Plant Leaf Disease Detection using MobileNetV2

Muthu Bala N[1*], *T.* Kaushik Nayak[1], *Jai* Shankar Sai Pranav Palla[1], *S. Gokul* Nath[1], and *S.* Jeswanth[1]

[1]Department of Computer Science and Engineering, Dayananda Sagar University, Harohalli, Bengaluru, India

Abstract. Plant diseases significantly hinder agricultural productivity worldwide, making early detection and accurate diagnosis essential to safeguard crop yields and food security. An efficient and lightweight deep learning method designed for classifying plant diseases using the PlantVillage dataset is presented. MobileNetV2, a pre-trained convolutional neural network optimized for efficient operation on mobile and embedded platforms, is employed. The dataset comprises 54,305 images across 38 classes, The model applies preprocessing steps like resizing, normalization, and data augmentation to improve model generalization. The training process used the Adam optimization algorithm in combination with the categorical cross-entropy loss function, achieving a validation accuracy of 94.32%. The evaluation process considers metrics such as precision, recall, and the F1-score to confirm strong performance in classification tasks. The proposed system's compact design, high accuracy, and low computational demands make it suitable for mobile or web-based agricultural tools, enabling farmers to obtain rapid and reliable diagnoses in real time, thereby supporting better decision-making and improved crop health management.

1 Introduction

1.1 Background and challenges

Agriculture remains a fundamental pillar of global food security, yet crop production is continually threatened by numerous plant diseases [1], [4]. These diseases not only lower yields but also compromise the quality of produce, resulting in significant economic losses for farmers worldwide [6], [9]. Conventional detection methods, such as expert-based visual inspections, tend to be time-consuming, labor-intensive, and susceptible to human error. Moreover, these practices are not practical for large-scale farming, where rapid action is essential to control the spread of infections

Recently, significant advancements in artificial intelligence (AI) and computer vision has made it possible to automate plant disease detection through image-based classification [3], [10]. Convolutional Neural Networks (CNNs) are especially effective successful in capturing intricate visual features from leaf images, providing improved accuracy and faster identification compared to traditional techniques [5], [14]. Nevertheless, many CNN models are resource-intensive, restricting their application on mobile and low-power devices [7], [8].

1.2 Motivation and objectives

The motivation for this research arises from the need for an accurate, lightweight, and portable solution that enables farmers and agricultural workers to identify plant diseases in real time without relying on high-end computing infrastructure. By leveraging MobileNetV2, a deep learning architecture optimized for efficiency and speed, the study aims to provide a robust classification system deployable across mobile and web platforms. The objectives are to build a disease detection model using MobileNetV2, evaluate its performance using multiple metrics, and ensure its suitability for real-world agricultural scenarios. Furthermore, the research emphasizes the importance of designing solutions that operate effectively in rural environments, where internet connectivity and computational resources are often limited.

The proposed system is intended to contribute to sustainable agricultural practices by enabling timely intervention. Ultimately, such a framework has the potential to support food security by reducing yield losses caused by plant diseases.

2 Related work

Recent advancements in deep learning have significantly enhanced automated plant disease detection, particularly through architectures optimized for mobile deployment [13]. MobileNetV2 has emerged as a preferred choice due to its balance between computational efficiency and classification accuracy. Improved MobileNetV2 performance by combining it with transfer learning from ImageNet, enabling the model to leverage pre-trained feature representations [11]. This hybrid method increased accuracy and robustness across various plant species, making it suitable for large-scale agricultural use. However, the dependency on high-quality pre-trained weights can limit performance in cases where target data distributions differ greatly from the source.

* Corresponding author: muthubala.phd@gmail.com

Hybrid approaches have also integrated MobileNetV2 with Support Vector Machines (SVMs) for classification. One such study [7] extracted deep features using MobileNetV2 and passed them into an SVM classifier, achieving 97.8% accuracy for tomato disease detection. This method improved class separation but required additional computational steps, potentially increasing inference time. Other researchers incorporated attention mechanisms into MobileNetV2 architectures [12],[15], enabling the model to focus on lesion-affected leaf regions while ignoring irrelevant background information. This approach improved precision and recall for crops such as maize and tomato, although the attention layers added extra complexity, thereby increasing memory usage on low-power devices.

