VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY
An Autonomous Institute Affiliated to University of Mumbai
Department of Computer Engineering

Project Report on

MedLeaf: AI-based Identification and Medicinal Value Assessment of Flora

In partial fulfillment of the Fourth Year, Bachelor of Engineering (B.E.) Degree in Computer Engineering at the University of Mumbai Academic Year 2024-25

Submitted by
Kevin Patel (D17A, 45)
Sanika Hadap (D17C, 22)
Tanvi Naik (D17C, 48)

Project Mentor
Dr.(Mrs.) Priya R. L

(2024-25)

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY
An Autonomous Institute Affiliated to University of Mumbai
Department of Computer Engineering

Certificate

This is to certify that ***Kevin Patel (D17A-45), Sanika Hadap (D17C-22) and Tanvi Naik (D17C- 48)*** of Fourth Year Computer Engineering studying under the University of Mumbai have satisfactorily completed the project on "***MedLeaf: AI-based Identification and Medicinal Value Assessment of Flora***" as a part of their coursework of PROJECT-II for Semester-VIII under the guidance of their mentor ***Prof. Priya R. L*** in the year 2024-25 .

This project report entitled ***MedLeaf: AI-based Identification and Medicinal Value Assessment of Flora*** by ***Kevin Patel, Sanika Hadap and Tanvi Naik*** is approved for the degree of **B.E. Computer Engineering**.

Programme Outcomes	Grade
PO1, PO2, PO3, PO4, PO5, PO6, PO7, PO8, PO9, PO10, PO11, PO12 PSO1, PSO2	

Date:

Project Guide: Dr. (Mrs.) Priya R. L

Project Report Approval
For
B. E (Computer Engineering)

This project report entitled **MedLeaf: AI-based Identification and Medicinal Value Assessment of Flora** by ***Kevin Patel, Sanika Hadap and Tanvi Naik*** is approved for the degree of **B.E. Computer Engineering.**

Internal Examiner

External Examiner

Head of the Department

Principal

Date:
Place: Mumbai

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

-- --

Kevin Patel(D17A-45) Sanika Hadap(D17C-22)

Tanvi Naik(D17C-48)

Date:

ACKNOWLEDGEMENT

We are thankful to our college Vivekanand Education Society's Institute of Technology for considering our project and extending help at all stages needed during our work of collecting information regarding the project.

It gives us immense pleasure to express our deep and sincere gratitude to Assistant Professor **Priya R. L.** for her kind help and valuable advice during the development of project synopsis and for her guidance and suggestions.

We are deeply indebted to the Head of the Computer Department **Dr. (Mrs.) Nupur Giri** and our Principal **Dr. (Mrs.) J.M. Nair,** for giving us this valuable opportunity to do this project.

We express our hearty thanks to them for their assistance without which it would have been difficult in finishing this project synopsis and project review successfully.

We convey our deep sense of gratitude to all teaching and non-teaching staff for their constant encouragement, support and selfless help throughout the project work. It is a great pleasure to acknowledge the help and suggestion, which we received from the Department of Computer Engineering.

We wish to express our profound thanks to all those who helped us in gathering information about the project. Our families too have provided moral support and encouragement several times.

Computer Engineering Department
COURSE OUTCOMES FOR B.E PROJECT

Learners will be to,

Course Outcome	Description of the Course Outcome
CO 1	Able to apply the relevant engineering concepts, knowledge and skills towards the project.
CO2	Able to identify, formulate and interpret the various relevant research papers and to determine the problem.
CO 3	Able to apply the engineering concepts towards designing solutions for the problem.
CO 4	Able to interpret the data and datasets to be utilized.
CO 5	Able to create, select and apply appropriate technologies, techniques, resources and tools for the project.
CO 6	Able to apply ethical, professional policies and principles towards societal, environmental, safety and cultural benefit.
CO 7	Able to function effectively as an individual, and as a member of a team, allocating roles with clear lines of responsibility and accountability.
CO 8	Able to write effective reports, design documents and make effective presentations.
CO 9	Able to apply engineering and management principles to the project as a team member.
CO 10	Able to apply the project domain knowledge to sharpen one’s competency.
CO 11	Able to develop a professional, presentational, balanced and structured approach towards project development.
CO 12	Able to adopt skills, languages, environment and platforms for creating innovative solutions for the project.

Index

List of Figures

Abstract

Medicinal plants have long served as vital components of traditional medicine and remain indispensable in modern pharmacological research. However, the accurate identification and study of these plants are often limited by the need for expert knowledge and time-intensive processes. The MedLeaf project addresses this challenge by integrating artificial intelligence with mobile technology to deliver a real-time, user-friendly solution for medicinal plant identification and information retrieval. At the core of MedLeaf lies a Convolutional Neural Network (CNN) model developed entirely from scratch, trained to recognize and classify various medicinal plant species based on leaf images. This approach ensures high accuracy and eliminates dependency on pre-trained models, allowing for customization and adaptability to specific user needs. Designed using a Human-Centered AI framework, the application focuses on accessibility and ease of interaction, especially for users such as Ayurvedic practitioners, medicinal plant harvesters, students, and enthusiasts with limited technical backgrounds. The mobile application, developed using Flutter, enables seamless operation on Android devices and supports image-based scanning, returning the plant's name along with its pharmacognostic details in textual form. These details are sourced through structured web scraping from authenticated sources, including government databases, trusted publications, and field-verified data. MedLeaf thus bridges the gap between conventional plant knowledge and modern technological solutions, offering a scalable, intelligent, and regionally adaptable tool for medicinal plant research, education, and practice.

Chapter 1: Introduction

1.1 Introduction

Medicinal plants have been central to traditional medicine and continue to be crucial for modern pharmacology. Accurate identification of these plants is vital for their effective use, ensuring the correct utilization of medicinal plants and preventing the use of incorrect or harmful species[1]. A large number of higher plants have been used as a source of drugs by mankind for several thousand years. It is estimated that 35,000 to 70,000 plant species have at one time or another been used in some cultures for medicinal purposes[2]. Traditional methods of plant identification are time-consuming and require extensive expertise. This project utilizes computer vision to enhance plant identification and find medical benefits. Using convolutional neural networks (CNNs), we aim to classify medicinal plants accurately based on images[3]. The model will be trained on datasets compiled from publicly available sources, web scraping, and reliable field data. Beyond identification, the project will analyze and present medicinal benefits using information from official websites, authoritative literature, and verified practitioners. This tool will assist Ayurvedic practitioners, medicinal plant harvesters, and students, providing a reliable resource for plant identification and benefit analysis.

1.2 Motivation for the project

The motivation behind MedLeaf is to bridge the knowledge gap in identifying medicinal plants and understanding their benefits. Traditional medicine and modern research both highlight the value of medicinal flora, yet accurate identification often requires specialized knowledge. This project aims to harness the power of AI and computer vision to create an accessible, reliable tool for identifying medicinal plants and assessing their therapeutic properties. By making this knowledge more accessible, MedLeaf supports Ayurvedic practitioners, harvesters, and students in utilizing these resources for health and wellness.

1.3 Problem Definition

Current methods for identifying medicinal plants are often labor-intensive and require specialized expertise, presenting significant barriers for practitioners, harvesters, and students. Traditional identification processes, which rely on detailed botanical knowledge and manual examination of plant characteristics, are both time-consuming and complex. This complexity limits accessibility and efficiency, making it challenging for non-experts and professionals alike. The growing shortage of skilled taxonomists and the broader "taxonomic crisis" further exacerbates these difficulties. Additionally, existing tools fall short in integrating detailed medicinal benefits information, which hinders users' ability to access comprehensive and relevant data effectively. With the rapid decline in biodiversity and increasing demand for accurate species identification, there is an urgent need for an integrated, user-friendly solution that bridges the gap between precise plant identification and extensive medicinal data. This solution should leverage modern technologies, such as computer vision and digital image processing, to enhance accessibility and efficiency in medicinal plant research and application, thus supporting both traditional practices and contemporary research.

1.4 Existing Systems

Many important studies and model-building projects have been undertaken to detect strokes or determine the likelihood of a stroke using a variety of machine-learning approaches using patient datasets and electronic health information. As a result, we conducted important research that provided us with clarity and direction when carrying out our duties. This motivated us to interact with doctors and seek their guidance, consult published articles, and take into account numerous other viewpoints that significantly advanced our understanding and helped streamline our work. As a result, we developed a thorough study on the referred papers that are briefly described below, based on the objective, observation, evaluation metrics, and system development.

1.5 Lacuna of Existing System

- Data Limitations
- Image Quality Dependency
- Overfitting Risks
- Species Misclassification
- Narrow Focus
- Lack of Real World Testing
- Environmental and Geographical Variability
- Low Accessibility of Systems

1.6 Relevance of the project

The MedLeaf project is highly relevant in today's context, where the need for sustainable healthcare solutions is more pressing than ever. With over 70% of the global population relying on medicinal plants for primary healthcare, accurate identification of these plants is crucial for ensuring safety and efficacy in traditional medicine practices. However, traditional methods of plant identification are time-consuming and require specialized expertise, making them less accessible to the general public. This project leverages computer vision and deep learning techniques to simplify and enhance the identification process, making it more efficient and accurate. By focusing on medicinal plants, the project supports biodiversity conservation, healthcare, and the preservation of ethnobotanical knowledge. Additionally, it addresses the growing demand for integrating artificial intelligence into healthcare, offering potential benefits for practitioners, researchers, and individuals who depend on medicinal plants for treatment. The real-time applicability of this project makes it highly relevant in rural or resource-constrained areas, where quick and accurate identification can have significant healthcare implications.

Chapter 2 : Literature Survey

A. Brief Overview of Literature Survey

Numerous studies and model-building initiatives have been conducted in recent years to identify medicinal plants and evaluate their pharmacological significance using machine learning and deep learning techniques. These approaches often leverage leaf image datasets and botanical information to enhance identification accuracy and automate medicinal value assessment. Inspired by these developments, we undertook a comprehensive review of relevant literature, which provided critical insights and direction for our project. This exploration enabled us to consult with experts, review scholarly articles, and incorporate diverse perspectives that refined our understanding and informed the overall design of the system.

Our study encompasses a variety of research papers and systematic reviews focused on plant identification, CNN model comparisons, dataset challenges, and the integration of NLP for medicinal benefit extraction. The core emphasis lies in using lightweight, efficient models for real-time identification, along with information retrieval systems that bridge the gap between traditional knowledge and modern AI applications. This review served as the foundation for developing our own application, MedLeaf, which leverages a custom-built DenseNet model and a Confidence Score Analyzer for precise classification, while also providing users with personalized information on plant uses via NLP-driven pharmacognosy modules.

2.1 Research Papers

Below are the research papers meticulously considered, reviewed and studied, contributing significantly to the refinement and development of our approach towards creating the MedLeaf project. These papers have served as guiding lights, shaping our understanding and strategies in addressing the challenges of medicinal plant identification. Through careful examination of these scholarly works, we have gained invaluable insights that have informed our decision-making processes and propelled us towards crafting a user-friendly mobile application. By leveraging the wealth of knowledge distilled from these studies, we aim to enhance medicinal plant identification practices and optimize the outcomes through innovative capabilities of 'MedLeaf'.

[5] **Assessing deep convolutional neural network models and their comparative performance for automated medicinal plant identification from leaf images.** (2024)

Methodology: The study utilized images resized to 224 × 224 pixels and employed various pretrained models, including VGG16, VGG19, DenseNet201, ResNet50V2, Xception, InceptionResNetV2, and InceptionV3, to conduct medicinal plant identification.

Performance: DenseNet201 exhibited superior performance, achieving an accuracy of 99.64% with public datasets and 97% with field images. The precision for public data reached 98.31%.

Outcomes: DenseNet201 emerged as the best-performing model for medicinal plant identification, demonstrating strong potential for applications in drug development and conservation initiatives.

Limitations: Misclassification issues were observed due to similarities in leaf morphology and environmental influences. Additionally, challenges arose from variations in species appearance based on age and habitat.

Inferences: The study highlights the effectiveness of DenseNet201 for medicinal plant identification, though it underscores the need for further refinement to address environmental and morphological variations. This approach can be crucial for real-world applications where plant identification is needed for conservation and pharmaceutical purposes.

[6] **Ethno medicine of Indigenous Communities: Tamil Traditional Medicinal Plants Leaf detection using Deep Learning Models** (2024)

Methodology: The study employed preprocessing techniques such as augmentation, noise removal using bilateral filtering, and grayscale conversion. Segmentation was carried out using a Hybrid Genetic Algorithm with Watershed (HGAW), followed by region detection, feature extraction, and identification utilizing a bilateral filter.

Performance: The deep learning model achieved an accuracy of 96.71% in recognizing traditional Tamil medicinal plants.

Outcomes: This research highlights the importance of integrating traditional ethno-medicinal knowledge with modern technological methods to improve healthcare practices and preserve valuable cultural knowledge.

Limitations: The model's performance is reliant on high-quality plant images and is constrained by the limited size of the dataset. Additionally, it has not been tested on real-world images and lacks a graphical user interface.

Inferences: The study demonstrates the potential of using advanced algorithms to identify medicinal plants, though improvements are necessary in dataset expansion and real-world testing. The absence of a user interface also limits practical applications, but the fusion of traditional and modern approaches has promising implications for future healthcare solutions.

[7] **Medicinal and poisonous plants classification from visual characteristics of leaves using computer vision and deep neural networks** (2024)

Methodology: The data were divided into five folds and augmented using FastAutoAugment (FAA). The images were processed by a ResNet variant incorporating both spatial and channel attention mechanisms. A Bayesian optimizer was employed to select optimal augmentation policies, enhancing classification accuracy.

Performance: The model variants achieved high classification accuracies: Tree-CA at 99.63%, Gated-CA at 99.07%, Mixed-CA at 98.13%, and GAP-CA at 96.23%.

Outcomes: The study showcases the integration of channel attention (CA) and spatial attention (SA) mechanisms into the model, improving classification performance.

Limitations: The model's accuracy was impacted by partially or fully covered leaves. Additionally, the study highlighted the need to expand the dataset and the lack of a graphical user interface.

Inferences: The combination of spatial and channel attention significantly enhanced classification accuracy, particularly in complex plant images. However, challenges like occluded leaves and dataset limitations indicate the need for further research. The absence of a user interface also restricts practical usability, calling for improvements to make the model more accessible for real-world applications.

[8] **Medicinal plant recognition based on Vision Transformer and BEiT** (2024)

Methodology: EfficientNetB0 was trained on 224x224 pixel images, while EfficientNetV2-S, Vision Transformer (ViT), and BEiT models were trained on 384x384 pixel images to optimize performance.

Performance: Among the models, BEiT achieved the highest accuracy of 99.14%, outperforming the other architectures.

Outcomes: The study suggests that transformer-based architectures, such as BEiT, can significantly enhance classification accuracy compared to conventional models.

Limitations: The model required high computational resources, and while higher image resolutions could potentially improve performance, this aspect was not explored in the study.

Inferences: The use of transformer-based models, particularly BEiT, shows great promise for improving classification accuracy in medicinal plant identification. However, the computational demands and unexplored potential of higher resolutions highlight areas for further research. Enhancing model efficiency without compromising accuracy could make this approach more feasible for broader applications.

[9] **<u>Medicinal Plants Identification Using Federated Deep Learning</u>** (2024)

- Methodology: The dataset was divided into training (50%), validation (10%), and testing (40%) sets. The data were distributed using both Independent and Identically Distributed (IID) and Non-IID methods. Models such as VGG16, ResNet50, ConvNext, and MaxVit were trained using Federated Averaging (FedAvg) and Federated Proximal (FedProx) algorithms.
- Performance: FL improved accuracy over 100 rounds, with the accuracy for IID data increasing from 88.56% to 88.93% and for Non-IID data from 67.81% to 68.76%.
- Outcomes: The study introduces a new challenge for classifying medicinal plants using Non-IID training data, demonstrating the potential of federated learning techniques.
- Limitations: The model required an extensive training time of 500 hours, which could be a significant constraint.
- **Inferences:** Federated learning shows promise in improving accuracy for both IID and Non-IID data distributions, particularly in complex tasks like medicinal plant identification. However, the high training time suggests a need for optimizing the process to make it more efficient, especially for real-world applications where faster results are crucial.

[10] **MTJNet: Multi-task joint learning network for advancing medicinal plant and leaf classification** (2024)

- Methodology: The images were reshaped to 224 × 224 × 3 dimensions, with Sobel edge detection and vein morphometric feature extraction applied. A Multi-Task Joint Learning Network (MTJNet) was used, combining local and global feature extraction along with dense layers for classification.
- Performance: The model achieved high performance metrics, including a precision of 99.60%, recall of 99.62%, accuracy of 99.71%, and an F1 score of 99.58%.
- Outcomes: The model outperformed existing methods, establishing itself as a promising tool for applications in both medical and industrial domains.
- Limitations: The model relied heavily on high-quality images, and there was a potential risk of overfitting due to the large number of features extracted.
- **Inferences:** This approach demonstrates the effectiveness of combining local and global feature extraction for medicinal plant identification, achieving excellent performance metrics. However, the dependency on high-quality images and the risk of overfitting highlight areas where the model could be refined, especially for practical applications in uncontrolled environments.

[11] **Varietal Discrimination of Guava (Psidium Guajava) Leaves Using Multi Features Analysis** (2024)

- Methodology: The images were resized to various resolutions, ranging from 128x128 to 1024x1024 pixels, and preprocessed using Hybrid Threshold Range-Based Segmentation (HTRS). A total of 47 features were extracted for classification.
- Performance: The study achieved an accuracy of 93.01% using the IBI algorithm for guava leaf classification.
- Outcomes: The research demonstrated the effectiveness of machine vision techniques for the classification of guava leaves, offering potential benefits to farmers, breeders, and the food industry.
- Limitations: The study focused exclusively on a single plant species, limiting the generalizability of its findings.
- **Inferences:** While the machine vision techniques proved effective in classifying guava leaves, the narrow focus on a single species suggests that further research is needed to extend these methods to other plants. Broadening the scope could lead to broader applications, especially in agricultural and industrial contexts.

[12] **Medicinal Plant Identification in Real-Time Using Deep Learning Model** (2023)

- Methodology: The images were resized to 224x224 pixels for training with the MobileNet model. Data augmentation techniques such as rotation, flipping, and zooming were applied to enhance the training dataset.
- Performance: The model achieved an accuracy of 98.3%, providing a real-time solution that is accessible through a mobile app for both experts and the general public.
- Outcomes: A mobile application was developed that enables users to capture images of leaves and receive real-time identification of medicinal plants.
- Limitations: The model is limited to identifying only six medicinal plant species from the Kaggle dataset.
- **Inferences:** It successfully demonstrates the use of mobile technology for real-time medicinal plant identification, making it convenient for a broad audience. However, the limitation of the model to just six plant species restricts its usability. Expanding the dataset and plant variety could significantly enhance the app's practicality and reach.

[13] **Medicinal Herbs Identification Using Deep Learning** (2023)

- Methodology: The images were resized to 299x299 pixels and augmented using flipping and rotation techniques. Transfer learning was applied using the pre-trained Xception model, with adjustments made to hyperparameters such as learning rate and the number of epochs.

- Performance: The system achieved a training accuracy of 93.34%, a validation accuracy of 96.79%, an average precision of 95.87%, and a recall of 96.25%. The model processed images and returned results in under 2 seconds.
- Outcomes: The study developed an AI-based deep learning model for the automated identification of medicinal plants.
- Limitations: The training process was computationally expensive, and the model requires further testing on real-world samples beyond the current dataset.
- **Inferences:** The implementation of transfer learning with the Xception model delivered impressive performance metrics for medicinal plant identification. But the high computational costs and need for real-world testing suggest that further optimization is needed to make the model more scalable and applicable in practical environments.

