Intelligent System for Medicinal Plant Identification and Personalized Health Recommendations

M.Kavitha
Department of CSE
Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology
Chennai, India
mkavi277@gmail.com

Karthikien
Department of CSE
Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology
Chennai, India
kkarthikien@gmail.com

Shivam Kumar
Department of CSE
Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology
Chennai, India
shivamashik888@gmail.com

Shanu Kumar
Department of CSE
Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology
Chennai, India
shanukumar61866@gmail.com

***Abstract*—In the present sectors of healthcare and herbal medicine, the integration of artificial intelligence with traditional medicinal knowledge offers a new approach to automated diagnosis and disease treatment recommendation. This research proposes a web application system for medicinal plant identification and disease treatment recommendation by using the deep learning techniques. The system consists of a MobileNetV2-based Convolutional Neural Network (CNN) for image-based medicinal leaf identification and a text-based query analysis module for disease treatment according to the symptoms. The input module consists of dual search function which allows plant identification through leaf image analysis and text-based query input for leaf name to image identification, ensuring that it has user-friendly interaction. Additionally, a prescription generation module automatically produces disease specific treatment recommendations based on patient details such as age, gender, and seasonal conditions. The system architecture is designed to use FastAPI for the backend, React for the frontend and is deployed on AWS EC2 for scaling and performance consistency. Through this experimental evaluation on various plant datasets combined with curated medical text data, this system achieves remarkable classification accuracy along with effective treatment mapping which ensures its practical usage and scalability in real-world scenarios.**

Keywords—Medicinal Leaf Identification, Disease Treatment Recommendation, MobileNetV2, Convolutional Neural Network (CNN), AWS EC2, FastAPI, React, Image Classification, Text-based Query, Web Application.

I. Introduction

The advancement of deep learning in healthcare has significantly improved the traditional medical practices providing an innovative solution for accurate diagnosis and treatment recommendations. This research study is designed to modernize traditional medicinal practices by using image-based plant classification and disease-specific treatment and prescription recommendations. The introduced system addresses the limitations of conventional methods which often rely on manual expertise, it is time consuming processes, and high dependency on specialists of that domain. So, by integrating the deep learning-based image classification and automated treatment recommendation, the system ensure precision and accessibility in healthcare diagnostics.

The core of the system relies on MobileNetV2, a lightweight and high-performance Convolutional Neural Network (CNN) model that can perform accurate plant identification from leaf images. So, the model has been chosen for its efficiency, enabling real-time classification while being computationally Applicable for deployment in a resource-constrained environment.

The model is trained on an extensively augmented dataset of medicinal leaf images, allowing it to capture diversified acceptable features under varying environmental conditions. The pre-processing stages includes resizing and normalization to ensure that the input images maintain uniformity and enhances the model's generalization capability. The trained model output shows the plant name of the leaves which is then mapped to relevant medicinal usage for disease treatment.

The backend structure is built using FastAPI which is a modern and high-performance Python web framework that facilitates seamless integration of the trained model with the user interface. This enables a rapid API development, ensuring low-latency model inference and data processing. Also, the React.js framework is employed for the frontend interface which offers an interactive and responsive user experience. The web application allows the users to either capture or upload leaf images and search for plant information by text input by facilitating multi-model user interaction. Upon image submission, the backend processes the data, the executes the deep learning model, and returns the plant leaf name along with its medicinal uses. This enables users to receive accurate and factual data on health recommendations in real-time.

The deployment has been conducted on AWS EC2 platform to ensure the scalability and accessibility of the system. As it provides dynamic resource allocation while ensuring the system can handle increased user traffic without compromising performance. Thus the integration of React.js with FastAPI over AWS EC2 would enable the real-time data flow while ensuring users to receive prompt responses based on their inputs. Furthermore, the system is also optimized to minimize the response time and facilitate the near instantaneous health recommendations. Hence pre-trained model would have a high accuracy in plant classification, providing a strong foundation for reliable disease treatment suggestions.

