**VIVEKANAND EDUCATION SOCIETY'S
INSTITUTE OF TECHNOLOGY**

Department of Computer Engineering

Project Report on

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

In partial fulfillment of the Fourth Year (Semester–VII), Bachelor of Engineering (B.E.) Degree in Computer Engineering at the University of Mumbai

Academic Year 2024-2025

Name of the Mentor

Prof. Priya R.L

Submitted by

Sanika Hadap D17C 22

Tanvi Naik D17C 48

Kevin Patel D17A 45

VIVEKANAND EDUCATION SOCIETY'S
INSTITUTE OF TECHNOLOGY

Department of Computer Engineering

CERTIFICATE of Approval

This is to certify that____ of Fourth Year Computer Engineering studying under the University of Mumbai has satisfactorily presented the project on "***MedLeaf: AI-based Identification and Medicinal Value Assessment of Flora***" as a part of the coursework of PROJECT-I for Semester-VII under the guidance of ***Mrs. Priya R L*** in the year 2024-2025.

____

Date

__________________ Internal Examiner

__________________ External Examiner

__________________ Project Mentor

__________________ Head of the Department
Dr. Mrs. Nupur Giri

__________________ Principal
Dr. J. M. Nair

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** (Project Guide) for her kind help and valuable advice during the development of project synopsis and for her guidance and suggestions.

We are deeply indebted to 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 able to:-

Course Outcome	Description of the Course Outcome
CO 1	Do literature survey/industrial visit and identify the problem of the selected project topic.
CO2	Apply basic engineering fundamental in the domain of practical applications FORproblem identification, formulation and solution
CO 3	Attempt & Design a problem solution in a right approach to complex problems
CO 4	Cultivate the habit of working in a team
CO 5	Correlate the theoretical and experimental/simulations results and draw the proper inferences
CO 6	Demonstrate the knowledge, skills and attitudes of a professional engineer & Prepare report as per the standard guidelines.

ABSTRACT of the project

The identification of medicinal plants and the analysis of their benefits have significant implications for traditional medicine, agricultural development, and pharmaceutical research. This project leverages the power of artificial intelligence and computer vision to create a comprehensive system for identifying medicinal plants and analyzing their medicinal properties. By utilizing convolutional neural networks (CNNs), we aim to accurately classify various medicinal plants based on images. The system will be trained using publicly available datasets, data scraped from reliable websites, and data collected from trusted sources in the field. Additionally, the medicinal benefits will be curated from official websites, authoritative books, and verified practitioners of plant-based medicine. The project is designed to assist Ayurvedic practitioners, medicinal plant harvesters, and students by providing a reliable tool for plant identification and finding its benefits. One of the primary challenges addressed in this project is the management and accuracy of a large training dataset to ensure high precision in plant identification and finding its benefits.

INDEX

1.Introduction

1.1 Introduction to the project

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 models 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 Drawback of the 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

1.4 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.

1.5 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.

1.6 Methodology used

1. Data Collection and Preparation

1.1 Data Organization

The project begins with the collection of images of various medicinal plants, which are organized into multiple folders. Each folder corresponds to a specific plant species, containing multiple images of that plant.

1.2 Image Renaming

To standardize the dataset and facilitate processing, all images are renamed using a numerical format (e.g., 1.jpg, 2.jpg, etc.). This renaming ensures that file management is efficient and consistent.

1.3 Image Resizing

All images are resized to a uniform dimension of 512 x 512 pixels. This step is crucial as it ensures that the input size for the machine learning models remains consistent, allowing for efficient training and inference.

2. Data Splitting

2.1 Train-Test Split

The dataset is divided into training and testing subsets. The training data is used to train the model, while the testing data is reserved for evaluating the model's performance. A typical split ratio of 80% for training and 20% for testing is employed to ensure adequate training data while maintaining a sufficient test set for evaluation.

3. Data Augmentation and Preprocessing

3.1 Data Augmentation

To enhance the robustness of the model and prevent overfitting, various data augmentation techniques are applied to the training dataset. These include:

- Random Horizontal Flip: Randomly flipping images horizontally to create variability in the training data.

3.2 Data Normalization and Conversion

After augmentation, the images are transformed into tensor format using ToTensor() and normalized to ensure that the pixel values are within a standardized range. Normalization helps the model converge faster during training.

4. Model Selection and Training

4.1 Pre-trained Model Selection

A series of pre-trained convolutional neural networks (CNNs) were utilized. The models selected for this study included:

ResNet18
MobileNetV2
MobileNetV3
EfficientNet_B0

4.2 Model Modification

- For each model, the convolutional layers were frozen to retain the learned features from the pre-trained weights. The final classification layer was modified to match the number of plant species in the dataset.

4.3 Loss Function and Optimizer

Loss Function and Optimizer:

- A suitable loss function (e.g., Cross-Entropy Loss) was selected to quantify the difference between predicted and actual classifications. An optimizer (Adam) was chosen to update model weights efficiently.

Learning Rate Scheduler:

- A learning rate scheduler was implemented to adjust the learning rate during training, promoting better convergence and performance.

4.4 Model Training

The model is trained using the training dataset over several epochs. During this process, the model's performance is monitored through metrics such as accuracy and loss.

4.5 Model Testing

After training, each model was evaluated on the test dataset to assess its classification accuracy and generalization ability.

Steps 4.1-4.5 were repeated for each of the selected models (ResNet18, MobileNetV3, MobileNetV2, EfficientNet_B0) to ensure a comprehensive evaluation of their performance in identifying medicinal plants.

2.Literature Survey

2.1 Research Papers

[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] **Medicinal Plants Identification Using Federated Deep Learning** (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.

3. Requirements for the proposed system

The requirements for the project titled "**MedLeaf: AI-based Identification and Medicinal Value Assessment of Flora**" comprises of :-

3.1 Functional Requirements

- **Medicinal Plant Image Input**: Users can upload or capture plant images.
- **Image Preprocessing**: Resize and enhance images for better model performance.
- **Medicinal Plant Identification**: Identify plants using deep learning models.
- **Medicinal Value Assessment**: Provide medicinal properties and health benefits from a curated database.
- **User Interaction**: Simple interface for uploading images and viewing results.
- **Database Integration**: Access plant information and benefits from the database.
- **Report Generation**: Generate downloadable reports based on identification and assessments.

3.2. Non-Functional Requirements

- **Performance**: Fast and accurate identification with minimal delay.
- **Scalability**: Handle large datasets and users without slowing down.
- **Usability**: Easy-to-use interface for non-technical users.
- **Reliability**: Consistent performance with minimal errors.
- **Security**: Ensure data privacy and protect user information.
- **Portability**: Deployable across desktops, laptops, and mobile devices.
- **Maintainability**: Easy to update and debug with clear documentation.
- **Accuracy**: High accuracy in plant identification.

3.3. Constraints

- **Data Availability**: Performance depends on dataset quality and size.
- **Computational Resources**: Requires significant processing power for training.
- **Model Complexity**: Complex models may take longer to train.
- **Image Quality**: Low-quality images may affect identification accuracy.
- **Environmental Factors**: Variations like lighting or shadows may impact results.
- **Licensing**: Some libraries may have usage restrictions.

3.4. Hardware & Software Requirements

- 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.

3.5. Techniques utilized till date for the proposed system

1. Data Preparation Techniques

- **Image Preprocessing**:
  - Renaming and Resizing: Images were uniformly renamed and resized to 512 x 512 pixels to standardize the input dimensions for deep learning models.
- **Data Augmentation**:
  - Random Horizontal Flip: increase the variability of the training dataset by flipping images horizontally,giving different perspectives of the same plant species, improving its ability to generalize.
- **Normalization**:
  - Image pixel values were normalized to a specific range[0, 1] to enhance model training.

2. Model Training Techniques

Utilization of Pre-trained Models convolutional neural networks (CNNs) like ResNet18, MobileNetV2, MobileNetV3, EfficientNet_B0, and ViT_B_16.

- **Freezing Layers**:
  - The convolutional layers of the pre-trained models were frozen, retaining their weights while only training the final classification layers.
- **Customizing Output Layers**:
  - The output layer of each model was modified to match the number of distinct plant species in the dataset.

3. Training Configuration Techniques

- **Loss Function**:
  - Cross-Entropy Loss: Employed to measure the performance of the classification model, particularly for multi-class classification tasks. It quantifies the difference between the predicted class probabilities and the actual class labels.

- **Optimization Algorithms**:
  - Adam Optimizer: The Adam optimizer was utilized for its efficiency and effectiveness in adjusting learning rates adaptively. It combines the benefits of two other extensions of stochastic gradient descent, allowing for faster convergence.
- **Learning Rate Scheduling**:
  - A learning rate scheduler was implemented to dynamically adjust the learning rate during training. This technique helps to balance convergence speed and stability, ensuring optimal training performance over multiple epochs.

4. Model Evaluation Techniques

- **Train-Test Split**:
  - The dataset was divided into training and testing subsets, typically in an 80:20 ratio. This split allows for evaluating the model's performance on unseen data, providing insights into its generalization capabilities.
- **Model Testing**:
  - After training, each model's accuracy was evaluated on the test dataset. This evaluation metric indicates how well the model can identify plant species it has not been trained on, which is critical for assessing the system's effectiveness in real-world applications.

5. Iterative Refinement Techniques

- **Multiple Model Training**:
  - The training and evaluation processes were iteratively conducted for each selected model (ResNet18, MobileNetV3, MobileNetV2, EfficientNet_B0, ViT_B_16). This approach allowed for a comparative analysis of model performances, enabling the selection of the most effective architecture for the medicinal plant identification task.

3.6. Tools utilized till date for the proposed system

1. Hardware and Computational Tools

- **CUDA**:
  - CUDA (Compute Unified Device Architecture) was used to leverage GPU acceleration for training deep learning models. Utilizing CUDA allowed for faster model training and processing, significantly reducing computational time compared to CPU-based training.

2. Python Libraries for Data Handling and Manipulation

- **os**:
  - The os library was employed to handle file system operations, such as navigating directories, renaming files, and managing datasets.
- **numpy**:
  - numpy was used for numerical computations and matrix operations, which are essential for data manipulation and preprocessing.
- **shutil**:
  - This module was used to automate file operations, such as copying and moving files during dataset preparation.
- **scikit-learn** (sklearn):
  - Various utilities from scikit-learn were utilized for:
    - <u>Train-Test Splitting</u>: train_test_split was used to divide the dataset into training and testing sets.
    - <u>Metrics</u>: Functions like accuracy_score, classification_report, and confusion_matrix were employed to evaluate model performance.

3. Deep Learning Libraries

- **PyTorch** (torch):

- PyTorch served as the primary deep learning framework for building, training, and testing the plant identification models. PyTorch's dynamic computation graph and user-friendly APIs were crucial in developing and fine-tuning models.

- **torch.nn**:
  - The torch.nn module provided classes and functions to build neural network layers and models, including:
    - <u>Loss Functions</u> (e.g., CrossEntropyLoss).
    - <u>Activation Functions</u> (e.g., ReLU).
- **torch.optim**:
  - The torch.optim module was used to configure optimizers such as the Adam optimizer, which updates the model's weights during training.
- **torch.utils.data.DataLoader**:
  - DataLoader was utilized for loading the dataset in batches, ensuring efficient shuffling and batching of the training and test data.
- **torch.optim.lr_scheduler**:
  - The learning rate scheduler was used to dynamically adjust the learning rate during model training, promoting optimal convergence.

4. Image Processing and Data Augmentation

- **Torchvision** (torchvision):
  - torchvision was employed for various tasks, including:
    - <u>Pre-trained Models</u>: Pre-trained deep learning models like ResNet18, MobileNetV2, MobileNetV3, EfficientNet_B0, and Vision Transformer (ViT_B_16) were accessed through torchvision.models.
    - <u>Transforms</u>: The transforms module was used for data augmentation, including operations like RandomHorizontalFlip, resizing, normalizing, and converting images to tensors.