[14] **DIMPSAR: Dataset for Indian medicinal plant species analysis and recognition** (2023)

- Composition: The dataset consists of 5,900 images from forty plant species, along with single-leaf images of eighty plant species. In total, 6,900 samples were collected, obtained in real-time conditions using smartphones.
- Image Characteristics: The resolution of the images captured ranged from 2,560×1,920 to 5,312×2,988 pixels, ensuring a variety of image qualities based on the conditions and devices used.
- Outcomes: The dataset collection involved visits to various botanical gardens in and around Karnataka and Kerala, providing a diverse array of plant species for the dataset.
- Disadvantages: The image quality varied significantly, leading to inconsistencies that could affect the accuracy and reliability of plant identification models trained on this dataset.
- **Inferences:** The dataset provides a rich source of real-time plant images captured in natural conditions, adding to its authenticity and diversity. However, the variation in image quality due to differences in camera resolutions and environmental conditions could hinder model performance. Future efforts could focus on standardizing the image collection process or applying preprocessing techniques to minimize the impact of image quality variations.

2.2 Patent Search

1. A DEEP LEARNING-BASED IMAGE RECOGNITION METHOD FOR LEAF DISEASES OF MEDICINAL PLANTS (CN110717451A)[15]

Inventor : Liu Yongguo, Li Qiaoqin, Yang Shangming Cai, Li Yang He Jiahuan

This invention presents a deep learning-based method to identify leaf diseases in medicinal plants. It involves collecting and enhancing disease images, resizing them to 299x299 pixels, and training a deep CNN that integrates Inception-I and depthwise separable convolutions. The model classifies disease types from leaf images, aiding planters in diagnosing issues more efficiently.

2. SYSTEM AND METHOD FOR PLANT IDENTIFICATION (US8577616B2)[16]

Inventor : Susan C. Dunlap

Methods of compiling a database of images of plant species and the use of the database to identify unknown plant species are described. Images of the apical complexes of the plant are obtained and stored in a database to allow a comparison of the apical complexes with unknown plant species. The invention provides a facile method for the identification of unknown or unidentified plant species.

2.3 Inference Drawn

Recent studies in medicinal plant identification using deep learning techniques have shown encouraging results, especially with architectures like DenseNet201, BEiT, and Xception. These models, when coupled with enhancements such as spatial and channel attention or the fusion of local and global features, have significantly boosted classification accuracy. They exhibit strong potential in recognizing complex patterns in plant leaves, even in naturally varying environments. Some studies also explored the integration of federated learning, which improves accuracy across decentralized datasets while maintaining data privacy. These advancements indicate the field's readiness to address critical challenges in healthcare, agriculture, and biodiversity through automated plant identification systems.

However, limitations across these approaches are evident. Many models rely heavily on curated, high-quality image datasets and struggle with occlusions, morphological variations, and inconsistent lighting in real-world scenarios. Transformer-based models like BEiT also present computational burdens, limiting their feasibility on lightweight or mobile platforms. Additionally, while a few attempts at real-time mobile implementation exist, most studies lack user interface integration or sufficient species diversity, making them less suitable for practical use. This highlights a significant gap between high-performance backend systems and their translation into accessible, end-user applications.

The need for standardized image collection, preprocessing techniques, and dataset diversification remains paramount. Broader species coverage and inclusion of region-specific flora will significantly improve real-world applicability and impact.

In summary, while current research has laid a strong technical foundation, future efforts must focus on enhancing model generalizability, optimizing computational performance, and creating practical, user-friendly tools that translate lab success into real-world solutions.

2.4 Comparison with Existing System

In comparison to existing plant identification systems and botanical databases, which frequently depend on predefined datasets or basic image-matching techniques, the MedLeaf project offers a more flexible and intelligent solution. Many existing platforms lack real-time identification capabilities, mobile-friendly interfaces, or integration with region-specific medicinal information, limiting their effectiveness in practical, field-level use.

To address these limitations, MedLeaf adopts a machine learning-driven approach by employing a Convolutional Neural Network (CNN) model trained from scratch on a curated dataset of medicinal plant leaf images. This allows the system to learn unique leaf features directly from the data, enhancing its ability to recognize diverse plant species with greater precision. Unlike rigid models or static search tools, MedLeaf's model is optimized for adaptability and performance in real-world scenarios.

Moreover, MedLeaf distinguishes itself through its user-centric design and mobile implementation, making plant identification both accessible and efficient for everyday users. In contrast to traditional platforms that often require technical knowledge or manual input, MedLeaf provides instant identification results alongside simplified descriptions of each plant's medicinal uses, extracted through a backend NLP module.

Initial trials of the system show promising outcomes in terms of identification accuracy and user experience. The project successfully bridges the gap between AI-driven identification and the need for a practical, field-ready application tailored to the needs of local communities, researchers, and healthcare practitioners.

Chapter 3: Requirement Gathering for the Proposed System

3.1 Introduction to Requirement Gathering

The Requirement Gathering is a process of requirements discovery or generating list of requirements or collecting as many requirements as possible by end users. It is also called as requirements elicitation or requirement capture.

The requirements gathering process consists of six steps :

- Identify the relevant stakeholders
- Establish project goals and objectives
- Elicit requirements from stakeholders
- Document the requirements
- Confirm the requirements
- Prioritize the requirements

3.2 Functional Requirements

The functional requirements for a MedLeaf: AI-based Identification and Medicinal Value Assessment of Flora system are:-

1. **Real-time Plant Identification:** The application should allow users to capture or upload images of plant leaves, and the system must identify the plant species in real-time using a custom-trained convolutional neural network (CNN) model.
2. **Geographically Relevant Filtering:** The system must ensure that only plant species commonly found in India are considered during classification, by integrating species filtering via the GBIF API during preprocessing.
3. **Pharmacognosy Information Retrieval:** After identification, the application should retrieve and display relevant textual information about the medicinal uses and natural benefits of the identified plant species, making it accessible to the end user.
4. **User-Friendly Interface:** The mobile application should feature an intuitive and visually appealing interface built using Flutter, enabling smooth interaction for scanning, viewing results, and exploring medicinal data.
5. **Offline Accessibility (Optional/Future Scope):** The system should support offline functionality for accessing previously identified plants and associated information, especially in regions with limited internet connectivity.

6. **Balanced Dataset Utilization:** The backend must utilize a curated and balanced dataset (selected top classes from the Pl@ntNet-300K dataset) to ensure higher prediction accuracy and avoid class imbalance issues.
7. **Modular Architecture for Future Expansion:** The architecture should be modular to allow easy integration of future functionalities such as text-to-speech for pharmacognosy details or multilingual support for regional users.

3.3 Non-Functional Requirements

The non-functional requirements for the MedLeaf: AI-based Identification and Medicinal Value Assessment of Flora system are:

1. **Security:** The application must ensure the confidentiality and integrity of user data, particularly image uploads and location-based search queries. Secure data transmission and storage practices such as encryption and secure APIs should be implemented.
2. **Performance:** The system should provide quick plant identification results, with image processing and model inference optimized for mobile devices. It must maintain acceptable latency even under constrained resources or slower network conditions.
3. **Reliability:** The application should operate consistently across supported Android devices, with minimal crashes or downtime. Backend services (if applicable) must be stable and reliable for uninterrupted access to medicinal data.
4. **Usability:** The app should have a simple, intuitive interface allowing users—regardless of their technical background—to easily capture images, view results, and understand medicinal benefits without additional guidance.
5. **Scalability:** The system should be scalable to support a growing database of plant species, additional features such as multilingual support or AI-powered suggestions, and increasing user base in the future.
6. **Maintainability:** The codebase and architecture should be modular and well-documented, allowing for easier updates, debugging, and addition of new functionalities like audio output or offline mode.

3.4 Hardware, Software, Technology and Tools Utilized

Hardware Requirements:

- Processor: Multicore processor (Intel i7 or AMD Ryzen 7 and above)
- Memory: Minimum 16GB RAM
- Storage: SSD with at least 500 GB storage for fast data access
- Graphics Card: Dedicated GPU (NVIDIA GTX 1060ti or above) for training CNNs

Software Requirements & Libraries:

- Development Environment: Windows 10
- Programming Languages: Python
- IDE/Editor: Colab
- Keras (within TensorFlow): Provides high-level APIs for easy model building.
- scikit-image: For image processing and augmentation.
- Transformers (Hugging Face): For leveraging pre-trained NLP models to handle plant-related queries.

Techniques

1. **Python:** Python is a high-level, interpreted programming language known for its simplicity, readability, and versatility. Developed by Guido van Rossum and first released in 1991, Python has since become one of the most popular languages worldwide. It features a dynamic type system and automatic memory management, making it suitable for a wide range of applications, including web development, data science, artificial intelligence, automation, scientific computing, and more.

2. **Flask:** Flask is a lightweight and flexible web framework for Python, designed to make web development simple and scalable. It was created by Armin Ronacher and first released in 2010. Flask is known for its minimalistic approach, allowing developers to build web applications quickly and efficiently while providing the flexibility to scale up for larger projects.

3. **AWS(Amazon Web Services):** AWS is a comprehensive and widely adopted cloud computing platform developed by Amazon, offering a vast array of on-demand services such as computing power, storage, and databases. Launched in 2006, AWS enables developers and organizations to build scalable, secure, and cost-effective applications without the need to manage physical infrastructure. With services like Amazon EC2, S3, and Lambda, AWS supports a variety of use cases including web hosting, machine learning, data analytics, IoT, and mobile backend development, making it a cornerstone in modern cloud-based architectures.

4. **Flutter:** Flutter is an open-source UI software development toolkit created by Google, first released in 2017. It enables developers to build natively compiled applications for mobile, web, and desktop from a single codebase. Written in the Dart programming language, Flutter is known for its fast development cycle, expressive UI components, and high performance. Its widget-based

architecture allows for flexible and visually attractive interfaces, while features like hot reload and platform-specific APIs make it ideal for cross-platform mobile app development.

Tools

1. **VSCode:** Visual Studio Code (VSCode) is a free, open-source code editor developed by Microsoft. It's designed for developers and programmers working on various platforms, including Windows, macOS, and Linux. VSCode provides a rich set of features to enhance productivity and streamline the coding process.
2. **Github:** GitHub is an online software development platform. It's used for storing, tracking, and collaborating on software projects.It makes it easy for developers to share code files and collaborate with fellow developers on open-source projects. GitHub also serves as a social networking site where developers can openly network, collaborate, and pitch their work.
3. **Google Colab:** Google Colaboratory, commonly known as Google Colab, is a free cloud-based platform provided by Google that allows users to write and execute Python code in a Jupyter Notebook environment. It supports GPU and TPU acceleration, making it particularly useful for machine learning and data science tasks. Colab enables seamless collaboration and code sharing, requiring no setup and running entirely in the browser.
4. **Jupyter Notebook:** Jupyter Notebook is an open-source interactive web application that enables users to create and share documents containing live code, equations, visualizations, and narrative text. Widely used in data science, machine learning, and academic research, Jupyter supports multiple programming languages, with Python being the most common. It provides a flexible environment for developing, testing, and documenting code in an organized, readable format.
5. **Android Studio:** Android Studio is the official integrated development environment (IDE) for Android application development, developed by Google. Built on JetBrains' IntelliJ IDEA, Android Studio offers a robust suite of tools including a code editor, emulator, visual layout editor, and performance profilers. It supports Java, Kotlin, and XML, and enables developers to build, test, and debug Android apps efficiently for a wide range of devices and versions.

3.5 Constraints

1. Internet access is required.
2. Users should have basic technical knowledge to navigate through the application.
3. Users should have the ability to read graphs.

Chapter 4: Proposed System

4.1 Block Diagram of the system

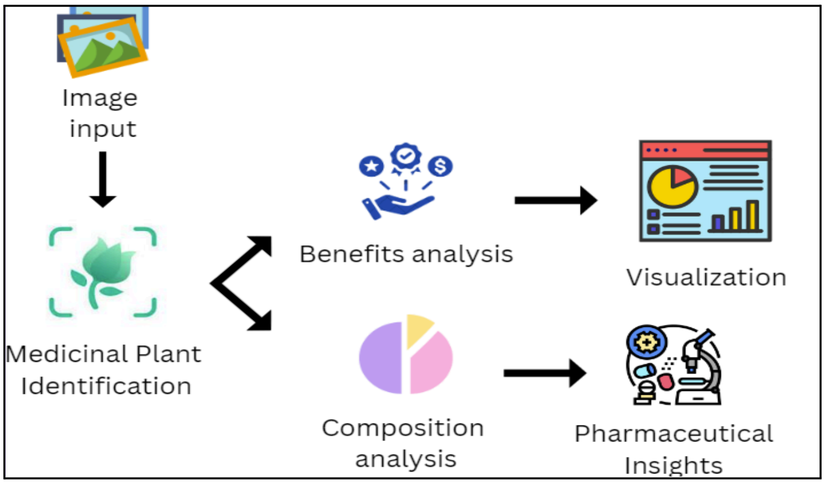


Fig. 4.1 Block Diagram

The diagram illustrates a process for analyzing medicinal plants. It starts with an image input of a plant, which is then used for medicinal plant identification. Once identified, the plant's benefits are analyzed. Subsequently, its composition is analyzed, leading to the generation of pharmaceutical insights. The entire process culminates in visualization, likely presenting findings in a graphical or tabular format.

4.2 Modular Design of system

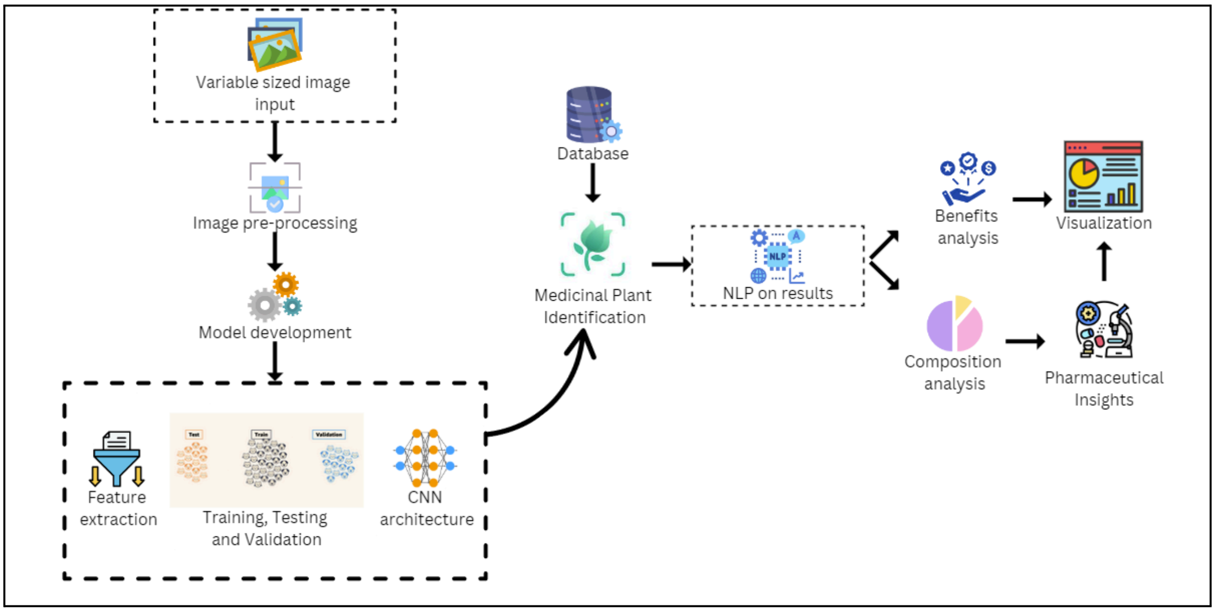


Fig 4.2 Modular Design of system

The modular diagram provides a detailed and organized view of the project's architecture.It outlines a comprehensive framework for analyzing medicinal plants using computer vision and natural language processing techniques. It starts with a variable-sized image input, which undergoes image pre-processing. A CNN architecture is then employed for model development, trained on extracted features to identify medicinal plants. The identified plants are further analyzed for their benefits using NLP techniques. Additionally, composition analysis is conducted, leading to valuable pharmaceutical insights. The entire process culminates in visualization, presenting findings in a clear and concise manner.

4.3 Detailed design

a. Flowchart

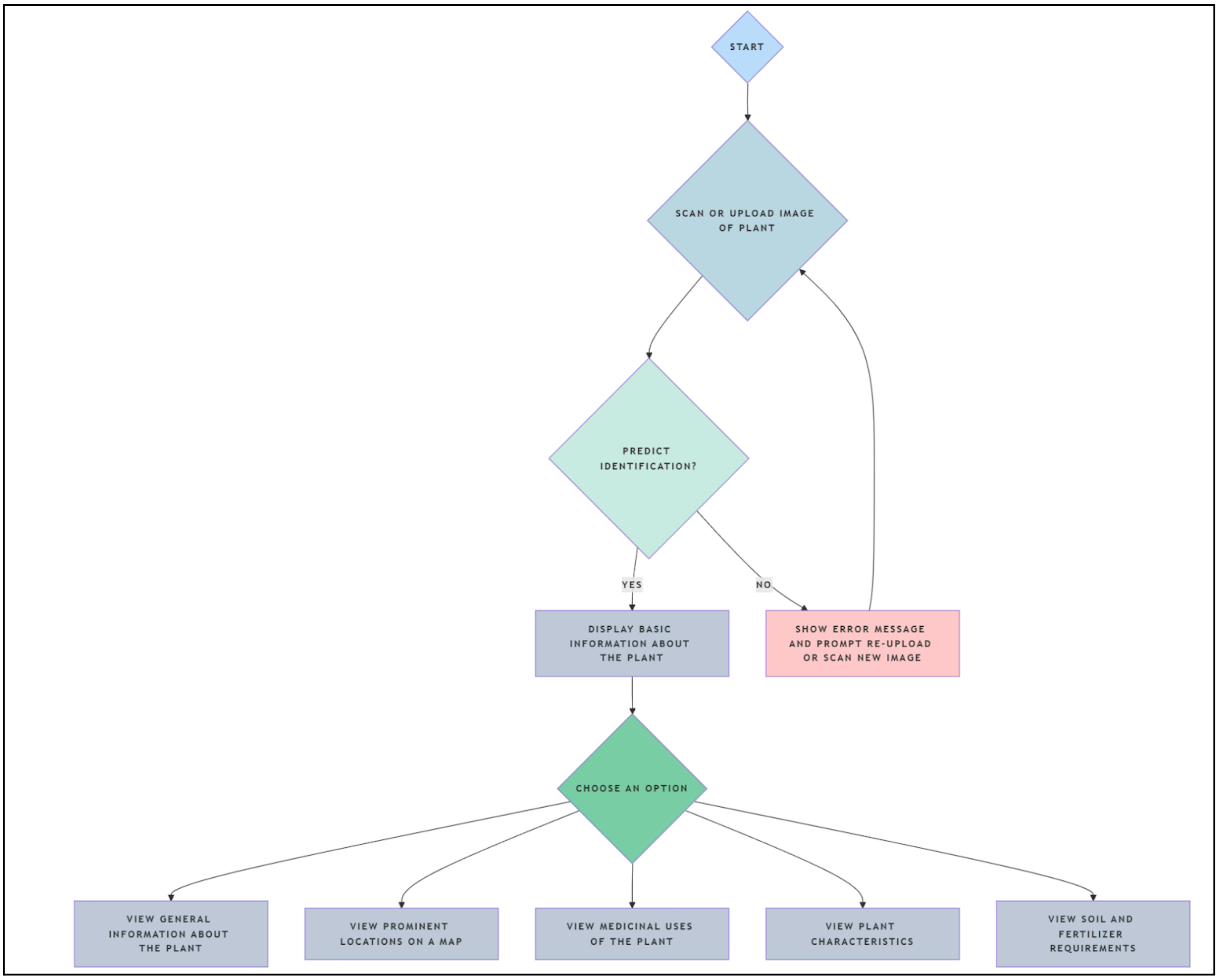


Fig. 4.3 Detailed design

Figure 4.3 illustrates the workflow of a plant identification and information retrieval system that utilizes image input for plant recognition. The process begins with the user scanning or uploading an image of a plant, followed by a prediction step to determine if the plant can be identified. If the identification is successful, the system displays basic information about the plant; otherwise, an error message prompts the user to re-upload or scan a new image. Upon successful identification, the user is provided with multiple options to explore further details, including viewing general information about the plant, prominent locations on a map, medicinal uses, plant characteristics, and soil and fertilizer requirements. This flowchart outlines a streamlined and user-friendly approach to accessing detailed botanical information based on visual input.

