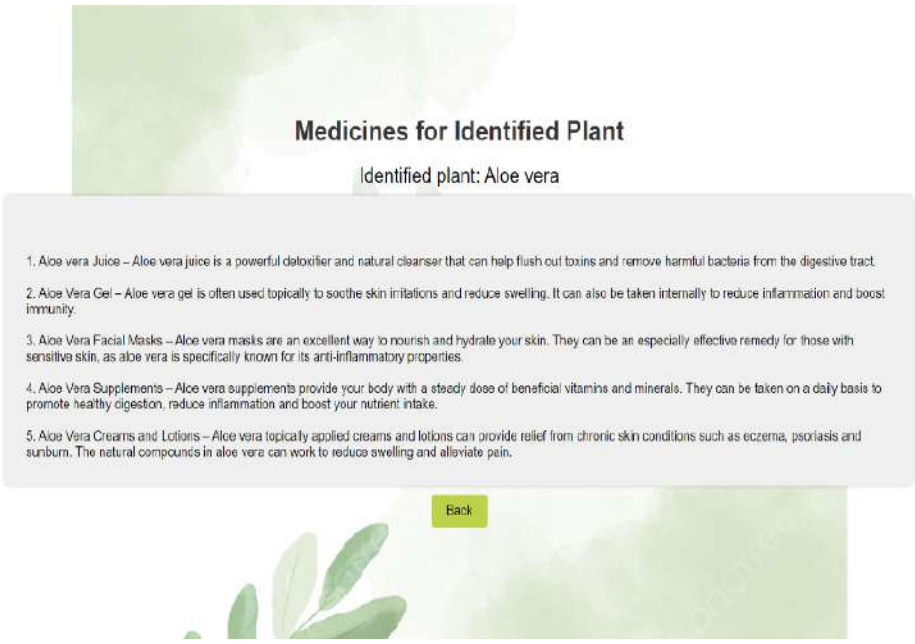

Figure 8.7: UI for Homepage

Chapter 9

Risks and Challenges

The development and implementation of a Plant Identification and Medicinal Recommendation Web Application present numerous risks and challenges that require careful consideration. These key considerations encompass various aspects, ranging from the accuracy and reliability of algorithms to legal and ethical concerns. Addressing these challenges is crucial for the application's success and ensuring the safety and well-being of its users. Here, we outline some of the primary risks and challenges that need to be navigated throughout the development and operation of the application.

- Accuracy and Reliability: One of the most significant challenges is ensuring the accuracy and reliability of the plant identification and medicine recommendation algorithms. The application's success heavily depends on providing correct information to users. Inaccurate plant identification or incorrect medicine recommendations could have severe consequences on users' health and well-being, leading to potential legal and reputational risks for the application.

- Data Privacy and Security: Handling sensitive user data, such as medical history and location, requires robust data privacy and security measures. The application must comply with relevant data protection regulations to safeguard user information from unauthorized access, breaches, or misuse.

- Liability and Legal Concerns: Providing medical recommendations implies a level of responsibility and potential liability for any adverse outcomes experienced by users following the app's advice. The application's developers must carefully consider disclaimers, terms of use, and appropriate warnings to mitigate legal risks.

- Database Management: Building a comprehensive database of plant species, medicinal properties, and potential interactions with other medications can be a complex and time-consuming task. Keeping the database up to date and accurate is crucial for the application's success.

- User Education: To ensure users use the application responsibly, it is essential to provide clear instructions and educational material on the limitations and potential risks of relying solely on the app's recommendations. Encouraging users to consult healthcare professionals before making any decisions based on the app's suggestions is crucial.

- Scaling and Performance: As the user base grows, the application should be able to handle increased traffic and usage. Ensuring scalability and optimal performance is vital to maintain a positive user experience.

- Updating Regulations: The regulations related to plant identification and medicine recommendations might change over time. Developers must stay updated with relevant laws and adapt the application accordingly to remain compliant.

- Ethical Considerations: The app must be designed ethically, ensuring that it promotes responsible use of medicinal information and respects cultural practices related to herbal medicine. Care must be taken to avoid promoting the overuse or exploitation of plant resources.

- User Engagement: Getting users to regularly use the application and provide feedback can be challenging. Without active engagement, the app's database may not receive the necessary user-generated content and updates, impacting the application's overall effectiveness.

Addressing these risks and challenges will contribute to the application's viability and trustworthiness, providing users with a valuable and safe tool for plant identification and medicinal recommendations. The successful management of these considerations will also foster a deeper appreciation and understanding of the natural world and its botanical diversity.

Chapter 10

Conclusion

The development and implementation of PlantSense, the Plant Identification and Medicine Recommendation Web Application have been a significant undertaking, aimed at providing users with a valuable tool for identifying plants and accessing relevant medicinal information. Throughout this mini-project report, we have outlined the key features, design considerations, and challenges associated with creating such an application.

The application's core functionality, plant identification, has been carefully developed and fine-tuned to ensure accuracy and reliability. Users can now confidently identify various plant species, empowering them to explore and interact with the natural world more effectively. The additional feature of medicine recommendation based on identified plants further enhances the application's utility and benefits, providing valuable insights into traditional and alternative medicine.

Looking ahead, future iterations of the Plant Identification and Medicine Recommendation Web Application hold great promise. By incorporating machine learning advancements and harnessing user-generated content, the accuracy and scope of plant identification and medicine recommendations can be further enhanced. Scaling the application to accommodate a growing user base is another crucial consideration to maintain optimal performance and seamless user experience. We will continuously strive to promote the responsible use of medicinal information, respecting cultural practices and safeguarding the environment by encouraging sustainable practices.

In conclusion, the Plant Identification and Medicine Recommendation Web Application represents a valuable tool for plant enthusiasts, botanists, and individuals interested in the world of herbal medicine. As we continue to refine and expand the application's capabilities, we aim to foster a deeper connection between humans and nature, ultimately contributing to a healthier and more sustainable world.

References

[1] Malik OA, Ismail N, Hussein BR, Yahya U. Automated Real-Time Identification of Medicinal Plants Species in Natural Environment Using Deep Learning Models-A Case Study from Borneo Region. Plants (Basel). 2022 Jul 27;11(15):1952. doi: 10.3390/plants11151952. PMID: 35956431; PMCID: PMC9370651.

[2] Weng SW, Chen BC, Wang YC, Liu CK, Sun MF, Chang CM, Lin JG, Yen HR. Traditional Chinese Medicine Use among Patients with Psoriasis in Taiwan: A Nationwide Population-Based Study. Evid Based Complement Alternat Med. 2016;2016:3164105. doi: 10.1155/2016/3164105. Epub 2016 Oct 16. PMID: 27822287; PMCID: PMC5086381.

[3] Sharma V, Sarkar IN. Bioinformatics opportunities for identification and study of medicinal plants. Brief Bioinform. 2013 Mar;14(2):238-50. doi: 10.1093/bib/bbs021. Epub 2012 May 15. PMID: 22589384; PMCID: PMC3603214.

[4] Mader,P.,et al.(2017).PlantNet:A deep learning based smart phone app for large scale plant identification.Ecological Informatics,40,24-29

[5] Wäldchen J, Mäder P. Plant Species Identification Using Computer Vision Techniques: A Systematic Literature Review. Arch Comput Methods Eng. 2018;25(2):507-543. doi: 10.1007/s11831-016-9206-z. Epub 2017 Jan 7. PMID: 29962832; PMCID: PMC6003396.

[6] Quattrocchi, U. (2012). CRC World Dictionary of Medicinal and Poisonous Plants: Common Names, Scientific Names, Eponyms, Synonyms, and Etymology (5 Volume Set). CRC Press.

[7] Jain, S., and Jain, R. (2013). Database of Medicinal Plants in India: Volume 1 (A-D). Today and Tomorrow's Printers and Publishers.

[8] V Zangeneh, E., Rahmati, M. and Mohsenzadeh, Y., 2020. Low resolution face recognition using a two-branch deep convolutional neural network architecture. Expert Systems with Applications, 139, p.112854.

[9] Priya, K., Ranjana, S., and Manimegala, R. (2020). MEDICINAL PLANT IDENTIFICATION USING ANDROID APPLICATION BASED ON LEAF IMAGE. European Journal of Molecular and Clinical Medicine, 7(9), 1496-1506. doi: 10.31838/ejmcm.07.09.161

[10] Malik OA, Ismail N, Hussein BR, Yahya U. Automated Real-Time Identification of Medicinal Plants Species in Natural Environment Using Deep Learning Models-A Case Study from Borneo Region. Plants (Basel). 2022 Jul 27;11(15):1952. doi: 10.3390/plants11151952. PMID: 35956431; PMCID: PMC9370651

[11] Pudaruth, Sameerchand and Mahomoodally, Fawzi and Kissoon, Noushreen and Chady, Fadil. (2021). MedicPlant: A mobile application for the recognition of medicinal plants from the Republic of Mauritius using deep learning in real-time. IAES International Journal of Artificial Intelligence (IJ-AI). 10. 938. 10.11591/ijai.v10.i4.pp938-947.

[12] Dandu Sai Satya Vamsi Krishna Raju , Ediga Vinod Kumar , Ediga Mahendra Goud , Dr. Sharmasth Vali Y "A Novel Approach to Identify Medicinal Plant Based On Image Processing" Iconic Research And Engineering Journals, 6(11)

[13] B Veselý, Ondřej and Vild, Ondřej and Stastny, Jiri. (2014). A Novel Approach for the Increase in Student's Learning Motivation. Řihák, Jiří and Vild, Ondřej and Veselý, Ondřej. (2019). Deeper evaluation of Plant.id.

[14] H Jackson, Ilya and Saenz, Maria. (2022). From Natural Language to Simulations: Applying GPT-3 Codex to Automate Simulation Modeling of Logistics Systems.

[15] Kapoor, L.D. (2001). Handbook of Ayurvedic Medicinal Plants: Herbal Reference Library (1st ed.). Routledge. https://doi.org/10.1201/9780203719473

Appendix A: Base Paper

plants

MDPI

Article

Automated Real-Time Identification of Medicinal Plants Species in Natural Environment Using Deep Learning Models—A Case Study from Borneo Region

Owais A. Malik [1,*], Nazrul Ismail [1], Burhan R. Hussein [1] and Umar Yahya [2]

[1] School of Digital Science, Universiti Brunei Darussalam, Jln Tungku Link, Gadong BE1410, Brunei; 20m8540@ubd.edu.bn (N.I.); 18h8338@ubd.edu.bn (B.R.H.)