The proposed system has undergone through an extensive experimental evaluation to validate its effectiveness in the plant classification and treatment recommendation. So, the overall parameters such as accuracy, precision, and recall are used here would evaluate the model's efficiency, ensuring that the system delivers a high-quality and reliable health solutions. Also, there is a Grad-CAM detection feature has been integrated into system to enhance the model clarity by highlighting the key leaf regions that influences the classification. This feature could increase the transparency that would make user to trust the automated recommendations.

This proposed system would offer a transformative approach in modernizing the traditional medicinal approach by using deep learning and cloud-based deployment techniques. On integration of MobileNetV2, FastAPI, and React.js it ensures a seamless, scalable, and user-friendly platform for accurate medicinal plant identification and disease treatment suggestions. This study would benefit the healthcare industry by giving actual information by demonstrating personalized treatment suggestions with low human intervention. For future enhancements it may also incorporate wider datasets, mobile deployment, and real-time database updates to further improve system efficiency and expand its applicability across diverse healthcare settings.

II. RELATED WORKS

With the progression of deep learning have accelerated substantial developments in the area of medicinal plant identification and its usage in personalized health advice. A number of research work have proven that innovative Convolutional Neural Networks (CNNs) can precisely categorize a plant species from images even with different altering environmental conditions. For instance, Adibaru Kiflie et al. [1] demonstrated a systematic review that demonstrates the use of extensive data enhancement and transfer learning to address issues that are related with small and unbalanced datasets. Their research has set a performance baseline by showing that employing pretrained models could significantly enhance the classification accuracy and minimize the necessity of training it from scratch.

Y. Sharrab et al. [2] has also identified those light-weight architectures such as MobileNetV2 that have found an excellent trade-off between computational efficiency and accuracy which is a desirable attribute for deployment on mobile and edge devices. Also, M. S. Rao et al. [3] improved the image processing by incorporating the Log-Gabor filters into the CNNs, thus enhancing texture feature extraction, and discriminating among visually ambiguous medicinal plants. Transfer learning has also emerged as a key strategy in this research area. The Indonesian medicinal plant detection system by M. A. Rahman et al. [4] successfully fine-tuned pre-trained models on a specific domain dataset to recognize the Indonesian medicinal plants with significant performance gains. Similarly, M. Sreedevi et al. [15] have combined transfer learning techniques with advanced data augmentation techniques to solve the limitations throwed by limited datasets underscoring the utility of pre-trained networks in specialized applications.

While earlier research explored integrating natural language processing (NLP) for mapping disease queries to plant recommendations, a recent effort have increasingly focused on image-based approaches which offered superior precision in species identification. Aashish Cheruvu et al. [6] introduced multi-model recommendation system that integrates visual and structured clinical data to improve the predictivity performance in disease diagnostics. However, our system prioritizes a streamlined approach that relies solely on image-based identification using MobileNetV2 and fetching health treatment and recommendation for that disease, thus simplifying it using pipeline and reducing dependency on textual data.

Explainability remains a serious component for building trust in AI-driven healthcare systems. J. Yao et al. [8] demonstrated an effectiveness of techniques such as Grad-CAM to visualize the key regions in leaf images that contribute to classification decisions providing transparency in clinical validation. Complementary methods such as SHAP that is discussed by Kavitha et al. [9] and Theju KV, et al. [10] quantify the impact of input features on model outputs in further enhancing interpretability. These techniques ensure that the system's predictions are both accurate and explainable, joining the user confidence and facilitating model improvements.

The work Taylor M et al. represents a joint analytical study. [11] and D. Li et al. research presented in [13] demonstrates multiple CNN architectures which illustrate the balance among model complexity accuracy and computational efficiency. H. Kukreja et al. Scalable deep learning-based plant recognition advancements gained an extra dimension through [14]'s exploration which supports real-world application potential. The collective body of these studies establishes a robust foundational platform for our proposed system. The MobileNetV2-based CNN powers our framework's high-precision image classification while a unified FastAPI backend together with a responsive React frontend operates on AWS EC2 to ensure scalability. The integrated architecture system delivers swift and dependable identification of medicinal plant leaves while providing tailored treatment recommendations to enable personalized healthcare solutions.