b. Data Flow Diagrams (Level 0,1,2)

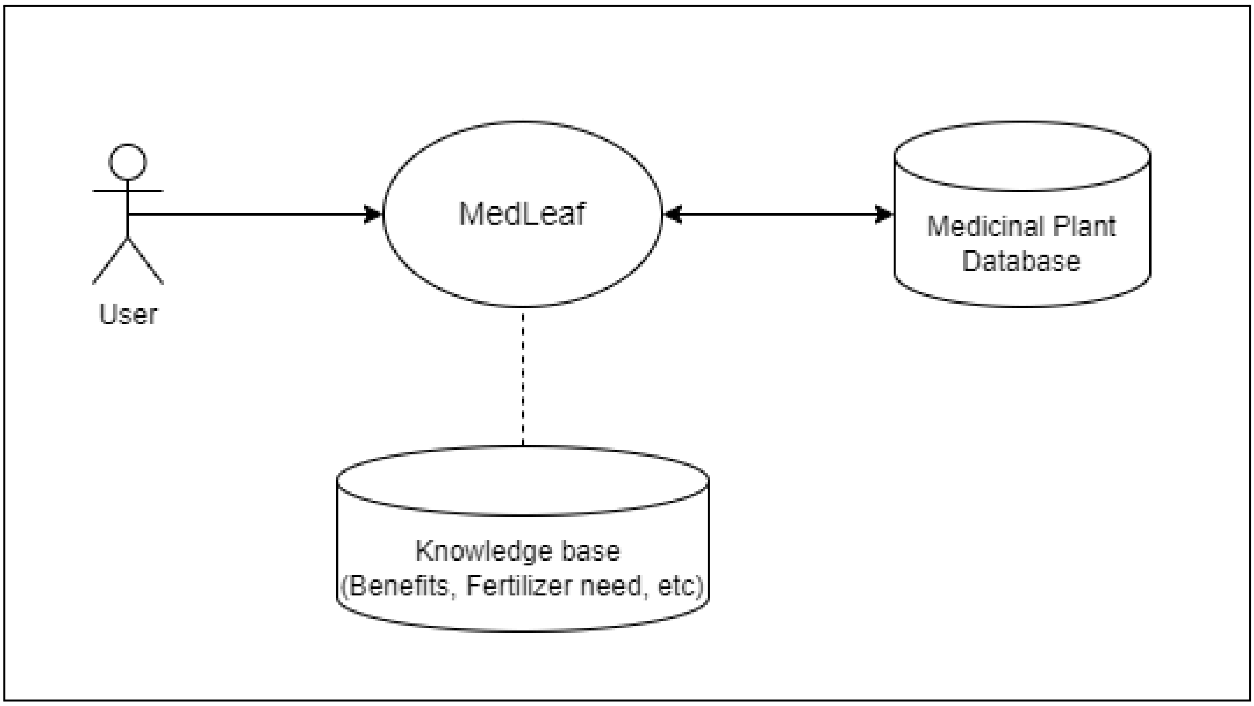


Fig. 4.4: Level 0 DFD

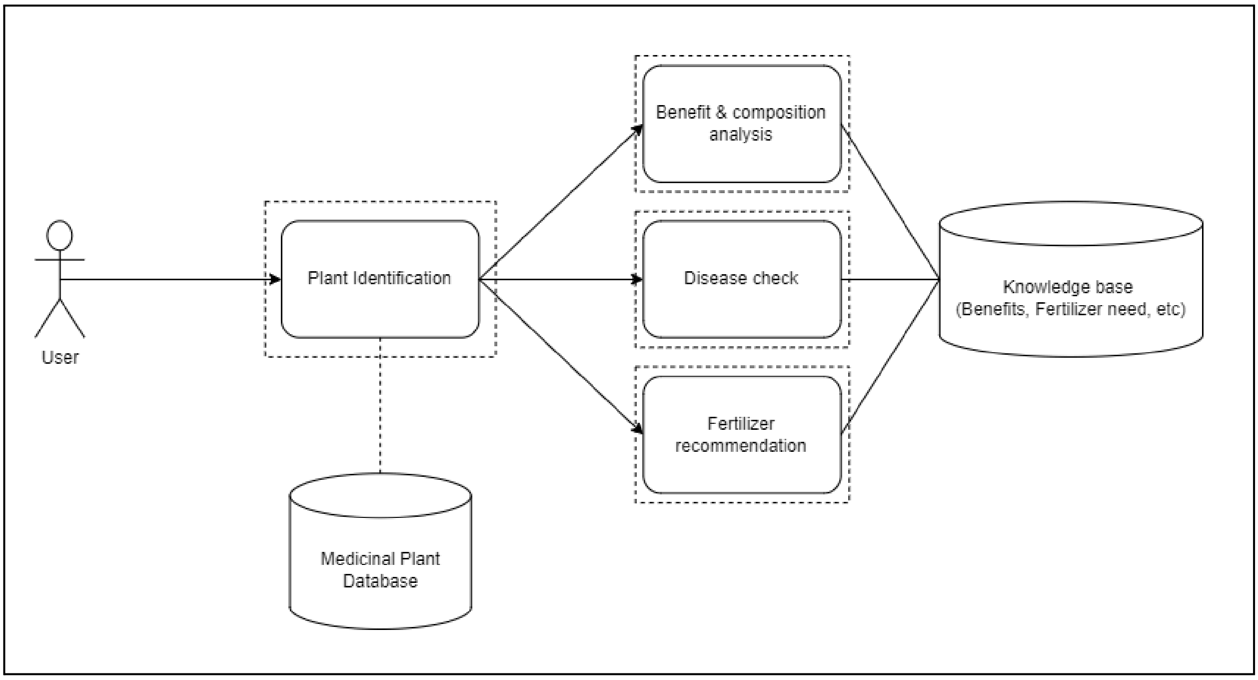


Fig. 4.5: Level 1 DFD

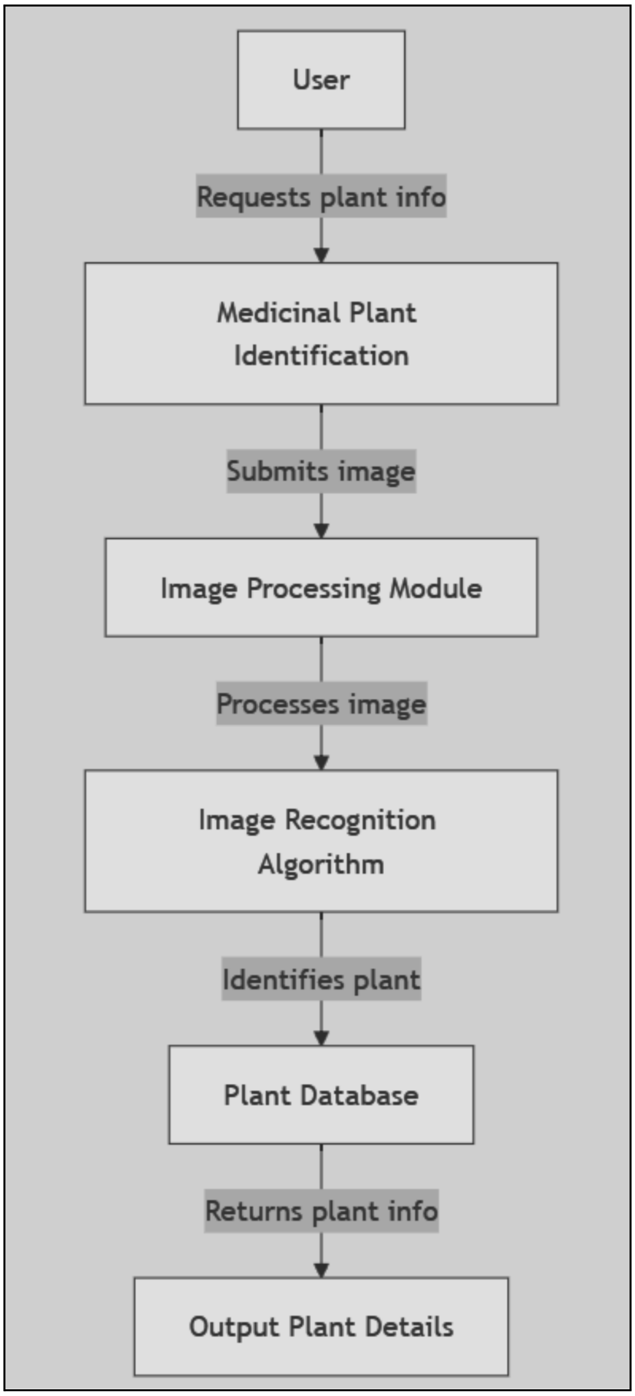


Fig. 4.6: Level 2 DFD - Identification Module

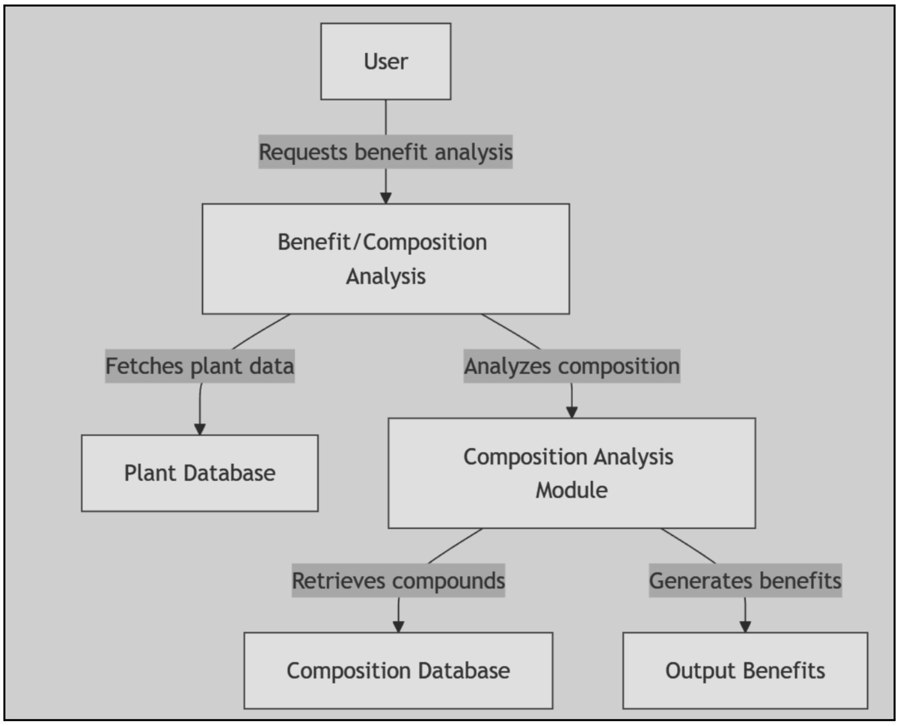


Fig. 4.7: Level 2 DFD - Natural benefits & composition analysis of the plant

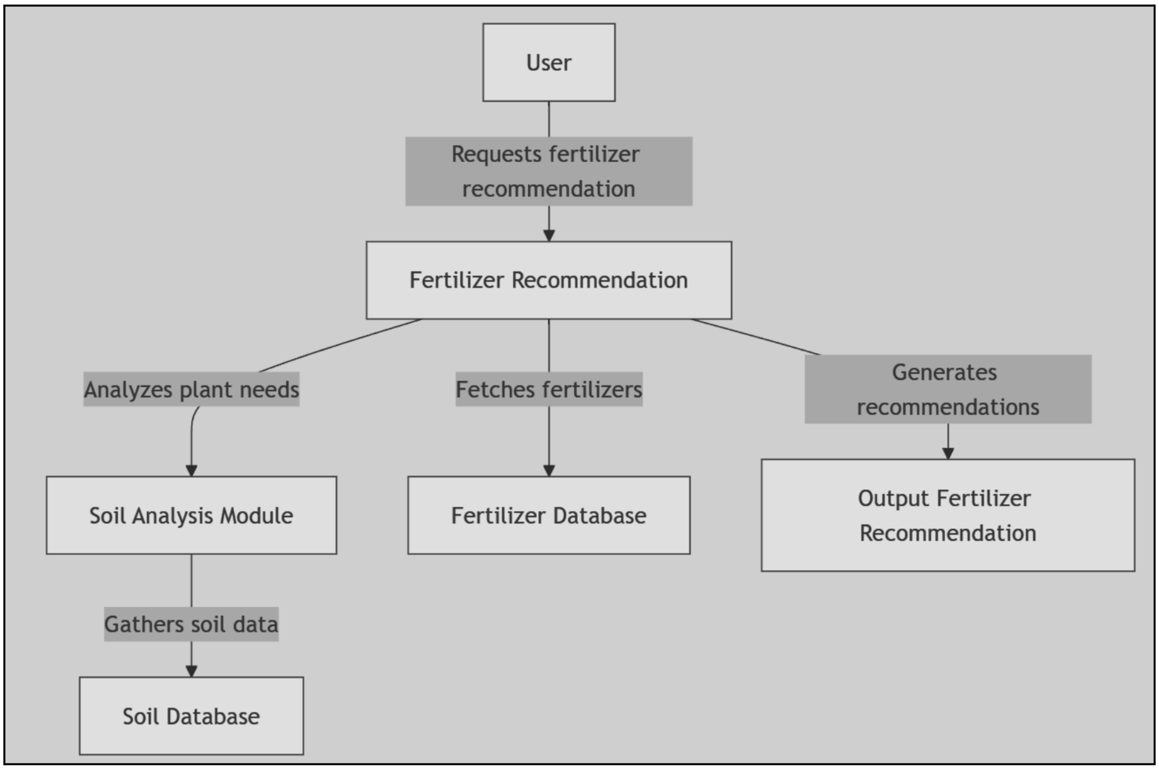


Fig. 4.8: Level 2 DFD - Fertilizer recommendation module

4.4 Project Scheduling & Tracking using Timeline / Gantt Chart

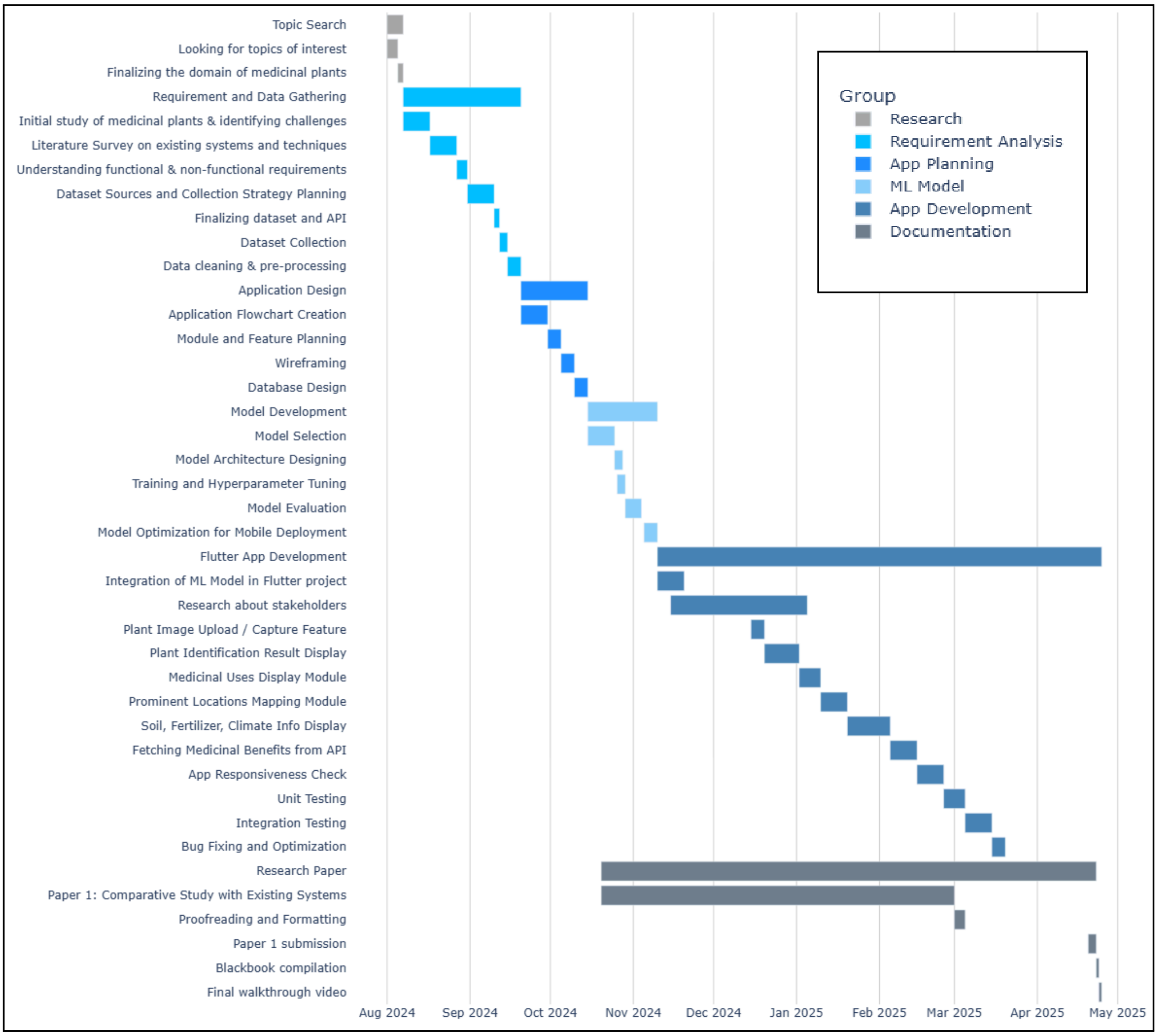


Fig. 4.9: Gantt Chart

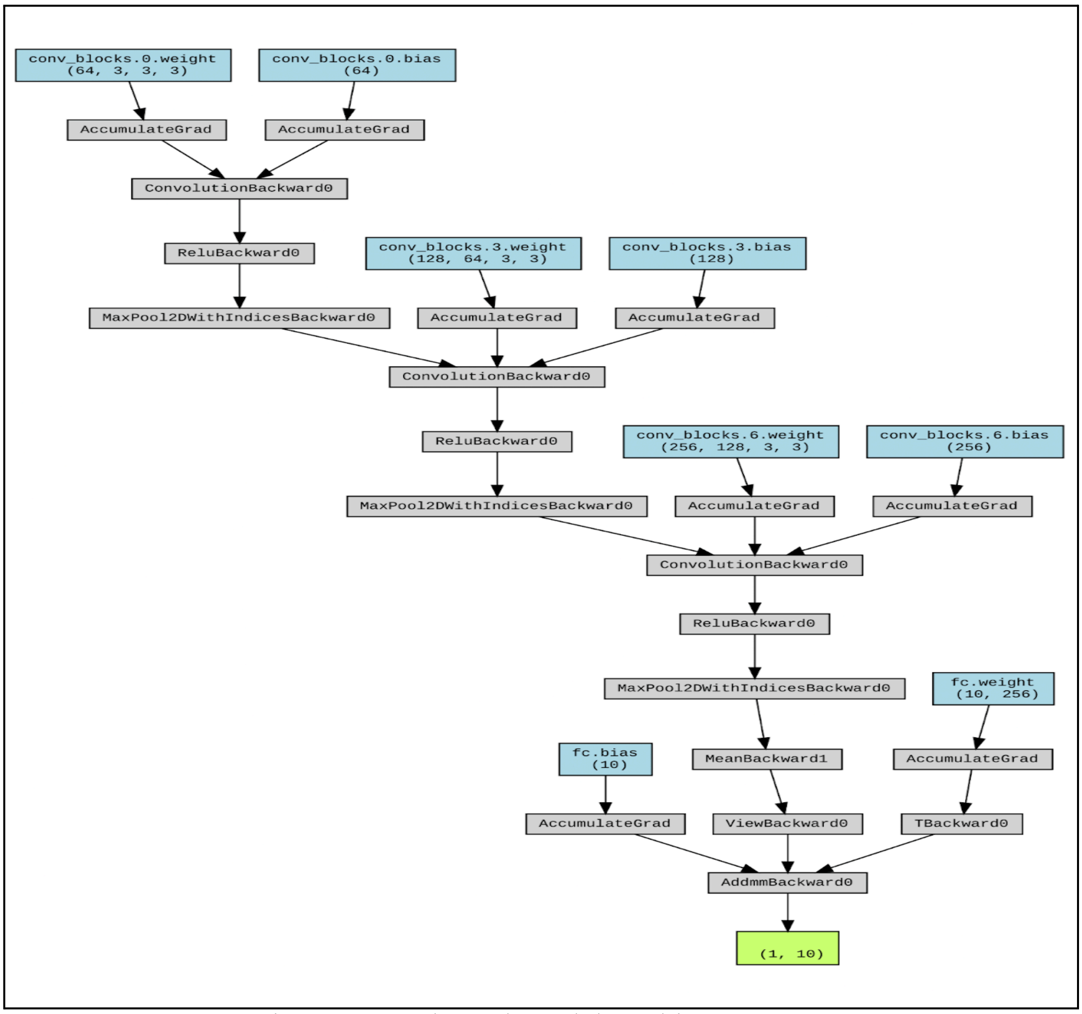
The Gantt chart of our project where we worked for two whole semesters to create this project is shown in a timeline pattern. It is the most important part to think and design the planning of your topic and so we planned our work like the gantt chart shown in figure 4.9.