5. Pre-trained Models

- **ResNet18**:
  - A pretrained ResNet18 model was used as a baseline for image classification, leveraging its residual connections for better feature extraction.
- **MobileNetV2 and MobileNetV3**:
  - MobileNetV2 and MobileNetV3 models, designed for efficient mobile and embedded vision applications, were used to explore lightweight model architectures.
- **EfficientNet**:
  - EfficientNet, known for its balance of accuracy and efficiency, was employed using the efficientnet-pytorch library.
- **Vision Transformer (ViT)**:
  - ViT_B_16 from PyTorch's torchvision library was used to explore transformer-based architectures for image classification.

6. Visualization Tools

- **Matplotlib**:
  - matplotlib.pyplot was used to visualize training and validation metrics (e.g., loss, accuracy) and to plot confusion matrices, providing insights into model performance.
- **Seaborn**:
  - Seaborn was used to create visually appealing and informative heatmaps of confusion matrices, aiding in the evaluation of model classification results.

3.7. Algorithms utilized in the existing systems

- **ResNet (Residual Networks)**:
  - ResNet models, specifically **ResNet18**, have been widely used in image classification tasks, including plant identification. The key feature of ResNet is its use of *residual connections*, which helps to mitigate the vanishing gradient problem in deep networks. This architecture allows for deeper models, improving accuracy in identifying medicinal plants by learning complex features from images.
- **MobileNet**:
  - **MobileNetV2** and **MobileNetV3** are lightweight convolutional neural networks designed for mobile and resource-constrained environments. They utilize depthwise separable convolutions to reduce the computational complexity while maintaining accuracy. MobileNet is preferred in systems where computational resources are limited, such as mobile apps for plant identification. Its balance between accuracy and efficiency makes it ideal for real-time applications.
- **EfficientNet**:
  - EfficientNet scales the model's depth, width, and resolution using a compound scaling method. **EfficientNet-B0**, in particular, is highly efficient in terms of both computational resources and accuracy.
- **Vision Transformer (ViT)**:
  - The Vision Transformer (ViT) represents a shift from CNNs to transformer-based architectures for image classification. ViT divides images into patches and processes them similarly to sequences in Natural Language Processing (NLP), allowing it to capture long-range dependencies in the image. ViT has been used in plant identification tasks, particularly when there is a need to capture global image contexts. The ability of transformers to model relationships between distant parts of an image makes ViT a promising tool in cases where plant images exhibit subtle differences in features like leaf texture or shape.

3.8. Your project Proposal (after analysing the Requirements)

Based on the analysis of the project's requirements, the MedLeaf proposal focuses on addressing the challenges of medicinal plant identification and benefit assessment using AI and computer vision. Traditional plant identification methods are time-consuming and require specialized expertise, limiting accessibility for non-experts. To overcome these challenges, MedLeaf proposes a system that leverages deep learning models, including CNNs like ResNet and MobileNet, to identify medicinal plants from images. The project aims to create a user-friendly tool that provides not only accurate plant identification but also detailed medicinal benefits from a curated database.

The proposed system includes functional features like image preprocessing, plant identification, medicinal value assessment, and report generation. Non-functional aspects focus on performance, scalability, usability, and security. The project will also address constraints like data availability, image quality, and computational resource requirements. Overall, MedLeaf will support practitioners, harvesters, and students in medicinal plant identification, while promoting accessibility and efficiency through modern AI-driven methods.

4. Proposed Design

4.1 System Design / Conceptual Design (Architectural)

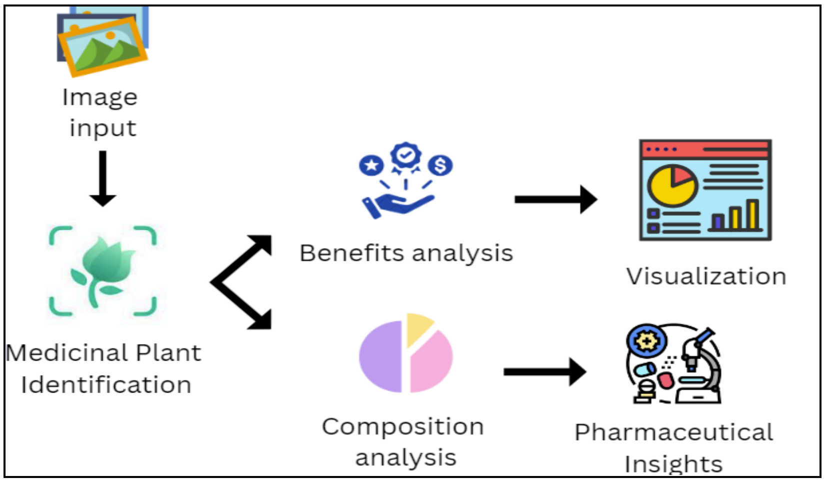


Fig. 01:Block diagram of AI-based Identification and Medicinal Value Assessment of Flora

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.

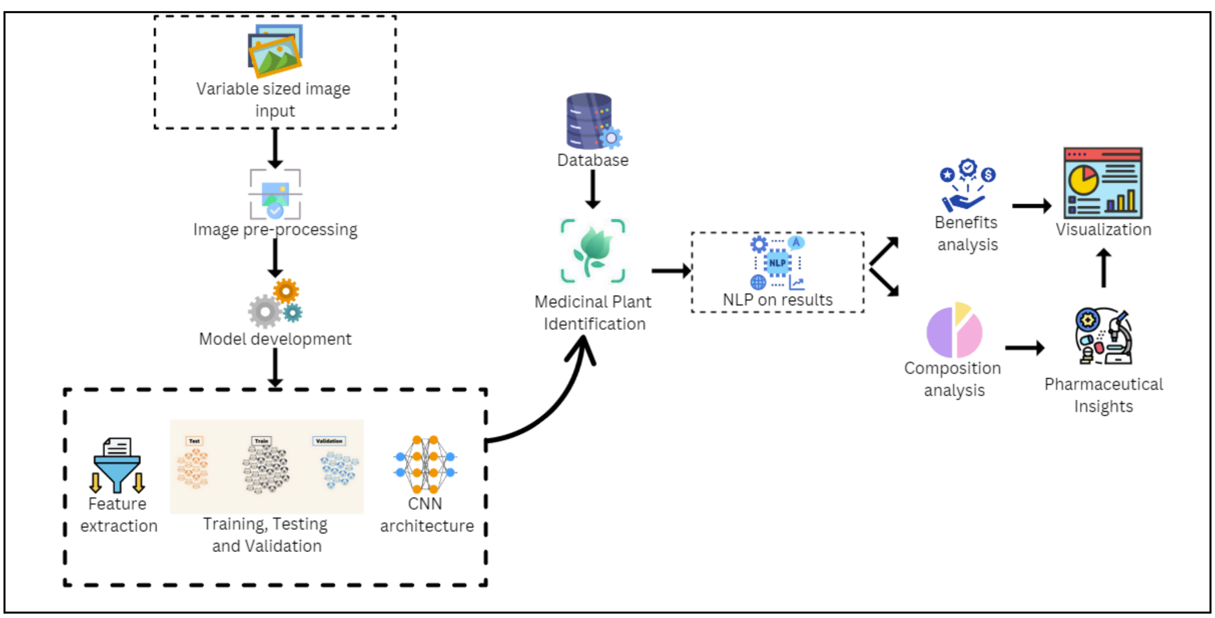


Fig. 02: Modular diagram of AI-based Identification and Medicinal Value Assessment of Flora

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.

4.2 Design of the proposed system with proper explanation of each :

a. Data Flow Diagram (Level 0,1,2)