III. Methodology

A. Dataset Collection

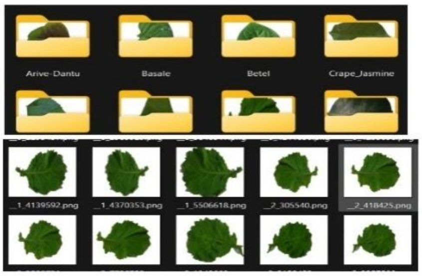

Fig. 1. Sample Leaf Dataset

The dataset comprises 15,000 medicinal leaf images alongside curated disease queries collected from multiple sources including Kaggle. TensorFlow along with Keras processes images through resizing operations and normalization techniques before applying augmentation methods. The MobileNetV2 model undergoes fine-tuning processes to perform plant classification alongside disease query mapping. Data augmentation techniques with transfer learning enhance performance as the full dataset achieves a size of about 2GB.

B. Proposed Methodology

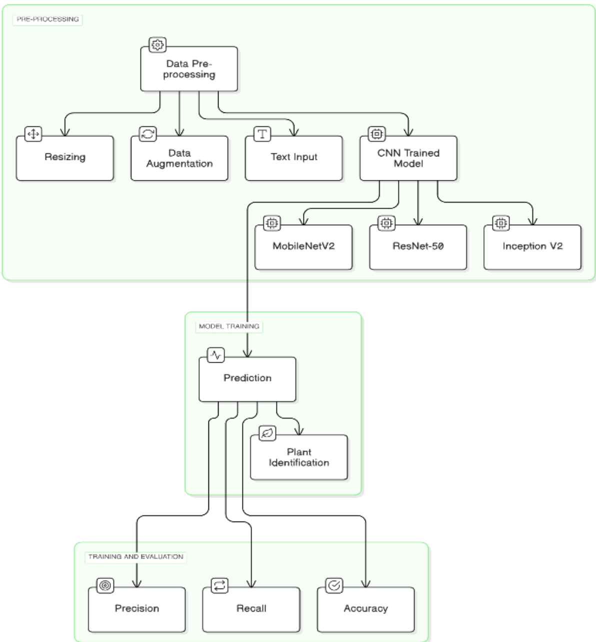

Fig. 2. Model architecture for Plant Leaf Identification

This Fig. (2) illustrates the Plant Identification and Disease Detection Recommendation System architecture diagram. Several stages collaborate in classifying plant species from leaf images and mapping disease-query-related recommendations to the recommended plants. The acquisition of raw leaf imagery and textual disease questions happens in the first phase. With pre-processing activities such as resizing, normalization, and augmentation, the images obtain the same input dimensions while diversity in the dataset increases.

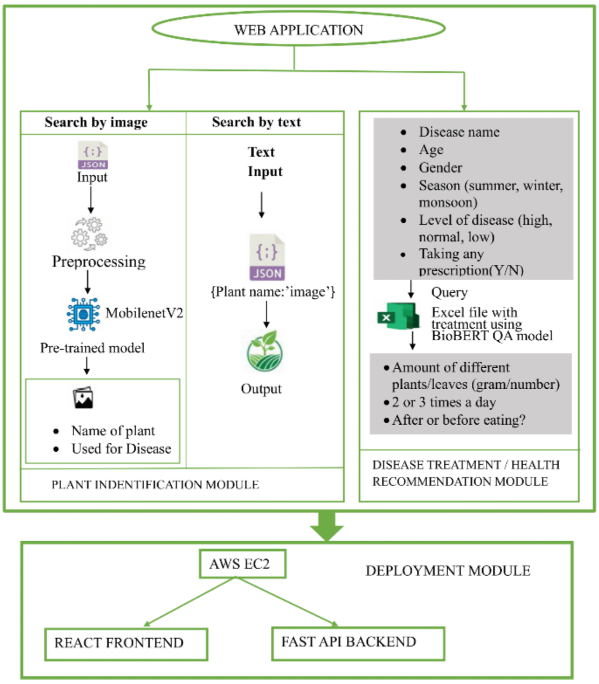

Fig. 3. System architecture for Plant Leaf Identification and Disease Detection System