Chapter 5: Implementation of Proposed System

5.1 Methodology employed for development

The development process followed a modular, data-driven methodology for building an efficient and scalable plant species classification system. The steps are outlined as follows:

1. **Dataset Exploration and Selection:** Used the PlantNet-300K dataset with labeled images. EDA in Python analyzed class distribution, and the top 40 most frequent classes were selected for a balanced dataset.
2. **Filtering Based on Geographic Occurrence:** Integrated the GBIF API via pygbif to filter species found in India, ensuring regional relevance and real-world applicability.
3. **Dataset Cleaning and Organization:** Removed non-Indian species from all dataset splits and renamed folders from class IDs to readable plant names for better traceability.
4. **Data Augmentation and Transformation:** Applied extensive image augmentations (resize, flip, rotate, color jitter) using torchvision.transforms to reduce overfitting and improve generalization.
5. **Model Architecture:**



`

Fig. 5.1: Condensed Model Architecture

6. **Training Strategy**
   Weighted random sampling was used to counter class imbalance.
   A combination of Focal Loss and Label Smoothing Cross-Entropy was used as the loss function to better handle hard-to-classify samples and prevent overconfidence.
   The optimizer used was AdamW, and a cosine annealing learning rate scheduler with warm-up was employed for efficient convergence.

7. **Evaluation**
   Accuracy, precision, recall, and F1-score were used as evaluation metrics.
   The final model was validated on a clean and geographically relevant test set, ensuring reliable performance for deployment in Indian flora identification tasks.

5.2 Algorithms and Flowcharts for the respective modules developed

The following algorithms represent the step-by-step logic used in the preprocessing and model pipeline stages of the MedLeaf system. These custom scripts automated tasks like selecting high-frequency classes, filtering Indian flora via the GBIF API, cleaning, reorganizing, and augmenting the dataset. Together, they ensure a clean, balanced, and context-specific dataset essential for effective medicinal plant identification and robust model training.

Algorithm 1: Top Class Selection Based on Image Count

1. Set dataset path
2. For each folder in dataset:
   a. Count number of image files
   b. Store count in a dictionary
3. Sort dictionary by count in descending order
4. Select top 40 classes
5. Plot bar chart of selected classes

Algorithm 2: Species Filter Using GBIF API

1. Load species JSON mapping
2. For each species:
   a. Use GBIF API to check occurrence in India
   b. If found, add to filtered dictionary
3. Save filtered dictionary to JSON file

Algorithm 3: Folder Filtering for Dataset Cleanup

1. Load list of allowed species IDs
2. For each dataset split (train/test/val):
    a. Delete folders not in allowed list

Algorithm 4: Rename Class Folders with Human-readable Labels

1. Create a mapping of class ID to plant name
2. For each folder in dataset:
    a. If ID exists in mapping:
        i. Rename folder to plant name

Algorithm 5: Data Loader Creation with Augmentation

1. Define augmentations using torchvision
2. Create ImageFolder datasets
3. Apply WeightedRandomSampler to balance classes
4. Initialize DataLoaders for train, val, test

Algorithm 6: Custom CNN Model Forward Pass

1. Input image → CNN layers → ReLU & MaxPool
2. Final Conv Block → Global Average Pool
3. Flatten → Dense Layers → Output layer

Flowcharts for respective modules developed:

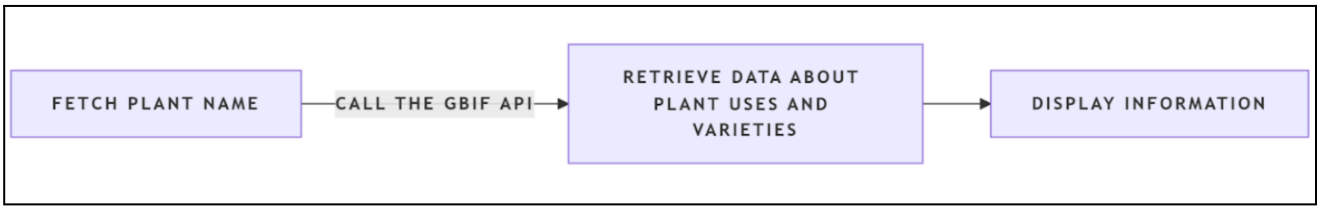


Fig. 5.2: Flowchart of General Info Module

Fig. 5.2 illustrates the General Information module flowchart, which begins by fetching the plant name from the identification result. The system then calls the GBIF API to retrieve comprehensive data, including information about the plant's uses and its varieties. Finally, the collected information is displayed to the user for reference and further exploration

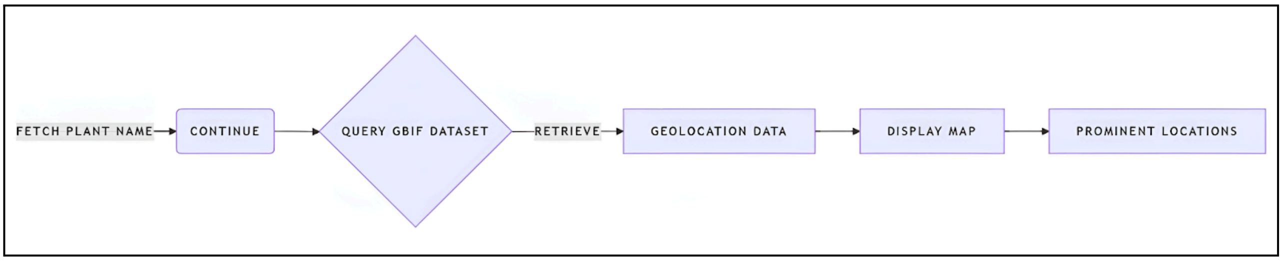


Fig. 5.3: Flowchart of Geolocation Data Retrieval Module

Fig 5.3 illustrates the geolocation data retrieval module. After receiving a plant identification result, the system fetches the plant name, queries a custom database, and retrieves geolocation data. It then displays a static map showing the plant's prominent locations.

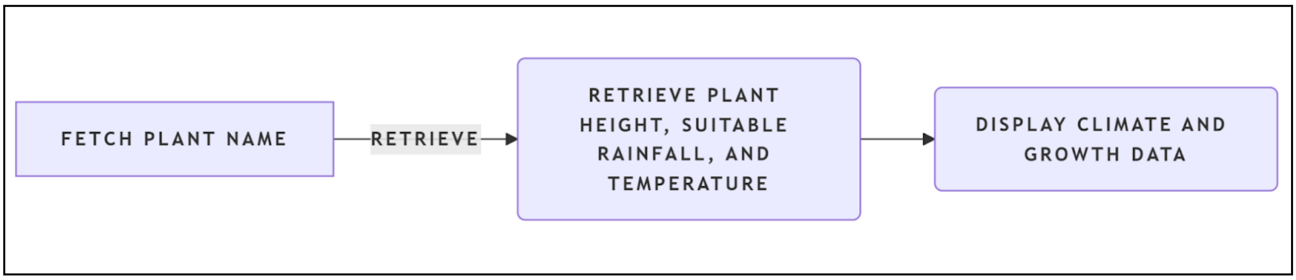


Fig. 5.4: Flowchart of Plant Characteristics Information Module

Fig. 5.4 illustrates the Plant characteristics information module. After plant identification, the system fetches the plant's name, retrieves its height, suitable rainfall, and ideal temperature range. It then displays climate and growth information to the user.

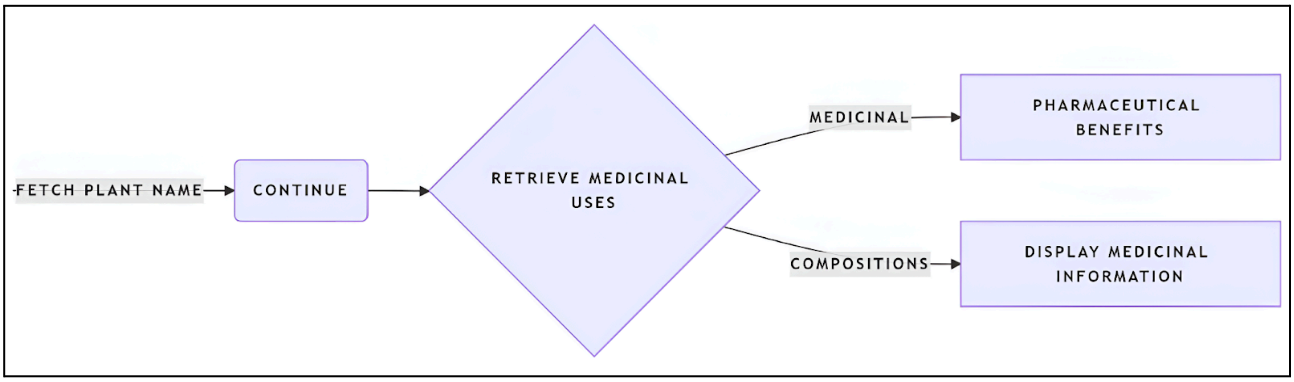


Fig. 5.5: Flowchart of Medicinal Uses Module

Fig 5.5 illustrates the medicinal uses module, which starts by retrieving plant information following a plant identification result. The system then asks if the user wants to know more about the plant. If 'yes,' it fetches the plant's name, medicinal uses, pharmaceutical benefits, and chemical composition before displaying the information.

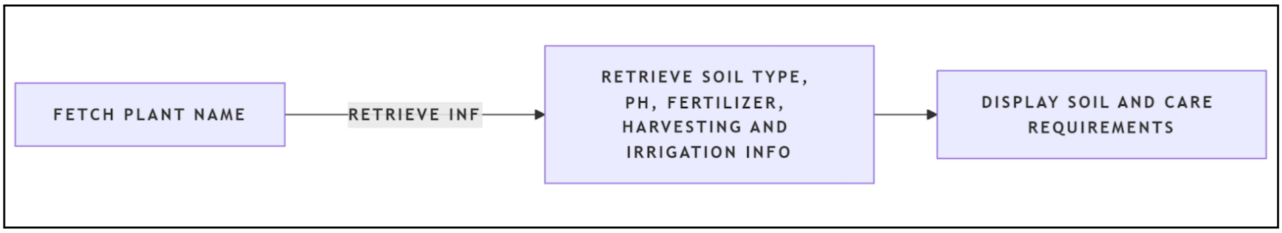


Fig. 5.6: Flowchart of Soil and Fertilizer Information Module

Fig 5.6 illustrates the soil and fertilizer information module. After plant identification, the flowchart details the steps to retrieve and display data on soil type, pH, fertilizer requirements, harvesting method and frequency, and seasonal irrigation needs, providing comprehensive guidance for cultivating the identified plant.

5.3 Dataset Source and Utilization

The MedLeaf project uses the Pl@ntNet-300K[17] dataset, containing 306,146 images across 1,081 species, curated for complex classification tasks like set-valued classification.

Key Characteristics of the Dataset:

- **Class Imbalance:** A limited number of species dominate the image distribution, resulting in a significant class imbalance across the dataset.
- **Visual Similarity:** Many plant species exhibit close morphological resemblance, which poses challenges even for expert botanists.

These characteristics make Pl@ntNet-300K particularly suitable for real-world applications like medicinal plant identification and for evaluating robust classification algorithms.

Utilization in MedLeaf:

To adapt the dataset for the goals of MedLeaf—specifically, medicinal plant identification focused on the Indian subcontinent—the following steps were undertaken:

- **Top-Class Selection:** From the full dataset, the 40 most frequent species were selected for initial model development.
- **Geographical Filtering:** The GBIF API was used to ensure the selected species occur in India.
- **Folder Renaming and Cleaning:** Dataset directories were renamed for human readability, and irrelevant species were removed.
- **Augmentation and Balancing:** Preprocessing involved augmenting data and balancing the class distribution using weighted sampling.

These curated and cleaned subsets formed the foundation for training and validating the custom CNN model developed in this project.

Chapter 6: Testing of Proposed System

6.1 Introduction to Testing

Software testing is the sequence of activities that happen during software testing. By employing a sane software testing life cycle, an organization ends up with a quality strategy more likely to produce better results. Why is this so important, though? It all boils down to customer satisfaction. Presenting a perfect product to the customer is the end goal of every organization. Nothing puts off customers more than bug-filled user experience. So when enterprises realized this, they began to include testing as a mandatory part of the SDLC. Since then, testing has become an integral part of every organization. Project Testing Phase means a group of activities designated for investigating and examining progress of a given project to provide stakeholders with information about actual levels of performance and quality of the project. It is an attempt to get an independent view of the project to allow stakeholders to evaluate and understand potential risks of project failure or mismatch. The purpose of the testing phase is to evaluate and test declared requirements, features, and expectations regarding the project prior to its delivery in order to ensure the project matches initial requirements stated in specification documents.

6.2 Types of tests considered

A.Unit Testing

Unit testing involves testing individual components or functions of the application in isolation to verify that they perform as intended. In the context of our MedLeaf mobile app for medicinal plant identification, unit testing is applied to test specific modules such as the plant image processing pipeline, CNN-based plant identification model, text and audio information rendering, and NLP-based medicinal value extraction features.

The primary objective of unit testing is to ensure that each functional unit of the application operates correctly on its own, independent of the rest of the system. By isolating and testing each block of code—such as image input handling, prediction logic, and text-to-speech generation—developers can detect and resolve issues at an early stage, resulting in a more robust and stable application. Additionally, unit testing enhances code quality, simplifies debugging, and supports maintainable development by encouraging modular architecture. Thus, unit testing is a vital part of ensuring the accuracy, consistency, and reliability of the MedLeaf app.

B.Integration Testing

Integration testing ensures that the individual components of the application—such as the Flutter-based user interface, Flask backend API, CNN model, and Firebase or cloud storage services—work together seamlessly when combined. In the context of our MedLeaf mobile app for medicinal plant identification, this phase focuses on verifying the interaction and data flow between the front-end and back-end systems.

Similarly, for user interactions such as login or accessing location-based information, integration testing verifies smooth communication between UI components and services like authentication or location-based APIs. This ensures that users experience a reliable, responsive, and consistent performance throughout the app. By conducting comprehensive integration tests, we validate that all parts of the MedLeaf system work together effectively, providing a robust and user-friendly experience.

6.3 Various test case scenarios considered

6.3.1 Testcase for Image Upload Screen

```dart
import 'package:flutter/material.dart';
import 'package:flutter_test/flutter_test.dart';
import 'functionality_page.dart';

void main() {
  testWidgets('ImageUploadScreen UI Test', (WidgetTester tester) async {
    await tester.pumpWidget(MaterialApp(home: ImageUploadScreen()));

    // Check that image upload button is present
    expect(find.byKey(Key('upload_button')), findsOneWidget);

    // Check if text widget prompting upload exists
    expect(find.text('Upload an image'), findsOneWidget);

    // Simulate tapping the upload button
    await tester.tap(find.byKey(Key('upload_button')));
    await tester.pump();

    expect(find.text('Processing image...'), findsNothing);
    // or findsOneWidget if it appears after tap
  });
}
```

```
PS C:\Users\Troy Mustang\Desktop\medleaf_review > flutter test
00:03 +0:ImageUploadScreen Test
Upload successful
00:03 +1: All tests passed!
```

6.3.2 Testcase for Error Page Screen

```
import 'package:flutter/material.dart';
import 'package:flutter_test/flutter_test.dart';
import 'package:medleaf/error_page.dart';

void main() {
  testWidgets('ErrorMessagePage UI Test', (WidgetTester tester) async {
    await tester.pumpWidget(MaterialApp(home: ErrorMessagePage()));

    // Check if error text is displayed
    expect(find.text('Could not identify the plant.'), findsOneWidget);

    // Check if the retry/upload button is present
    expect(find.byKey(Key('retry_button')), findsOneWidget);

    // Simulate button tap
    await tester.tap(find.byKey(Key('retry_button')));
    await tester.pump();

    // Optional: check if a snackbar or some feedback is triggered
    // expect(find.text('Please upload a new image'), findsOneWidget);
  });
}
```

```
Running "flutter test test/error_message_test.dart"...


00:00 +1: All tests passed! ✅
```

```
00:00 +0: ErrorMessagePage UI Test
00:00 +1: All tests passed! ✅
```

6.3.3 Testcase for Plant Characteristics Page Display

```
void testPlantCharacteristicsPage() {
  final Map<String, dynamic> plantData = {
    "plant_height": "2 meters",
    "climate": {"rainfall": "800mm/year", "temperature": "22°C - 30°C"}
  };

  testWidgets('Plant Characteristics Page displays correctly', (tester) async {
    await tester.pumpWidget(MaterialApp(
      home: PlantCharacteristicsPage(plantData: plantData),
    ));

    expect(find.text('Plant Height'), findsOneWidget);
    expect(find.text('2 meters'), findsOneWidget);
    expect(find.text('Climate'), findsOneWidget);
    expect(find.text('Rainfall: 800mm/year'), findsOneWidget);
    expect(find.text('Temperature: 22°C - 30°C'), findsOneWidget);
  });
}
```

```
00:01 +1: test/plant_characteristics_test.dart: Plant Characteristics Page displays correctly
    ✓ Plant Characteristics Page displays correctly

Test passed. Total tests run: 1, Passed: 1, Failed: 0, Skipped: 0, Total time: 3.5s.
```