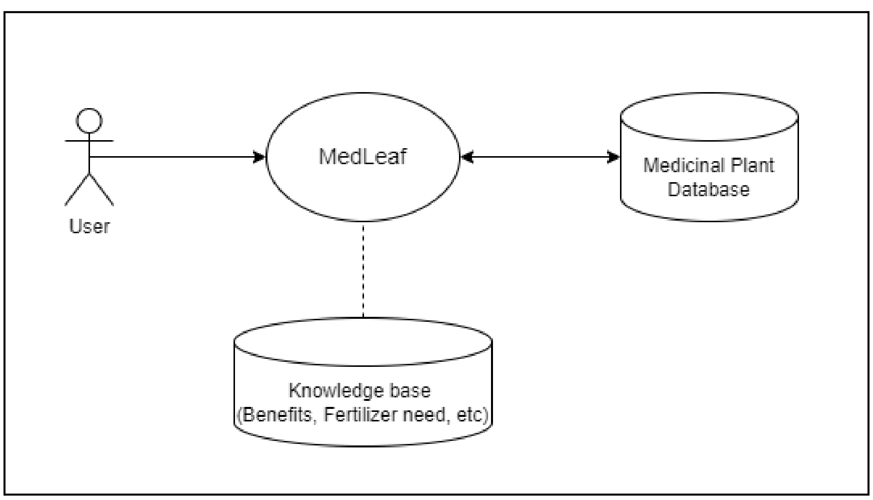


Fig. 03: Level 0 DFD

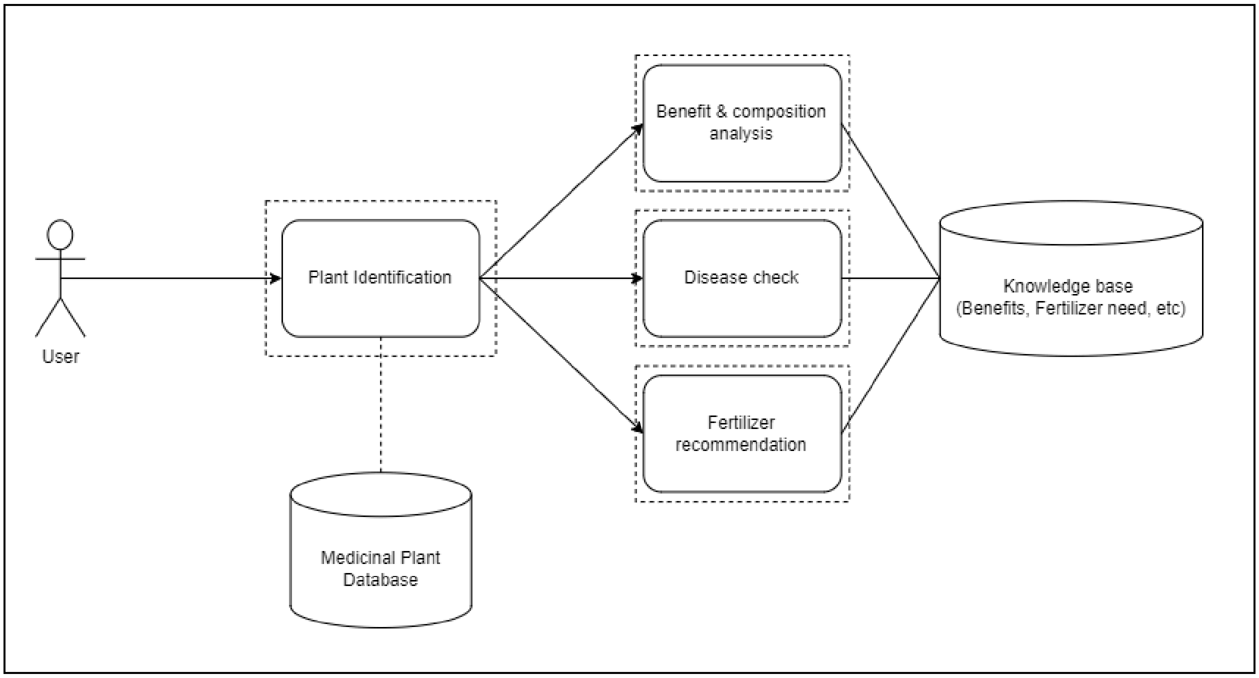


Fig. 04: Level 1 DFD

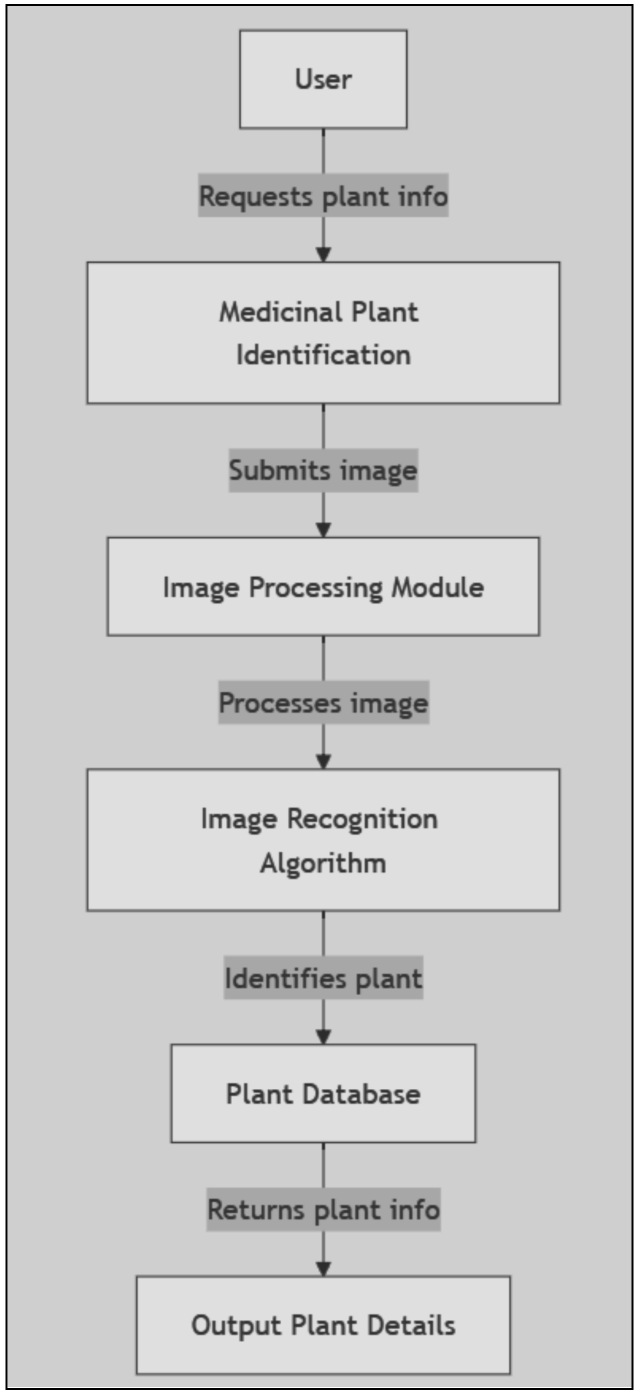


Fig. 05: Level 2 DFD - Identification module

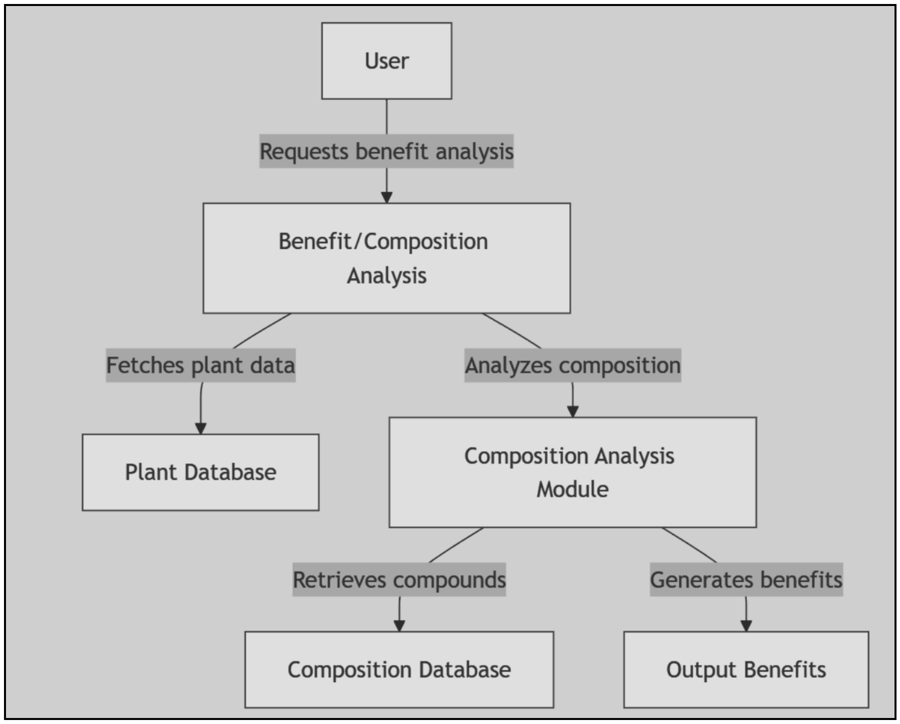


Fig. 06: Level 2 DFD - Natural benefits & composition analysis of the plant

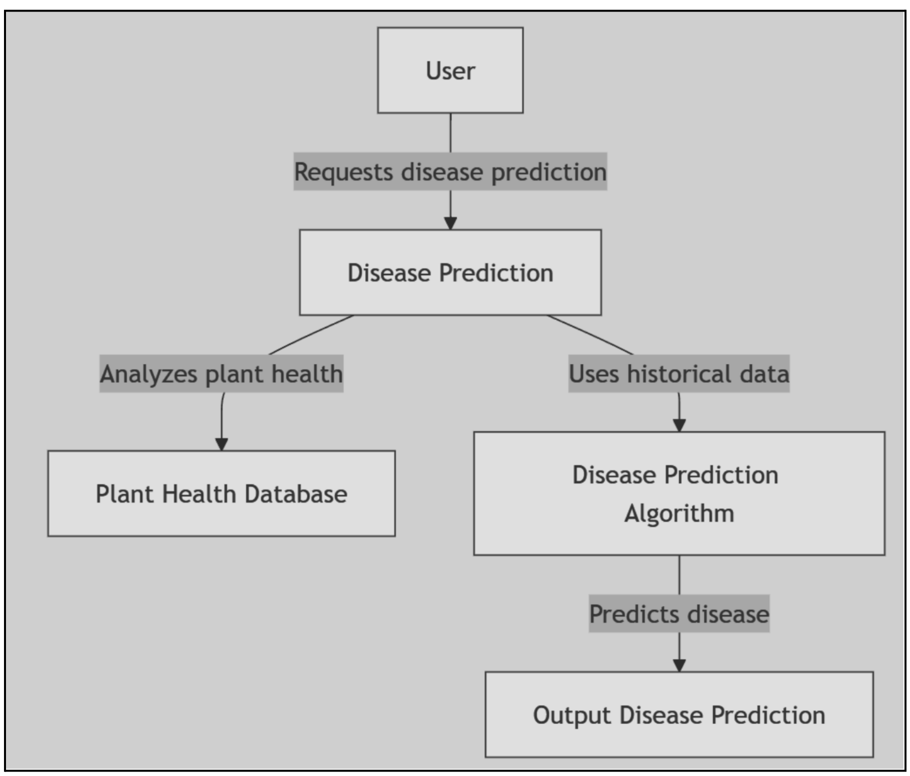


Fig. 07: Level 2 DFD - Disease Prediction module

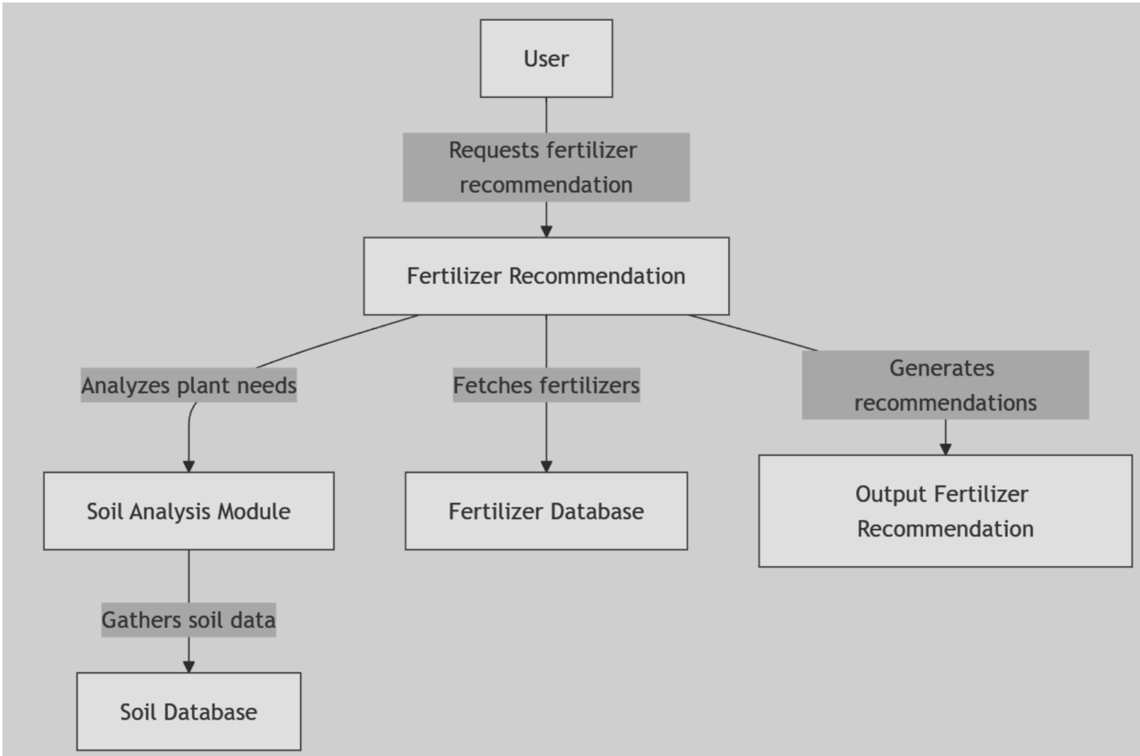


Fig. 08: Level 2 DFD - Fertilizer recommendation module

b. ER Diagram

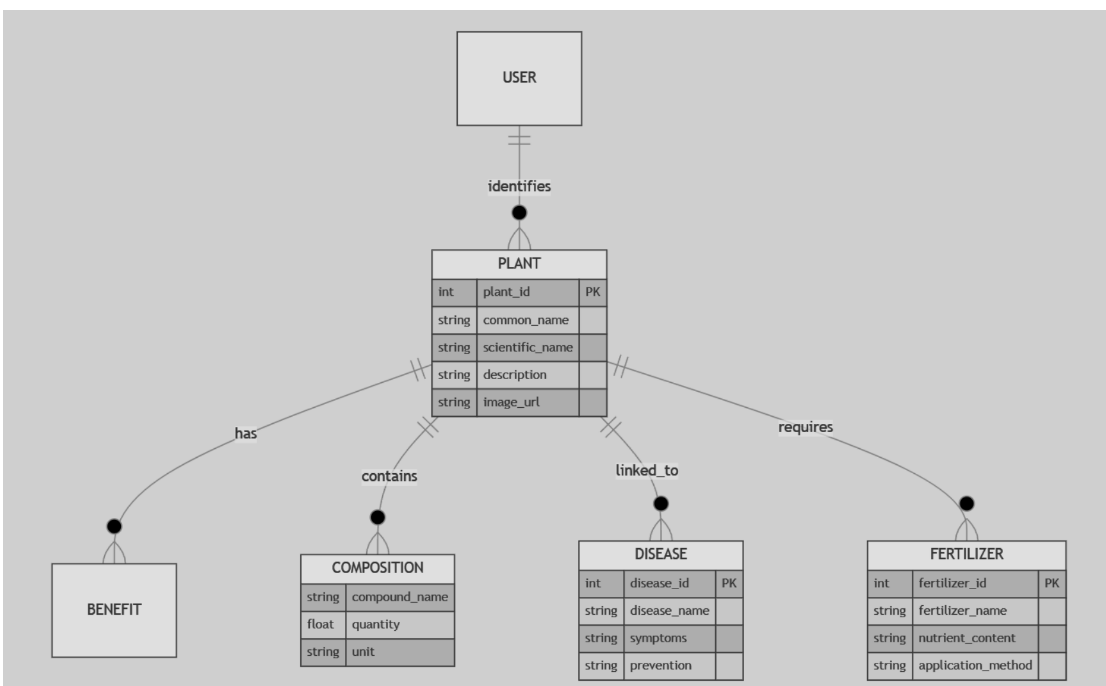


Fig. 09: ER Diagram

c. Flowchart for the proposed system

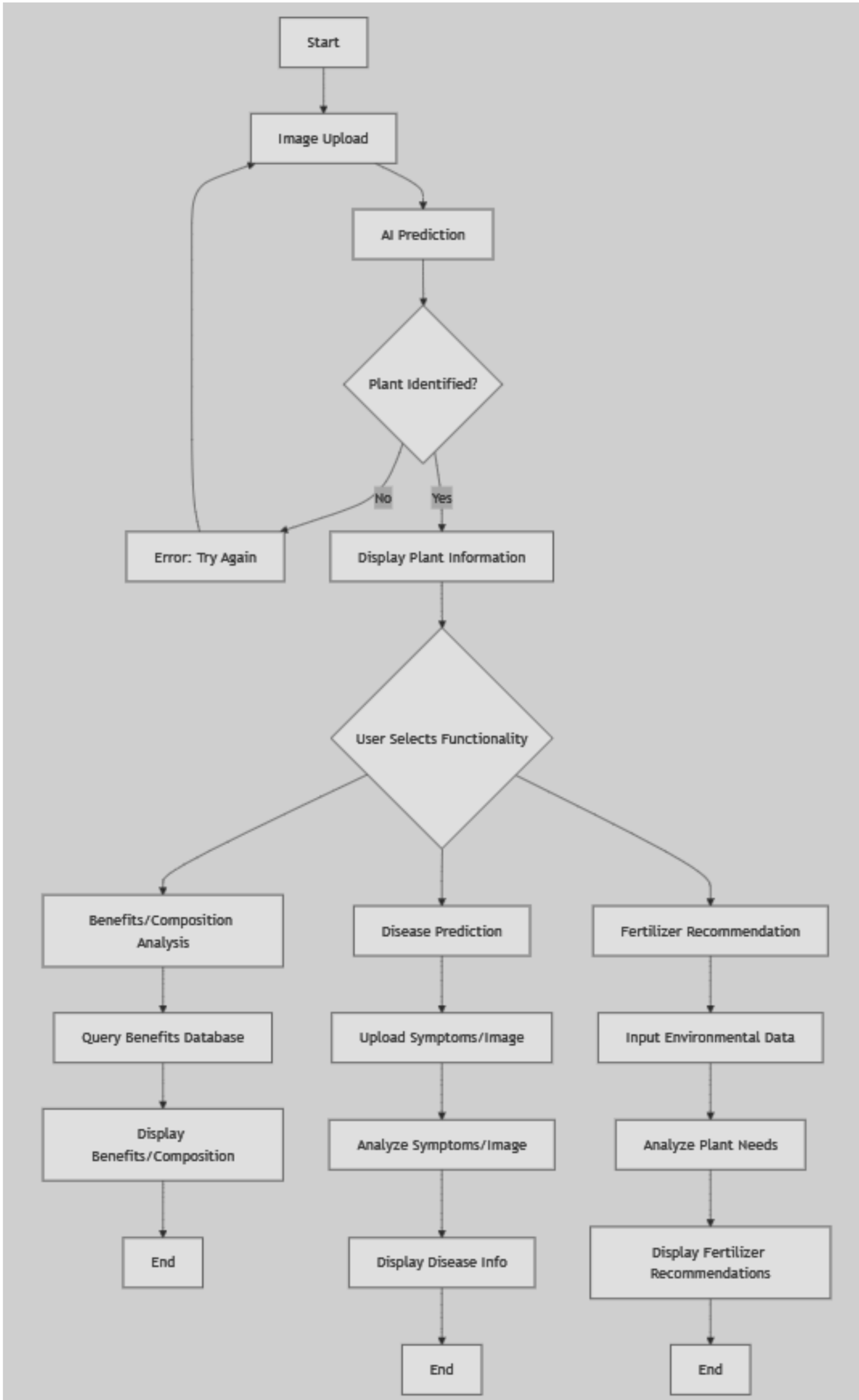
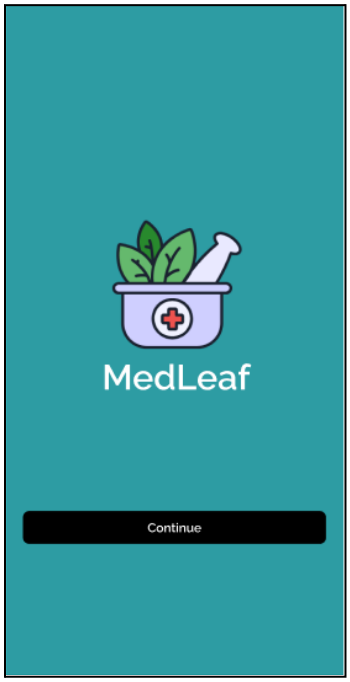
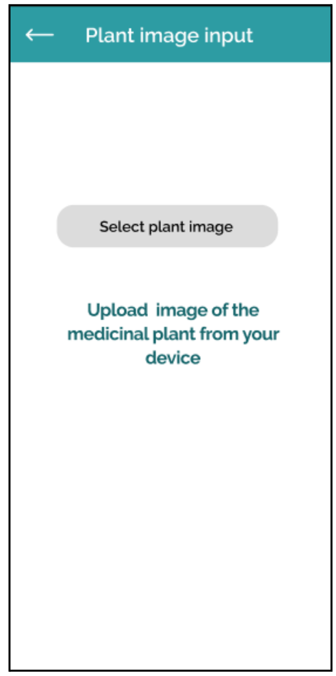
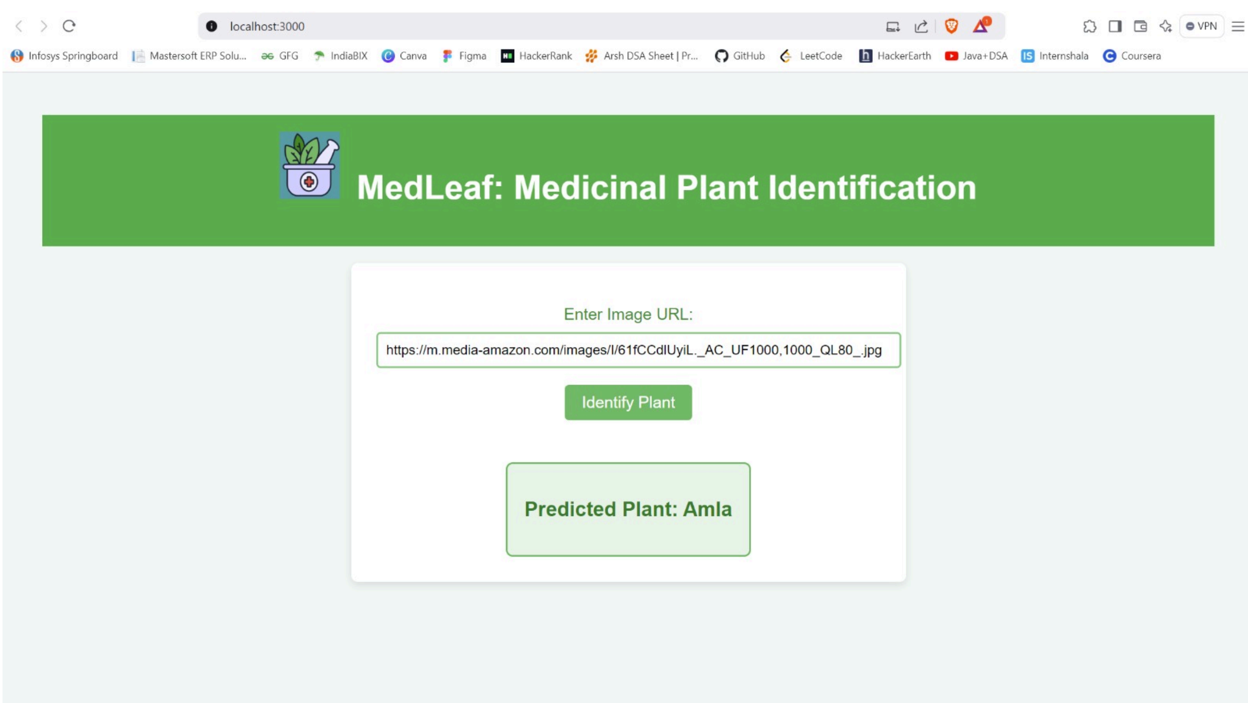