The Fig. (3) shown architecture describes a holistic web-based system aimed at medicinal plant identification and disease treatment suggestions. system comprises three core modules: Plant Identification Module, Disease Treatment Module, and an AWS EC2 infrastructure-based Deployment Module. In the Plant Identification Module, the users can upload the image of the leaf or enter a text query to retrieve associated plant information. The image input is analyzed with a CNN, for example, MobileNetV2, to derive features and classify them to recognize the plant properly. The text query is analyzed with a natural language model to retrieve botanical information to match plant species. The Disease Treatment Module includes patient-specific parameters such as age and disease status to provide personalized prescription information. Dosage and treatment plans are exported using an Excel-based output for ease of access. The deployment in AWS EC2 offers low-latency, scalable access with a FastAPI backend for data communication and a React frontend with an easy-to-use interface. The integrated framework offers effective, accurate, and flexible medicinal plant identification and treatment suggestion.

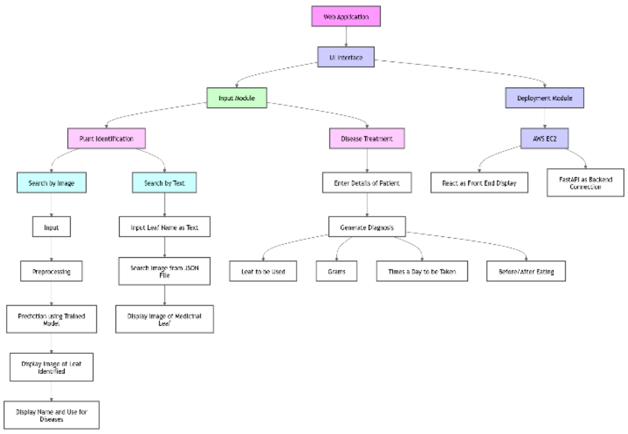

Fig. 4. Flowchart for Plant Identification and Disease Detection System

The Fig. (4) flowchart is organized around a root "Web Application" node with child nodes in the form of an "Input Module," a "UI Interface," and a "Deployment Module" on AWS EC2. Beneath the "Input Module" are "Plant Identification" and "Disease Touchpoint" sub-branches. For "Plant Identification," the user can "Search by Image" or "Search by Text," submitting an image or calling a JSON file. The system accepts these inputs, showing a view of the medicinal leaf or usage data. At the same time, the "Disease Touchpoint" provides for entering patient data, creating a diagnosis, and marking times during the day or before/after meals. The UI Interface provides for an intuitive interface, facilitating real-time interaction for both modules. AWS EC2's "Deployment Module" manages React as the front and FastAPI as the backend, providing reliable performance. Through bringing these together, the flowchart outlines a medicinal plant-based solution for health suggestions, appropriate to general-based user requirements and applications. Such synergy between data harvesting and user feedback is the merit of an amalgamated design. This integrated approach fosters synergy among data acquisition and user interaction.

IV. Experimental Findings

A. Algorithm

Input: Plant Identification or Disease Treatment

Step 1: Collect a full set of high-resolution medicinal leaf images.
-Resize each image to 224×224 pixels and normalize pixel values to the [0, 1] range.
-Apply data augmentation techniques (e.g., rotation, flipping, brightness/contrast adjustments, zooming) to enhance data diversity.

Step 2: Splitting the image dataset into train (70%), validate (15%), and test (15%) subsets to ensure balanced class distribution and reliable evaluation.