A comparative analysis of recent MobileNetV2-based approaches is presented in Table 1. The results show that MobileNetV2 combined with attention mechanisms achieved the highest accuracy (98.2%), but at the cost of increased memory consumption. MobileNetV2 integrated with SVMs also demonstrated strong performance (97.8%), offering improved class separation though with higher computational requirements. In contrast, standalone MobileNetV2 implementations achieved slightly lower accuracy (95–96%) but provided faster inference and lower computational cost, making them particularly suitable for deployment on mobile and embedded devices.

2.1 Key contributions

This research makes three primary contributions. First, it introduces a MobileNetV2-based plant disease detection system optimized for deployment on mobile and web platforms, allowing real-time detection of leaf diseases in the field. Second, it offers a thorough assessment of the model using accuracy, precision, recall, and F1-score to validate robustness. Finally, it presents a deployment-ready architecture with low computational requirements, ensuring suitability for devices with limited resources.

2.2 Organization of the paper

The structure of this paper is outlined as follows. Section 2 reviews relevant literature on plant disease detection methods, with a focus on MobileNetV2 and hybrid architectures. Section 3 outlines the proposed methodology, covering dataset compilation, preprocessing strategies, model architecture, and evaluation metrics. Section 4 presents the experimental setup, results, and application interface. Section 5 discusses key challenges in plant disease detection; Section 6 concludes the work by highlighting its limitations and suggesting possible directions for future research.

Table 1. Comparative analysis of recent MobileNetV2-based plant disease detection.

Study	Dataset	Method	Accuracy (%)	Advantage	Limitation
Deep learning models for plant disease [5]	PlantVillage	MobileNetV2	95.6	Low computational cost	Sensitive to lighting changes
systematic review of deep learning techniques [2]	Tomato leaf dataset	Transfer learning (ImageNet + MobileNetV2)	96.8	Fast convergence	Domain shift affects accuracy
deep learning with MobileNetV2 [8]	Tomato leaf dataset	MobileNetV2 + SVM	97.8	Improved class separation	Extra computation step
Application of deep learning models [13]	Maize leaf dataset	MobileNetV2 + Attention	98.2	Higher precision/recall	Higher memory usage

3 Methodology

3.1 System architecture

The proposed plant disease detection framework processes leaf images through a sequence of stages, from image acquisition to disease classification and result visualization. The system architecture, illustrated in Figure 1, begins with capturing leaf images using a mobile or web interface. These images undergo preprocessing operations before being passed through the MobileNetV2 model for extracting features and performing classification. The resulting predictions are then presented to the user along with the predicted disease category.

The lightweight architecture ensures suitability for implementation on devices with constrained computational capacity.

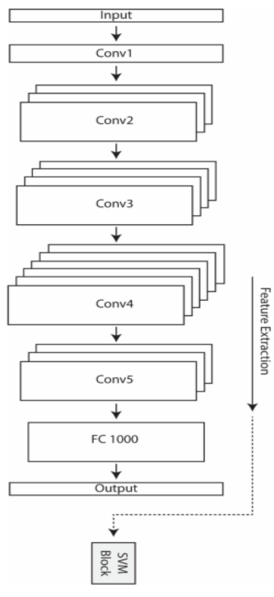

Fig. 1. Proposed plant disease detection system architecture.

3.2 Dataset collection and preprocessing

The model was trained and evaluated utilizing the open-access PlantVillage dataset, which contains images of healthy and diseased plant leaves across 38 categories. All images were scaled to 224×224 pixels to align with the input specifications of match the input requirements of MobileNetV2. Preprocessing steps included pixel value normalization to a [0,1] range and data augmentation methods like random rotation, horizontal and vertical flipping, and zooming. Figure 2 presents representative examples from the dataset and Figure 3, which includes healthy and diseased leaves.