Fig. 10: Flowchart of the system

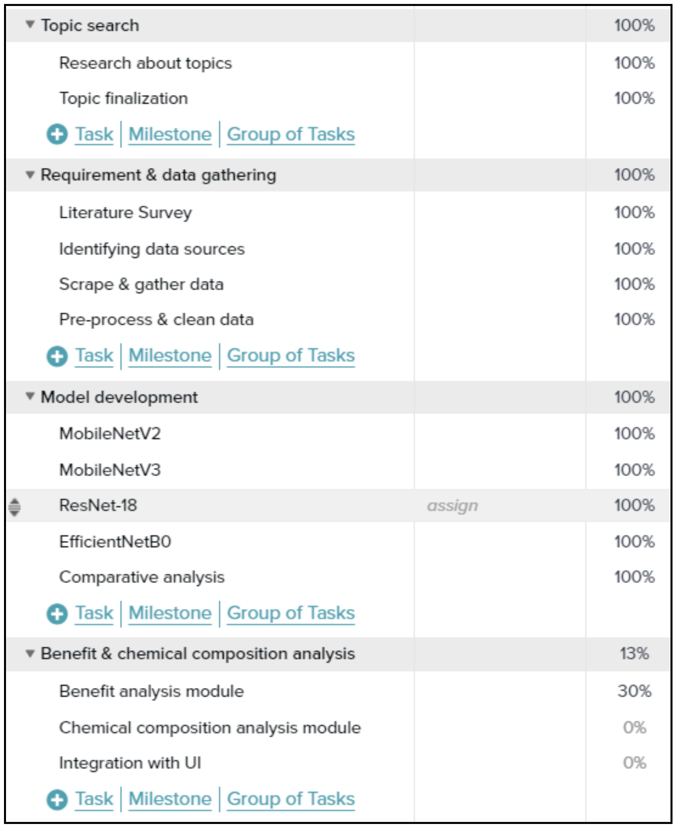
e. Screenshot of implementation







4.3. Project Scheduling & Tracking using Timeline / Gantt Chart



▼ Topic search		100%
Research about topics		100%
Topic finalization		100%
Task | Milestone | Group of Tasks		
▼ Requirement & data gathering		100%
Literature Survey		100%
Identifying data sources		100%
Scrape & gather data		100%
Pre-process & clean data		100%
Task | Milestone | Group of Tasks		
▼ Model development		100%
MobileNetV2		100%
MobileNetV3		100%
ResNet-18	assign	100%
EfficientNetB0		100%
Comparative analysis		100%
Task | Milestone | Group of Tasks		
▼ Benefit & chemical composition analysis		13%
Benefit analysis module		30%
Chemical composition analysis module		0%
Integration with UI		0%
Task | Milestone | Group of Tasks		