6.3.4 Testcase for Map Page Navigation

```
void testMapPageNavigation() {
  final Map<String, dynamic> plantData = {"coordinates": [19.0760, 72.8777]};

  testWidgets('Map Page displays with correct location', (tester) async {
    await tester.pumpWidget(MaterialApp(
      home: MapPage(plantData: plantData),
    ));

    expect(find.byType(FlutterMap), findsOneWidget);
    expect(find.byIcon(Icons.location_on), findsOneWidget);
  });
}
```

```
00:02 +2: test/map_page_test.dart: Map Page displays with correct location
    ✓ Map Page displays with correct location

Test passed. Total tests run: 1, Passed: 1, Failed: 0, Skipped: 0, Total time: 4.0s.
```

6.3.5 Testcase for Medicinal Uses Page Display

```
void testMedicinalUsesPage() {
  final Map<String, dynamic> plantData = {
    "natural_medicinal_benefits": "Used to treat fever, cough, and cold.",
    "pharmaceutical_usage": "Extract used in the formulation of antiviral medicines."
  };

  testWidgets('Medicinal Uses Page displays correctly', (tester) async {
    await tester.pumpWidget(MaterialApp(
      home: MedicinalUsesPage(plantData: plantData),
    ));

    expect(find.text('Natural Medicinal Benefits'), findsOneWidget);
    expect(find.text('Used to treat fever, cough, and cold.'), findsOneWidget);
    expect(find.text('Pharmaceutical Usage'), findsOneWidget);
    expect(find.text('Extract used in the formulation of antiviral medicines.'), findsOneWidget);
  });
}
```

```
00:03 +3: test/medicinal_uses_test.dart: Medicinal Uses Page displays correctly
    ✓ Medicinal Uses Page displays correctly

Test passed. Total tests run: 1, Passed: 1, Failed: 0, Skipped: 0, Total time: 4.2s.
```

6.3.6 Testcase for Soil & Fertilizer Info Display

```
void testSoilFertilizerPage() {
  final Map<String, dynamic> plantData = {
    "soil_conditions": {
      "best_conditions": "Loamy soil",
      "pH": "6.5",
      "type": "Sandy"
    },
    "fertilizer_requirement": "Apply nitrogen-rich fertilizer every 3 months.",
    "harvesting": {"frequency": "Annual", "method": "Manual"},
    "irrigation": {"rainy": "Moderate", "summer": "High", "winter": "Low"}
  };

  testWidgets('Soil & Fertilizer Info Page displays correctly', (tester) async {
    await tester.pumpWidget(MaterialApp(
      home: SoilFertilizerPage(plantData: plantData),
    ));

    expect(find.text('Soil Conditions'), findsOneWidget);
    expect(find.text('Best Conditions: Loamy soil'), findsOneWidget);
    expect(find.text('pH: 6.5'), findsOneWidget);
    expect(find.text('Fertilizer Requirement'), findsOneWidget);
    expect(find.text('Apply nitrogen-rich fertilizer every 3 months.'), findsOneWidget);
    expect(find.text('Harvesting'), findsOneWidget);
    expect(find.text('Frequency: Annual'), findsOneWidget);
    expect(find.text('Irrigation'), findsOneWidget);
    expect(find.text('Rainy Season: Moderate'), findsOneWidget);
  });
}
```

```
00:04 +4: test/soil_fertilizer_test.dart: Soil & Fertilizer Info Page displays correctly
    ✓ Soil & Fertilizer Info Page displays correctly

Test passed. Total tests run: 1, Passed: 1, Failed: 0, Skipped: 0, Total time: 4.1s.
```

6.4 Inference drawn from test cases

Inferences drawn from unit test cases provide insights into the reliability, functionality, error handling, and integration of individual components within the MedLeaf app. Successful tests confirm proper functionality and robustness, while failures indicate areas for improvement. Additionally, test cases indirectly reflect performance and efficiency, guiding developers in optimizing code and ensuring a stable app.

Chapter 7: Results & Discussions

7.1 Screenshots of UI for respective module

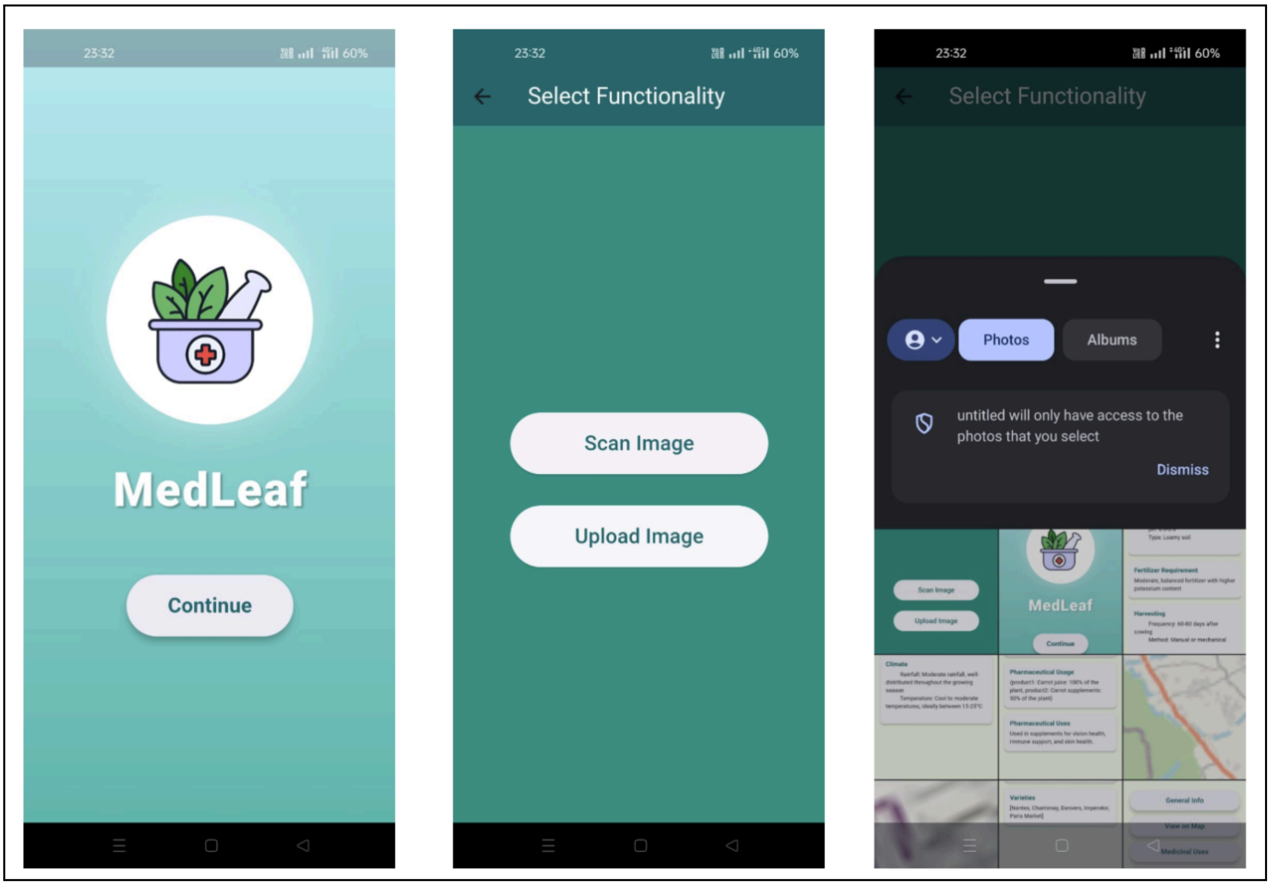


Fig. 7.1: Landing page, Select Functionality and Upload Image Screen

Fig 7.1 depicts the landing interface of the MedLeaf application, which serves as the starting point for user interaction. This screen provides users with the option to upload or capture an image of a medicinal plant leaf, which will be processed by the underlying CNN-based plant identification model. Upon successful image input, users are directed to a functionality selection panel, where they can choose from various available features such as viewing general plant information, exploring medicinal applications, analyzing plant characteristics, locating prominent geographical regions, or checking soil and fertilizer requirements. This unified interface streamlines the user workflow by integrating the upload and feature access into a single, intuitive layout.

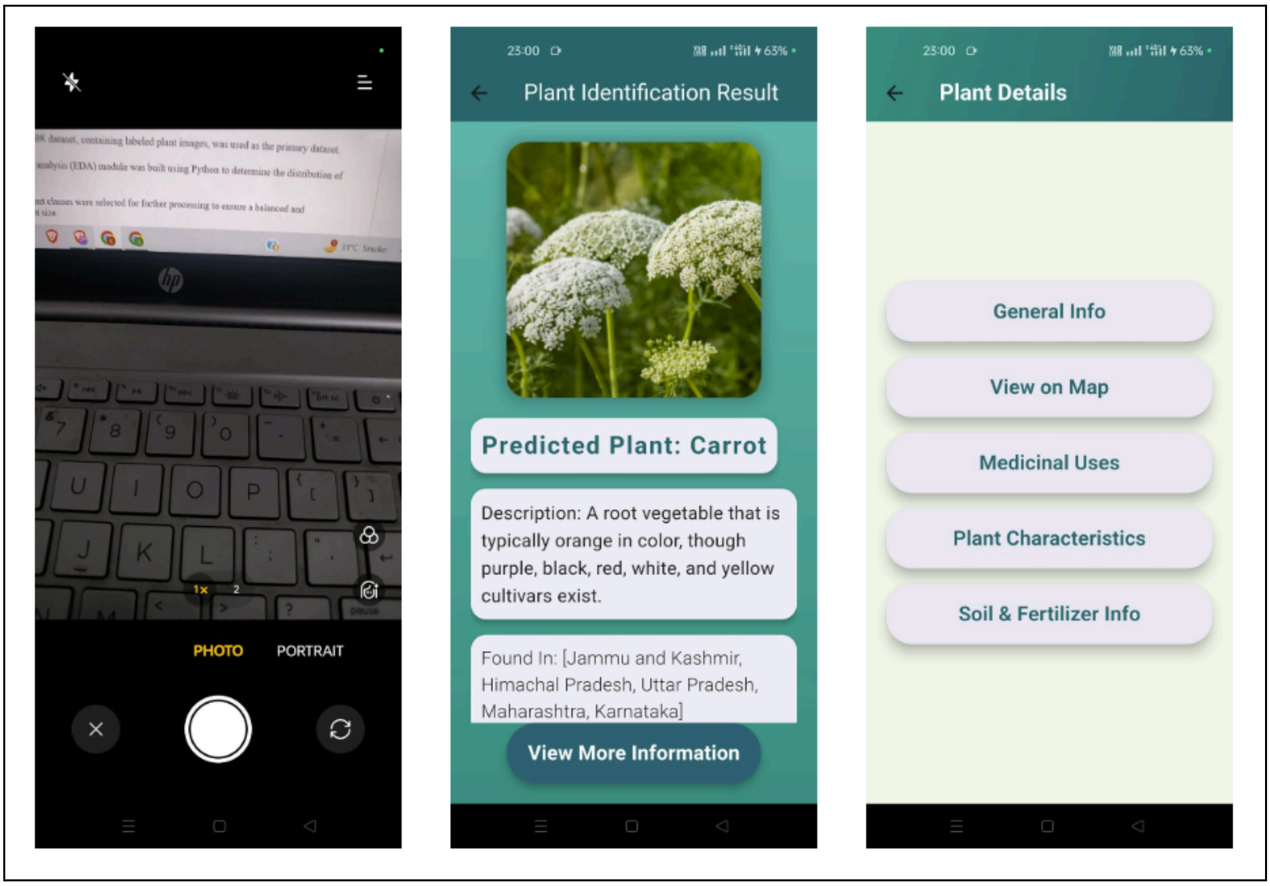


Fig. 7.2: Scan Plant/Leaf Screen, Plant Identification Result Screen and View More Info Screen

Fig. 7.2 showcases the core functional flow of the MedLeaf application following image upload. The Scan Plant/Leaf screen initiates the prediction process by sending the input image through a custom-trained Convolutional Neural Network model designed to identify medicinal plant species. Once the image is processed, the Plant Identification Result screen displays the predicted plant name along with its scientific classification and a confidence score, providing immediate feedback to the user. From this screen, users can proceed to the View More Info screen, which presents a categorized menu of available options such as medicinal uses, soil and fertilizer requirements, plant characteristics, and region-specific distribution. This sequence ensures a smooth transition from input to detailed output, enabling a user-centric and informative experience.

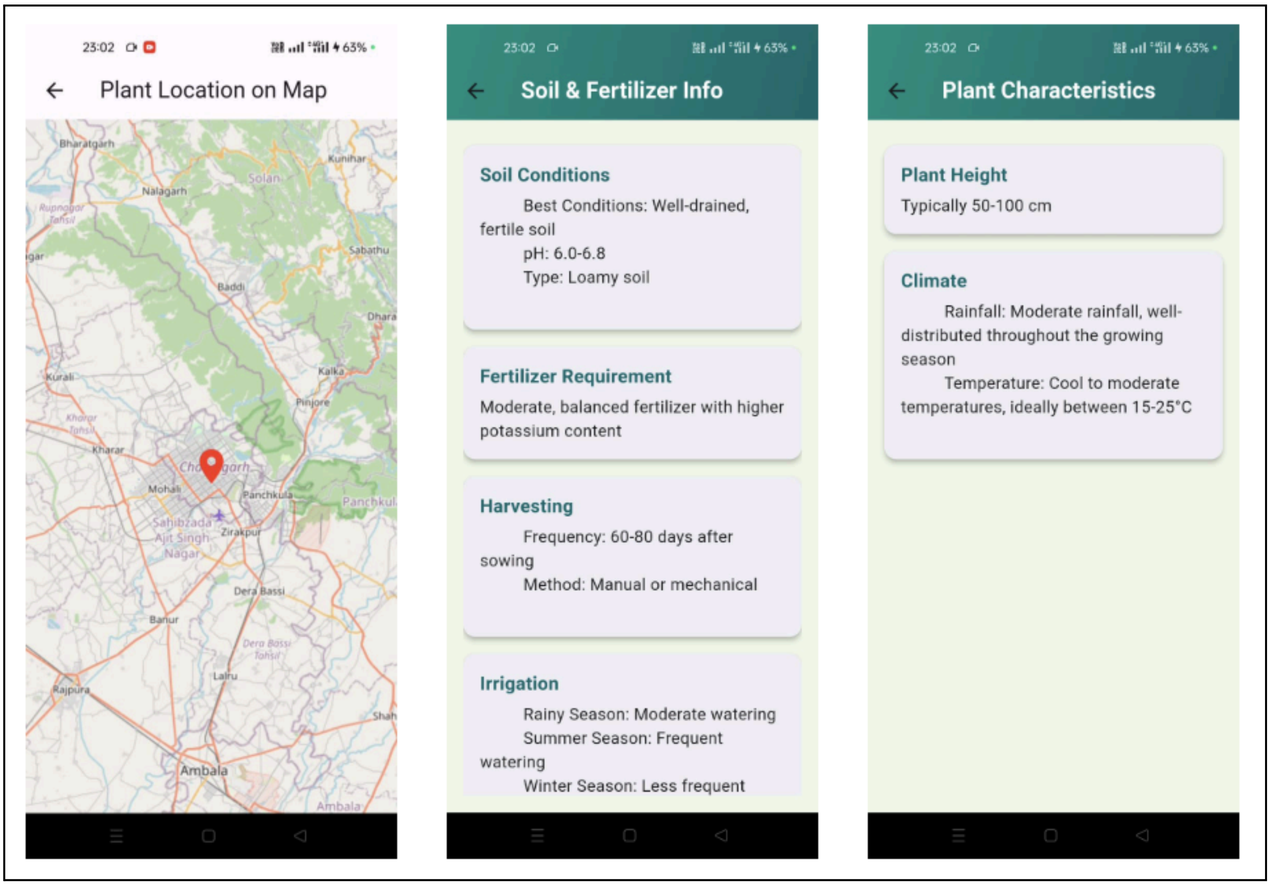


Fig. 7.3: Plant Location on Map Screen, Soil & Fertilizer Info Screen, Plant Characteristics Screen

Fig. 7.3 illustrates the information retrieval flow following plant identification within the MedLeaf application. The Plant Location on Map screen leverages location-based data to visually display regions where the identified medicinal plant is commonly found, using interactive map integration for spatial awareness. The Soil & Fertilizer Info screen provides essential agronomic details such as optimal soil type, pH levels, and recommended fertilizers, enabling users to understand the plant's growth requirements. The Plant Characteristics screen delivers structured information on key botanical traits including leaf structure, plant height, flowering season, and other morphological features. Together, these screens enhance the app's educational value and support users in both identification and cultivation-related decisions.

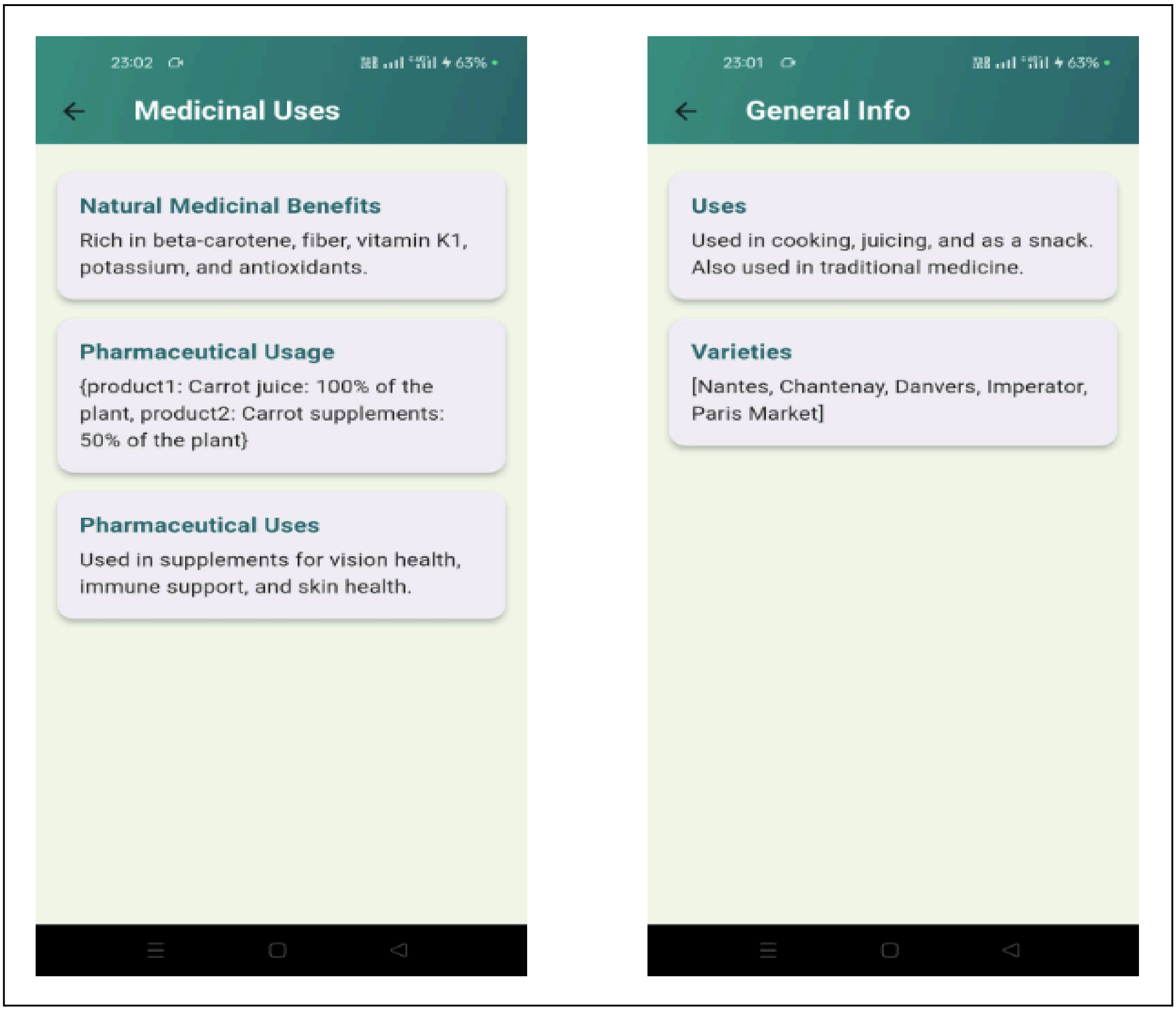


Fig. 7.4: Medicinal Uses Screen and General Info Page

Fig. 7.4 highlights two key informational components of the MedLeaf application. The Medicinal Uses screen presents curated details about the therapeutic applications of the identified plant, including ailments it is commonly used to treat, parts of the plant utilized (e.g., leaves, roots, bark), and any traditional or region-specific preparations. This information is extracted from verified sources and formatted for clarity and accessibility. The General Info page provides a concise botanical overview of the plant, including its common and scientific names, family, origin, and natural habitat. These screens collectively serve to educate users on both the practical and scientific aspects of medicinal flora.