[2] Department of Computer Science and Information Technology, Islamic University in Uganda, Kampala P.O. Box 7689, Uganda; umar.yahya@ieee.org

* Correspondence: owais.malik@ubd.edu.bn

Abstract: The identification of plant species is fundamental for the effective study and management of biodiversity. In a manual identification process, different characteristics of plants are measured as identification keys which are examined sequentially and adaptively to identify plant species. However, the manual process is laborious and time-consuming. Recently, technological development has called for more efficient methods to meet species' identification requirements, such as developing digital-image-processing and pattern-recognition techniques. Despite several existing studies, there are still challenges in automating the identification of plant species accurately. This study proposed designing and developing an automated real-time plant species identification system of medicinal plants found across the Borneo region. The system is composed of a computer vision system that is used for training and testing a deep learning model, a knowledge base that acts as a dynamic database for storing plant images, together with auxiliary data, and a front-end mobile application as a user interface to the identification and feedback system. For the plant species identification task, an EfficientNet-B1-based deep learning model was adapted and trained/tested on a combined public and private plant species dataset. The proposed model achieved 87% and 84% Top-1 accuracies on a test set for the private and public datasets, respectively, which is more than a 10% accuracy improvement compared to the baseline model. During real-time system testing on the actual samples, using our mobile application, the accuracy slightly dropped to 78.5% (Top-1) and 82.6% (Top-5), which may be related to training data and testing conditions variability. A unique feature of the study is the provision of crowdsourcing feedback and geo-mapping of the species in the Borneo region, with the help of the mobile application. Nevertheless, the proposed system showed a promising direction toward real-time plant species identification system.

Keywords: deep learning; medicinal plants; species identification; computer vision; real-time system; mobile application

Citation: Malik, O.A.; Ismail, N.; Hussein, B.R.; Yahya, U. Automated Real-Time Identification of Medicinal Plants Species in Natural Environment Using Deep Learning Models—A Case Study from Borneo Region. *Plants* **2022**, *11*, 1952. https://doi.org/10.3390/plants11151952

Academic Editors: Ana Barradas, Jorge Marques da Silva, Pedro Mariano, Tae-Hyuk Ahn and Georgios Koubouris

Received: 15 June 2022
Accepted: 22 July 2022
Published: 27 July 2022

1. Introduction

Plant species' identification is a challenging task that has a key role in effectively studying biodiversity and investigating unknown species. The manual identification of plant species is a time-consuming process and requires a lot of expertise in the field. Automated identification systems based on computer vision and machine learning techniques provide an alternative and assistive approach for this task. These systems are useful, but their accuracy varies due to the diversity of the species. A recent survey highlighted the growing application of machine learning (ML) and deep learning (DL) for plants' identification through leaves [1]. Integrated with Mobile applications, ML and DL techniques are increasingly being applied to distinguish between diseased and healthy plants [2–7], identification and classification of herbs and medicinal plants [3–10], classification of both generic plants and specific plant species [11–19], identification of crop-specific diseases [20], and general identification of plants to guide field tours [21].

Convolutional neural networks (CNNs) and the various deep CNN models have been reported to be the most commonly used methods in the automation process of plant-classification tasks [1]. An automated diagnosis of the 10 most common tomato leaf diseases, using a mobile application, was conducted in Reference [2], using the MobileNet model of CNN, and an accuracy of up to 89% was reported. Similar to Reference [2], where a single crop (tomato) was used, researchers in Reference [7] trained and deployed Residual Neural Networks (ResNets), a deep version of CNN, in a custom-built mobile application to classify wheat disease in the wild, reporting classification accuracy of up to 96%. ResNet and Xception Networks, in combination with the YOLO object detection framework, have also been used to detect early blight tomato disease, with an accuracy of over 99% reported in Reference [3]. DL Neural Networks have also been deployed through an android-based mobile application to detect different common diseases in terrestrial plants found in the Philippines, with an accuracy of 80% [4]. B5 and B4 models of the EfficientNet DL model are reported in Reference [5], which achieved an accuracy of over 99% in the classification for 38 different diseases in various plants. However, the classification was performed in offline mode, and it was not deployed and tested over a mobile application. More recently, transfer learning has been successfully deployed in a mobile application to enable the detection of tomato leaf diseases, reporting detection accuracy of over 96%, using the EffecientNet-B0 DL model [6]. The results of these studies demonstrate the efficacy of deploying ML and DL techniques in mobile applications to detect, classify, and identify plant diseases by using plant leaves.

Medicinal plants and herbs have continued to be used for the traditional management of various illnesses in many societies since time immemorial [22]. With advancements in artificial intelligence (AI) and different Information and Communication Technologies (ICTs), the need to automatically identify medicinal plants from the thousands of plant species can only continue to grow. Researchers in Indonesia [8] utilized local binary patterns to extract the leaf texture of 30 different medicinal plants and then applied probabilistic Neural Networks to automatically classify the herbal leaves, achieving a classification accuracy of just over 56%. In Reference [9], the researchers utilized a support vector machine (SVM) and DL Neural Networks to automatically classify 20 different herbs found in Malaysia, using a mobile application. The mobile application reported spending only 2 s for processing the input leaf image and returning the classification results, with a classification accuracy of 93%. Similarly, in Reference [10], a fusion of fuzzy local binary pattern and fuzzy color histogram, using product decision rules, was performed to enable the automatic identification of 51 medicinal plant species commonly found in Indonesia. Probabilistic Neural Networks were used to classify color histograms, reportedly achieving an accuracy of just over 74%. The promising results achieved in these studies clearly highlight the plausibility of utilizing ML and DL models in mobile applications to automate the classification of medicinal plants.

It has been reported that there could be over 450,000 different plants, with one-third of them facing extinction [23]. The easy and automatic identification of plant species is, therefore, a crucial step toward their preservation. The scientific research community continues to make efforts toward the realization of this step. In Reference [11], the researchers built a joint-classifier by using Backpropagation Neural Networks and weighted K-NN and deployed it in an android-based mobile application to enable automatic classification of 220 plant species (angiosperms) found in China. The joint classifier reportedly achieved a classification accuracy of nearly 93%. K-NN was similarly utilized in Reference [12] to identify 32 different plant species common in Mauritius, using leaf-shape features and a color histogram; accuracy of just over 87% was reportedly achieved. In Reference [13], a custom-built android-based mobile application was able to identify a tree from 126 tree species common in the French Mediterranean area, using tree leaves, with an accuracy of up to 90%. Researchers achieved this through segmenting tree leaf images to form feature space and then used histogram intersection to predict the class. Similarly, a custom-built mobile application with a back-end classifier, using SIFT features with the Bag of Words model and SVM, was able to classify 20 different plant species common in Sri Lanka with an accuracy of 96.48% [14]. Furthermore, CNN models, including VGG19, MobileNet, and MobileNetV2, deployed in a mobile application were able to identify 33 different types of leaves common in East Hokkaido in Japan, with MobileNetV2 achieving an accuracy of over 99% [15]. In Reference [16], the researchers also deployed a CNN model in an android-based mobile application to classify natural images of leaves belonging to 129 different species crowdsourced from all over the world. Single image classification took 2.5 s, obtaining a Top-3 test error of above 60%. Similarly, in Reference [17], the researchers presented a mobile application that was able to classify plant leaves belonging to five different plant species common in India, based on leaf color and shape. To perform classification by using leaf shapes, the extracted morphological features of the leaves were categorized by using the Sobel and Otsu methods, while the color-based classification was performed by using the dominant-color method. Finally, but not least, in Reference [18], the results of a mobile application for leaf classification utilizing a CNN were presented. This study experimented with different CNN architectures' performances in classifying 15 leaf species of plants common in California and North Carolina. Deep CNNs are reported to have achieved the highest classification accuracy of 81.6%, with the mobile application completing the classification task in 2.5 s after spending about 5.2 s loading the query leaf image in the NN classifier from the gallery. The results reported in the abovementioned works from the literature demonstrate the promising efficiency of mobile application classifiers that were developed by using the ML and DL models to facilitate the automatic identification and classification of plant species. Not only is this a crucial step toward the conservation of species risking extinction, but such mobile applications could eventually serve as field guides during tours to various plantations in the wild. In Reference [16], the researchers demonstrated how a mobile application utilizing leaf morphological features and angle code histogram was able to serve as a tour guide in a wild field with six different plant species, in the USA, with an error rate between 17% and 53%. While this error rate appears to be high, it is a promising result to begin with for a field tour in the open wild.

The current study aimed to develop a mobile application to enable the automatic identification of medicinal plants and plants predominantly found in the Borneo region in real time. The present study builds on the strengths of promising methods established in related works from the literature, while also addressing the weaknesses (gaps) identified in the same. Most of the reviewed studies either utilized their own generated small-sized datasets for training and assessing their ML and DL models or entirely (i.e., training and testing) used some of the open existing image datasets (such as ImageCLEF, Flavia, ICL Plantae, PlantVillage, CVIP100, Flavia, Swedish, etc.), potentially resulting in overfitted and underfitted models, respectively. Additionally, many of the reviewed studies considered clean leaf images captured against a white background, a condition that is unlikely in a real-world setting in which leaf images are often captured with a non-clean background. Moreover, studies that reported excellent (over 95%) classification accuracies mainly involved classifying a single plant (e.g., tomato), thereby limiting their application to a wide range of species. Additionally, the feedback of the mobile application end-user on observed classification results was not taken into account to further enrich the classifier's knowledge base for future classification. The development of the mobile application proposed in the current study is, therefore, a valuable addition to the existing efforts toward the preservation of the herbs and medicinal plants native to the Borneo region, a region primarily believed to be the most species-rich area in the world [24], yet with the immense threat of extinction to some of its plant species [25,26]. The current study experimented with and optimized different EffecientNet deep learning models, as transfer learning has mainly been singled-out to produce the best classification results for real-time multiclass image classification tasks [1,5]. A unique feature of the study is the provision of crowdsourcing feedback and geo-mapping of the species in the Borneo region with the help of the mobile application. This is important for the region since several local plant species (with their local names and benefits) are unknown to the experts. The system provides an adaptive learning approach where the models can be updated based on the newly collected data and people's feedback.

Our contributions in this study are highlighted as follows:

- We have proposed a machine vision system that is capable of automating the identification of medicinal plant species in real time.
- We have developed an end-to-end computer vision system with a convolutional neural network (CNN) model to identify medicinal plant species when given an image.
- The system works in real time and can accurately identify different plant species given by simply taking a picture with a mobile camera or uploading an existing image from a device.
- The system provides a feedback mechanism and a knowledge base as a means to continuous lifelong learning of the models to produce a robust plant species identification system.

2. Materials and Methods

2.1. Proposed System

The overall flow of the proposed system is presented in Figure 1. The system is composed of three main components: (1) a computer vision and deep-learning-based plant species classifier, (2) a knowledge base as a central repository for plant information together with auxiliary and feedback data, and (3) a mobile front-end that provides a user interface to the end-user to interact with the system and displaying of classification results. Details for each component of the system are explained in the subsequent subsections.