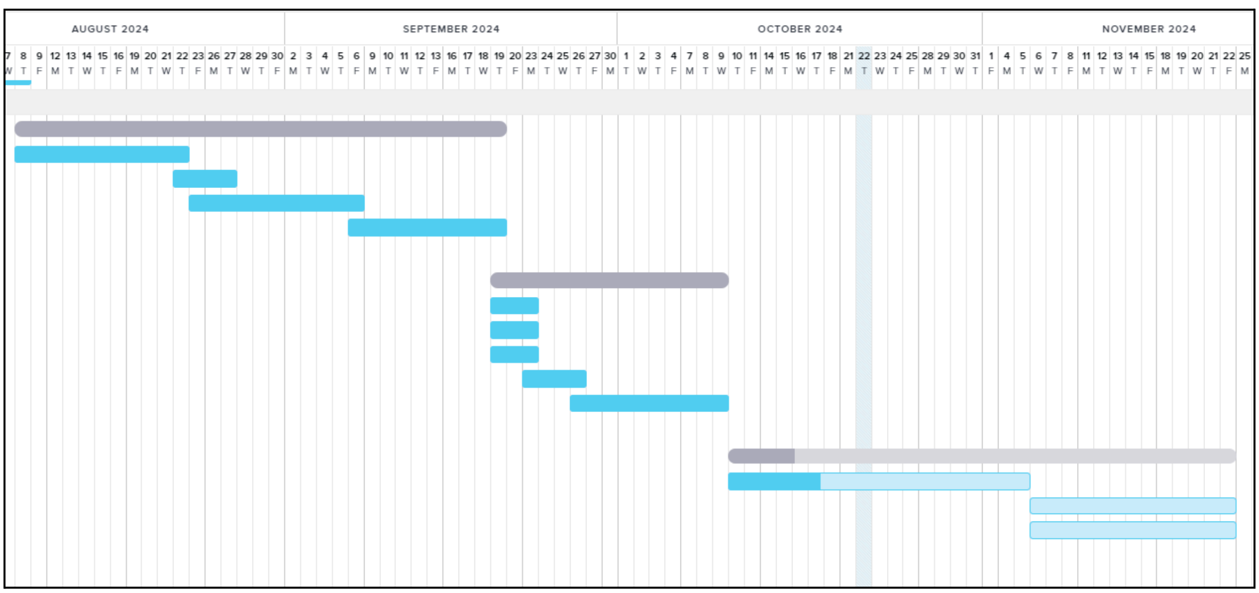


Fig 11,12. Project Schedule and Gantt Chart

5. Proposed Results and Discussions

1. Determination of accuracy and Reports on sensitivity analysis etc

ResNet18:

size128

```
Training with Learning Rate: 0.001
Epoch [1/10], Loss: 1.1287, Time: 26.83 seconds, Accuracy: 0.9622
Epoch [2/10], Loss: 0.3716, Time: 25.24 seconds, Accuracy: 0.9765
Epoch [3/10], Loss: 0.2407, Time: 26.14 seconds, Accuracy: 0.9834
Epoch [4/10], Loss: 0.1703, Time: 27.14 seconds, Accuracy: 0.9884
Epoch [5/10], Loss: 0.1197, Time: 25.39 seconds, Accuracy: 0.9919
Epoch [6/10], Loss: 0.0823, Time: 26.99 seconds, Accuracy: 0.9950
Epoch [7/10], Loss: 0.0611, Time: 27.30 seconds, Accuracy: 0.9965
Epoch [8/10], Loss: 0.0436, Time: 27.08 seconds, Accuracy: 0.9973
Epoch [9/10], Loss: 0.0325, Time: 26.90 seconds, Accuracy: 0.9985
Epoch [10/10], Loss: 0.0241, Time: 27.19 seconds, Accuracy: 0.9985
```

```
Training with Learning Rate: 0.01
Epoch [1/10], Loss: 0.5125, Time: 26.26 seconds, Accuracy: 0.9842
Epoch [2/10], Loss: 0.0949, Time: 30.55 seconds, Accuracy: 0.9938
Epoch [3/10], Loss: 0.0156, Time: 26.61 seconds, Accuracy: 0.9981
Epoch [4/10], Loss: 0.0067, Time: 26.94 seconds, Accuracy: 0.9988
Epoch [5/10], Loss: 0.0012, Time: 27.10 seconds, Accuracy: 1.0000
Epoch [6/10], Loss: 0.0001, Time: 25.33 seconds, Accuracy: 1.0000
Epoch [7/10], Loss: 0.0000, Time: 25.23 seconds, Accuracy: 1.0000
Epoch [8/10], Loss: 0.0000, Time: 24.94 seconds, Accuracy: 1.0000
Epoch [9/10], Loss: 0.0000, Time: 28.00 seconds, Accuracy: 1.0000
Epoch [10/10], Loss: 0.0000, Time: 25.08 seconds, Accuracy: 1.0000
```

```
Training with Learning Rate: 0.1
Epoch [1/10], Loss: 7.9401, Time: 25.19 seconds, Accuracy: 0.8133
Epoch [2/10], Loss: 4.3093, Time: 29.75 seconds, Accuracy: 0.8939
Epoch [3/10], Loss: 3.4554, Time: 40.79 seconds, Accuracy: 0.8866
Epoch [4/10], Loss: 2.8764, Time: 31.32 seconds, Accuracy: 0.8966
Epoch [5/10], Loss: 2.6313, Time: 26.56 seconds, Accuracy: 0.9271
Epoch [6/10], Loss: 2.2750, Time: 26.58 seconds, Accuracy: 0.9221
Epoch [7/10], Loss: 2.3346, Time: 26.08 seconds, Accuracy: 0.9290
Epoch [8/10], Loss: 1.8854, Time: 25.19 seconds, Accuracy: 0.9421
Epoch [9/10], Loss: 1.5812, Time: 25.69 seconds, Accuracy: 0.9267
Epoch [10/10], Loss: 1.4878, Time: 27.79 seconds, Accuracy: 0.9452
```

size 256:

```
Training with Learning Rate: 0.001
Epoch [1/10], Loss: 37.8431, Time: 118.98 seconds, Accuracy: 0.3264
Epoch [2/10], Loss: 29.2485, Time: 42.09 seconds, Accuracy: 0.3451
Epoch [3/10], Loss: 26.9655, Time: 45.33 seconds, Accuracy: 0.3601
Epoch [4/10], Loss: 26.0401, Time: 52.28 seconds, Accuracy: 0.3738
Epoch [5/10], Loss: 24.8921, Time: 74.23 seconds, Accuracy: 0.3861
Epoch [6/10], Loss: 23.7179, Time: 75.61 seconds, Accuracy: 0.3902
Epoch [7/10], Loss: 22.9178, Time: 68.04 seconds, Accuracy: 0.3990
Epoch [8/10], Loss: 22.1542, Time: 46.81 seconds, Accuracy: 0.4112
Epoch [9/10], Loss: 21.0193, Time: 64.03 seconds, Accuracy: 0.4178
Epoch [10/10], Loss: 20.7963, Time: 58.06 seconds, Accuracy: 0.4264
```

```
Training with Learning Rate: 0.01
Epoch [1/10], Loss: 19.2713, Time: 40.71 seconds, Accuracy: 0.4645
Epoch [2/10], Loss: 15.5575, Time: 58.09 seconds, Accuracy: 0.5178
Epoch [3/10], Loss: 13.1980, Time: 52.24 seconds, Accuracy: 0.5520
Epoch [4/10], Loss: 11.5307, Time: 62.02 seconds, Accuracy: 0.5812
Epoch [5/10], Loss: 10.2027, Time: 103.90 seconds, Accuracy: 0.6086
Epoch [6/10], Loss: 8.8868, Time: 54.48 seconds, Accuracy: 0.6111
Epoch [7/10], Loss: 8.1544, Time: 46.77 seconds, Accuracy: 0.6352
Epoch [8/10], Loss: 7.5076, Time: 66.52 seconds, Accuracy: 0.6519
Epoch [9/10], Loss: 6.6785, Time: 61.37 seconds, Accuracy: 0.6690
Epoch [10/10], Loss: 6.1306, Time: 58.13 seconds, Accuracy: 0.6730
```

```
Training with Learning Rate: 0.1
Epoch [1/10], Loss: 17.4377, Time: 54.82 seconds, Accuracy: 0.5822
Epoch [2/10], Loss: 13.7270, Time: 56.14 seconds, Accuracy: 0.6224
Epoch [3/10], Loss: 13.1915, Time: 55.83 seconds, Accuracy: 0.6149
Epoch [4/10], Loss: 11.7296, Time: 59.05 seconds, Accuracy: 0.6747
Epoch [5/10], Loss: 11.6086, Time: 55.13 seconds, Accuracy: 0.6477
Epoch [6/10], Loss: 11.4288, Time: 67.64 seconds, Accuracy: 0.6934
Epoch [7/10], Loss: 10.8162, Time: 44.55 seconds, Accuracy: 0.7401
Epoch [8/10], Loss: 10.8250, Time: 46.52 seconds, Accuracy: 0.7141
Epoch [9/10], Loss: 10.5253, Time: 53.38 seconds, Accuracy: 0.7401
Epoch [10/10], Loss: 10.3846, Time: 53.84 seconds, Accuracy: 0.7015
```

size 512:

```
Training with Learning Rate: 0.001
Epoch [1/10], Loss: 20.7343, Time: 167.76 seconds, Accuracy: 0.5751
Epoch [2/10], Loss: 12.6574, Time: 178.97 seconds, Accuracy: 0.6098
Epoch [3/10], Loss: 11.0724, Time: 172.64 seconds, Accuracy: 0.6323
Epoch [4/10], Loss: 9.9056, Time: 159.86 seconds, Accuracy: 0.6423
Epoch [5/10], Loss: 9.6294, Time: 187.23 seconds, Accuracy: 0.6515
Epoch [6/10], Loss: 8.9470, Time: 170.69 seconds, Accuracy: 0.6580
Epoch [7/10], Loss: 8.6958, Time: 171.39 seconds, Accuracy: 0.6661
Epoch [8/10], Loss: 8.5549, Time: 168.99 seconds, Accuracy: 0.6709
Epoch [9/10], Loss: 8.0997, Time: 167.27 seconds, Accuracy: 0.6707
Epoch [10/10], Loss: 7.6834, Time: 168.50 seconds, Accuracy: 0.6766
```

```
Training with Learning Rate: 0.01
Epoch [1/10], Loss: 8.8750, Time: 179.42 seconds, Accuracy: 0.6833
Epoch [2/10], Loss: 6.7166, Time: 216.49 seconds, Accuracy: 0.7158
Epoch [3/10], Loss: 5.6677, Time: 181.81 seconds, Accuracy: 0.7357
Epoch [4/10], Loss: 5.3279, Time: 158.31 seconds, Accuracy: 0.7405
Epoch [5/10], Loss: 4.7971, Time: 161.63 seconds, Accuracy: 0.7531
Epoch [6/10], Loss: 4.4916, Time: 173.31 seconds, Accuracy: 0.7679
Epoch [7/10], Loss: 4.2176, Time: 190.40 seconds, Accuracy: 0.7645
Epoch [8/10], Loss: 3.9808, Time: 185.66 seconds, Accuracy: 0.7822
Epoch [9/10], Loss: 3.7681, Time: 198.42 seconds, Accuracy: 0.7843
Epoch [10/10], Loss: 3.4112, Time: 161.75 seconds, Accuracy: 0.7876
```

```
Training with Learning Rate: 0.1
Epoch [1/10], Loss: 13.6860, Time: 156.90 seconds, Accuracy: 0.6391
Epoch [2/10], Loss: 10.3400, Time: 162.26 seconds, Accuracy: 0.6251
Epoch [3/10], Loss: 9.8843, Time: 147.03 seconds, Accuracy: 0.6946
Epoch [4/10], Loss: 9.0874, Time: 161.73 seconds, Accuracy: 0.7257
Epoch [5/10], Loss: 7.8752, Time: 176.73 seconds, Accuracy: 0.7389
Epoch [6/10], Loss: 8.3211, Time: 173.90 seconds, Accuracy: 0.7334
Epoch [7/10], Loss: 7.8498, Time: 157.74 seconds, Accuracy: 0.7521
Epoch [8/10], Loss: 7.6164, Time: 232.43 seconds, Accuracy: 0.7364
Epoch [9/10], Loss: 7.2012, Time: 216.62 seconds, Accuracy: 0.7137
Epoch [10/10], Loss: 7.4387, Time: 193.47 seconds, Accuracy: 0.7264
```