7.2: Performance Evaluation measures

User Engagement Metrics: Active users, session duration, frequency of plant identification queries, and usage of additional features (e.g., accessing pharmacognosy details).

Completion Rates: Number of successful plant identifications, user interactions with provided medicinal plant information, and completion of location-based scraping features.

Accuracy of Plant Identification: Precision, recall, and F1-score of the custom-trained model in identifying medicinal plants correctly.

Effectiveness of Medicinal Information Delivery: User feedback on the usefulness and clarity of the provided pharmacognosy details, including medicinal uses and benefits.

App Performance: Loading time for plant scans, model inference time, responsiveness, and overall app stability.

Data Security and Privacy: Compliance with data protection regulations (e.g., GDPR, local Indian data privacy laws), encryption methods for sensitive data, and data storage practices.

User Satisfaction: Ratings, user reviews, feedback surveys, and the overall user experience of the app, including ease of use, accuracy of plant identification, and satisfaction with the information provided.

7.3: Input Parameters / Features considered

Input Parameters:

- **Plant Image:** Users upload an image of a plant leaf for identification.
- **Location Data:** Users can optionally provide their location to enable location-based plant identification and data scraping for medicinal plants specific to their region.
- **Plant Characteristics:** Users can input additional plant details such as size, shape, color, and any distinguishing features of the leaf or plant.
- **User Feedback:** Allow users to provide feedback on the accuracy of the identification, the usefulness of the plant information, and suggestions for improvement.

Features:

- **Plant Identification Tool:** Utilize a deep learning-based model to identify plants based on the uploaded leaf image and associated data (size, shape, etc.).
- **Medicinal Information Module:** Provide detailed textual pharmacognosy information for each identified plant, including medicinal uses, benefits, and potential side effects.
- **Location-Based Plant Recommendations:** Offer plant suggestions based on the user's location, taking into account regional plants known for their medicinal properties.
- **User Feedback System:** Include a feedback system for users to report issues, suggest improvements, and validate plant identification accuracy.
- **Medicinal Plant Encyclopedia:** Build a comprehensive library or encyclopedia of medicinal plants accessible to users for reference.
- **Seamless Navigation and UI:** Design a user-friendly interface with intuitive navigation for easy plant identification and exploration.

7.4: Comparison of results with existing systems

Aspect	Other System	MedLeaf (Our System)
Identification Model	Pre-trained models (e.g., DenseNet, ResNet)	Custom CNN model, trained from scratch
Region Focus	Global, no specific region focus	Tailored for India, includes local medicinal plants
Features	Leaf and flower identification	Leaf scan, textual pharmacognosy details, location-based data scraping
App Performance	Moderate performance	Fast loading time, responsive UI

7.5: Inference drawn

The project aimed to revolutionize medicinal plant identification and its associated benefits by developing a user-friendly mobile app. The primary goal was to provide accurate, real-time identification of medicinal plants and deliver detailed pharmacognosy information to users, empowering them with knowledge about the plants' medicinal properties. By utilizing a custom CNN model trained specifically for local flora, the app ensures high accuracy in identifying plants. For users, it offers easy navigation from plant identification to comprehensive details about medicinal uses, benefits, and applications. The app also integrates location-based data scraping to enhance the relevance of the information provided. Ultimately, MedLeaf enhances the accessibility and awareness of medicinal plants, aiding in traditional medicine practices and empowering users with actionable knowledge for improved health and well-being.

Chapter 8 : Conclusion

8.1 Limitation

1. **Data Availability:** The accuracy and reliability of the MedLeaf app depend significantly on the quality and diversity of the dataset used to train the plant identification model. Limited availability of high-resolution images for certain regional or rare medicinal plants can affect identification accuracy. Additionally, lack of structured pharmacognosy data for lesser-known species can restrict the app's ability to provide comprehensive information.
2. **User Adoption:** The effectiveness of the app is influenced by user engagement and trust. Some users, especially those unfamiliar with AI-based tools or smartphone apps, may hesitate to rely on the system for medicinal plant identification. Furthermore, traditional practitioners might prefer conventional methods over a mobile application.
3. **Technical Constraints:** MedLeaf may face technical limitations related to device compatibility, camera quality (affecting image input), and inconsistent internet access, especially in rural areas. Ensuring smooth real-time processing across different Android devices, while maintaining security and privacy of user-collected data, remains a critical challenge.

8.2 Conclusion

The MedLeaf application serves as an intelligent assistant for users seeking to identify medicinal plants and understand their natural benefits. By simply capturing an image of a plant or leaf, users can receive accurate identification results along with detailed pharmacognosy information. This not only promotes awareness about medicinal flora but also supports the use of traditional plant-based remedies in a modern, accessible format. The app maintains a structured record of user interactions and identified plants, enabling easy reference and continuous learning. Through its Human-Centered AI approach, MedLeaf aims to bridge the gap between technology and traditional knowledge, empowering users with accessible, reliable, and regionally relevant plant information.

8.3 Future Scope

1. **Audio and Multilingual Support:** Future iterations of MedLeaf can incorporate audio playback of pharmacognosy details and support for regional languages, making the app

more accessible to diverse user groups, including those with visual impairments or limited literacy.

2. **Offline and Location-Aware Functionality:** To improve usability in remote or low-connectivity areas, the app can offer offline plant identification and deliver location-specific plant suggestions based on GPS data.
3. **Augmented Reality and Visual Learning:** Integration of Augmented Reality (AR) can provide users with an interactive 3D experience of plant structures, enhancing their understanding of medicinal features in a more engaging way.
4. **Expanded Database and Community Contributions:** The app can scale to include a wider range of medicinal plants across India and introduce features for user-contributed images and feedback, encouraging a community-driven database.
5. **Integration with Traditional Medicine and Research Collaboration:** Future updates may include natural remedy suggestions based on Ayurveda and naturopathy, along with potential collaborations with botanical and research institutions to ensure the reliability of medicinal data.
6. **Continuous Model Enhancement:** By adopting continual learning techniques, the identification model can evolve to recognize new plant species and visual variations without requiring complete retraining.

References

[1] "Plant leaf Disease detection, classification, and diagnosis Using Computer Vision and Artificial intelligence: A review," IEEE Journals & Magazine | IEEE Xplore, 2024. https://ieeexplore.ieee.org/document/10458943

[2] "International trade in non-wood forest products: An overview - XI. Medicinal plants." https://www.fao.org/4/x5326e/x5326e0e.htm

[3] J. Wäldchen and P. Mäder, "Plant Species Identification Using Computer Vision Techniques: A Systematic Literature review," *Archives of Computational Methods in Engineering*, vol. 25, no. 2, pp. 507–543, Jan. 2017, doi: 10.1007/s11831-016-9206-z. https://link.springer.com/article/10.1007/s11831-016-9206-z

[4] P. K. Pang and K. H. Lim, "Review on automatic plant identification using Computer Vision approaches," *IOP Conference Series Materials Science and Engineering*, vol. 495, p. 012032, Jun. 2019, doi: 10.1088/1757-899x/495/1/012032. https://iopscience.iop.org/article/10.1088/1757-899X/495/1/012032/pdf#:~:text=Computer%20vision%20consists%20of%20two,the%20accuracy%20of%20the%20classification

[5] Dey, Biplob et al. "Assessing deep convolutional neural network models and their comparative performance for automated medicinal plant identification from leaf images." Heliyon vol. 10,1 e23655. 13 Dec. 2023, https://doi.org/10.1016/j.heliyon.2023.e23655

[6] G.B. Govindaprabhu, M. Sumathi. "Ethno medicine of Indigenous Communities: Tamil Traditional Medicinal Plants Leaf detection using Deep Learning Models" Procedia Computer Science, Volume 235, 2024, doi:10.1016/j.procs.2024.04.108

[7] Azadnia Rahim et al. "Medicinal and poisonous plants classification from visual characteristics of leaves using computer vision and deep neural networks" Ecological Informatics, Volume 82, 2024, https://doi.org/10.1016/j.ecoinf.2024.102683

[8] Duy Tran Nguyen Nhut et al, "Medicinal plant recognition based on Vision Transformer and BEiT" Procedia Computer Science, Volume 234, 2024, https://doi:0.1016/j.procs.2024.02.165

[9] Khanh Le Dinh Viet, Khiem Le Ha, Trung Nguyen Quoc, and Vinh Truong Hoang. 2024. Medicinal Plants Identification Using Federated Deep Learning. Procedia Comput. Sci. 234, C (2024), 247–254. https://doi.org/10.1016/j.procs.2024.02.171

[10] Shubham Sharma, Manu Vardhan, "MTJNet: Multi-task joint learning network for advancing medicinal plant and leaf classification", Knowledge-Based Systems, Volume 299, 2024, https://doi:10.1016/j.knosys.2024.112147 .

[11] Asim, M. et al. (2024) 'Varietal Discrimination of Guava (Psidium Guajava) Leaves Using Multi Features Analysis', International Journal of Food Properties, 26(1), pp. 179–196. doi: 10.1080/10942912.2022.2158863.

[12] S. Kavitha, T. Satish Kumar, E. Naresh, Vijay H. Kalmani, Kalyan Devappa Bamane, and Piyush Kumar Pareek. 2023. Medicinal Plant Identification in Real-Time Using Deep Learning Model. SN Comput. Sci. 5, 1 (Nov 2023). https://doi.org/10.1007/s42979-023-02398-5

[13] Bammannavar, A.M. , Y, M. and Rashid, M. 2024. Medicinal Herbs Identification Using Deep Learning. QTanalytics Publication (Books). (Jul. 2023), 24–35. DOI:https://doi.org/10.48001/978-81-966500-7-0-3.

[14] Pushpa B R, N. Shobha Rani, "DIMPSAR: Dataset for Indian medicinal plant species analysis and recognition" Data in Brief, Volume 49, 2023, https://doi.org/10.1016/j.dib.2023.109388.

[15] Google. (n.d.). CN110717451A - a deep learning-based image recognition method for leaf diseases of medicinal plants. Google Patents. https://patents.google.com/patent/CN110717451A/en?oq=CN110717451A

[16] Google. (n.d.-b). US8577616B2 - system and method for plant identification. Google Patents. https://patents.google.com/patent/US8577616/en23

[17] C. Garcin, A. Joly, P. Bonnet, Maximilien Servajean, and J. Salmon, "Pl@ntNet-300K image dataset," Zenodo (CERN European Organization for Nuclear Research), Apr. 2021, doi: https://doi.org/10.5281/zenodo.4726653.

Appendix

1.a) Paper 1

Comparative Analysis of Lightweight Deep Learning Models for Medicinal Plant Identification

Dr. Priya R. L.
Department of Computer Engineering, Vivekanand Education Society's Institute Of Technology (Affiliated to the University of Mumbai)
Mumbai, India
ORCID: 0000-0002-7755-578X

Kevin Patel
Department of Computer Engineering, Vivekanand Education Society's Institute Of Technology (Affiliated to the University of Mumbai)
Mumbai, India
ORCID: 0009-0000-4369-9964

Tanvi Naik
Department of Computer Engineering, Vivekanand Education Society's Institute Of Technology (Affiliated to the University of Mumbai)
Mumbai, India
ORCID: 0009-0002-1388-5058

Sanika Hadap
Department of Computer Engineering, Vivekanand Education Society's Institute Of Technology (Affiliated to the University of Mumbai)
Mumbai, India
ORCID: 0009-0003-9860-3060

Acknowledgments

The authors express their gratitude to Vivekanand Education Society's Institute of Technology, Mumbai, India for providing the necessary resources and support for this research.

Funding Information

No funding was received for this manuscript.

Conflict of Interest Statement

The authors declare that there is no conflict of interest regarding the publication of this paper.

Comparative Analysis of Lightweight Deep Learning Models for Medicinal Plant Identification

Abstract

The identification of medicinal plants is crucial for traditional medicine, biodiversity conservation, and agricultural development. With the advent of deep learning, lightweight neural networks have emerged as efficient solutions for real-time plant recognition on resource-constrained devices. This paper presents a comparative study of ResNet18, MobileNetV2, MobileNetV3_Small, MobileNetV3_Large, and EfficientNetB0, evaluating their performance in medicinal plant identification based on leaf images. The models were assessed on training and validation accuracy, loss convergence, computational efficiency, and epoch time stability. Experimental results indicate that MobileNetV3_Large achieved the highest accuracy (97.41%) and lowest training loss, making it the most effective model overall, while EfficientNetB0 exhibited the fastest convergence but required higher computational resources. MobileNetV3_Small balanced efficiency and accuracy, whereas ResNet18 and MobileNetV2 showed relatively lower stability and performance. The findings highlight the trade-offs between accuracy, computational cost, and deployment feasibility, offering insights for optimizing deep learning-based plant identification. This study underscores the potential of lightweight neural networks for real-time applications and suggests future improvements through hyperparameter tuning, dataset expansion, and model optimization to enhance efficiency further.

Keywords: **Medicinal Plant Identification, Lightweight Deep Learning Models, ResNet, MobileNet, EfficientNetB0**

1. Introduction

Medicinal plants have played a crucial role in traditional healing practices and continue to be essential in modern pharmacology. Accurate identification is necessary to maximize their therapeutic potential and prevent risks associated with misidentifying toxic species [1]. Studies estimate that between 35,000 and 70,000 plant species have been used medicinally across various cultures [2]. Notably, over 50,000 plant species are utilized for medicinal purposes worldwide [3], and approximately 40% of pharmaceuticals are derived from natural compounds found in plants [4]. However, conventional identification methods often require extensive botanical expertise, making them labor-intensive and less accessible.

With advancements in machine learning and computer vision, automated plant identification has emerged as a promising solution. Convolutional Neural Networks (CNNs) have demonstrated high accuracy in image classification tasks, making them suitable for medicinal plant recognition. However, real-time or mobile applications demand lightweight models that optimize computational efficiency without significantly compromising accuracy.

This study presents a comparative analysis of lightweight CNN architectures, including MobileNetV2, MobileNetV3, EfficientNet-B0, and ResNet-18. The models are trained on datasets of Indian medicinal plants, with performance evaluated based on training accuracy, validation accuracy, training loss, validation loss and epoch time.

By assessing the trade-offs between computational complexity and classification performance, this research aims to identify the most effective lightweight model for medicinal plant identification. The findings can contribute to advancements in botany, healthcare, and herbal medicine, providing insights for real-world applications in mobile-based plant identification systems.

2. Literature Review

The rapid advancement of deep learning has enabled efficient medicinal plant identification through convolutional neural networks (CNNs) and transformer-based models. However, deploying such models on resource-constrained devices like smartphones requires a balance between accuracy, computational efficiency, and model size. Several studies have explored lightweight architectures, including MobileNet, EfficientNet, and DenseNet, aiming to optimize performance while reducing memory and processing demands. This section reviews existing research on lightweight deep learning models for medicinal plant identification, analyzing their methodologies, performance metrics, and limitations. Furthermore, it highlights key challenges and identifies gaps that necessitate further exploration in the field.

S. Kavitha et al. [5] conducted a study utilizing the MobileNet model to identify medicinal plants by resizing images to 224x224 pixels and employing data augmentation techniques such as rotation, flipping, and zooming. The model achieved an impressive accuracy of 98.3%, enabling a mobile application that allows users to capture leaf images and receive real-time identification. While the application demonstrates the effectiveness of mobile technology for accessible medicinal plant identification, it is limited to just six species from the Kaggle dataset. Expanding the dataset and variety of plants could enhance the app's usability and reach, making it more beneficial for users.

Nguyen Van Hieu and Ngo Le Huy Hien [6] evaluated several deep learning models, including MobileNetV2 and ResNetV2, for classifying Vietnamese plant species. MobileNetV2 achieved the highest accuracy of 83.9%, highlighting its potential for real-time plant identification applications. However, the study also pointed out limitations in generalizability due to dataset constraints, emphasizing the need for broader data collection.

Mulugeta et al. [7] provided a systematic review of deep learning methodologies for medicinal plant classification over the past five years. Analyzing 67 research papers, they found that 83.8% of studies employed transfer learning with pre-trained CNN models such as ResNet, MobileNet, and EfficientNet, while 64.5% used CNNs as primary classifiers. A key observation was the reliance on private datasets, with 67.7% of studies utilizing non-publicly available image collections, thereby limiting reproducibility. The review highlighted emerging trends such as attention mechanisms, transformer-based architectures, and multi-modal learning, suggesting that future research should focus on explainability and real-world implementation of medicinal plant identification models.

Abdollahi [8] proposed a region-specific deep learning-based approach for classifying medicinal plants native to Ardabil, Iran. The study utilized a custom-built CNN model trained on a dataset comprising 50 different medicinal plant species captured in varying lighting conditions and environmental settings. The model, incorporating data augmentation techniques such as rotation, scaling, and brightness adjustments, achieved a classification accuracy of 93.2%, outperforming MobileNetV2 but slightly trailing behind EfficientNet-B0. The study underscores the significance of developing localized deep learning models for medicinal plant identification, particularly for regions with unique biodiversity.

Jamie R. Sykes, Katherine Denby, and Daniel W. Franks [9] introduced PhytNet, a custom CNN architecture designed for plant classification tasks. The research highlights that traditional models like ResNet18 and EfficientNet often underperform on specialized datasets. PhytNet demonstrated superior performance with lower computational costs, making it suitable for rapid plant identification.

Ghosh and Singh [10] proposed a Principal Component Analysis (PCA) based hybrid transfer learning model for classifying medicinal plants through leaf identification. Utilizing the VGG16 pre-trained model combined with PCA for feature extraction, the approach achieved a test accuracy of 95.25% on a dataset comprising 30 different species of medicinal plants.

Malik et al. [11] focused on real-time medicinal plant recognition in the Borneo region, where plant species are often misidentified due to high biodiversity and morphological similarities. Their deep learning pipeline integrated MobileNetV3-Small and EfficientNet-B0 for on-device classification using edge computing. The dataset, consisting of over 12,000 annotated plant images, captured variations in illumination, occlusion, and seasonal changes. Experimental results revealed that EfficientNet-B0 achieved the highest classification accuracy of 95.7%, while MobileNetV3-Small balanced accuracy with computational efficiency, making it suitable for mobile applications.

Musyaffa et al. [12] developed IndoHerb, a large-scale dataset and classification model designed for recognizing Indonesian medicinal plants. The study employed transfer learning techniques on CNN architectures such as ResNet-18, DenseNet, and MobileNetV3-Large, leveraging pre-trained ImageNet weights for improved feature extraction. The manually annotated dataset of over 20,000 images ensured high-quality training data. Results indicated that ConvNeXt and EfficientNet-B0 outperformed other models, achieving 92.5% and 91.3% accuracy, respectively. The study emphasized the role of transfer learning in improving medicinal plant identification, particularly in low-resource settings.