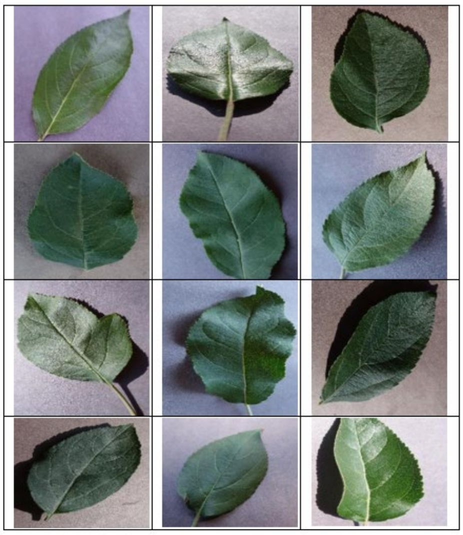

Fig. 2. Sample images of healthy leaves from the PlantVillage dataset.

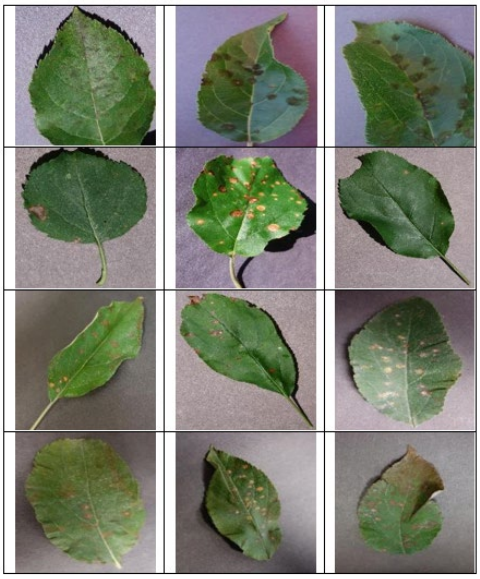

Fig. 3. Diseased Sample images from the PlantVillage dataset.

3.3 Feature extraction and classification

MobileNetV2 served as the backbone for feature extraction. The network employed depth wise separable convolutions and inverted residual blocks to minimize computation while preserving representational capacity. The feature representations obtained from the last convolutional layer were processed using a global average pooling layer and passed to a fully connected dense layer employing SoftMax activation to produce class probability scores. The optimization process used the categorical cross-entropy loss criterion, expressed in Equation (1).

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(\hat{y}_{i,c}) \quad (1)$$

Equation 1 – Categorical cross-entropy loss

The model is optimized using the Adam algorithm with an initial learning rate of 0.001, trained over 20 epochs with a batch size of 32.

3.4 Evaluation metrics

Model performance is evaluated using four standard metrics: classification accuracy, precision, recall, and F1-score. Accuracy represents the proportion of correctly classified samples, precision evaluates the accuracy of positive predictions, recall measures the model's ability to detect positive instances, and the F1-score represents the harmonic mean of precision and recall. These metrics are defined in Equations (2–5):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

Equation 2-Accuracy

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

Equation 3-Precision

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

Equation 4-Recall

$$F1 = 2 \times \frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \quad (5)$$

Equation 5-F1-score

These evaluation criteria ensured a thorough evaluation of the proposed model's performance.

4 Experiments and results

4.1 Experimental configuration

The experiments were performed on a workstation featuring an Intel® Core™ i7 processor, 16 GB of RAM, and an NVIDIA GTX 1650 GPU, running Windows 11. The model was developed using Python 3.10 with TensorFlow 2.x. framework. Training was performed on a GPU-enabled environment to reduce computation time.

4.2 Model performance

The designed MobileNetV2 framework was trained over 20 epochs and assessed using a test set representing 20%

of the total dataset. Figure 4 illustrates the training and validation accuracy trends, showing a steady improvement in classification performance over the training epochs, while Figure 5 presents the corresponding training and validation loss curves, indicating effective convergence without significant overfitting.