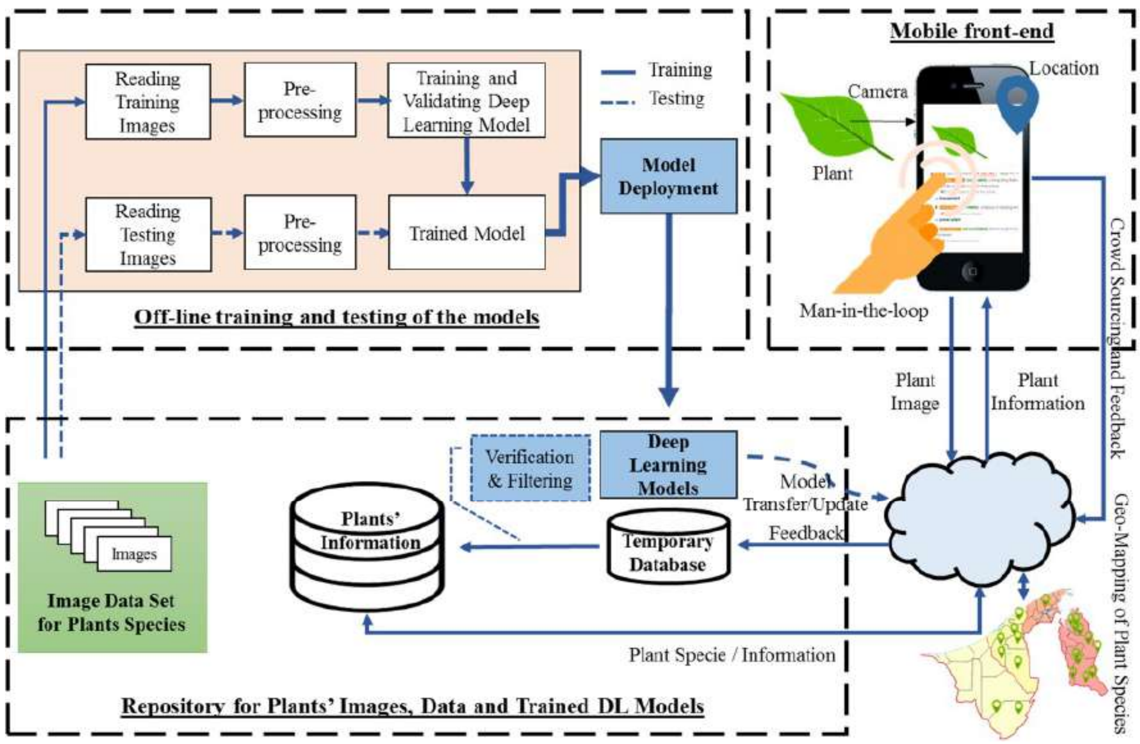

Figure 1. Mobile application system's architecture.

2.2. Classification Model

To develop the real-time plant species identification system, a number of deep learning models were trained and tested on the given datasets. The details of the datasets, pre-processing steps, and training/testing of the models are described below.

2.2.1. Datasets

PlantCLEF 2015

PlantCLEF 2015 [27] is a plant identification challenge dataset that aims to build an image-based plant identification system and evaluates methods and systems at a very large scale that adapts to real-world conditions. The dataset was constructed through a participatory community platform in 2011, consisting of thousands of collaborators. PlantCLEF 2015 consists of curated images from many different contributors, cameras, areas, periods of the year, and individual plants. More precisely, the PlantCLEF 2015 dataset comprises 113,205 pictures belonging to 41,794 observations of 1000 species of trees, herbs, and ferns living in Western European regions. Each image corresponds to one of the seven types of views in the meta-data (entire plant, fruit, leaf, flower, stem, branch, and leaf scan) and is associated with a scientific name.

In this study, the PlantCLEF 2015 dataset was used as an auxiliary dataset to create our plant identification model to improve our classification result. We have further extracted images relevant to our application, that is, the images containing only leaf-related information (e.g., leaf, leaf-scan, and the entire plant). Thus, the extracted dataset consisted of 23,708 images. The hold-out set is built by partitioning 90% of the dataset for training and testing, while the validation data are set to be roughly 10% of the whole dataset (please see Table 1 for more detailed statistics).

Table 1. Number of images extracted from PlantCLEF 2015 dataset.

	Total	**Leaf**	**Leaf-Scan**	**Entire Plant**
Train	18,949	6122	9522	3305
Test	2380	767	1190	423
Validation	2379	781	1134	464

UBD Botanical Garden Dataset

The dataset was laboriously collected from the UBD Botanical Garden, using a DSLR camera with manually annotated labels (i.e., Scientific classification of plants), together with their descriptions. The main originality of the plants in the database is that these plants are mainly native to the Borneo regions, comprising primarily medicinal plants and native plants.

The dataset consists of 106 species with a total of 2097 images collected by us (Table 2). Plants' images were captured from different angles, and instances of plants with a variety of growth conditions were collected. Images are mostly concentrated on the leaf, with only a small fraction of the dataset containing flowers. Examples of images taken from the dataset are illustrated in Figure 2.

Table 2. Numbers of images for UBD Botanical dataset.

Train	1691
Test	157
Validation	249
Total	2097

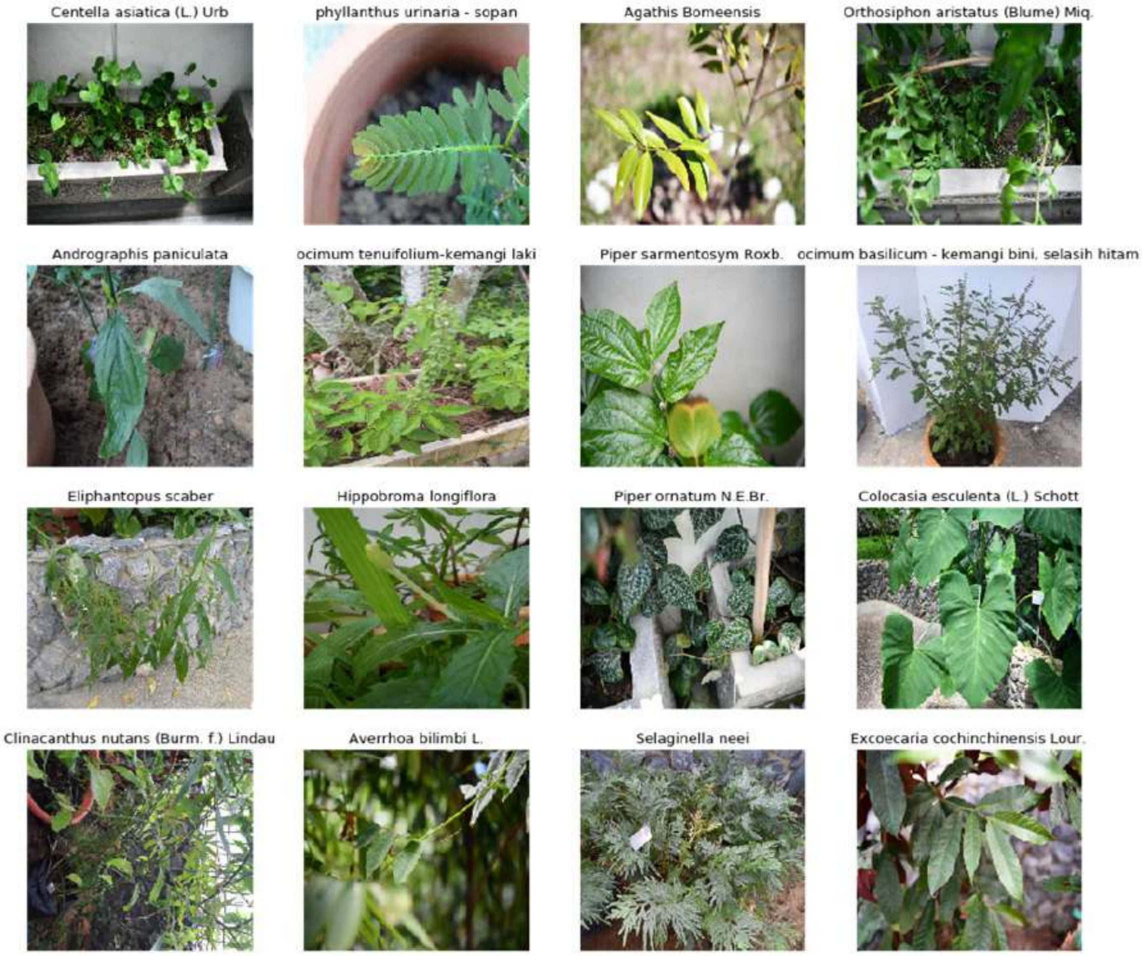

Figure 2. Image samples taken from UBD Botanical dataset.

2.2.2. Training Details

Due to the insufficient size of our dataset, we opted for transfer learning with ImageNet [28] pretrained weights. We used the EfficientNet-B1 [29] variant for deployment and added dropouts of 0.6 and a softmax layer downstream of the network. A post-training quantization of Float16 was employed by using the TensorFlow lite, resulting in 1.95 M parameters after quantization. The network was trained for 100 epochs using the Google Cloud GPU compute engine. We initially trained the model by using ImageNet weights to fine-tune to PlantCLEF 2015 dataset and later merged both datasets (i.e., PlantCLEF and UBD datasets) and performed fine-tuning of the weights in an incremental manner for the locally collected species. A total of 18,949 training Images were used from PlantCLEF and a total of 20,640 images after merging both datasets. These images were resized to 224×224 px with an AutoAugment [30] augmentation policy.

AutoAugment policies are the optimal augmentation policies that improve the overall performance of models using Reinforcement Learning (RL) based on the ImageNet dataset. A policy consists of 5 sub-policies, and each sub-policy applies two image operations in sequence. Each image operation has two parameters: the probability of applying it and the magnitude of the operation. Applying the same optimal augmentation policy to different datasets while using pretrained ImageNet weights yielded an improvement in the accuracy of the models when experimented by us. Thus, we opted for ImageNet weights as our basis for transfer learning with AutoAugment policies in training our models.

To counter the imbalanced distribution of samples from our datasets, we employed two cost-sensitive learning methods to train our model: computing the class weights for each class and focal loss [29]. To summarize, we trained our models by using transfer learning, using ImageNet weights with AutoAugment optimal augmentation policy and cost-sensitive learning methods applied. Figure 3 highlights the imbalance distribution for both datasets with a ratio.

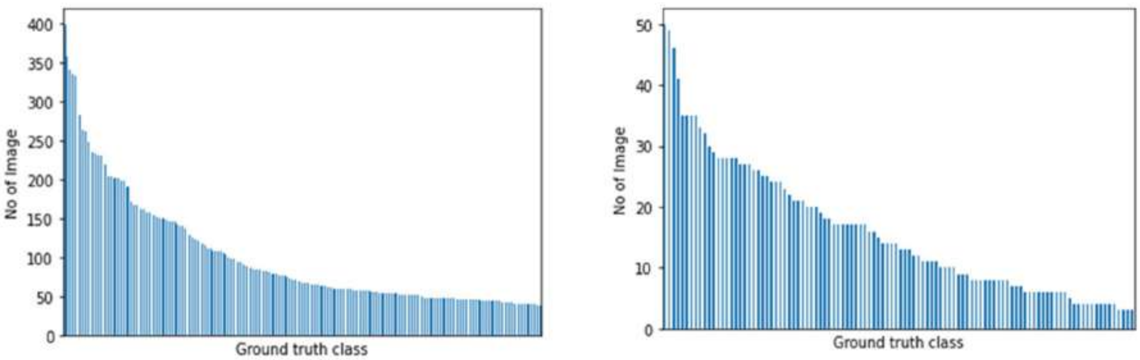

Figure 3. Class distribution for PlantCLEF 2015 (**left**) and UBD Botanical dataset (**right**). Ground truth class names were removed from the *x*-axis for brevity.