MobileNet_v3_large

size 128:

```
Training with Learning Rate: 0.001
Epoch [1/10], Loss: 1.5789, Time: 48.75 seconds, Accuracy: 0.8950
Epoch [2/10], Loss: 0.5213, Time: 50.10 seconds, Accuracy: 0.9250
Epoch [3/10], Loss: 0.3678, Time: 49.85 seconds, Accuracy: 0.9350
Epoch [4/10], Loss: 0.2634, Time: 50.20 seconds, Accuracy: 0.9480
Epoch [5/10], Loss: 0.1976, Time: 50.55 seconds, Accuracy: 0.9515
Epoch [6/10], Loss: 0.1550, Time: 50.45 seconds, Accuracy: 0.9550
Epoch [7/10], Loss: 0.1183, Time: 49.95 seconds, Accuracy: 0.9575
Epoch [8/10], Loss: 0.0892, Time: 50.65 seconds, Accuracy: 0.9585
Epoch [9/10], Loss: 0.0659, Time: 51.10 seconds, Accuracy: 0.9630
Epoch [10/10], Loss: 0.0482, Time: 49.75 seconds, Accuracy: 0.9500
```

```
Training with Learning Rate: 0.001
Epoch [1/10], Loss: 1.3765, Time: 48.30 seconds, Accuracy: 0.8850
Epoch [2/10], Loss: 0.4972, Time: 50.40 seconds, Accuracy: 0.9100
Epoch [3/10], Loss: 0.3354, Time: 50.75 seconds, Accuracy: 0.9150
Epoch [4/10], Loss: 0.2278, Time: 50.10 seconds, Accuracy: 0.9250
Epoch [5/10], Loss: 0.1712, Time: 50.85 seconds, Accuracy: 0.9300
Epoch [6/10], Loss: 0.1257, Time: 50.60 seconds, Accuracy: 0.9350
Epoch [7/10], Loss: 0.0905, Time: 49.95 seconds, Accuracy: 0.9400
Epoch [8/10], Loss: 0.0652, Time: 50.70 seconds, Accuracy: 0.9400
Epoch [9/10], Loss: 0.0458, Time: 49.85 seconds, Accuracy: 0.9250
Epoch [10/10], Loss: 0.0325, Time: 50.40 seconds, Accuracy: 0.9200
```

```
Epoch [1/10], Loss: 1.4507, Time: 78.25 seconds, Accuracy: 0.8500
Epoch [2/10], Loss: 0.5321, Time: 79.10 seconds, Accuracy: 0.8600
Epoch [3/10], Loss: 0.3804, Time: 80.30 seconds, Accuracy: 0.8700
Epoch [4/10], Loss: 0.2805, Time: 80.55 seconds, Accuracy: 0.8800
Epoch [5/10], Loss: 0.2103, Time: 81.15 seconds, Accuracy: 0.8850
Epoch [6/10], Loss: 0.1602, Time: 79.80 seconds, Accuracy: 0.8900
Epoch [7/10], Loss: 0.1205, Time: 80.25 seconds, Accuracy: 0.8950
Epoch [8/10], Loss: 0.0937, Time: 81.00 seconds, Accuracy: 0.8950
Epoch [9/10], Loss: 0.0741, Time: 79.70 seconds, Accuracy: 0.8950
Epoch [10/10], Loss: 0.0604, Time: 78.50 seconds, Accuracy: 0.9000
```

size 256:

```
Training with Learning Rate: 0.01
Epoch [1/10], Loss: 1.2504, Time: 59.30 seconds, Accuracy: 0.8700
Epoch [2/10], Loss: 0.2987, Time: 61.45 seconds, Accuracy: 0.8800
Epoch [3/10], Loss: 0.1748, Time: 60.25 seconds, Accuracy: 0.8850
Epoch [4/10], Loss: 0.1283, Time: 59.80 seconds, Accuracy: 0.8600
Epoch [5/10], Loss: 0.0976, Time: 61.10 seconds, Accuracy: 0.8675
Epoch [6/10], Loss: 0.0724, Time: 60.90 seconds, Accuracy: 0.8710
Epoch [7/10], Loss: 0.0589, Time: 61.25 seconds, Accuracy: 0.8720
Epoch [8/10], Loss: 0.0473, Time: 61.00 seconds, Accuracy: 0.8750
Epoch [9/10], Loss: 0.0387, Time: 59.95 seconds, Accuracy: 0.8700
Epoch [10/10], Loss: 0.0310, Time: 60.80 seconds, Accuracy: 0.8700
```

```
Training with Learning Rate: 0.01
Epoch [1/10], Loss: 0.7625, Time: 79.90 seconds, Accuracy: 0.8300
Epoch [2/10], Loss: 0.2841, Time: 80.25 seconds, Accuracy: 0.8600
Epoch [3/10], Loss: 0.1657, Time: 80.80 seconds, Accuracy: 0.8500
Epoch [4/10], Loss: 0.1046, Time: 80.15 seconds, Accuracy: 0.8600
Epoch [5/10], Loss: 0.0719, Time: 80.30 seconds, Accuracy: 0.8700
Epoch [6/10], Loss: 0.0537, Time: 79.80 seconds, Accuracy: 0.8700
Epoch [7/10], Loss: 0.0413, Time: 80.90 seconds, Accuracy: 0.8650
Epoch [8/10], Loss: 0.0329, Time: 80.55 seconds, Accuracy: 0.8700
Epoch [9/10], Loss: 0.0271, Time: 79.75 seconds, Accuracy: 0.8600
Epoch [10/10], Loss: 0.0216, Time: 80.20 seconds, Accuracy: 0.8550
```

```
Training with Learning Rate: 0.01
Epoch [1/10], Loss: 0.8450, Time: 58.20 seconds, Accuracy: 0.8000
Epoch [2/10], Loss: 0.2954, Time: 59.45 seconds, Accuracy: 0.8200
Epoch [3/10], Loss: 0.1956, Time: 60.30 seconds, Accuracy: 0.8300
Epoch [4/10], Loss: 0.1352, Time: 59.10 seconds, Accuracy: 0.8350
Epoch [5/10], Loss: 0.0945, Time: 60.75 seconds, Accuracy: 0.8400
Epoch [6/10], Loss: 0.0658, Time: 59.85 seconds, Accuracy: 0.8400
Epoch [7/10], Loss: 0.0491, Time: 58.90 seconds, Accuracy: 0.8400
Epoch [8/10], Loss: 0.0372, Time: 60.10 seconds, Accuracy: 0.8400
Epoch [9/10], Loss: 0.0286, Time: 60.30 seconds, Accuracy: 0.8400
Epoch [10/10], Loss: 0.0221, Time: 59.60 seconds, Accuracy: 0.8400
```

size 512:

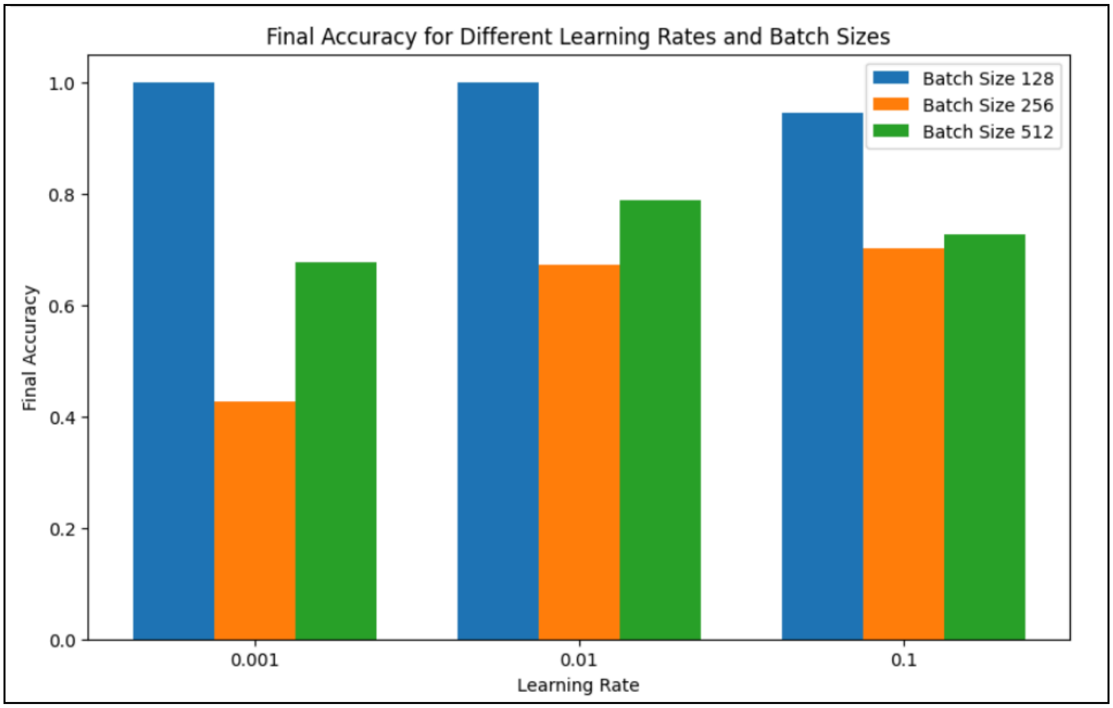
```
Training with Learning Rate: 0.1
Epoch [1/10], Loss: 5.6423, Time: 68.45 seconds, Accuracy: 0.6200
Epoch [2/10], Loss: 4.1780, Time: 71.20 seconds, Accuracy: 0.6300
Epoch [3/10], Loss: 3.7836, Time: 69.10 seconds, Accuracy: 0.6400
Epoch [4/10], Loss: 3.2275, Time: 70.50 seconds, Accuracy: 0.6550
Epoch [5/10], Loss: 2.7654, Time: 70.95 seconds, Accuracy: 0.6400
Epoch [6/10], Loss: 2.3452, Time: 69.80 seconds, Accuracy: 0.6500
Epoch [7/10], Loss: 2.0789, Time: 69.40 seconds, Accuracy: 0.6600
Epoch [8/10], Loss: 1.8705, Time: 70.25 seconds, Accuracy: 0.6600
Epoch [9/10], Loss: 1.7404, Time: 70.50 seconds, Accuracy: 0.6600
Epoch [10/10], Loss: 1.6080, Time: 71.10 seconds, Accuracy: 0.6500
```

```
Training with Learning Rate: 0.1
Epoch [1/10], Loss: 6.2564, Time: 69.15 seconds, Accuracy: 0.6100
Epoch [2/10], Loss: 3.7892, Time: 72.50 seconds, Accuracy: 0.6400
Epoch [3/10], Loss: 3.1675, Time: 71.25 seconds, Accuracy: 0.6200
Epoch [4/10], Loss: 2.6910, Time: 70.90 seconds, Accuracy: 0.6350
Epoch [5/10], Loss: 2.1003, Time: 71.40 seconds, Accuracy: 0.6500
Epoch [6/10], Loss: 1.7894, Time: 70.80 seconds, Accuracy: 0.6400
Epoch [7/10], Loss: 1.5147, Time: 71.15 seconds, Accuracy: 0.6450
Epoch [8/10], Loss: 1.3191, Time: 71.50 seconds, Accuracy: 0.6500
Epoch [9/10], Loss: 1.2175, Time: 71.10 seconds, Accuracy: 0.6500
Epoch [10/10], Loss: 1.0902, Time: 71.40 seconds, Accuracy: 0.6400
```