Siyu Quan et al. [13] introduced MS-Net, an enhanced MobileNetV3 incorporating skip connections optimized by an improved whale optimization algorithm. The model achieved a 99.8% accuracy on the PlantVillage dataset and 97.8% on an apple leaf disease dataset with complex backgrounds, demonstrating high performance with fewer parameters.

Ghosh et al. [14] introduced PSR-LeafNet, a neural network combining P-Net, S-Net, and R-Net for extracting leaf features. Utilizing the minimum redundancy maximum relevance approach, the model achieved accuracies of 97.12% on the MalayaKew dataset, 98.10% on the IMP dataset, and 95.88% on the Flavia dataset. This framework demonstrates the effectiveness of integrating machine learning techniques with deep neural networks in medicinal plant identification.

Yao et al. [15] presented a novel multi-prediction deep learning framework for simultaneous plant identification and disease classification using leaf images. They categorized existing approaches into multi-label, multi-task, and multi-output learning, assessing their effectiveness on benchmark datasets such as LeafSnap and PlantVillage. The proposed Generalized Stacking Multi-output CNN (GSMo-CNN) integrated ResNet-18 and MobileNetV3-Small as feature extractors, achieving a significant improvement in classification accuracy compared to single-task CNN models. The study demonstrated that multi-output models can improve generalization by learning shared representations across tasks.

Beikmohammadi et al. [16] proposed SWP-LeafNET, a multistage approach modeling botanists' behavior in leaf identification through three deep learning-based models. Evaluated on the Flavia and MalayaKew datasets, the method achieved accuracies of 99.67% and 99.81%, respectively. The study emphasizes the importance of interpretable and reliable systems in automated plant identification.

Pushpa et al. [17] proposed HybNet, a hybrid deep learning approach for medicinal plant species identification, addressing challenges posed by environmental variations and complex leaf structures. The study introduced three hybrid models integrating convolutional neural networks (CNNs) to enhance feature extraction and classification accuracy. The first model combined VGG16 and MobileNet, achieving 85.85% accuracy using a KNN classifier, while the second model integrated MobileNet with ResNet50, improving accuracy to 88%. The third model, leveraging MobileNetV2 with Squeeze and Excitation (SE) layers, demonstrated superior performance with 94.24% accuracy. Their experiments, conducted on a self-created real-time dataset, highlighted the efficacy of feature recalibration in improving model performance.

Tan et al. [18] introduced a deep learning-based approach for the rapid identification of Chinese medicinal plants (CMPs) using high-resolution visual feature extraction. The authors developed a hybrid supervised pre-training network incorporating Masked Autoencoders (MAE) with a parallel classification branch to enhance feature representation. A self-created dataset of multiple CMP varieties was used for training, with novel preprocessing techniques such as random cropping and shadow augmentation to improve model robustness. The model integrated Vision Transformers (ViT) with MBConv layers to capture both local and global features, achieving an impressive 98.73% accuracy, surpassing conventional CNN and Transformer-based models.

3. Methodology

3.1 Overview of the Approach

This study aims to evaluate and compare the performance of five lightweight deep learning models for medicinal plant identification: **ResNet-18, MobileNetV2, MobileNetV3-Small, MobileNetV3-Large, and EfficientNet-B0**. The methodology involves image acquisition through dataset collection, preprocessing, model selection, classification, prediction and comparative analysis of results. Each model is assessed based on classification accuracy, computational efficiency, and suitability for real-world applications such as mobile and embedded systems.

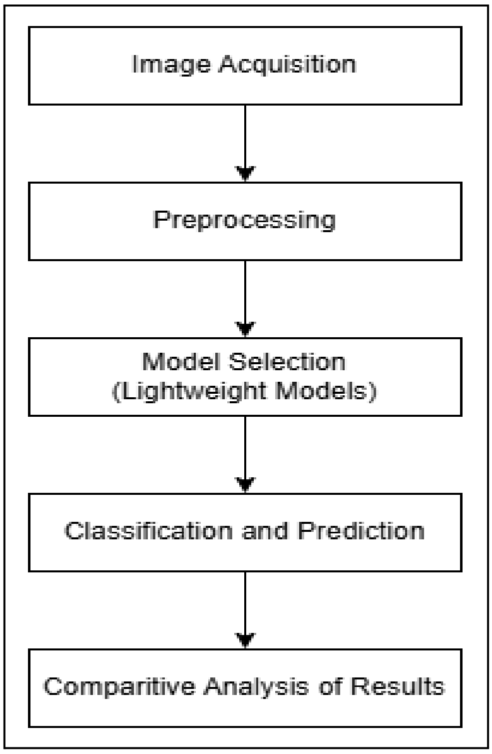


Fig. 01: Block Diagram of the Proposed Approach

As shown properly the proposed approach follows a structured pipeline for medicinal plant identification. It begins with **Image Acquisition**, where plant images are collected for analysis. These images undergo **Preprocessing**, including noise reduction, resizing, and normalization to enhance quality. Next, **Feature Extraction** is performed to derive relevant characteristics from the images, aiding in model training. The extracted features are then utilized for **Model Selection**, focusing on lightweight models to ensure efficiency. The chosen model is used for **Classification and Prediction**, identifying plant species based on learned patterns. Finally, the results are assessed through **Comparative Analysis of Results**, ensuring the model's effectiveness in real-world applications.

3.2 Dataset Description

Indian Medicinal plant datasets [19] is a repository that consists of medicinal plants images. The images are captured with varying backgrounds without any environment constraints. The details of the dataset are as follows:

- **Dataset: Medicinal Leaf dataset**
  - Min Size: (450, 600)
  - Max Size: (3480, 4640)
  - Image Count: 6900
  - Image Count Range per class: (8, 177)

- **Dataset: Medicinal plant dataset**
  - Min Size: (600, 270)
  - Max Size: (548, 600)
  - Image Count: 5945
  - Image Count Range per class: (115, 187)

- **Medicinal Leaf Dataset Plant Names:**
  Aloevera, Amla, Amruthaballi, Arali, ashoka, Astma_weed, Badipala, Balloon_Vine, Bamboo, Beans, Betel, Bhrami, Bringaraja, camphor, Caricature, Castor, Catharanthus, Chakte, Chilly, Citron lime (herelikai), Coffee, Common rue(naagdalli), Coriender, Curry, Doddpathre, Drumstick, Ekka, Eucalyptus, Ganigale, Ganike, Gasagase, Ginger, Globe Amarnath, Guava, Henna, Hibiscus, Honge, Insulin, Jackfruit, Jasmine, kamakasturi, Kambajala, Kasambruga, kepala, Kohlrabi, Lantana, Lemon, Lemongrass, Malabar_Nut, Malabar_Spinach, Mango, Marigold, Mint, Neem, Nelavembu, Nerale, Nooni, Onion, Padri, Palak(Spinach), Papaya, Parijatha, Pea, Pepper, Pomoegranate, Pumpkin, Raddish,

Rose, Sampige, Sapota, Seethaashoka, Seethapala, Spinach1, Tamarind, Taro, Tecoma, Thumbe, Tomato, Tulsi, Turmeric

- **Medicinal Plant Dataset Plant Names:** Aloevera, Amla, Amruta_Balli, Arali, Ashoka, Ashwagandha, Avacado, Bamboo, Basale, Betel, Betel_Nut, Brahmi, Castor, Curry_Leaf, Doddapatre, Ekka, Ganike, Gauva, Geranium, Henna, Hibiscus, Honge, Insulin, Jasmine, Lemon, Lemon_grass, Mango, Mint, Nagadali, Neem, Nithyapushpa, Nooni, Pappaya, Pepper, Pomegranate, Raktachandini, Rose, Sapota, Tulasi, Wood_sorel

3.3 Data Preprocessing

The dataset consists of medicinal plant leaf images collected from publicly available sources and manually curated repositories. The dataset is preprocessed to ensure consistency and enhance model performance. The preprocessing steps include:

- **Resizing**: All images are resized to 512 x 512 pixels to match the input requirements of the selected models.
- **Augmentation:** A variety of random transformations, such as resizing, flipping, rotation, brightness adjustment, and contrast enhancement, are applied to the images. These augmentations help improve the model's generalizability by introducing variations in the training data, preventing overfitting and enabling the model to better recognize patterns in different conditions.
- **Data Splitting**: The dataset is divided into **72% training, 8% validation, and 20% testing** to ensure unbiased model evaluation.
    - We apply normalization using ImageNet pre-trained model mean and standard deviation values:
        - Mean: [0.485, 0.456, 0.406]
        - Standard Deviation: [0.229, 0.224, 0.225]

3.4 Selected Models

Five lightweight deep learning models are selected for comparative analysis. These models are chosen due to their efficiency, reduced computational requirements, and effectiveness in image classification tasks.

3.4.1 ResNet-18 [20]

ResNet-18 is a compact model from the ResNet family, designed to address the vanishing gradient issue using residual connections. With 18 layers, it remains computationally efficient while preserving strong feature extraction. Its ability to learn deep hierarchical representations makes it a popular choice for image classification tasks.

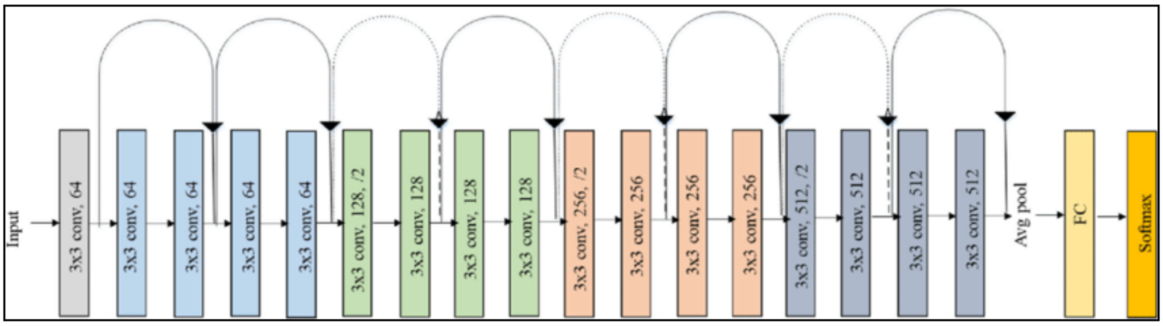


Fig. 02: ResNet-18 Architecture[21]

The figure 2 above depicts the architecture of ResNet-18, highlighting its residual connections and 18-layer structure.

3.4.2 MobileNetV2 [22]

MobileNetV2 is a lightweight deep learning model designed for mobile and embedded systems. It utilizes depthwise separable convolutions to minimize parameters and computational load. Furthermore, its inverted residual blocks enhance accuracy while boosting inference speed, making it ideal for real-time plant identification.

3.4.3 MobileNetV3-Small [23]

MobileNetV3-Small is a compact version of MobileNetV3, specifically designed for ultra-low-power devices. It integrates squeeze-and-excitation (SE) layers with hard-swish activation, improving performance while minimizing computational complexity. This model is ideal for applications requiring real-time inference on resource-limited hardware.

3.4.4 MobileNetV3-Large [23]

MobileNetV3-Large is a more powerful version of MobileNetV3, designed for higher accuracy while still maintaining efficiency. It combines depthwise convolutions, SE layers, and lightweight attention mechanisms to enhance feature extraction. This model balances accuracy and computational efficiency, making it suitable for real-time medicinal plant classification.

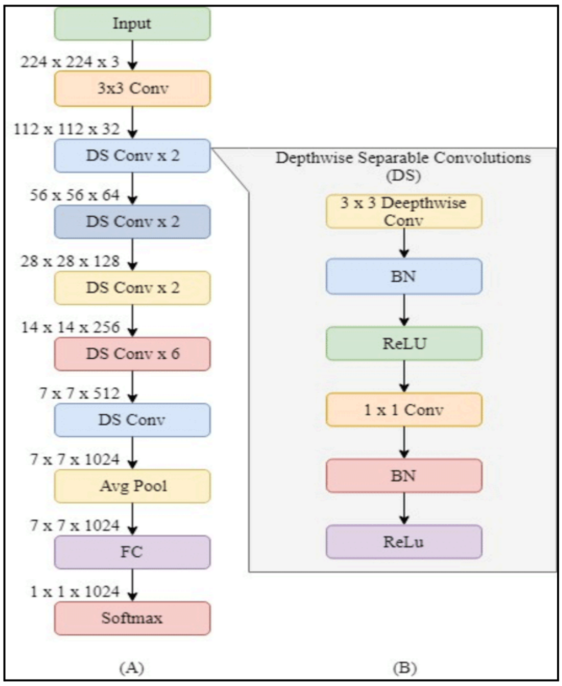


Fig. 03: MobileNet Architecture[24]

Figure 3 above presents the MobileNet architecture: (A) the overall framework and (B) a detailed explanation of the DS layer.

3.4.5 EfficientNet-B0 [25]

EfficientNet-B0 is a highly optimized deep learning model that uses compound scaling which adjusts depth, width, and resolution to maintain balance, achieving high accuracy with fewer parameters than conventional CNNs. EfficientNet-B0 is designed to maximize accuracy while minimizing computational costs, making it a strong candidate for lightweight medicinal plant identification.

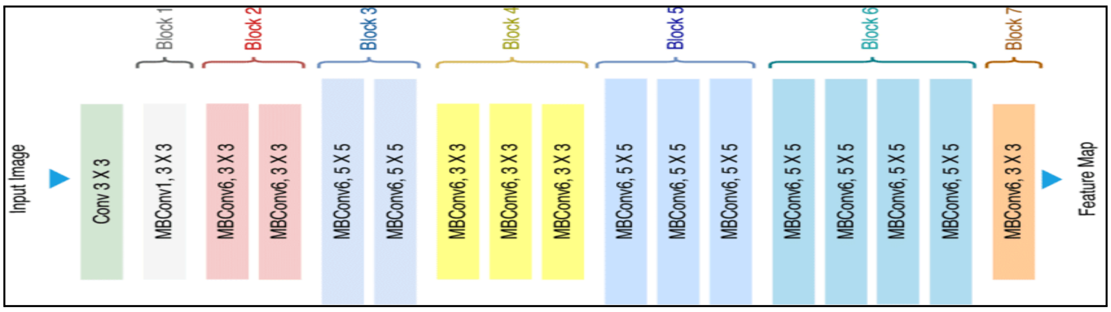


Fig. 04: EfficientNet-B0 Architecture[26]

Figure 4 above illustrates the architecture of EfficientNet-B0, with MBConv as its fundamental building block.

3.5 Experimental Setup

The models are implemented using **TensorFlow and PyTorch** and trained on an **NVIDIA GPU** to accelerate computations. The training setup includes:

- **Batch Size:** A batch size of 32 is used to ensure efficient training while maintaining good generalization performance.
- **Optimizer:** Adam optimizer is used with a learning rate of 3e-4. Adam is known for its adaptive learning rates, which help achieve faster convergence.
- **Epochs:** The models are trained for up to 100 epochs, incorporating early stopping. to avoid overfitting. If the validation loss remains unchanged for five successive epochs, training stops early.
- **Loss Function:** The models are trained using Cross-Entropy Loss, which is suitable for multi-class classification tasks.

3.6 Evaluation Metrics

To ensure a comprehensive comparison, the models are evaluated using the following metrics:

- **Accuracy**: Measures the overall correctness of model predictions.
- **Precision**: Evaluates the proportion of correctly classified plant species.
- **Recall**: Assesses the model's ability to correctly identify all instances of each species.
- **F1-Score**: Provides the balanced average of precision and recall.
- **Inference Time**: Measures time required for a single image prediction, crucial for real-time applications.

4. Comparative Analysis

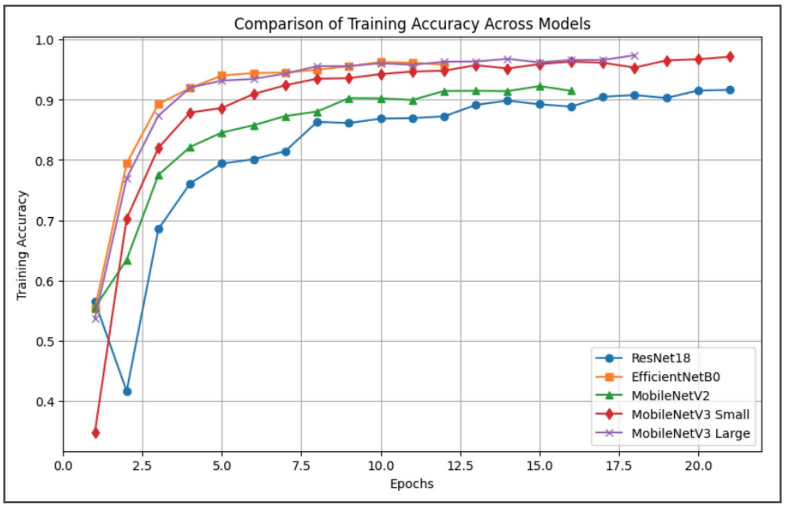


Fig 05: Training Accuracy Trends Across Models

Figure 5 above is a line chart showing training accuracy over epochs for ResNet18, EfficientNetB0, MobileNetV2, MobileNetV3_Small, and MobileNetV3_Large, highlighting their convergence speed and final accuracy.

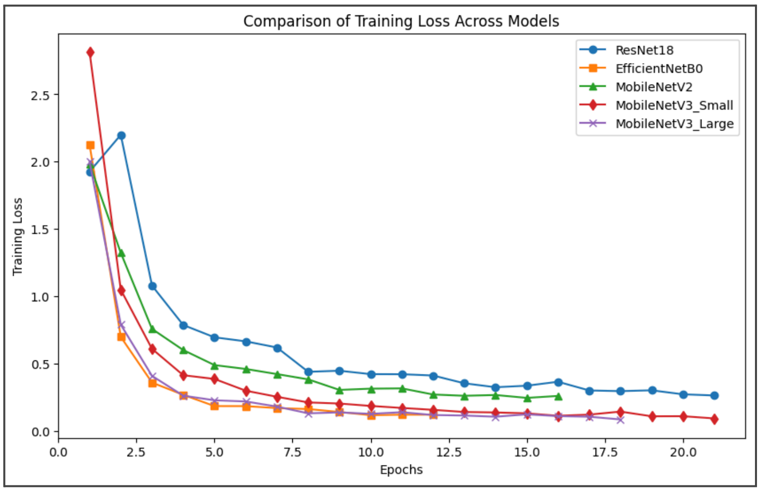


Fig. 06: Training Loss Trends Across Models

Figure 6 above depicts a line chart showing the decline in training loss over epochs, reflecting how effectively each model minimizes error during training.

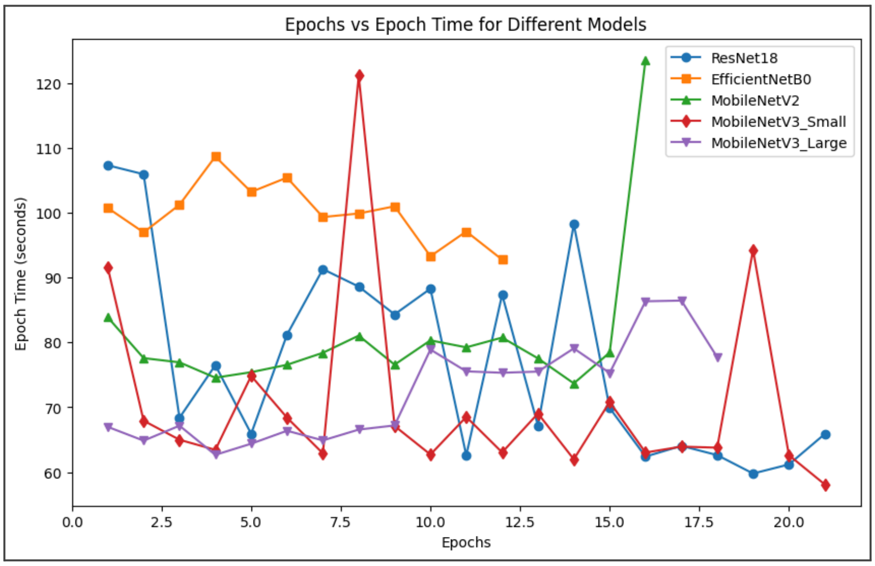


Fig. 07: Epoch Time Analysis of Models

Figure 7 above is a line chart depicting epoch time variations across the five models, showcasing differences in computational efficiency and processing speed.