Class Weighted Function

The classical way of training neural networks by using backpropagation involves updating the model weights with respect to errors being made by the model. This method fails when we have imbalanced training samples where examples from each class are treated the same, meaning that, for imbalanced datasets, the model is prone to performing well only for the majority class; that is, it is biased toward the majority class samples.

The backpropagation algorithm can be updated to consider the misclassification of each class by using a cost-sensitive loss function that incorporates the error in proportion to the number of samples present in the training samples. The effect of adding this class-weighted function allows the neural network to learn the minority classes such that the model will be penalized more when misclassification of the minority class occurs. We defined the following algorithm in computing the class weights for our dataset as follows:

def class_weights(Y_{train}, α) #returns a dict. of class weights.
counter = dict. containing the no. of samples per class from Y_{train}.
if $\alpha > 0$:
p = max(counter.values()) * α
for class in counter.keys():
counter[class] += p
majority = max(counter.values())
return {class: majority/$count_{class} \forall class \in Y_{train}$}

The α parameter represents the smoothing parameter, which balances the class weights between the majority and minority classes. It determines how much it is to penalize the model when misclassification for the minority class occurs. Setting $\alpha = 1$ equates to applying equal class weightage with respect to the number of species present in the dataset and setting $\alpha > 0$ equates to making the minority class weights higher. We defined α to be 0.4 for both the PlantCLEF 2015 and UBD Botanical datasets and trained by using the multiclass cross-entropy loss.

Focal Loss

The focal loss was introduced in Reference [29] for the object detection task, which deals with the sparse set of foreground examples present in the datasets and prevents the vast number of easy negatives from overwhelming the detector during training. The loss function was reshaped to down-weight easy examples and, thus, focused training on hard negatives by adding a modulating factor $(1 - p_t)^\gamma$ to the cross-entropy loss, with tunable focusing parameter $\gamma \geq 0$ (Equation (1)). In Reference [29], the authors experimented with $\gamma \in [0, 5]$, where $\gamma = 2$ worked best in their experiment.

$$FL(p_t) = -(1 - p_t)^\gamma log(p_t) \tag{1}$$

The properties of focal loss are as follows:

- As $p_t \rightarrow 1$—meaning that the model is confident that a given sample belongs to class t—the modulating factor goes to 0 given $\gamma > 0$, resulting in the down-weight of the loss of the easy examples during training.
- Parameter γ, or the focusing parameter, controls the rate at which easy examples are down-weighted. As $\gamma \in [0, 5]$, $\gamma = 0$ is equivalent to the standard cross-entropy loss with no class weights assigned. As $\gamma \rightarrow 5$, the modulating factor grows exponentially, resulting in the increase of down-weight of easy examples.

We followed the same approach as the one described in Reference [29] for defining the γ to be 2 to both of our datasets. The experimented results for different γ values are also shown in Tables 2 and 3.

Table 3. Performance evaluation on PlantCLEF 2015 test set.

	Top-1 Acc. (%)	**Top-5 Acc. (%)**	**Sensitivity (%)**	**Specificity (%)**
Baseline	73.5	79.4	43.2	53.5
Class weighted ($\alpha = 0.2$)	81.9	87.4	61.5	64.6
Class weighted ($\alpha = 0.4$)	83.2	92.4	79.5	77.6
Focal Loss ($\gamma = 2.0$)	83.8	92.2	76.5	74.6
Focal Loss ($\gamma = 5.0$)	84.0	89.4	76.5	74.6

2.2.3. Hyperparameter Tuning

Learning Rate Finder

Following the super convergence approach for neural networks, the learning rate range test (LR range test) was performed to find the optimal LR_{min} and LR_{max} to be used in the Cyclical Learning Rate (CLR) [31,32]. The losses for each mini-batch were plotted with respect to the LR range of $10^{-10} < LR < 1$, increasing exponentially on each mini-batch. The LR range test serves as a guide for how well a network performs through a range of learning rates and spot values that effectively train the network. The stopping criteria for the LR range test was set at a condition when the current loss is 4 times greater than the previous loss, as seen in Figure 4 at $LR > 10^2$ regime for the PlantCLEF 2015 dataset. The "triangular" learning rate policy rule was opted for this study where learning rates oscillate in a triangular cycle from LR_{min} to LR_{max} in equal lengths of steps size. We set the LR_{min} and LR_{max} to be 10^{-2} to 10^{-1} and 10^{-4} to 10^{-2} for the PlantCLEF 2015 and UBD Botanical datasets, respectively.

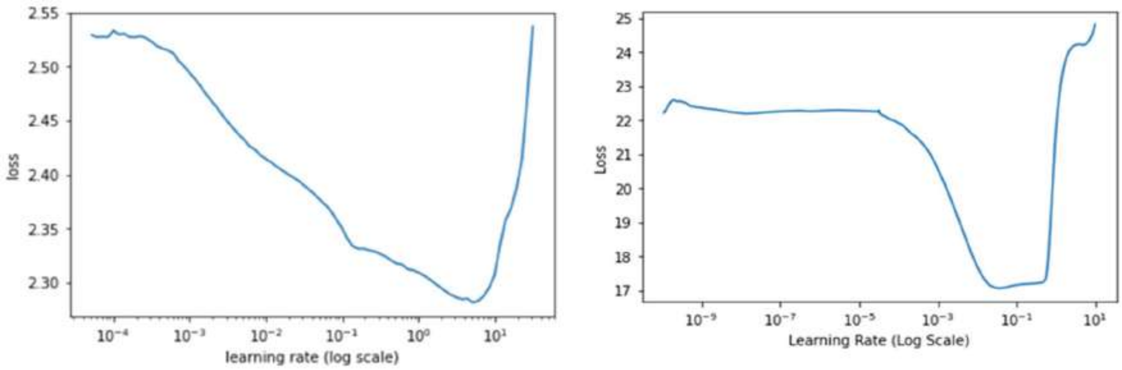

Figure 4. Learning rate range test with the exponential moving average for PlantCLEF 2015 (**Left**) and UBD Botanical datasets (**right**).

2.3. Mobile Application Details

We developed a mobile application dedicated to the production of our collected botanical dataset through image-based plant identification. The application currently supports Android (API level 16 and above), allowing images of a plant via gallery upload or captured by the camera to be sent to the web server to retrieve a list of predicted species made by our model. Figure 5 shows a graphical user interface of our mobile application system. Below are the details of the architecture of the mobile application system.

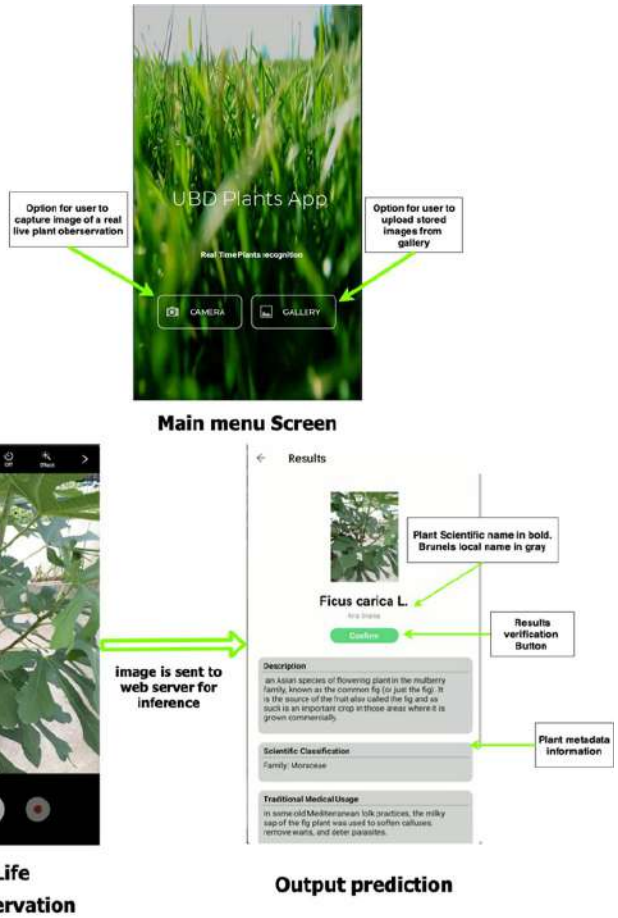

Figure 5. Screenshot of GUI mobile application.

2.3.1. Web API and Front-End

Data exchanges between the back end (web server) and the front-end (client-side) are managed through a REST-full web API, using JSON data format and jpeg images. The list of candidate species predicted for a searched plant observation is displayed in increasing order of confidence level, and the results of only Top-3 predictions are shown to the end-user. In this study, for testing the real-time classification of the species, the Google cloud compute engine server was used as the backend (web server) to deploy the deep learning model, and the custom-developed mobile applications worked as the front-end.

2.3.2. Species Prediction

The predictions made by the classifier are based on the scientific or botanical names of the plant. Metadata information (i.e., family, genus classification, plant usage, etc.) is further extracted from a database via a web server. If a candidate species has a weak confidence level by the model, the observation may not correspond to one species inside our database (rejection class). As a result, the list of returned Top-3 predicted results is not displayed, and the end-users are allowed to suggest the name of the observation.

2.3.3. User Response Information Storage

Each top-k label returned by the model may not be relevant or accurate; as a result, we have implemented a verification mechanism that allows the end-users to verify the returned results and store the response. Information stored includes one or several plant images tagged with provenance data (device, author, date, etc.). Each observation is associated with a geo-location and one or more determinations, i.e., possible species names proposed by the author him/herself, and/or by other annotators, and/or by automated classifiers. Observations are stored within a NoSQL database called MongoDB. The observations are later verified by domain experts and can be used to improve the model performance as a continuous lifelong learning process for plant species identification.

2.4. Knowledge Base

A knowledge base (KB) is a centralized repository for information related to a particular field or domain. In this study, a KB was created in order to manage the information related to the plant species' profiles and other ancillary data. The KB contains different types of information, including raw and processed image data, domain knowledge, learning models (trained intelligent methods), geo-location of images transferred by users for classification, feedback data, etc. The designed KB is not a static collection of information; instead, it acts as a dynamic resource that has the capacity to learn and evolve with time when new plant images are presented, and new classification/labels are added to the system. Thus, as an integral component of the plant species classification system, this KB has been used to optimize the collection, organization, and retrieval of relevant information for plants. The internal implementation of the KB is a hybrid storage model consisting of a database, image files, and classification models.

3. Experiments and Results

The trained classification models were evaluated in two stages: Initially, the models were tested offline on a hold-out test set from both datasets. In the next phase, the models were deployed to the mobile application for real-time species identification, wherein their performance was assessed.