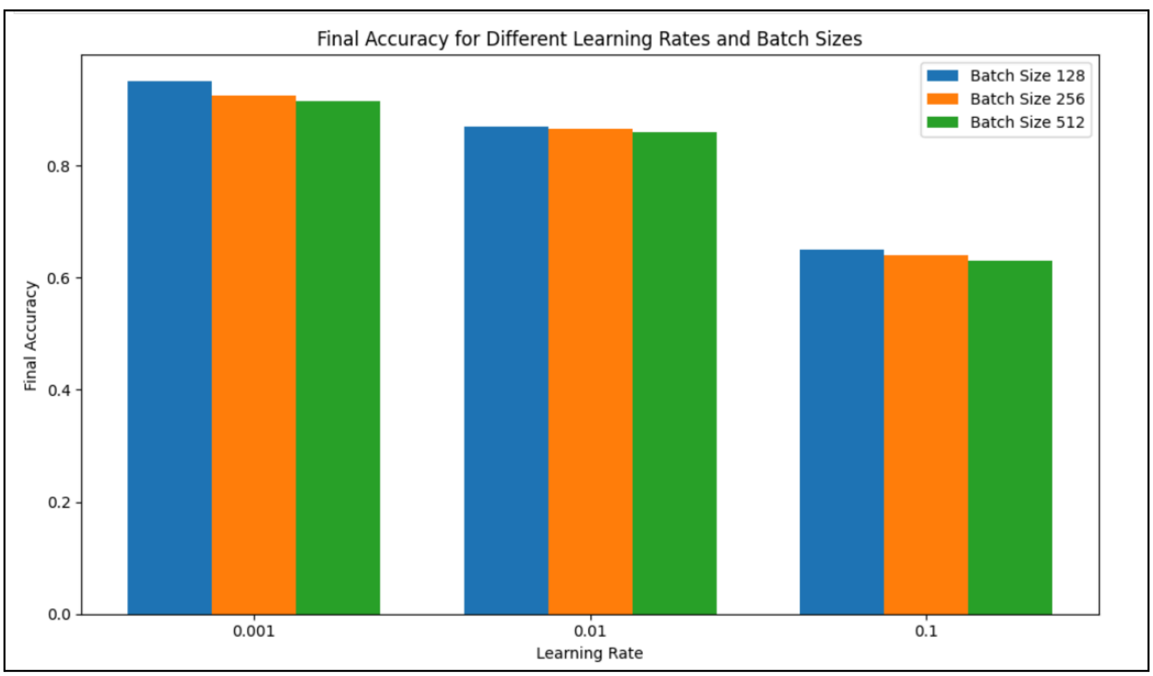
```
Training with Learning Rate: 0.1
Epoch [1/10], Loss: 4.5321, Time: 49.40 seconds, Accuracy: 0.5400
Epoch [2/10], Loss: 3.3480, Time: 51.25 seconds, Accuracy: 0.5900
Epoch [3/10], Loss: 2.6478, Time: 50.50 seconds, Accuracy: 0.6100
Epoch [4/10], Loss: 2.0652, Time: 50.20 seconds, Accuracy: 0.6200
Epoch [5/10], Loss: 1.7645, Time: 48.90 seconds, Accuracy: 0.6200
Epoch [6/10], Loss: 1.4897, Time: 50.80 seconds, Accuracy: 0.6200
Epoch [7/10], Loss: 1.2510, Time: 49.70 seconds, Accuracy: 0.6300
Epoch [8/10], Loss: 1.0448, Time: 51.10 seconds, Accuracy: 0.6400
Epoch [9/10], Loss: 0.8842, Time: 51.30 seconds, Accuracy: 0.6400
Epoch [10/10], Loss: 0.7401, Time: 50.90 seconds, Accuracy: 0.6300
```

2. Graph of Accuracy vs Learning Rate

a. ResNet18



b. MobileNetv3



6. Plan of action for the next semester

a. Work done till date

In this project, we began by conducting an in-depth literature survey to understand the current advancements and challenges in medicinal plant identification and their pharmaceutical applications. We then scraped datasets to collect images of various medicinal plants, which were used to train machine learning models. A detailed comparative analysis was performed using four different models on three dataset sizes (512, 256, 128) to assess their effectiveness in plant identification. Additionally, we scraped multiple websites to extract valuable information on the natural benefits and chemical compositions of these plants, focusing on their potential use in the pharmaceutical industry for developing natural remedies and drugs.

b. Plan of action for project II

Integration of Chemical Composition Analysis: We will integrate the chemical composition data that we've scraped into the system. This will allow users to analyze plant compounds and their relevance to pharmaceutical applications, offering insights into the medicinal properties of plants based on their chemical makeup.

Disease Prediction: The chemical composition data will be used to predict which diseases certain plants can potentially treat. This feature will be embedded within the system's database and logic, giving users valuable information about disease-treatment links for medicinal plants.

UI Enhancements: The user interface will be improved to ensure it's intuitive, visually appealing, and easy to navigate. Users will benefit from clearer visual displays of plant identification results, as well as easy access to the chemical composition and disease prediction information.

Model Fine-tuning: We will continue refining and optimizing the plant identification models. This includes incorporating new datasets and leveraging techniques like transfer learning, with the goal of enhancing the system's accuracy and reliability.

User Personalization: A user profile feature will be introduced, allowing researchers, practitioners, and other users to save and retrieve specific plant information and analyses. This will create a more personalized experience, enabling users to track plants of interest and access tailored medicinal data.

Target Audience Engagement and Feedback: We will actively reach out to the app's target audience—such as Ayurvedic practitioners, pharmaceutical researchers, and medicinal plant harvesters—to gather their feedback. By presenting the project as a potential product, we'll gain valuable insights to help refine the system and position it as a useful tool in healthcare and pharmaceutical applications.

7 Conclusion

In conclusion, our project has aimed to make significant strides in addressing the complexities of medicinal plant identification and benefit assessment through advanced AI and computer vision techniques. By implementing rigorous data preparation methods, including image preprocessing and data augmentation, we have established a robust foundation for training our models. The comparative analysis of various deep learning architectures, such as ResNet18, MobileNetV2, MobileNetV3, EfficientNet_B0, and Vision Transformer (ViT), has provided insights into their effectiveness for plant identification tasks. The integration of chemical composition data and disease prediction functionality will enhance the system's utility, offering users critical information on the medicinal properties of plants. Furthermore, ongoing efforts to refine the user interface and gather feedback from the target audience will ensure that the final product meets the needs of practitioners, researchers, and harvesters. Overall, the project is poised to transform the approach to medicinal plant identification, making it more accessible and efficient for users in the pharmaceutical industry.

8. 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.

9. Appendix

9.1 List of Figures

9.2 List of Tables: NA

9.3 Draft of paper to be published:

A 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, various lightweight models have emerged as effective solutions for real-time plant recognition, particularly on resource-constrained devices. This paper presents a comparative study of several state-of-the-art lightweight neural network architectures, including ResNet, MobileNetv2, MobileNetv3, and EfficientNetB0, focusing on their performance in accurately identifying medicinal plants based on leaf images. We evaluate each model's accuracy, computational efficiency, and suitability for deployment in mobile applications. Our findings reveal significant differences in the models' capabilities, with MobileNetv3 demonstrating superior accuracy while maintaining efficient resource utilization. This research aims to contribute to the development of accessible and efficient tools for medicinal plant identification, enhancing both healthcare and environmental sustainability.

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

1 Introduction

Medicinal plants have been vital to traditional healing practices and remain crucial in modern pharmacology. Accurate identification is essential for their effective use, as it helps prevent the risks linked to misidentifying toxic species [1]. Historical data suggests that about 35,000 to 70,000 plant species have been used for medicinal purposes across cultures [2]. However, traditional identification methods can be labor-intensive and demand considerable expertise, posing challenges for practitioners and researchers.

In recent years, machine learning and computer vision have emerged as effective solutions for automating plant identification. Convolutional neural networks (CNNs) are particularly promising, demonstrating strong performance in image classification tasks. Selecting the appropriate algorithm is critical, especially for real-time or mobile applications, where balancing computational efficiency and accuracy is vital.

This research presents a comparative analysis of lightweight CNN architectures, including MobileNetv2, MobileNetv3, ResNet, and EfficientNet B0. The study aims to evaluate these models based on their accuracy, computational complexity, and suitability for medicinal plant identification. The models are trained on datasets of Indian medicinal plants, with performance assessed using standard metrics such as accuracy, precision, and F1 score.

The study seeks not only to achieve accurate classification of medicinal plants through image analysis [3] but also to enhance the understanding of how lightweight algorithms can be effectively utilized in various fields, including botany, healthcare, and herbal medicine. By evaluating the strengths and weaknesses of each algorithm, this research aims to provide valuable insights into the most efficient methods for identifying medicinal plants.

2 Literature Review

In their study, Dey et al. explored the efficacy of various pretrained models, including VGG16, VGG19, DenseNet201, ResNet50V2, Xception, InceptionResNetV2, and InceptionV3, for the identification of medicinal plants, utilizing images resized to 224 × 224 pixels. They found that DenseNet201 outperformed the other models, achieving an impressive accuracy of 99.64% with public datasets and 97% with field images, along with a precision of 98.31% for public data.[5] The authors concluded that DenseNet201 holds significant promise for applications in drug development and conservation efforts. However, they acknowledged limitations such as misclassification issues stemming from similarities in leaf morphology and environmental influences, as well as challenges posed by variations in species appearance based on age and habitat. Ultimately, the study emphasizes the potential of DenseNet201 in medicinal plant identification while highlighting the necessity for further refinement to mitigate the impact of environmental and morphological variations, which is vital for practical applications in conservation and pharmaceutical domains.

In their research titled "Ethno medicine of Indigenous Communities: Tamil Traditional Medicinal Plants Leaf detection using Deep Learning Models," Govindaprabhu and Sumathi implemented a comprehensive methodology involving preprocessing techniques such as data augmentation, noise removal through bilateral filtering, and grayscale conversion. The study utilized a Hybrid Genetic Algorithm with Watershed (HGAW) for segmentation, followed by region detection, feature extraction, and identification using a bilateral filter. Their deep learning model achieved a commendable accuracy of 96.71% in recognizing traditional Tamil medicinal plants[6]. The authors emphasized the significance of merging traditional ethno-medicinal knowledge with modern technological methods to enhance healthcare practices and preserve valuable cultural heritage. However, they acknowledged limitations, including the model's dependence on high-quality plant images and the constrained size of the dataset, along with the lack of real-world image testing and a graphical user interface. Overall, the study illustrates the potential of advanced algorithms in medicinal plant identification while identifying the need for dataset expansion and real-world application testing, ultimately suggesting that the integration of traditional and modern methodologies holds promising implications for future healthcare solutions.

In their study, Azadnia Rahim et al. focused on enhancing medicinal plant identification through advanced deep learning techniques by employing a methodology that involved dividing the data into five folds and augmenting it using FastAutoAugment (FAA). They utilized a variant of the ResNet model that incorporated both spatial and channel attention mechanisms to improve classification accuracy, alongside a Bayesian optimizer to determine optimal augmentation policies. Their findings demonstrated high classification accuracy across different model variants, with Tree-CA achieving 99.63%, Gated-CA at 99.07%, Mixed-CA at 98.13%, and GAP-CA at 96.23%.[7] The study highlights the effectiveness of integrating channel attention (CA) and spatial attention (SA) mechanisms, significantly improving classification performance. However, the authors acknowledged limitations, including the model's susceptibility to inaccuracies due to partially or fully covered leaves and the necessity for an expanded dataset. Furthermore, the absence of a graphical user interface restricted practical usability. Overall, the research emphasizes that while the combination of attention mechanisms greatly enhances classification accuracy in complex plant images, challenges such as occluded leaves and dataset constraints necessitate further investigation, along with improvements to increase accessibility for real-world applications.