Step 3: Load MobileNetV2 pre-trained on ImageNet with include_top = False as a feature extractor.
-Process images through convolutional layers where feature maps are computed as $Z = (X * W) + b$ and apply the ReLU activation function as: $f(Z) = \max(0, Z)$.
-Apply global average pooling and attach fully connected layers to classify images into medicinal plant categories.
Compile the model using the categorical crossentropy loss and Adam optimizer.

Step 4: Train the CNN model on the training data while monitoring accuracy and loss on the validation set.
-Utilize early stopping to avert overfitting and then evaluate the final model on the test dataset.

Step 5: Develop a FastAPI backend that accepts image uploads.
- For each image, use the MobileNetV2 module to predict the plant class and map the output to a recommended treatment using a rule-based function.

Step 6: Integrate Grad-CAM to produce heatmaps highlighting salient image regions that influenced the classification.(Optional)
- Optionally, apply SHAP to quantify the contribution of key features in the recommendation mapping.(Optional)

Step 7: Deploy the complete system as a web service on AWS EC2 with a responsive React frontend for both desktop and mobile interfaces.
-Continuously monitor performance using real-world feedback and retrain models periodically to maintain high accuracy and adaptability.

Pseudocode:

```
START
# Define dataset paths and configuration parameters
image_data_dir ← "/path/to/plant_images"
train_path ← concatenate (image_data_dir, "train")
test_path ← concatenate (image_data_dir, "test")
batch_size ← 32
image_size ← (224, 224)

# Method to preprocess and load images
function_load.and.preprocess_images(folder, image_size, batch_size)
    images ← empty list
    filenames ← list of filenames in folder
    for each filename in filenames do
        img_path ← concatenate (folder, filename)
        img ← load_image_from(img_path)
        img ← resize_image(img, image_size)
        img ← normalize_image(img)
        add img to images
    end for
    return images
end function

# Load training and testing image data
train_images ← load_and_preprocess_images(train_path, image_size, batch_size)
test_images ← load_and_preprocess_images(test_path, image_size, batch_size)
```

```
26:
27: # Build MobileNetV2-based CNN for plant identification
28: base_model ← MobileNetV2(weights='imagenet', include_top=False, input_shape = (224,224,3))
29: Freeze layers in base_model
30: x ← GlobalAveragePooling2D(base_model.output)
31: x ← Dense (1024, activation='relu')(x)
32:predictions←Dense(number_of_classes, activation='softmax')(x)
33: C_model ← Model (inputs=base_model.input, outputs=predictions)
34:C_model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
35:
36: # Train current model using training dataset
37: C_model.train(train_images, train_labels, epochs=60, batch_size=batch_size)
38:
39: # Evaluate the model using test data
40:evaluation_metrics←C_model.evaluate(test_images, test_labels)
41: print ("Evaluation Metrics: ", evaluation_metrics)
42:
43: # Save the trained model to file
44: save C_model as 'trademed_mobilenet.h5'
45:
46: # Define unified backend functions for prediction and prescription generation
47: function predict_from_image(image_file): preprocess image_file, prediction ← CNN_model.predict(image_file) class_index ← argmax(prediction)
plant_name ← map class_index to plant name
  return plant_name, confidence
48:
49: function predict_from_text(text_input): clean text_input (non-case sensitive, remove symbols/ numbers) map text to corresponding plant name using predefined mapping\n return plant_name
50:
51: function generate_prescription(user_details):
# user_details: Age, Gender, Season, Disease Level, Prescription flag
 create prescription_data from user_details
  save prescription_data to Excel file
  return download link for Excel file
52: # Integrate functions in a FastAPI backend and React UI
# Endpoints include /predict-image, /predict-text, /generate-prescription
53:
54: # Deploy the FastAPI application along with the React frontend on AWS EC2
# Ensure scalability and low latency
55: EXIT
```

For Fig. (5) the MobileNetV2 model following accuracy graph is displayed in the Fig.4. Both training accuracy and validation accuracy are included, with the Y-axis ranging from 0.90 to 1.00 (90% to 100%) and the X-axis denoting the number of epochs (0 to 60). By the 60th epoch, the trained model has achieved a high training accuracy close to 98% and a validation accuracy of around 96%. These findings are perfect with prior research that highlights the effectiveness of lightweight CNN architectures such as MobileNetV2 for medicinal plant classification in resource-constrained environments [1,2]. The strong performance and close alignment between training and validation accuracies indicates the robust generalization showing the model can handle variations in leaf appearance and environmental conditions without significant overfitting.