We selected accuracy, sensitivity (true-negative rate), and specificity (true-positive rate) as evaluation metrics, computed as the average for all class labels present in the target dataset. Specificity presents the proportion of the predicted negative classes which were correctly predicted. On the other hand, sensitivity presents a proportion of the true-positive classes which were correctly identified.

The equations for these metrics are given below:

$$Specificity = \frac{\sum_{i=1}^{k} TP_i}{\sum_{i=1}^{k}(TP_i + FN_i)} \quad (2)$$

$$Sensitivity = \frac{\sum_{i=1}^{k} TN_i}{\sum_{i=1}^{k}(TN_i + FP_i)} \quad (3)$$

$$Accuracy = \frac{\sum_{i=1}^{k} \frac{TP_i+TN_i}{TP_i+TN_i+FP_i+FN_i}}{k} \quad (4)$$

where k represents total class labels; and TN, FN, FP, and TP are the numbers of the true-positive, true-negative, false-positive, and false-negative predictions for the considered class, respectively.

3.1. Offline Testing of the Classification Models (for Both Datasets)

We trained a baseline model by applying the cost-sensitive learning method to our highly imbalanced dataset with standard cross-entropy loss and applied the same training techniques as outlined in Section 2.2.2. The training and validation loss curves for the model are shown in Figure 6. We performed an experiment for a total of three runs and reported its average evaluation metrics, as shown in Tables 3 and 4 for both datasets. The results shown in these tables are based on 2380 test samples for PlantCLEF 2015 and 2537 samples when merged (PlantCLEF and UBD Botanical dataset), respectively.

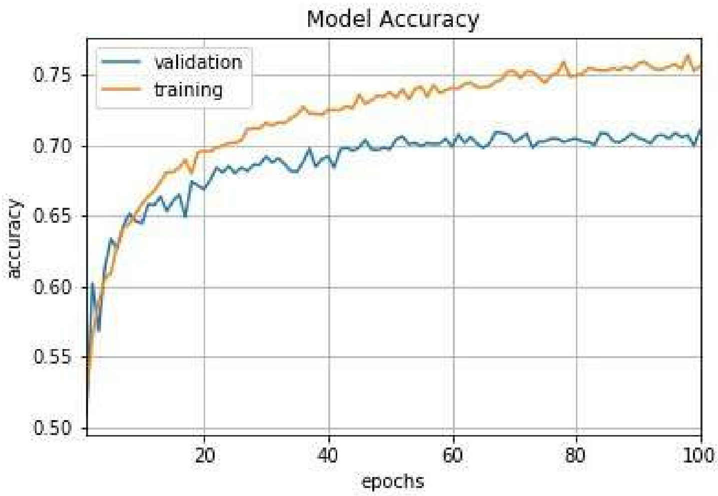

Figure 6. Training and validation loss curves for the deep learning model.

Table 4. Performance evaluation on UBD Botanical + PlantCLEF test set.

	Top-1 Accuracy (%)	Top-5 Accuracy (%)	Sensitivity (%)	Specificity (%)
Baseline	63.4	72.4	43.2	53.5
Class weighted ($\alpha = 0.2$)	83.5	87.4	61.5	64.6
Class weighted ($\alpha = 0.4$)	85.5	92.4	73.5	77.6
Focal Loss ($\gamma = 2.0$)	83.5	92.4	71.5	77.6
Focal Loss ($\gamma = 5.0$)	87.5	86.4	70.5	74.6

The baseline model showed roughly 80% in Top-5 accuracy. However, it performs poorly in classifying the true-positive and true-negative samples as visible from the low sensitivity and specificity rate. The baseline model was found to be biased toward the majority classes; hence, it had a high accuracy rate (80%) and a low rate of sensitivity or recall (43.2%). Both cost-sensitive learning methods, including the class weighted and focal loss, showed an improvement in the classification performance for the imbalanced dataset, as indicated by the significant increase in the sensitivity and specificity values. With the class weighted function applied, the model is being penalized more and able to make a correct prediction with an improvement of sensitivity rate from 43% to 79%, while using focal loss has the effect of having the model recognize classes that do not correspond to its labels. We later performed an evaluation on the test samples on the merged dataset and ran a total of three runs. The results shown in Table 4 are based on a total of 251 species with 2537 images.

The baseline model appeared to show no improvement, albeit adding more samples from our collected dataset. This is mainly due to the imbalance in the nature of the dataset, as shown in the distribution graph in Figure 3. After applying the cost-sensitive learning methods and adding more external samples, a slight incremental improvement in performance was found. In the next section, we outlined the test performed on real-life plant observation and reported its performance.

3.2. Mobile Application Testing

Based on the species identification performance shown in Tables 3 and 4, we selected two models for testing where both are trained on the merged dataset with the highest sensitivity for each of the respective cost-learning methods, i.e., focal loss with γ = 2.0 and class weighted function with α = 0.4. The real-life plants' observation test samples used for this study are the 121 species which are found at the UBD Botanical Garden.

Due to the lack of variability in our dataset, both models' performance deteriorates upon real-time testing, as shown in Table 5. The accuracy of using focal loss is comparable to offline testing, with an average of 80% accuracy for both Top-1 and Top-5 accuracy. However, the sensitivity drops for both methods' classes weighted function and focal loss from 80% to 66.1% and 75.2%, respectively. The resultant of this performance may have been caused by the failure of the model to generalize the dynamic change of state of the plants (i.e., growth cycle), which can be solved when more samples are present in the dataset.

Table 5. Performance evaluation on real-life plants observation at UBD Botanical Garden.

	Top-1 Accuracy (%)	Top-5 Accuracy (%)	Sensitivity (%)	Specificity (%)
Focal Loss ($\gamma = 2.0$)	78.5	82.6	75.2	77.7
Class weighted ($a = 0.4$)	62.8	70.2	66.1	68.6

Some examples of the real-time testing of the developed mobile application are shown in Figure 7. In the real-time classification system, the image taken through the mobile camera is uploaded to the cloud, and the classification results are sent back to the mobile device. The corresponding information, including the plant's scientific name, description, medical usage, precautions, local family name, common name, and origin, is retrieved from the online database and displayed to the user. Since the system also facilitates the geo-mapping of the species, the longitude and latitude information, where the user took the image, can also be saved in the knowledge base. Moreover, a confirmation option is provided for the users to give feedback to improve the system further, and the domain expert verifies this at the end. This option helps us continuously improve the classification model for plant species identification. Figure 8 depicts an example of the system when used in the offline mode. It also shows a scenario where a plant image can be classified into multiple species with different probability values. The class with the highest probability value is shown first, while the rest are shown in decreasing order of probability values. Similar to the real-time system, a feedback mechanism is provided to report the misclassification of the plant species.

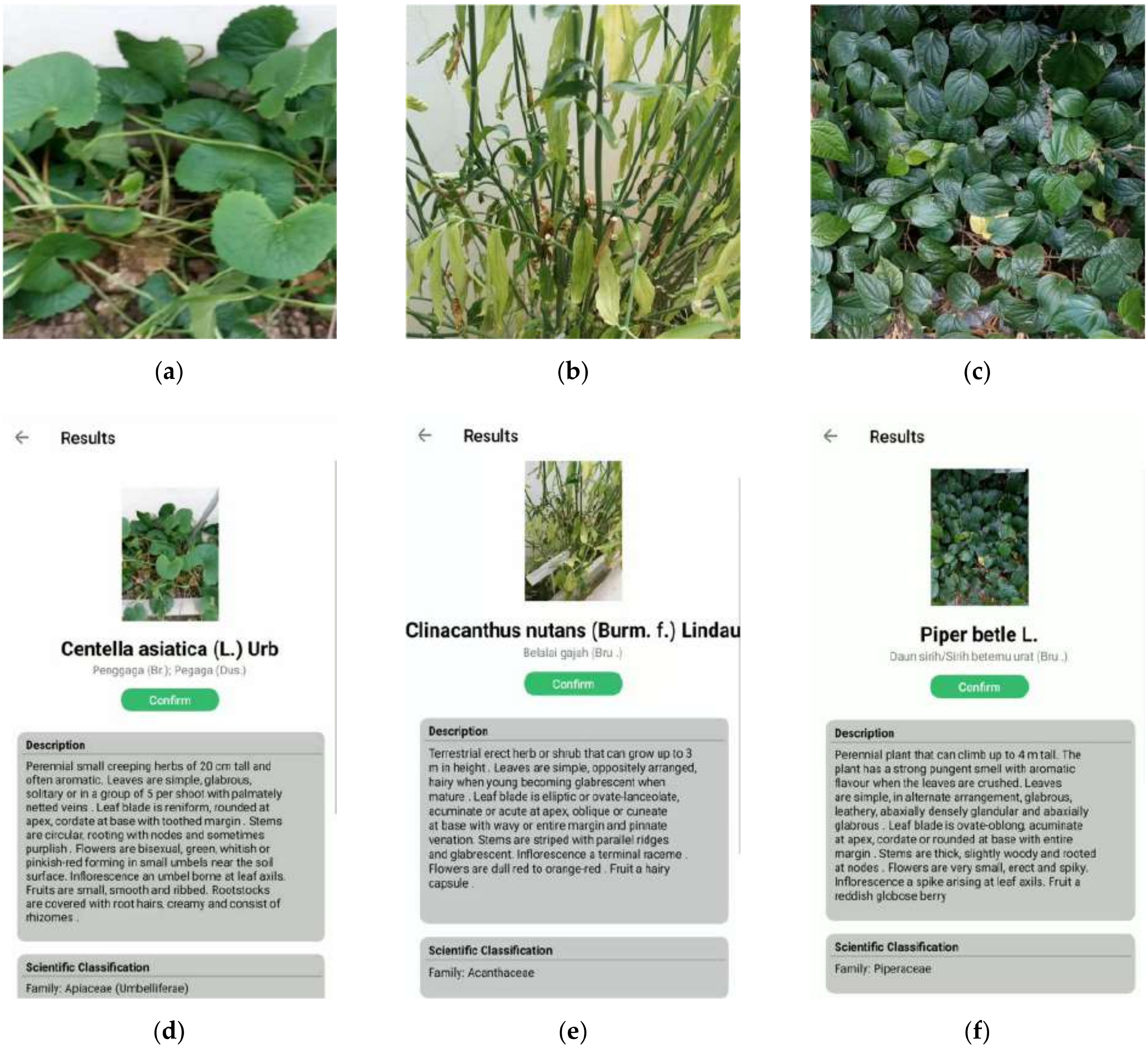

Figure 7. Examples of real-time classification of some species, using the mobile application: (**a**) *Centella asiatica* (L.) Urb; (**b**) *Clinacanthus nutans* (Burm. f.) Lindau; (**c**) *Piper betle* L.; (**d**) real-time classification of *Centella asiatica* (L.) Urb; (**e**) real-time classification of *Clinacanthus nutans* (Burm. f.) Lindau; (**f**) real-time classification of *Piper betle* L.