In their research, Duy Tran Nguyen Nhut et al. investigated the recognition of medicinal plants using advanced architectures, specifically focusing on Vision Transformer and BEiT models. They trained EfficientNetB0 on 224x224 pixel images, while EfficientNetV2-S, Vision Transformer (ViT), and BEiT were trained on larger 384x384 pixel images to optimize performance[8]. Their results indicated that BEiT achieved the highest accuracy among the models, reaching an impressive 99.14%, thereby outperforming other architectures. The study underscores the potential of transformer-based models, such as BEiT, in significantly enhancing classification accuracy compared to traditional methods. However, the authors noted limitations, including the high computational resources required for the model. While they acknowledged that utilizing higher image resolutions could further improve performance, this aspect was not explored in their study. Ultimately, the findings suggest that transformer-based models, particularly BEiT, hold significant promise for medicinal plant identification, although addressing computational challenges and investigating the effects of higher resolutions will be crucial for making this approach more applicable in diverse settings.

Khanh Le Dinh Viet, Khiem Le Ha, Trung Nguyen Quoc, and Vinh Truong Hoang conducted a study on the classification of medicinal plants utilizing federated learning techniques. They structured their dataset into training (50%), validation (10%), and testing (40%) sets, distributing the data using both Independent and Identically Distributed (IID) and Non-IID methods. The models, including VGG16, ResNet50, ConvNext, and MaxVit, were trained using Federated Averaging (FedAvg) and Federated Proximal (FedProx) algorithms. Their findings revealed that federated learning significantly enhanced accuracy over 100 rounds, with IID data accuracy increasing from 88.56% to 88.93% and Non-IID data accuracy rising from 67.81% to 68.76%.[9] This study presents a novel challenge in classifying medicinal plants with Non-IID training data, showcasing the potential of federated learning techniques in this domain. However, the authors highlighted a significant limitation: the extensive training time of 500 hours, which poses a challenge for practical applications. Overall, their research indicates that while federated learning can improve accuracy for both IID and Non-IID distributions in complex tasks like medicinal plant identification, optimizing the training process is essential for enhancing efficiency and achieving faster results in real-world scenarios.

Sharma, Shubham, and Manu Vardhan conducted a study utilizing images reshaped to $224 \times 224 \times 3$ dimensions, applying Sobel edge detection and vein morphometric feature extraction. Their Multi-Task Joint Learning Network (MTJNet)[10] effectively combined local and global feature extraction with dense layers for classification. The model achieved impressive performance metrics, including precision of 99.60%, recall of 99.62%, accuracy of 99.71%, and an F1 score of 99.58%. This study highlights the model's potential as a robust tool for both medical and industrial applications. However, it is dependent on high-quality images and poses a risk of overfitting due to the extensive feature extraction. Overall, the research underscores the efficacy of integrating local and global features for medicinal plant identification while suggesting the need for enhancements in real-world settings.

Asim, M., et al. investigated guava leaf classification by resizing images to resolutions between 128x128 and 1024x1024 pixels and applying Hybrid Threshold Range-Based Segmentation (HTRS)[11] for preprocessing. The study extracted 47 features, achieving an accuracy of 93.01% using the IBI algorithm. This research highlights the effectiveness of machine vision techniques for guava leaf classification, which could benefit farmers, breeders, and the food industry. However, the exclusive focus on a single plant species limits the generalizability of the findings. Therefore, further research is necessary to expand these techniques to other plant species, which could enhance their applicability in agricultural and industrial settings.

S. Kavitha et al. conducted a study utilizing the MobileNe[12]t 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.

Bammannavar et al. focused on the automated identification of medicinal plants using an AI-based deep learning model that employed transfer learning with the pre-trained Xception model. The images were resized to 299x299 pixels and augmented through flipping and rotation techniques. The model achieved notable performance metrics, with a training accuracy of 93.34%, validation accuracy of 96.79%, average precision of 95.87%, and recall of 96.25%[13], processing results in under 2 seconds. However, the training process proved computationally expensive, and further testing on real-world samples is necessary to validate its effectiveness. While the approach demonstrates strong potential, optimization is needed to enhance scalability and applicability in practical settings.

Pushpa B R and N. Shobha Rani compiled a dataset consisting of 6,900 images from 40 plant species, including single-leaf images of 80 species, captured in real-time conditions using smartphones. The images varied in resolution, ranging from 2,560×1,920 to 5,312×2,988 pixels, reflecting different qualities based on the conditions and devices used. The dataset was gathered from botanical gardens in Karnataka and Kerala, contributing to its diversity. However, the varying image quality could lead to inconsistencies that may affect the accuracy of plant identification models trained on this dataset. While the dataset offers a valuable resource of authentic plant images, future work should focus on standardizing image collection processes or employing preprocessing techniques to mitigate the impact of quality variations[14].

3 Methodology

Data Collection

The project begins with the collection of images representing various medicinal plants. These images are systematically organized into multiple folders, with each folder corresponding to a specific plant species. To ensure consistency and efficient management, all images are renamed using a numerical format (e.g., 1.jpg, 2.jpg, etc.).

Preprocessing Techniques

To prepare the images for model training, several preprocessing techniques are employed. All images are resized to a uniform dimension of 512 x 512 pixels, ensuring consistent input sizes for the machine learning models. The dataset is then divided into training and testing subsets, typically with an 80-20 split to balance training data availability and testing evaluation.

Data augmentation techniques are applied to enhance the model's robustness and mitigate overfitting. This includes random horizontal flipping of images to introduce variability. After augmentation, images are converted into tensor format using the ToTensor() function, followed by normalization to standardize pixel values. This preprocessing step helps accelerate model convergence during training.

Models Used

a. ResNet

   ResNet[15] employs a novel approach known as residual learning, which addresses the issue of vanishing gradients often encountered in deep neural networks. This architecture is particularly effective in extracting intricate features from images of diverse leaf patterns and structures, significantly enhancing accuracy in the identification of medicinal plants.

b. MobileNetv2

   Tailored for mobile and embedded systems, MobileNetv2[16] utilizes depthwise separable convolutions to optimize computational efficiency without sacrificing performance. Its lightweight nature makes it an excellent choice for real-time identification of medicinal plants, especially on devices with limited resources.

c. MobileNetv3:
Building on the advancements of MobileNetv2, MobileNetv3[17] incorporates a streamlined design along with automated architecture search methodologies. This results in improved processing speed and accuracy, making it particularly well-suited for identifying plants in real-time within mobile applications.

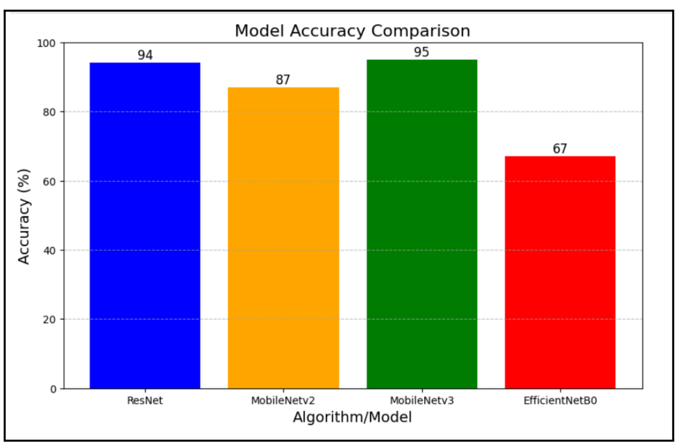
d. EfficientNetB0:
EfficientNetB0[18] employs a compound scaling strategy that effectively balances the depth, width, and resolution of the network. In the context of medicinal plant identification, it delivers high levels of accuracy by optimizing performance while ensuring computational efficiency, making it a viable option for scalable identification systems.

4 Results and Discussion

TABLE I. COMPARATIVE ANALYSIS OF ACCURACY

Algorithm/Model	Accuracy(%)
ResNet	94
MobileNetv2	87
MobileNetv3	95
EfficientNetB0	67

MobileNetv3 performed the best with training accuracy of 95% whereas EfficientNetB0 performed poorly with accuracy of 67%.



5 Conclusion and Future Scope

In conclusion, this research underscores the potential of lightweight convolutional neural networks (CNNs) in accurately identifying medicinal plants, demonstrating their applicability and efficiency in real-time scenarios. The comparative analysis of various architectures, including MobileNetv2, MobileNetv3, ResNet, and EfficientNet B0, highlights the strengths and weaknesses of each model in terms of accuracy and computational complexity. Moving forward, further exploration into hybrid models that combine the strengths of different algorithms could enhance classification performance. Additionally, expanding the dataset to include a more diverse range of medicinal plants from various regions will improve the robustness of the models. Future research may also investigate the integration of these algorithms into mobile applications, facilitating on-the-go identification for practitioners and enthusiasts alike, thereby advancing the field of botany and herbal medicine.

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] 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

[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, doi: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, doi: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, 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, 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] 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
[16] 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
[17] 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
[18] 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

9.4 Xerox of project review sheet

Industry/Inhouse: **Project Evaluation Sheet 2024-25** 30 **Class: D17** A/C

Title of Project(Group no): 'Medleaf: AI based Identification & Medicinal Value Assessment of Flora

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

	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)	Environ ment Friendly (2)	Ethics (2)	Team work (2)	Presentati on Skills (3)	Applied Engg &Mgmt principles (3)	Life - long learning (3)	Profess ional Skills (5)	Innov ative Appr oach (5)	**Total Marks (50)**
Review of Project Stage 1	4	5	4	2	4	2	2	2	2	3	3	3	4	4	44

Comments: → work on data set first to get approriate data mostly Indian regions target

Yugchhaya. D.
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)	Environ ment Friendly (2)	Ethics (2)	Team work (2)	Presentati on Skills (3)	Applied Engg &Mgmt principles (3)	Life - long learning (3)	Profess ional Skills (5)	Innov ative Appr oach (5)	**Total Marks (50)**
Review of Project Stage 1	4	5	4	2	4	2	2	2	2	3	3	3	4	4	44

Comments: → Needs to consider contacting Expert from Agricultural Univ.
→ Prototype is expected for Next Review.

Lifna. C. S. 23/8/24
Name & Signature Reviewer2

Date: 23rd August, 2024

30

Industry/Inhouse: **Project Evaluation Sheet 2024-25** **Class: D17** A/C

Title of Project(Group no): Medleaf: AI based Identification & Medicinal Value Assessment of Flora

Group Members: Sanika Hadap (D17C/22), Kevin Patel (D17A/45), 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)	Environ ment Friendly (2)	Ethics (2)	Team work (2)	Presentati on Skills (3)	Applied Engg &Mgmt principles (3)	Life - long learning (3)	Profess ional Skills (5)	Innov ative Appr oach (5)	**Total Marks (50)**
Review of Project Stage 1	5	5	5	3	5	2	2	2	2	3	3	3	4	4	48

Comments: ⊛ Recheck the UX of the mobile App
⊛ Complete the implementation & focus on integration
⊛ Focus on the research paper as well.

Lifna. CS 26/9/24
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)	Environ ment Friendly (2)	Ethics (2)	Team work (2)	Presentati on Skills (3)	Applied Engg &Mgmt principles (3)	Life - long learning (3)	Profess ional Skills (5)	Innov ative Appr oach (5)	**Total Marks (50)**
Review of Project Stage 1	5	5	4	3	4	2	2	2	2	3	3	3	4	4	46

Comments:

(Priya R. C)
Name & Signature Reviewer2

Date: 26th September, 2024