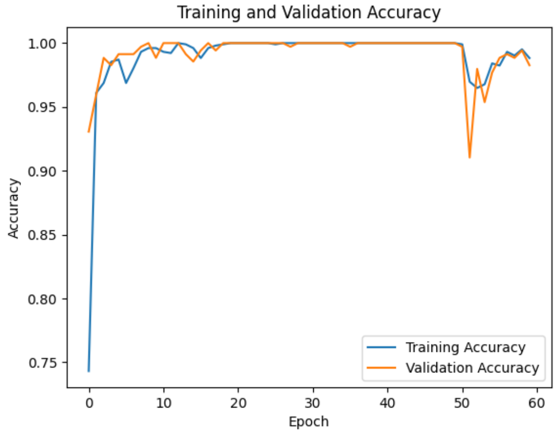

Fig 5: Training using MobileNetV2 model

The ResNet50 model accuracy graph is presented in Fig. (6) Training accuracy and validation accuracy are plotted together with the Y-axis ranging from 0.1 to 0.7 (10% to 70%) for accuracy and the X-axis for epochs (0 to 60). The model achieves a final training accuracy of approximately 70% and a validation accuracy of approximately 60% at the 60th epoch. This performance reflects that although ResNet50's deep architecture is very well-suited to extract high-end features, its overall generalization is below par—perhaps because of overfitting or data nature of the dataset. Similar behaviors are observed in studies where deeper networks require larger and more varied datasets to achieve their full potential [7]. Comparative studies also indicate that although ResNet50 is robust in feature extraction, its computational costliness and susceptibility to small training datasets can create a performance gap between training and validation accuracies[13]. These results underscore the importance of dataset diversity and proper model regularization in using ResNet50 in fields such as medicinal plant classification.

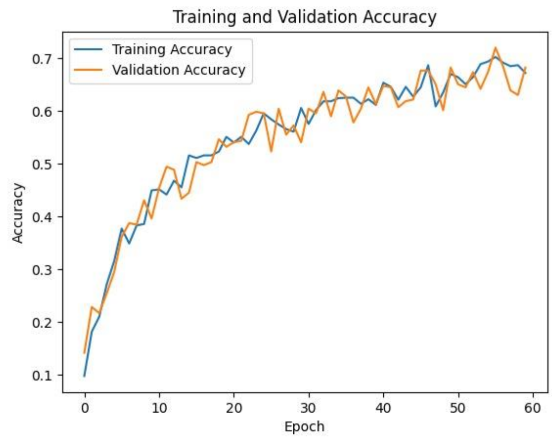

Fig 6: Model Training using ResNet50

The Fig. (7) InceptionV2 model accuracy graph is displayed in the figure. Training and validation accuracies are given. The Y-axis spans from 75% to 90%, with mid markers at 80% and 85%, and the X-axis represents epochs ranging from 10 to 60 with an interval of 10. On the 60th epoch, the model achieves a closing training accuracy of approximately 90% and a validation precision of approximately 85%. This indicates that InceptionV2 has learned consistently, with stable learning and excellent generalization to unseen samples. The small disparity between the training and validation accuracies indicates that the model learns high-resolution leaf details without excessive overfitting, due to its multi-scale feature extraction architecture. These outcomes are in line with highlighting the strength of the Inception architecture for complicated image classification problems [11, 14]. Overall, the performance of InceptionV2 in this application warrants its potential in automatic medicinal plant identification, particularly when supplemented with large scale data augmentation.