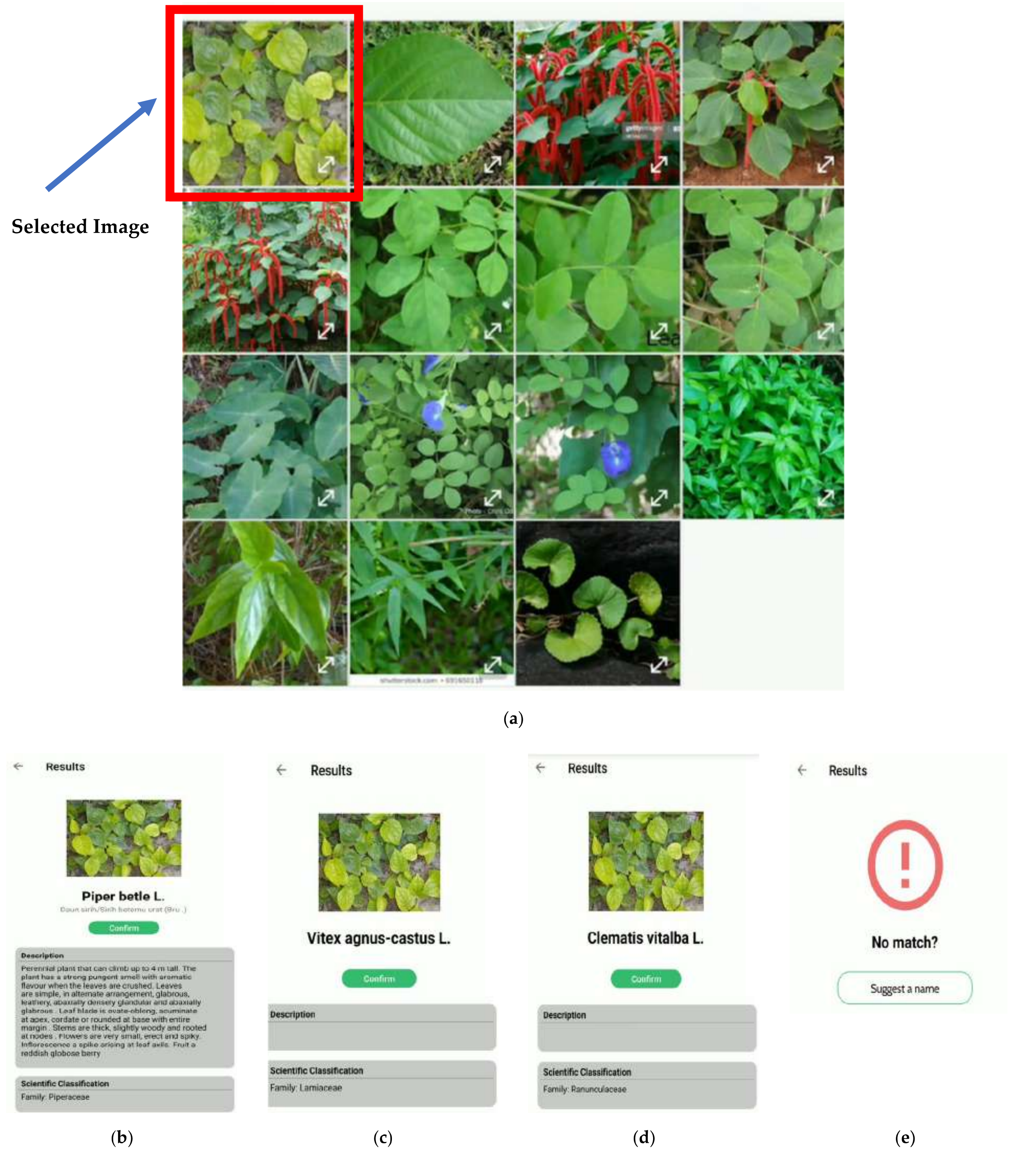

Figure 8. Examples of offline classification of a plant by using the mobile application and feedback option for correction/suggestion for the plant name. (**a**) Images from the gallery for offline testing: (**b**) first matching, (**c**) second matching, (**d**) third matching, and (**e**) feedback option.

In contrast to several previous promising studies in which either open-access datasets [2,3,5] or localized primary datasets [4,20] were exclusively utilized, the deep learning models trained and tested in the current study used over 25,000 images, consisting of both secondary open-access datasets, as well as a primary dataset of images collected from a local botanical garden in Brunei with medicinal plants native to Borneo region. Therefore, this dataset diversity adds to the reproducibility confidence of the obtained results under real-time conditions and the likely generalizability of the trained models. Additionally, the average real-time in-the-wild classification accuracy of 80% for both Top-1 and Top-5 accuracies obtained in the current study is lower than the classification accuracies of 88.4% and 99.9% previously reported in References [2,3], respectively; it should be noted, however, that the latter studies were dealing with diseased leaf detection for only one type of crop (tomatoes) in contrast to the present study, with over 250 medicinal plants species. Furthermore, as a way of building on previous related works [2,4,6,17,20,21] that aimed to train and deploy deep learning plants identification models on mobile applications, the current work has introduced a new feature by enabling confirmation and, thus, verification of the classification results by the mobile application user, as shown in Figure 5. This new feature introduced in this work aids the continuous improvement of the cloud-hosted knowledge base, particularly by domain experts and native users of medicinal plants, thereby facilitating incremental learning of the deployed deep learning model.

4. Conclusions and Future Work

In this study, a deep-learning-based system was proposed to perform a real-time species identification of medicinal plants found in the Borneo region. The proposed system addressed some of the key challenges when training deep learning models, such as small training samples with long-tile class imbalance distribution of the species data. Techniques such as class weighting and the use of focal loss function were applied to improve the learning process of the model. The results showed that the proposed system could significantly improve the performance of the deep learning model by more than 10% accuracy compared to the baseline model. However, performance accuracy was slightly dropped when the system was tested on the actual samples by using the developed mobile application in real time. In the future, we intend to further improve the system's performance by improving the sample collection of the training data. Furthermore, to make the system more useful, we intend to increase the number of species, as the Borneo region is a high species diversity spot.

Author Contributions: Conceptualization, O.A.M.; methodology, O.A.M., N.I. and B.R.H.; software, N.I. and B.R.H.; validation, N.I., O.A.M., B.R.H. and U.Y.; formal analysis, N.I., B.R.H. and O.A.M.; investigation, N.I. and O.A.M.; resources, N.I. and O.A.M.; data curation, N.I. and O.A.M.; writing—original draft preparation, U.Y., N.I., B.R.H. and O.A.M.; writing—review and editing, U.Y., N.I., O.A.M. and B.R.H.; visualization, N.I. and B.R.H.; supervision, O.A.M.; project administration, O.A.M.; funding acquisition, O.A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Universiti Brunei Darussalam under research grant number [UBD/RSCH/1.4/FICBF(b)/2018/011].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available upon request.

Acknowledgments: We express our gratitude to Ferry Slik from Universiti Brunei Darussalam for accessing the UBD Botanical Garden data.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sachar, S.; Kumar, A. Survey of feature extraction and classification techniques to identify plant through leaves. *Expert Syst. Appl.* **2021**, *167*, 114181. [CrossRef]
2. Elhassouny, A.; Smarandache, F. Smart mobile application to recognize tomato leaf diseases using Convolutional Neural Networks. In Proceedings of the International Conference of Computer Science and Renewable Energies (ICCSRE 2019), Agadir, Morocco, 22–24 July 2019; IEEE: Piscatway, NJ, USA, 2019. [CrossRef]
3. Chakravarthy, A.S.; Raman, S. Early Blight Identification in Tomato Leaves using Deep Learning. In Proceedings of the 2020 International Conference on Contemporary Computing and Applications (IC3A), Lucknow, India, 5–7 February 2020; IEEE: Piscatway, NJ, USA, 2020; pp. 154–158. [CrossRef]
4. Valdoria, J.C.; Caballeo, A.R.; Fernandez, B.I.D.; Condino, J.M.M. iDahon: An Android Based Terrestrial Plant Disease Detection Mobile Application through Digital Image Processing Using Deep Learning Neural Network Algorithm. In Proceedings of the 4th International Conference on Information Technology, Bangkok, Thailand, 24–25 October 2019; IEEE: Piscatway, NJ, USA, 2019; pp. 94–98. [CrossRef]
5. Dyrmann, M.; Karstoft, H.; Midtiby, H.S. Plant species classification using deep convolutional neural network. *Biosyst. Eng.* **2016**, *151*, 72–80. [CrossRef]
6. Bir, P.; Kumar, R.; Singh, G. Transfer Learning based Tomato Leaf Disease Detection for mobile applications. In Proceedings of the International Conference on Computing, Power and Communication Technologies (GUCON), Greater Noida, India, 2–4 October 2020; IEEE: Piscatway, NJ, USA, 2020; pp. 34–39. [CrossRef]
7. Picon, A.; Alvarez-Gila, A.; Seitz, M.; Ortiz-Barredo, A.; Echazarra, J.; Johannes, A. Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild. *Comput. Electron. Agric.* **2019**, *161*, 280–290. [CrossRef]
8. Prasvita, D.S.; Herdiyeni, Y. MedLeaf: Mobile Application for Medicinal Plant Identification Based on Leaf Image. *Int. J. Adv. Sci. Eng. Inf. Technol.* **2013**, *3*, 103. [CrossRef]
9. Muneer, A.; Fati, S.M. Efficient and Automated Herbs Classification Approach Based on Shape and Texture Features using Deep Learning. *IEEE Access* **2020**, *8*, 196747. [CrossRef]
10. Herdiyeni, Y.; Wahyuni, N.K.S. Mobile application for Indonesian medicinal plants identification using fuzzy local binary pattern and fuzzy color histogram. In Proceedings of the International Conference on Advanced Computer Science and Information Systems (ICACSIS), Depok, Indonesia, 1–2 December 2012; IEEE: Piscatway, NJ, USA, 2012; pp. 301–306.
11. Cheng, Q.; Zhao, H.; Wang, C.; Du, H. An Android Application for Plant Identification. In Proceedings of the 4th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 14–16 December 2018; IEEE: Piscatway, NJ, USA, 2018; pp. 60–64. [CrossRef]
12. Munisami, T.; Ramsurn, M.; Kishnah, S.; Pudaruth, S. Plant Leaf Recognition Using Shape Features and Colour Histogram with K-nearest Neighbour Classifiers. *Procedia Comput. Sci.* **2015**, *58*, 740–747. [CrossRef]
13. Zhao, Z.-Q.; Ma, L.-H.; Cheung, Y.; Wu, X.; Tang, Y.; Chen, C. ApLeaf: An efficient android-based plant leaf identification system. *Neurocomputing* **2015**, *151*, 1112–1119. [CrossRef]
14. Priyankara, H.A.C.; Withanage, D.K. Computer assisted plant identification system for Android. In Proceedings of the Moratuwa Engineering Research Conference (MERCon), Moratuwa, Sri Lanka, 7–8 April 2015; IEEE: Piscatway, NJ, USA, 2015; pp. 148–153. [CrossRef]
15. Akiyama, T.; Kobayashi, Y.; Sasaki, Y.; Sasaki, K.; Kawaguchi, T.; Kishigami, J. Mobile Leaf Identification System using CNN applied to plants in Hokkaido. In Proceedings of the 8th Global Conference on Consumer Electronics (GCCE), Osaka, Japan, 15–18 October 2019; IEEE: Piscatway, NJ, USA, 2019; pp. 324–325. [CrossRef]
16. Jassmann, T.J.; Tashakkori, R.; Parry, R.M. Leaf classification utilizing a convolutional neural network. In Proceedings of the SoutheastCon 2015, Fort Lauderdale, FL, USA, 9–12 April 2015; IEEE: Piscatway, NJ, USA, 2015; pp. 1–3. [CrossRef]
17. Zaid, M.; Akhtar, S.; Patekar, S.A.; Sohani, M.G. A Mobile Application for Quick Classification of Plant Leaf Based on Color and Shape. *Int. J. Mod. Trends Eng. Res.* **2015**, *2*, 2393–8161.
18. Jassmann, T.J. Mobile Leaf Classification Application Utilizing A CNN. Master's Thesis, Appalachian State University, Boone, NC, USA, 2015.
19. Hussein, B.R.; Malik, O.A.; Ong, W.-H.; Slik, J.W.F. Automated Classification of Tropical Plant Species Data Based on Machine Learning Techniques and Leaf Trait Measurements. In *Computational Science and Technology*; Springer: Singapore, 2020; pp. 85–94. [CrossRef]
20. Ngugi, L.C.; Abelwahab, M.; Abo-Zahhad, M. Tomato leaf segmentation algorithms for mobile phone applications using deep learning. *Comput. Electron. Agric.* **2020**, *178*, 105788. [CrossRef]
21. Knight, D.; Painter, J.; Potter, M. *Automatic Plant Leaf Classification for a Mobile Field Guide—An Android Application*; Stanford University: Stanford, CA, USA, 2010.
22. Petrovska, B.B. Historical review of medicinal plants' usage. *Pharmacogn. Rev.* **2012**, *6*, 1–5. [CrossRef] [PubMed]
23. Pimm, S.L.; Joppa, L.N. How many plant species are there, where are they, and at what rate are they going extinct? *Ann. MO Bot. Gard.* **2015**, *100*, 170–176. [CrossRef]
24. Barthlott, W. Borneo—The Most Species-Rich Area in the World! 2005. Available online: https://phys.org/news/2005-05-borneo-species-rich-area-world.html (accessed on 19 January 2021).