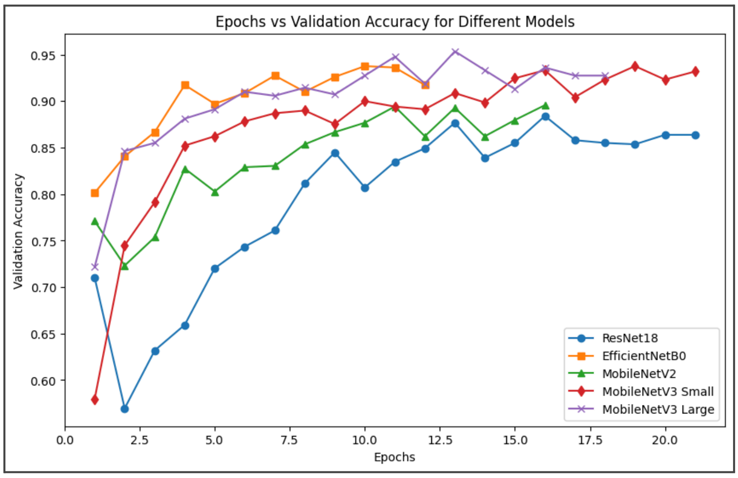


Fig. 08: Validation Accuracy Trends Across Models

Figure 8 above depicts a line chart illustrating the progression of validation accuracy over epochs, comparing the generalization performance of each model.

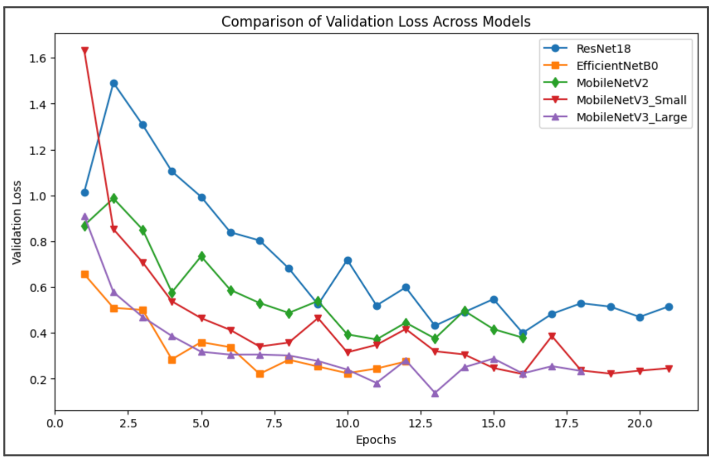


Fig. 09: Validation Loss Trends Across Models

Figure 9 above is a line chart representing the validation loss reduction over epochs, indicating convergence stability and generalization capability.

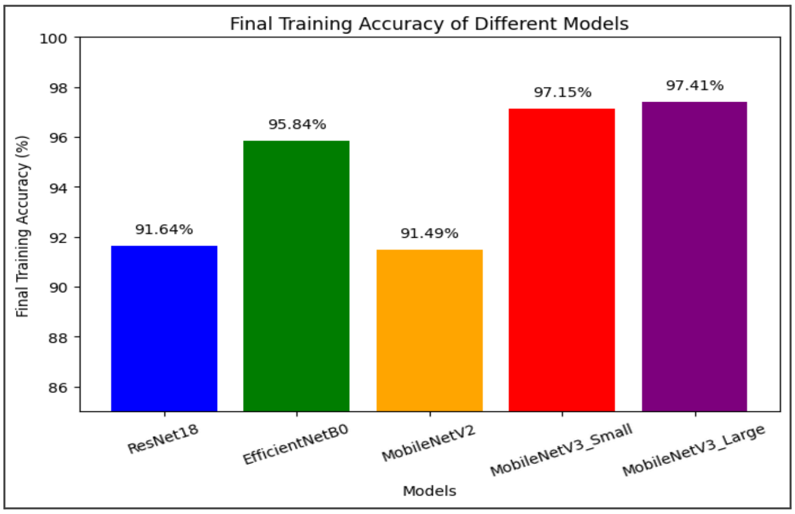


Fig. 10: Final Training Accuracy of Models

Figure 10 above depicts a bar graph comparing the final training accuracy of the five models, highlighting the superior performance of MobileNetV3 variants.

5. Discussion

The comparative analysis of **ResNet18, EfficientNetB0, MobileNetV2, MobileNetV3_Small, and MobileNetV3_Large** highlights distinct performance patterns across multiple dimensions, including **accuracy, loss trends, training efficiency, and computational stability.**

5.1 Model Performance Trends

EfficientNetB0 achieves high accuracy early in training, reaching over **90% accuracy by epoch 5** and stabilizing at **~96% by epoch 12**, making it ideal for applications where rapid convergence is critical. MobileNetV3 (both Small and Large) steadily reach **~97% accuracy**, demonstrating their suitability for mobile applications with a balance between efficiency and precision. In contrast, **ResNet18 exhibits initial instability**, requiring **21 epochs** to stabilize at **91.6%,** which suggests the need for careful hyperparameter tuning. MobileNetV2, while demonstrating **consistent learning, plateaus earlier** than the other models, indicating limited capacity for further improvement.

5.2 Loss Convergence and Stability

Examining the **training and validation loss trends,** EfficientNetB0 and MobileNetV3_Large demonstrate the **fastest and most stable convergence**, with MobileNetV3_Large achieving the **lowest final loss (0.0876).** ResNet18, despite an initial **high loss of 1.93**, steadily decreases to **0.26 by epoch 21**, highlighting a more gradual but effective learning process. MobileNetV2 shows **consistent loss reduction**, though it stabilizes earlier, reinforcing its limited adaptability.

5.3 Computational Efficiency and Training Time

Efficiency variations become evident through **epoch time analysis**. ResNet18 displays **high variance (59.78s–107.35s)**, making it less predictable. EfficientNetB0, while achieving fast convergence, **demands higher computational resources (peaking at 108.70s** per epoch). MobileNetV3_Large is the most efficient model, maintaining stable processing times (62.66s–86.47s), making it the optimal choice for balancing **computational cost and accuracy.**

5.4 Overall Insights

MobileNetV3_Large emerges as the **top-performing model**, offering the highest accuracy (97.41%) while maintaining **computational efficiency and stable training times.** EfficientNetB0 excels in **fast training,** making it suitable for real-time applications where quick convergence is required. MobileNetV3_Small provides **comparable accuracy with lower computational overhead**, making it another strong choice for mobile implementations. ResNet18 and MobileNetV2, while **competent, require trade-offs**—ResNet18 for its stability concerns and MobileNetV2 for its **limited improvement capacity.**

Conclusion

This study presents a comparative analysis of lightweight deep learning models for medicinal plant identification, evaluating ResNet18, MobileNetV2, MobileNetV3_Small, MobileNetV3_Large, and EfficientNetB0 based on their accuracy, loss convergence, training efficiency, and computational requirements. The results indicate that MobileNetV3_Large achieves the highest accuracy (97.41%) with the lowest training loss, making it the most suitable for real-time applications. EfficientNetB0 demonstrates rapid convergence but requires higher computational resources, whereas MobileNetV3_Small balances efficiency and performance effectively. ResNet18 and MobileNetV2 exhibit stable learning trends but show slightly lower accuracy and require careful tuning for optimal performance.

Despite their effectiveness, lightweight models still face challenges in handling diverse environmental conditions, varying lighting, and complex leaf structures. Future work could focus on improving model generalization through larger and more diverse datasets, optimizing architectures for better trade-offs between accuracy and efficiency, and integrating self-supervised learning techniques to enhance feature extraction. Additionally, incorporating explainable AI methods could improve model interpretability, making them more reliable for practical use in plant identification.

References

[1] "Plant leaf Disease detection, classification, and diagnosis Using Computer Vision and Artificial intelligence: A review," IEEE Journals & Magazine |IEEE Xplore, 2024. https://ieeexplore.ieee.org/document/10458943

[2] "International trade in non-wood forest products: An overview - XI. Medicinal plants." https://www.fao.org/4/x5326e/x5326e0e.html

[3] U. Schippmann, D. J. Leaman, and A. B. Cunningham, "Impact of cultivation and gathering of medicinal plants on biodiversity: Global trends and issues," in Biodiversity and the Ecosystem Approach in Agriculture, Forestry and Fisheries, FAO, 2002. [Online]. https://www.fao.org/4/AA010E/AA010E02.htm

[4] U.S. Forest Service, "Medicinal botany," U.S. Department of Agriculture. [Online]. https://www.fs.usda.gov/wildflowers/ethnobotany/medicinal/index.shtml

[5] S. Kavitha, T. Satish Kumar, E. Naresh, Vijay H. Kalmani, Kalyan Devappa Bamane, and Piyush Kumar Pareek. 2023. Medicinal Plant Identification in Real-Time Using Deep Learning Model. SN Comput. Sci. 5, 1 (Nov 2023). https://doi.org/10.1007/s42979-023-02398-5

[6] N. V. Hieu and N. L. H. Hien, "Automatic Plant Image Identification of Vietnamese species using Deep Learning Models," International Journal of Engineering Trends and Technology, vol. 68, no. 4, pp. 25–31, Apr. 2020, doi: https://doi.org/10.14445/22315381/ijett-v68i4p205s.

[7] Mulugeta, A. K., Sharma, D. P., & Mesfin, A. H. (2024). Deep learning for medicinal plant species classification and recognition: a systematic review. Frontiers in Plant Science, 14, 1286088. https://www.frontiersin.org/journals/plant-science/articles/10.3389/fpls.2023.1286088/full

[8] Abdollahi, J. (2022). Identification of medicinal plants in Ardabil using deep learning. In 2022 27th International Computer Conference, Computer Society of Iran (CSICC) (pp. 1-6). IEEE. https://ieeexplore.ieee.org/document/9780493

[9] J. R. Sykes, K. J. Denby, and D. W. Franks, "Tailoring convolutional neural networks for custom botanical data," Applications in Plant Sciences, vol. 13, no. 1, Oct. 2024, doi: https://doi.org/10.1002/aps3.11620.

[10] Ghosh, S., & Singh, A. (2022). Identification of medicinal plant using hybrid transfer learning technique. Indonesian Journal of Electrical Engineering and Computer Science, 31(3), 1605-1615.https://ijeecs.iaescore.com/index.php/IJEECS/article/viewFile/31573/17570

[11] Malik, O. A., Ismail, N., Hussein, B. R., & Yahya, U. (2022). Automated real-time identification of medicinal plant species in natural environment using deep learning models—A case study from Borneo region. Plants, 11(15), 1952. https://pmc.ncbi.nlm.nih.gov/articles/PMC9370651/

[12] Musyaffa, M. S. I., Yudistira, N., Rahman, M. A., & Batoro, J. (2023). IndoHerb: Indonesia medicinal plants recognition using transfer learning and deep learning. arXiv preprint arXiv:2308.01604. https://pmc.ncbi.nlm.nih.gov/articles/PMC11629298/

[13] S. Quan, J. Wang, Z. Jia, M. Yang, and Q. Xu, "MS-Net: a novel lightweight and precise model for plant disease identification," Frontiers in Plant Science, vol. 14, Oct. 2023, doi: https://doi.org/10.3389/fpls.2023.1276728.

[14] Ghosh, M., Mondal, S., & Ghosh, S. (2024). PSR-LeafNet: A deep learning framework for identifying medicinal plants using leaf images. Big Data and Cognitive Computing, 8(12), 176. https://www.mdpi.com/2504-2289/8/12/176

[15] Yao, J., Tran, S. N., Garg, S., & Sawyer, S. (2023). Deep learning for plant identification and disease classification from leaf images: Multi-prediction approaches. arXiv preprint arXiv:2310.16273. https://arxiv.org/abs/2310.16273

[16] Beikmohammadi, A., Faez, K., & Motallebi, A. (2020). SWP-LeafNET: A novel multistage approach for plant leaf identification based on deep CNN. arXiv preprint arXiv:2009.05139. https://arxiv.org/abs/2009.05139

[17] B. R. Pushpa, S. Jyothsna, and S. Lasya, “HybNet: A hybrid deep model for medicinal plant species identification,” MethodsX, vol. 14, p. 103126, 2025. https://www.sciencedirect.com/science/article/pii/S2215016124005776

[18] Tan, C., Tian, L., Wu, C. et al. Rapid identification of medicinal plants via visual feature-based deep learning. Plant Methods 20, 81 (2024). https://doi.org/10.1186/s13007-024-01202-6

[19] R, P. B. (2023, May 5). Indian medicinal leaves image datasets. Mendeley Data. https://data.mendeley.com/datasets/748f8jkphb/3

[20] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770-778. https://arxiv.org/abs/1512.03385

[21] Ramzan, Farheen & Khan, Muhammad Usman & Rehmat, Asim & Iqbal, Sajid & Saba, Tanzila & Rehman, Amjad & Mehmood, Zahid. (2019). A Deep Learning Approach for Automated Diagnosis and Multi-Class Classification of Alzheimer’s Disease Stages Using Resting-State fMRI and Residual Neural Networks. Journal of Medical Systems. 44. 10.1007/s10916-019-1475-2. https://www.researchgate.net/publication/336642248

[22] Sandler, M., Howard, A. G., Zhu, M., Zhmoginov, A., & Chen, L. (2018). Mobilenetv2: Inverted Residuals and Linear Bottlenecks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4510-4520. https://arxiv.org/abs/1801.04381

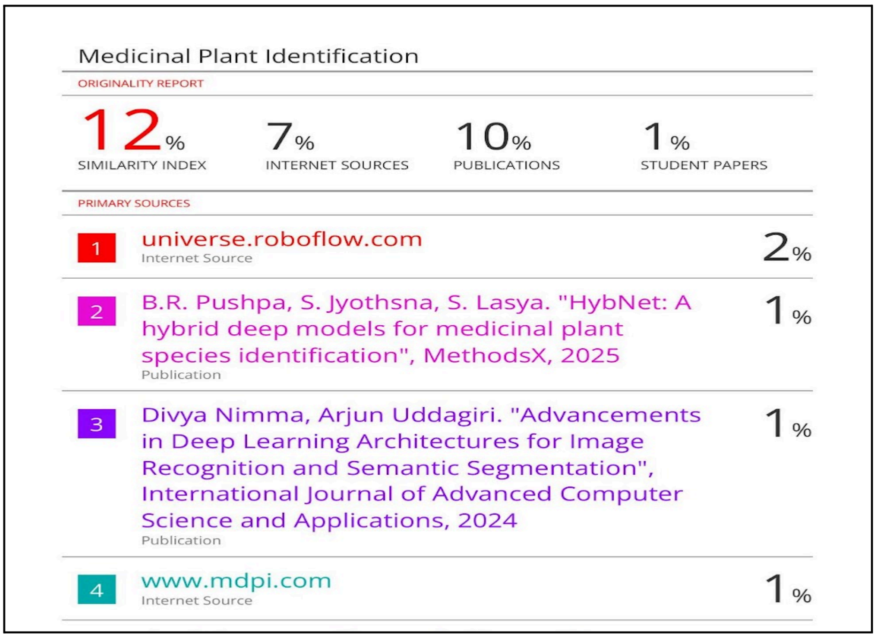
[23] Howard, A. G., Sandler, M., Chu, G., Chen, L., & Chen, Y. (2019). Searching for MobileNetV3. Proceedings of the IEEE International Conference on Computer Vision (ICCV), 1314-1324. https://arxiv.org/abs/1905.02244

[24] Phiphitphatphaisit, Sirawan & Surinta, Olarik. (2020). Food Image Classification with Improved MobileNet Architecture and Data Augmentation. https://www.researchgate.net/publication/341129298

[25] Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Proceedings of the 36th International Conference on Machine Learning (ICML), 6105-6114. https://arxiv.org/abs/1905.11946

[26] Ahmed, Tashin & Sabab, Noor. (2022). Classification and Understanding of Cloud Structures via Satellite Images with EfficientUNet. SN Computer Science. 3. 10.1007/s42979-021-00981-2. https://www.researchgate.net/publication/356981443

1.b) Plagiarism report of Paper 1



Medicinal Plant Identification

ORIGINALITY REPORT

12%	7%	10%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	universe.roboflow.com Internet Source	2%
2	B.R. Pushpa, S. Jyothsna, S. Lasya. "HybNet: A hybrid deep models for medicinal plant species identification", MethodsX, 2025 Publication	1%
3	Divya Nimma, Arjun Uddagiri. "Advancements in Deep Learning Architectures for Image Recognition and Semantic Segmentation", International Journal of Advanced Computer Science and Applications, 2024 Publication	1%
4	www.mdpi.com Internet Source	1%

1.c) Project Review Sheets

Project Review Sheet 1:

Inhouse/ Industry_Innovation/Research: Inhouse

Sustainable Goal: 03 : Good Health & Well Being

Project Evaluation Sheet 2024 - 25

Class: D17 ~~A/B~~/C

Group No.: 30

Title of Project: Medleaf : AI-based Identification & Medicinal Value Assessment of Flora

Group Members: Kevin Patel (D17A 45), Sanika Hadap (D17C 22), Tanvi Naik (D17C 48)

Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (2)	Applied Engg&Mgmt principles (3)	Life-long learning (3)	Professional Skills (3)	Innovative Approach (3)	Research Paper (5)	Total Marks (50)
5	5	4	2	4	2	2	2	2	2	3	3	3	3	5	47

Comments: Productize the project & Revenue Model

Lifna CS 1/3/25

Name & Signature Reviewer1

Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (2)	Applied Engg&Mgmt principles (3)	Life-long learning (3)	Professional Skills (3)	Innovative Approach (3)	Research Paper (5)	Total Marks (50)
5	4	4	2	4	2	2	2	1	2	3	3	3	3	5	46

Comments: → show more locations (geo) for plant locate at multiple location. → show clear Revenue model
→ Improvize GUI.

Date: 1st March, 2025

Jugchhaya

Name & Signature Reviewer 2

Project Review Sheet 2:

30

Project Evaluation Sheet 2024 - 25

Title of Project: Medleaf : AI Based Identification & Medicinal value assessment of flora

Group Members: Kevin Patel (D17A-45), Sanika Hadap (D17C-22), Tanvi Naik (D17C-48)

Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (2)	Applied Engg&Mgmt principles (3)	Life-long learning (3)	Professional Skills (3)	Innovative Approach (3)	Research Paper (5)	Total Marks (50)
4	4	4	3	4	2	2	2	2	2	3	3	3	3	4	45

Comments:

R.L.

Name & Signature Reviewer1

Inhouse/ Industry_Innovation/Research:

Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (2)	Applied Engg&Mgmt principles (3)	Life-long learning (3)	Professional Skills (3)	Innovative Approach (3)	Research Paper (5)	Total Marks (50)
4	4	4	3	4	2	2	2	2	2	3	3	3	3	4	45

Comments: ① Complete integration
② Prepare a walkthrough video

Date: 1st April,2025

Lifna CS 1/4/25

Name & Signature Reviewer 2