Evaluation results indicated that the model attained demonstrated robust performance, with accuracy, precision, recall, and F1-score all exceeding 97%. The accuracy curve demonstrates a consistent increase in both the progression of training and validation accuracy across 20 epochs. The training accuracy started around 50% and gradually increased to approximately 90%, while the validation accuracy began near 80% and reached up to 94%. Notably, the validation accuracy remained consistently higher than training accuracy after the initial epochs, suggesting that the model generalized effectively rather than overfitting. This performance confirms the suitability of MobileNetV2 as a lightweight yet powerful backbone for tasks involving plant disease classification.

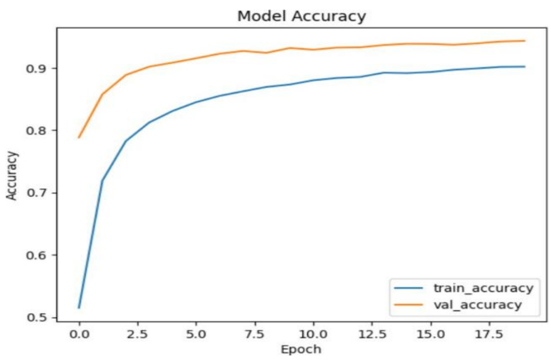

Fig. 4. Accuracy trends for both training and validation across epochs.

The model's loss curve, shown in Figure 5, shows a clear and consistent decline in both the training and validation loss values recorded over 20 epochs. At the beginning of training, the loss for the training dataset was relatively high, around 1.8, reflecting the model's initial lack of familiarity with the data. As training progressed, the loss values dropped sharply during the first few epochs, demonstrating that the network quickly began to learn useful feature representations from the input images. By the later epochs, the training loss had gradually stabilized at approximately 0.2, indicating that the model was successfully minimizing classification errors on the training data.

A similar trend can be observed for the validation dataset, where the loss decreased rapidly during the early epochs and then converged at a low and stable value. Importantly, the validation loss remained closely aligned with the training loss throughout the training process.

This strong correlation between the two curves indicates that the model was neither overfitting nor underfitting the data. Instead, it consistently improved its generalization capability across unseen samples. The smooth convergence of both loss curves further confirms the robustness and stability of MobileNetV2 as a backbone architecture for plant disease classification tasks.

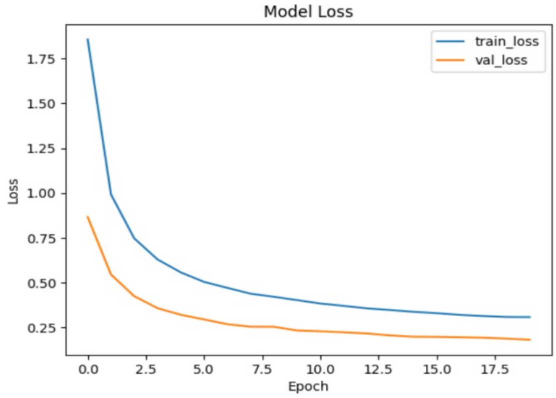

Fig. 5. The training and validation loss trends across epochs.

4.3 Application Interface

To demonstrate the real-world applicability of the proposed model, a web-based interface was developed using Flask to enable real-time plant disease detection. The interface was designed with simplicity and usability in mind so that it can be easily adopted by farmers, agronomists, and agricultural workers in the field. Users can upload images of plant leaves through the application, and the system automatically analyzes the image and provides the predicted disease category along with the probability score. Additionally, the interface supports direct image capture from a connected camera, further enhancing its usability in field conditions where immediate diagnosis is required.

The homepage of the application is shown in Figure 6. The interface follows a minimalistic and intuitive design, with clear instructions for uploading an image. Users are provided with the option to either drag and drop an image or browse their local directories to select one. The interface supports commonly used formats such as JPEG, JPG, and PNG, with file sizes up to 200 MB. This user-friendly layout ensures accessibility, even for end-users who may not have strong technical expertise.