25. Saw, L.G.; Chua, L.S.L.; Suhaida, M.; Yong, W.S.Y.; Hamidah, M. Conservation of some rare and endangered plants from Peninsular Malaysia. *Kew Bull.* **2010**, *65*, 681–689. [CrossRef]
26. Rautner, M.; Hardiono, M. *Borneo: Treasure Island at Risk*; WWF: Gland, Switzerland, 2005; pp. 1–80.
27. Joly, A.; Göeau, H.; Glotin, H.; Spampinato, C.; Bonnet, P.; Vellinga, W.P.; Planqué, R.; Rauber, A.; Palazzo, S.; Fisher, B.; et al. LifeCLEF 2015: Multimedia life species identification challenges. In Proceedings of the 6th International Conference of the CLEF Association (CLEF'15), Toulouse, France, 8–11 September 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 462–483. [CrossRef]
28. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; IEEE: Piscatway, NJ, USA, 2009; pp. 248–255. [CrossRef]
29. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection (RetinaNet). *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *42*, 318–327. [CrossRef] [PubMed]
30. Cubuk, E.D.; Zoph, B.; Mane, D.; Vasudevan, V.; Le, Q.V. AutoAugment: Learning Augmentation Policies from Data. *arXiv* **2019**, arXiv:1805.09501.
31. Smith, L.N. Cyclical Learning Rates for Training Neural Networks. In Proceedings of the Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017; IEEE: Piscatway, NJ, USA, 2017; pp. 464–472.
32. Loshchilov, I.; Hutter, F. Fixing Weight Decay Regularization in Adam. *arXiv* **2017**, arXiv:1711.05101.

Appendix B: Sample Code

//plt.html

```html
<!DOCTYPE html>
<html>
<head>
  <title>PLANT IDENTIFICATION</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <h1>PLANT IDENTIFICATION</h1>

  <div class="upload-container">
    <input type="file" id="image-upload" accept="image/*">
    <button id="submit-btn" onclick="identifyPlant()">Identify
Plant</button>
    <button id="capture-btn" onclick="takePhoto()">Take Photo</button>
    <button id="view-medicines-btn" onclick="viewMedicines()">View
Medicines</button>
    <button id="get-plant-btn" onclick="getPlant()">Get Plant</button>
  </div>

  <div id="result-container">
    <h2>RESULT</h2>
    <div id="result"></div>
  </div>

  <div id="download-container">
    <a id="download-link" download="captured_photo.jpg" style="display:
none;">Download Captured Photo</a>
  </div>


  <video id="video" autoplay></video>


  <script src="script1.js"></script> <!-- Include the external JavaScript
file -->
</body>
</html>
```

//style.css

```css
/* styles.css */
*
{
    margin: 0;
    padding: 0;
```

```css
    box-sizing: border-box;
    font-family: 'Poppins', sans-serif;
}


body {
  width: 100%;
  height: 100vh;
  background-image: url('wallpaperflare.com_wallpaper.jpg'); /* Replace
with your background image */
  background-size: contain;
  background-repeat: no-repeat;
  background-position: center;
}


h1 {
  text-align: center;
  color: #ffffff;
  padding: 20px;
}


.upload-container {
  text-align: center;
  margin-top: 50px;
}


#image-upload {
  border: 2px dashed #ccc;
  padding: 20px;
  background-color:#c4fb84;
  color: #333;
  font-size: 16px;
  cursor: pointer;
}


#submit-btn,
#capture-btn,
#view-medicines-btn {
  padding: 10px 20px;
  font-size: 16px;
  background-color: #bdd248;
  border: none;
  border-radius: 4px;
```

```css
  margin: 10px;
  cursor: pointer;
}

/* Add CSS for the "Get Plant" button */
#get-plant-btn {
  padding: 10px 20px;
  font-size: 16px;
  background-color: #bdd248; /* Change the background color as desired */
  color: #000000; /* Change the text color as desired */
  border: none;
  border-radius: 4px;
  margin: 10px;
  cursor: pointer;
  transition: background-color 0.3s ease; /* Add a smooth transition
effect on hover */
}

/* Add hover effect for the button */
#get-plant-btn:hover {
  background-color: #ffffff; /* Change the background color on hover */
}

#submit-btn:hover {
  background-color: #ffffff; /* Change the background color on hover */
}

#capture-btn:hover {
  background-color: #ffffff; /* Change the background color on hover */
}

#view-medicines-btn:hover {
  background-color: #ffffff; /* Change the background color on hover */
}



#download-container {
  position: absolute;
  top: 0;
  right: 0;
  margin: 10px;
}





#result-container {
```

```
  text-align: center;
  margin-top: 30px;
  color:#ffffff;
}


#result {
  margin-top: 10px;
  font-size: 18px;
}


#video {
  border: 1px solid #ccc;
  box-shadow: 0 0 5px rgba(0, 0, 0, 0.3);
  max-width: 100%;
  height: auto;
}
```

//script1.js

```
/* styles.css */
*
{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Poppins', sans-serif;
}


body {
  width: 100%;
  height: 100vh;
  background-image: url('wallpaperflare.com_wallpaper.jpg'); /* Replace
with your background image */
  background-size: contain;
  background-repeat: no-repeat;
  background-position: center;
}


h1 {
  text-align: center;
  color: #ffffff;
```

```css
  padding: 20px;
}


.upload-container {
  text-align: center;
  margin-top: 50px;
}


#image-upload {
  border: 2px dashed #ccc;
  padding: 20px;
  background-color:#c4fb84;
  color: #333;
  font-size: 16px;
  cursor: pointer;
}


#submit-btn,
#capture-btn,
#view-medicines-btn {
  padding: 10px 20px;
  font-size: 16px;
  background-color: #bdd248;
  border: none;
  border-radius: 4px;
  margin: 10px;
  cursor: pointer;
}

/* Add CSS for the "Get Plant" button */
#get-plant-btn {
  padding: 10px 20px;
  font-size: 16px;
  background-color: #bdd248; /* Change the background color as desired */
  color: #000000; /* Change the text color as desired */
  border: none;
  border-radius: 4px;
  margin: 10px;
  cursor: pointer;
  transition: background-color 0.3s ease; /* Add a smooth transition
effect on hover */
}

/* Add hover effect for the button */
```

```css
#get-plant-btn:hover {
  background-color: #ffffff; /* Change the background color on hover */
}

#submit-btn:hover {
  background-color: #ffffff; /* Change the background color on hover */
}

#capture-btn:hover {
  background-color: #ffffff; /* Change the background color on hover */
}

#view-medicines-btn:hover {
  background-color: #ffffff; /* Change the background color on hover */
}


#download-container {
  position: absolute;
  top: 0;
  right: 0;
  margin: 10px;
}




#result-container {
  text-align: center;
  margin-top: 30px;
  color:#ffffff;
}


#result {
  margin-top: 10px;
  font-size: 18px;
}


#video {
  border: 1px solid #ccc;
  box-shadow: 0 0 5px rgba(0, 0, 0, 0.3);
  max-width: 100%;
  height: auto;
}
```

//medicines.html

```html
<!DOCTYPE html>
<html>
<head>
  <title>View Medicines</title>
  <link rel="stylesheet" type="text/css" href="stylem.css">
  <style>
    /* Center the content vertically and horizontally */
    body {
      display: flex;
      flex-direction: column;
      justify-content: center;
      align-items: center;
      height: 100vh;
      margin: 0;
    }

    /* Add some spacing between elements */
    h1, #medicines-list, #openai-response, button {
      margin: 10px;
    }

    /* Styles for the OpenAI response container */
    #openai-response-container {
      background-color: #f0f0f0;
      padding: 20px;
      border-radius: 10px;
      box-shadow: 0 0 5px rgba(0, 0, 0, 0.2);
      max-width: 80%; /* Limit the width of the response container */
    }

    /* Increase the font size of the identified plant text */
    #identified-plant-text {
      font-size: 24px;
    }
  </style>
</head>
<body>
  <h1>Medicines for Identified Plant</h1>
  <!-- List the medicines here -->
  <div id="medicines-list"></div>

  <!-- Display the OpenAI response -->
  <div id="openai-response-container">
    <p id="openai-response"></p>
  </div>
```

```
  <!-- Add a back button to return to plt.html -->
  <button onclick="goBack()">Back</button>

  <script>
    function goBack() {
      // Go back to plt.html when the back button is clicked
      window.history.back();
    }

    // Retrieve the identified plant name from the URL query parameter
    const urlParams = new URLSearchParams(window.location.search);
    const identifiedPlant = urlParams.get('plant');

    //const identifiedPlant = "Aloe vera";

    // Display the identified plant name on the page
    const plantElement = document.createElement('p');
    plantElement.innerText = 'Identified plant: ' + identifiedPlant;
    plantElement.id = 'identified-plant-text'; // Add an ID to the element
    document.getElementById('medicines-list').appendChild(plantElement);

    const openaiApiKey =
"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";

    function generateResponse(prompt) {
      return
fetch("https://api.openai.com/v1/engines/text-davinci-003/completions", {
        method: "POST",
        headers: {
          "Content-Type": "application/json",
          "Authorization": `Bearer ${openaiApiKey}`,
        },
        body: JSON.stringify({
          prompt: prompt,
          max_tokens: 4000,
          stop: null,
          // temperature: 0.5,
        }),
      })
        .then((response) => response.json())
        .then((data) => data.choices[0].text)
        .catch((error) => console.error("Error:", error));
    }

    const prompt = identifiedPlant + " medicines";
```

```
    generateResponse(prompt)
      .then((response) => {
        // Display the OpenAI response on the page
        const openaiResponseElement =
document.getElementById('openai-response');
        openaiResponseElement.innerText = response;
      })
      .catch((error) => {
        console.error("Error:", error);
      });

  </script>
</body>
</html>
```