Fig 7: Model Training using InceptionV2

The Fig. (8) training accuracy and validation accuracy are compared with the bar graph of three CNN models in our medicinal plants recognition system: MobileNetV2, ResNet50 and InceptionV2. MobileNetV2 provides improved performance with 98% training accuracy and 96% validation accuracy to identify strong generalization even under changing leaf conditions. ResNet50's training accuracy is 70% and validation accuracy of 60%, i.e., its deeper model may suggest higher data enlargement or fine-tuning to overcome probable underfitting. InceptionV2 achieves end accuracies of 90% on training and 85% on validation indicating strong learning and efficient multi-scale feature learning. These are consistent with current literature which suggest advantages of light-weight structures to on-device deployment and constraints that deeper networks suffer from due to decreased data diversity. Overall, the graph indicates the vital significance of choosing a right model structure with respect to available computational resource, data set size, and desired accuracy to make the system deliver accurate and reliable medicinal plant recognition for clinical and practical deployment.

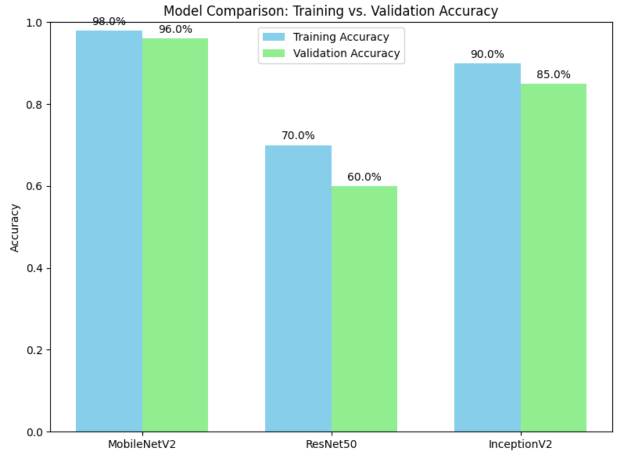

Fig 8: Comparison between MobileNetV2, ResNet50 and InceptionV2.

V. Conclusion

This research provides an integrated framework for automated identification of medicinal plants and individual health recommendations system based on state-of-the-art deep learning techniques. MobileNetV2 is utilized in image-based plant classification, the system properly classifies medicinal species from leaf images. The FastAPI backend incorporated combines the disease classification module with a rule-based disease to-plant mapping engine to deliver real-time and scalable recommendations. Also, explainability methods e.g., Grad-CAM and SHAP offer clear explanations into model decisions and thus enhancing clinical trust. Experimental tests set up high accuracy and stable generalization across broad imaging conditions. Future work will focus on enhancing dataset diversity, like other contextual variables, and calibrating recommendation logic to enhance system accuracy and flexible strategy. Overall, our strategy offers a scalable and effective solution that bridges the conventional medicinal knowledge with modern AI, remarkably revolutionizing diagnostic support and treatment regimens.

References

[1] Adibaru Kiflie Mulugeta, Durga Prasad Sharma, and Abebe Haile Mesfin, “Deep learning for medicinal plant species classification and recognition: a systematic review,” *Frontiers in Plant Science*, vol. 14, Jan. 2024, doi: https://doi.org/10.3389/fpls.2023.1286088.

[2] Y. Sharrab, D. Al-Fraihat, M. Tarawneh and A. Sharieh, "Medicinal Plants Recognition Using Deep Learning," 2023 International Conference on Multimedia Computing, Networking and Applications (MCNA), Valencia, Spain, 2023, pp. 116-122, doi: 10.1109/MCNA59361.2023.10185880.

[3] Praveen Kumar Sekharamantry, M. S. Rao, Y. Srinivas, and Archana Uriti, “PSR-LeafNet: A Deep Learning Framework for Identifying Medicinal Plant Leaves Using Support Vector Machines,” *Big Data and Cognitive Computing*, vol. 8, no. 12, pp. 176–176, Dec. 2024, doi: https://doi.org/10.3390/bdcc8120176.