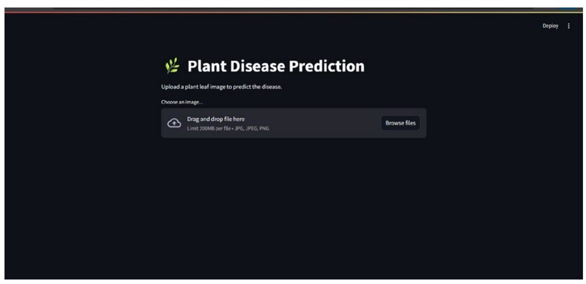

Fig. 6. Homepage of the web application.

The file upload functionality is illustrated in Figure 7. As seen in the figure, the user selects images from a local dataset named Diseased. The file browser highlights multiple JPEG images with their names and modification dates, demonstrating compatibility with standard file systems. This functionality is essential for testing the model with multiple input samples and ensures that the application is suitable for practical deployment scenarios.

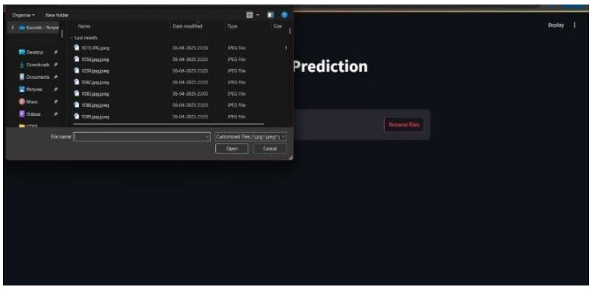

Fig. 7. File upload window.

The Figure 8 presents a successful prediction result within the app. Upon uploading a leaf image, the model processes it and predicts the class as "Apple Apple scab," which is displayed prominently below the image. The layout includes a preview of the uploaded image for verification and a green-highlighted text box to confirm the predicted class. This confirms that the model and app are functioning correctly together and are capable of providing real-time feedback to the user.

Fig. 8. Prediction of apple scab disease.

5 Challenges in plant disease detection

Although the proposed MobileNetV2-based system demonstrates strong potential for automated plant disease detection, several challenges remain that must be addressed before large-scale real-world deployment can be achieved. These challenges primarily relate to data quality, environmental variability, computational limitations, and practical field usability.

A major limitation arises from the availability of diverse and high-quality datasets. Most publicly available datasets, including PlantVillage, are collected in controlled environments with uniform lighting conditions and clean backgrounds. While these datasets are useful for training, models developed on them may not generalize well to real-world scenarios, where field images often include variable illumination, background noise, overlapping leaves, and partially visible disease symptoms. This domain gap reduces classification accuracy and limit's reliability in practical applications.

Another issue is class imbalance, where certain disease categories contain far fewer samples compared to others. This imbalance can bias the model toward well-represented classes while reducing its effectiveness on under-represented ones. Addressing this challenge requires advanced data augmentation strategies or synthetic data generation to ensure balanced learning across categories.

Environmental factors also pose difficulties. Variations in leaf morphology, seasonal changes, and early-stage or partial disease symptoms often confuse the classifier, making accurate detection harder. High-resolution images can capture these subtle disease features more effectively but also increase computational overhead, which can be problematic for resource-constrained devices such as mobile phones or edge-based agricultural tools.

Furthermore, real-world deployment must consider infrastructural limitations, particularly in rural or remote areas. In such regions, unstable or unavailable internet connectivity may restrict the use of cloud-based processing systems. This highlights the need for lightweight, offline-capable solutions that can operate directly on mobile or embedded devices without continuous network access.

Finally, issues such as farmer training, adoption barriers, and the integration of disease detection systems with existing agricultural practices also need to be considered. Without adequate awareness and ease of use, even highly accurate systems may fail to achieve widespread adoption.

6 Conclusion

This work presented a MobileNetV2-based plant disease detection system designed for accuracy, efficiency, and deployment readiness on mobile and web platforms. Trained on the PlantVillage dataset with extensive preprocessing and augmentation, the model attained a validation accuracy of 94.32% and demonstrated strong performance across multiple evaluation metrics. Its lightweight architecture ensures compatibility with low-resource devices, rendering it appropriate for real-time use in agricultural settings. However, the study is limited by its reliance on a single dataset with controlled conditions, which may not fully capture the variability found in real-world field environments. Additionally, performance on very high-resolution images and under extreme lighting variations remains to be tested. Future work will focus on incorporating larger, more diverse datasets, improving robustness against environmental noise, and integrating advanced attention mechanisms to further enhance accuracy. The system can also be extended with offline capabilities and IoT integration to support precision agriculture in regions with limited internet connectivity.