//stylem.css

```
/* style.css */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: 'Poppins', sans-serif;
}

body {
    width: 100%;
    height: 100vh;
    background-image:
url('pngtree-light-green-branches-and-leaves-watermark-floral-plant-backgr
ound-picture-image_1445958.png'); /* Replace with your background image */
    background-size: contain;
    background-repeat: no-repeat;
    background-position: center;
  }

h1 {
  text-align: center;
  margin-top: 20px;
  color: #333;
}

ul {
  list-style: none;
  padding: 0;
  margin-top: 20px;
}
```

```css
li {
  padding: 10px;
  border-bottom: 1px solid #ccc;
  font-size: 16px;
  color: #333;
}

button {
  display: block;
  margin: 20px auto;
  padding: 10px 20px;
  font-size: 16px;
  background-color: #bdd248;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}

button:hover {
  background-color: #a2bb36;
}

button:focus {
  outline: none;
}
```

Appendix C: CO-PO and CO-PSO Mapping

COURSE OUTCOMES:

After completion of the course the student will be able to

SL. NO	**DESCRIPTION**	**Blooms' Taxonomy Level**
CO1	Identify technically and economically feasible problems (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO2	Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO3	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools & advanced programming techniques (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO4	Prepare technical report and deliver presentation (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO5	Apply engineering and management principles to achieve the goal of the project (Cognitive Knowledge Level: Apply)	Level 3: Apply

CO-PO AND CO-PSO MAPPING

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO3
CO1	3	3	3	3		2	2	3	2	2	2	3	2	2	2
CO2	3	3	3	3	3	2		3	2	3	2	3	2	2	2
CO3	3	3	3	3	3	2	2	3	2	2	2	3			2
CO4	2	3	2	2	2			3	3	3	2	3	2	2	2
CO5	3	3	3	2	2	2	2	3	2		2	3	2	2	2

3/2/1: high/medium/low

JUSTIFICATIONS FOR CO-PO MAPPING

MAPPING	LOW/ MEDIUM/ HIGH	JUSTIFICATION
100003/CS622T.1-PO1	**HIGH**	Identify technically and economically feasible problems by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
100003/CS622T.1-PO2	**HIGH**	Identify technically and economically feasible problems by analysing complex engineering problems reaching substantiated conclusions using first principles of mathematics.
100003/CS622T.1-PO3	**HIGH**	Design solutions for complex engineering problems by identifying technically and economically feasible problems.
100003/CS622T.1-PO4	**HIGH**	Identify technically and economically feasible problems by analysis and interpretation of data.
100003/CS622T.1-PO6	**MEDIUM**	Responsibilities relevant to the professional engineering practice by identifying the problem.
100003/CS622T.1-PO7	**MEDIUM**	Identify technically and economically feasible problems by understanding the impact of the professional engineering solutions.
100003/CS622T.1-PO8	**HIGH**	Apply ethical principles and commit to professional ethics to identify technically and economically feasible problems.
100003/CS622T.1-PO9	**MEDIUM**	Identify technically and economically feasible problems by working as a team.
100003/CS622T.1-PO10	**MEDIUM**	Communicate effectively with the engineering community by identifying technically and economically feasible problems.
100003/CS622T.1-P011	**MEDIUM**	Demonstrate knowledge and understanding of engineering and management principles by selecting the technically and economically feasible problems.
100003/CS622T.1-PO12	**HIGH**	Identify technically and economically feasible problems for long term learning.
100003/CS622T.1-PSO1	**MEDIUM**	Ability to identify, analyze and design solutions to identify technically and economically feasible problems.
100003/CS622T.1-PSO2	**MEDIUM**	By designing algorithms and applying standard practices in software project development and Identifying technically and economically feasible problems.
100003/CS622T.1-PSO3	**MEDIUM**	Fundamentals of computer science in competitive research can be applied to Identify technically and economically feasible problems.
100003/CS622T.2-PO1	**HIGH**	Identify and survey the relevant by applying the knowledge of mathematics, science, engineering fundamentals.

100003/CS622T.2-PO2	**HIGH**	Identify, formulate, review research literature, and analyze complex engineering problems get familiarized with software development processes.
100003/CS622T.2-PO3	**HIGH**	Design solutions for complex engineering problems and design based on the relevant literature.
100003/CS622T.2-PO4	**HIGH**	Use research-based knowledge including design of experiments based on relevant literature.
100003/CS622T.2-PO5	**HIGH**	Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes by using modern tools.
100003/CS622T.2-PO6	**MEDIUM**	Create, select, and apply appropriate techniques, resources, by identifying and surveying the relevant literature.
100003/CS622T.2-PO8	**HIGH**	Apply ethical principles and commit to professional ethics based on the relevant literature.
100003/CS622T.2-PO9	**MEDIUM**	Identify and survey the relevant literature as a team.
100003/CS622T.2-PO10	**HIGH**	Identify and survey the relevant literature for a good communication to the engineering fraternity.
100003/CS622T.2-PO11	**MEDIUM**	Identify and survey the relevant literature to demonstrate knowledge and understanding of engineering and management principles.
100003/CS622T.2-PO12	**HIGH**	Identify and survey the relevant literature for independent and lifelong learning.
100003/CS622T.2-PSO1	**MEDIUM**	Design solutions for complex engineering problems by Identifying and survey the relevant literature.
100003/CS622T.2-PSO2	**MEDIUM**	Identify and survey the relevant literature for acquiring programming efficiency by designing algorithms and applying standard practices.
100003/CS622T.2-PSO3	**MEDIUM**	Identify and survey the relevant literature to apply the fundamentals of computer science in competitive research.
100003/CS622T.3-PO1	**HIGH**	Perform requirement analysis, identify design methodologies by using modern tools & advanced programming techniques and by applying the knowledge of mathematics, science, engineering fundamentals.
100003/CS622T.3-PO2	**HIGH**	Identify, formulate, review research literature for requirement analysis, identify design methodologies and develop adaptable & reusable solutions.

100003/CS622T.3-PO3	**HIGH**	Design solutions for complex engineering problems and perform requirement analysis, identify design methodologies.
100003/CS622T.3-PO4	**HIGH**	Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
100003/CS622T.3-PO5	**HIGH**	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools.
100003/CS622T.3-PO6	**MEDIUM**	Perform requirement analysis, identify design methodologies and assess societal, health, safety, legal, and cultural issues.
100003/CS622T.3-PO7	**MEDIUM**	Understand the impact of the professional engineering solutions in societal and environmental contexts and Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions.
100003/CS622T.3-PO8	**HIGH**	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions by applying ethical principles and commit to professional ethics.
100003/CS622T.3-PO9	**MEDIUM**	Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.
100003/CS622T.3-PO10	**MEDIUM**	Communicate effectively with the engineering community and with society at large to perform requirement analysis, identify design methodologies.
100003/CS622T.3-PO11	**MEDIUM**	Demonstrate knowledge and understanding of engineering requirement analysis by identifying design methodologies.
100003/CS622T.3-PO12	**HIGH**	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change by analysis, identify design methodologies and develop adaptable & reusable solutions.
100003/CS622T.3-PSO3	**MEDIUM**	The ability to apply the fundamentals of computer science in competitive research and prior to that perform requirement analysis, identify design methodologies.
100003/CS622T.4-PO1	**MEDIUM**	Prepare technical report and deliver presentation by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
100003/CS622T.4-PO2	**HIGH**	Identify, formulate, review research literature, and analyze complex engineering problems by preparing technical report and deliver presentation.

100003/CS622T.4-PO3	**MEDIUM**	Prepare Design solutions for complex engineering problems and create technical report and deliver presentation.
100003/CS622T.4-PO4	**MEDIUM**	Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions and prepare technical report and deliver presentation.
100003/CS622T.4-PO5	**MEDIUM**	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools and Prepare technical report and deliver presentation.
100003/CS622T.4-PO8	**HIGH**	Prepare technical report and deliver presentation by applying ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
100003/CS622T.4-PO9	**HIGH**	Prepare technical report and deliver presentation effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.
100003/CS622T.4-PO10	**HIGH**	Communicate effectively with the engineering community and with society at large by prepare technical report and deliver presentation.
100003/CS622T.4-PO11	**MEDIUM**	Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work by prepare technical report and deliver presentation.
100003/CS622T.4-PO12	**HIGH**	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change by prepare technical report and deliver presentation.
100003/CS622T.4-PSO1	**MEDIUM**	Prepare a technical report and deliver presentation to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas.
100003/CS622T.4-PSO2	**MEDIUM**	To acquire programming efficiency by designing algorithms and applying standard practices in software project development and to prepare technical report and deliver presentation.
100003/CS622T.4-PSO3	**MEDIUM**	To apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs by preparing technical report and deliver presentation.
100003/CS622T.5-PO1	**HIGH**	Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
100003/CS622T.5-PO2	**HIGH**	Identify, formulate, review research literature, and analyze complex engineering problems by applying engineering and management principles to achieve the goal of the project.

100003/CS622T.5-PO3	**HIGH**	Apply engineering and management principles to achieve the goal of the project and to design solutions for complex engineering problems and design system components or processes that meet the specified needs.
100003/CS622T.5-PO4	**MEDIUM**	Apply engineering and management principles to achieve the goal of the project and use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
100003/CS622T.5-PO5	**MEDIUM**	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools and to apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO6	**MEDIUM**	Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities by applying engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO7	**MEDIUM**	Understand the impact of the professional engineering solutions in societal and environmental contexts, and apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO8	**HIGH**	Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice and to use the engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO9	**MEDIUM**	Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings and to apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO11	**MEDIUM**	Demonstrate knowledge and understanding of engineering and management principles and apply these to one’s own work, as a member and leader in a team. Manage projects in multidisciplinary environments and to apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO12	**HIGH**	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change and to apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PSO1	**MEDIUM**	The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas. Apply engineering and management principles to achieve the goal of the project.

100003/CS622T.5-PSO2	**MEDIUM**	The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry and to apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PSO3	**MEDIUM**	The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur and apply engineering and management principles to achieve the goal of the project.