[4] Muhammad, Novanto Yudistira, M. A. Rahman, A. H. Basori, A. Besse, and Jati Batoro, “IndoHerb: Indonesia Medicinal Plants Recognition using Transfer Learning and Deep Learning,” *Heliyon*, vol. 10, no. 23, pp. e40606–e40606,Nov.2024, doi:https://doi.org/10.1016/j.heliyon.2024.e40606.

[5] U. Bhimavarapu, M. Sreedevi, N. Chintalapudi, and G. Battineni, “Physical Activity Recommendation System Based on Deep Learning to Prevent Respiratory Diseases,” *Computers*, vol. 11, no. 10, p. 150, Oct. 2022, doi: https://doi.org/10.3390/computers11100150.

[6] Aashish Cheruvu, “Multimodal Recommender System in the Prediction of Disease Comorbidity,” *arXiv (Cornell University)*, pp. 79–82, Sep. 2022, doi: https://doi.org/10.1109/transai54797.2022.00020.

[7] Y. Lin, M. Li, Z. Zhu, Y. Feng, L. Xiao, and Z. Chen, “Research on Disease Prediction Model Construction Based on Computer AI Deep Learning Technology,” *2024 IEEE 2nd International Conference on Image Processing and Computer Applications (ICIPCA)*, vol. 12, pp. 1341–1345, Jun. 2024, doi: https://doi.org/10.1109/icipca61593.2024.10709274.

[8] J. Yao, S. N. Tran, S. Garg, and S. Sawyer, “Deep Learning for Plant Identification and Disease Classification from Leaf Images: Multi-prediction Approaches,” *ACM computing surveys*, vol. 56, no. 6, pp. 1–37, Feb. 2024, doi: https://doi.org/10.1145/3639816.

[9] S. Kavitha, T. Satish Kumar, E Naresh, V. H. Kalmani, Kalyan Devappa Bamane, and Piyush Kumar Pareek, “Medicinal Plant Identification in Real-Time Using Deep Learning Model,” *SN Computer Science*, vol. 5, no. 1, Dec. 2023, doi: https://doi.org/10.1007/s42979-023-02398-5.

[10] Naik P, Shraddha KB, Gowda SS, Theju KV, Ruthvika R. “HerbId–A medicinal plant identification and recommendation model using machine learning algorithms”, In AIP Conference Proceedings 2024 Jun 18 (Vol. 3122, No. 1). Jul. 2022, doi: https://doi.org/10.3390/plants11151952.

[11] Smith J, Brown R, Taylor M. Rapid Identification of Medicinal Plants via Visual Feature-Based Classification. Plant Methods. 2024;20(1):202. doi:10.1186/s13007-024-01202-6.

[12] G. D. Patil, P. R. Gumble, and A. I. Rokade, “Analytical Comparison of Medicinal Plant Species Identification Using Artificial Intelligence,” pp. 1–6, Nov. 2024, doi: https://doi.org/10.1109/delcon64804.2024.10867055.

[13] D. Li *et al.*, “Deep learning and machine intelligence: New computational modeling techniques for discovery of the combination rules and pharmacodynamic characteristics of Traditional Chinese Medicine,” *European Journal of Pharmacology*, vol. 933, pp. 175260–175260, Oct. 2022, doi: https://doi.org/10.1016/j.ejphar.2022.175260.

[14] H. Kukreja *et al.*, “Web-Based Application for Medicinal Plant Identification Using Transfer Learning Approach,” *Lecture notes in networks and systems*, pp. 167–175, Jan. 2024, doi: https://doi.org/10.1007/978-981-97-4228-8_12.

[15] Kalaiselvi P, E. C, Aburoobha A, Nishanth J, Gopika S, and Kabilan M, “Deep Learning Technique for Medicinal Plant Leaf Identification: Using Fine-Tuning of Transfer Learning Model,” pp. 1–6, Oct. 2024, doi: https://doi.org/10.1109/icpects62210.2024.10780185.