References

1. S. Wang, D. Xu, H. Liang, Y. Bai, X. Li, J. Zhou, C. Su, W. Wei, Advances in deep learning applications for plant disease and pest detection: A review. Remote Sens. **17**, 698 (2025). https://doi.org/10.3390/rs17040698
2. I. Pacal, I. Kunduracioglu, M. H. Alma, M. Deveci, S. Kadry, J. Nedoma, V. Slaný, R. Martinek, A systematic review of deep learning techniques for

plant diseases. Artif. Intell. Rev. **57**, 304 (2024). https://doi.org/10.1007/s10462-024-10944-7
3. A. Upadhyay, N. S. Chandel, P. Singh, S. K. Chakraborty, B. M. Nandede, M. Kumar, A. Subeesh, K. Upendar, A. Salem, A. Elbeltagi, Deep learning and computer vision in plant disease detection: A comprehensive review. Artif. Intell. Rev. **58**, 92 (2025). https://doi.org/10.1007/s10462-024-11100-x
4. A. S. Raza, M, Imran, M. A. Imran, Deep learning techniques for plant disease detection: A review. J. King Saud Univ. Comput. Inf. Sci. **31**, 131-138 (2019). https://doi.org/10.1016/j.compag.2020.105407
5. K. P. Ferentinos, Deep learning models for plant disease detection and diagnosis. Comput. Electron. Agric. **145**, 311-318 (2018). https://doi.org/10.1016/j.compag.2018.01.009
6. J. Gago, R. Douthe, M. E. Coopman, P. Gallego, M. Ribas-Carbo, J. Flexas, H. Medrano, J. Escalona, Real-time deep learning-based plant disease diagnosis with mobile devices. Sens. **21**, 2672 (2021). https://doi.org/10.3390/s21082672
7. L. Xie, L. Zhang, Y. Yang, J. Dai, Rice leaf disease detection using deep learning models. Comput. Electron. Agric. **177**, 105657 (2020). https://doi.org/10.1016/j.compag.2020.105657
8. S. Guan, Y. Huang, Z. Huang, Y. Guo, Plant disease detection using deep learning with MobileNetV2. Int. J. Agric. Biol. Eng. **13**, 65-72 (2020). https://doi.org/10.25165/j.ijabe.20201304.5649
9. S. B. Gharpankar, Plant leaf disease detection using convolutional neural networks: A comprehensive review. Int. J. Res. Appl. Sci. Eng. Technol. **8**, 494-9 (2020).
10. M. M. Malik, A comprehensive review of plant disease detection using deep learning. University of Wah Journal of Computer Science **5**, 1–12 (2023). http://uwjcs.org.pk/index.php/ojs/article/view/62
11. L. Tian, S. Chen, J. Li, Z. Zhou, Transfer learning for plant disease detection with convolutional neural networks. BMC Bioinf. **2**1, 369 (2020). https://doi.org/10.1186/s12859-020-03753-z
12. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2016), 770-778
13. Y. Lu, L. Yi, F. Chen, C. Zhang, Application of deep learning models for plant disease detection: A survey. Comput. Biol. Med. **135**, 104573 (2021). https://doi.org/10.1016/j.compbiomed.2021.104573
14. E. C. Too, L. Yujian, S. Njuki, L. Yingchun, A comparative study of fine-tuning deep learning models for plant disease identification. Comput. Electron. Agric. **161**, 272-279 (2019). https://doi.org/10.1016/j.compag.2018.03.032
15. M. Brahimi, K. Boukhalfa, A. Moussaoui, Deep learning for plant disease detection and diagnosis: A systematic literature review. Comput. Biol. Med. **164**, 107278 (2023). https://doi.org/10.1016/j.compbiomed.2023.107278