AI-powered Brinjal Leaf Disease Identification: Towards Smarter and Faster Crop Protection

by

Sarmin Sultana Ruma
ID: CSE2103024168

MD. Kamruzzaman
ID: CSE2103024039

Injamul Reza Piash
ID: CSE2103024025

MD. Obaydur Rahman
ID: CSE2103024030

Supervised by
Arifur Rahaman

Submitted in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science and Engineering



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)

September 2025

APPROVAL

The thesis titled **"AI-powered Brinjal Leaf Disease Identification: Towards Smarter and Faster Crop Protection"** submitted by Sarmin Sultana Ruma (CSE2103024168),MD. Kamruzzaman (CSE2103024039), Injamul Reza Piash (CSE2103024025) and MD. Obaydur Rahman (CSE2103024030) to the Department of Computer Science and Engineering, Sonargaon University (SU), has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

Board of Examiners

Arifur Rahaman
Assistant Professor
Department of Computer Science and Engineering
Sonargaon University (SU)

Supervisor

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 1

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 2

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 3

DECLARATION

We, hereby, declare that the work presented in this report is the outcome of the investigation performed by us under the supervision of **Arifur Rahaman, Assistant Professor** Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh. We reaffirm that no part of this thesis has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

(Arifur Rahaman)
Supervisor

Signature

Sarmin Sultana Ruma
ID: CSE2103024168

MD. Kamruzzaman
ID: CSE2103024039

Injamul Reza Piash
ID: CSE2103024025

MD. Obaydur Rahman
ID: CSE2103024030

ABSTRACT

The accurate detection and classification of brinjal (eggplant) leaf diseases are crucial for sustainable agriculture, as early diagnosis can prevent crop loss and improve yield quality. Traditional manual identification is often time-consuming, error-prone, and reliant on expert knowledge, underscoring the need for automated and intelligent solutions. This study presents a comparative evaluation of three pre-trained convolutional neural network models MobileNetV2, DenseNet121, and EfficientNetB4 applied to brinjal leaf disease detection using transfer learning, fine-tuning, data augmentation, and class balancing strategies. The dataset, encompassing multiple disease categories including Healthy, Insect Pest, Leaf Spot, Mosaic Virus, Small Leaf, White Mold, and Wilt, was preprocessed and augmented to ensure balanced representation. Experimental results indicate that DenseNet121 achieved 96.74% accuracy, demonstrating the benefits of dense feature connectivity, while EfficientNetB4 attained the highest accuracy of 98.57%, leveraging compound scaling and larger input resolution at the cost of increased computational demand. MobileNetV2 achieved 98.29% accuracy with only 3.5 million parameters, offering superior inference speed suitable for real-time and mobile applications. Regularization techniques such as dropout, L2 weight decay, and class weighting were integrated to mitigate overfitting and improve minority class recall, enhancing model robustness. This work highlights the trade-offs between accuracy, computational efficiency, and deployment feasibility, providing a practical framework for precision agriculture. The findings suggest MobileNetV2 as an optimal choice for field-level applications, EfficientNetB4 for high-performance scenarios, and DenseNet121 as a balanced alternative, while future work will focus on scaling datasets, mobile deployment, and integrating explainable AI to improve trust and usability.

ACKNOWLEDGMENT

At the very beginning, we would like to express my deepest gratitude to the Almighty Allah for giving us the ability and the strength to finish the task successfully within the schedule time.

We are auspicious that we had the kind association as well as supervision of **Arifur Rahaman**, Assistant Professor Department of Computer Science and Engineering, Sonargaon University whose hearted and valuable support with best concern and direction acted as necessary recourse to carry out our project.

We would like to convey our special gratitude to **Prof. Bulbul Ahamed,** Head, Department of Computer Science and Engineering, for his kind concern and precious suggetions.

We are also thankful to all our teachers during our whole education, for exposing us to the beauty of learning.

Finally, our deepest gratitude and love to my parents for their support, encouragement, and endless love.

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DL	Deep Learning
FN	False Negative
FP	False Positive
GPU	Graphics Processing Unit
IoT	Internet of Things
KNN	K-Nearest Neighbor
ML	Machine Learning
RAM	Random Access Memory
RF	Random Forest
ROC-AUC	Receiver Operating Characteristic – Area Under Curve
ROM	Read Only Memory
SVM	Support Vector Machine
TP	True Positive

TABLE OF CONTENTS

Title			Page No.
DECLARATION …….……..			iii
ABSTRACT ……...…….			iv
ACKNOWLEDGEMENT …….……			v
LIST OF ABBREVIATION …….…….			vi
CHAPTER 1			1 – 6
INTRODUCTION TO BRINJAL LEAF DISEASE			
1.1	Introduction ……..…….....		1
1.2	Why it's Important?……...…….…...		1
1.3	Objectives of Thesis work……..……....		2
1.4	Literature Review…….…….....		2
1.5	Real Life Applications……..…….……		4
1.6	How It Works? ……..		4
1.7	Contribution of Thesis Work ……		5
	1.7.1	Dataset Development ……..	5
	1.7.2	Enhanced Learning Strategy(Data Augmentation & Transfer Learning) …….	5
	1.7.3	Lightweight Architecture Evaluation ……..	6
	1.7.4	Comprehensive Performance Analysis ……...	6
	1.7.5	Farmer-Centric AI Solution ……	6
CHAPTER 2			7 – 10
FOUNDATIONS OF ARTIFICIAL INTELLIGENCE IN AGRICULTURE			
2.1	Introduction …….….....		7
2.2	Application of Artificial Intelligence ……..…….…….		9
2.3	Challenges of Artificial Intelligence …….		9
CHAPTER 3			11 – 15
MACHINE LEARNING APPROACHES FOR PLANT DISEASE DETECTION			
3.1	Introduction…….....		11

3.2 Application of Machine Learning ……. 12
3.2.1 Computer Vision for Disease Detection…….…..... 13
3.2.2 Predictive Analytics and Future Predictions ……... 13
3.2.3 Time-Series and Forcasting Models …… 13
3.2.4 Multi-Class Disease Classification ……. 14
3.2.5 Mobile and Web-Based Decision Support Systems …… 14
3.3 Challenges of Machine Learning …….. 14
3.3.1 Data Availability and Quality…….…….…..... 14
3.3.2 Class Imbalance …….…….……..... 14
3.3.3 Computational Requirements …….. 15
3.3.4 Model Generalization and Robustness …… 15
3.3.5 Interpretability and Trust ……. 15

CHAPTER 4 16 – 25

DEEP LEARNING ARCHITECTURES FOR BRINJAL LEAF DISEASE CLASSIFICATION

4.1 Introduction …….……. 16
4.2 Workflow of the Project ……... 16
4.3 How Deep Learning Works in This Project? …… 17
4.4 Convolutional Neural Network ……. 17
4.5 MobileNetV2 …… 18
4.5.1 Key Innovations in MobileNetV2: ……..…… 18
4.5.2 Why MobileNetV2 for Brinjal Leaf Disease Detection? …… 19
4.6 DenseNet121 20
4.6.1 Key Features of DenseNet121 …… 20
4.6.2 Why DenseNet121 for Brinjal Leaf Disease Detection? …… 21
4.7 EfficientNetB4 …….. 21
4.7.1 Key Features of EfficientNetB4 …….. 22
4.7.2 Why EfficientNetB4 for Brinjal Leaf Disease Detection? ….. 23
4.8 Deep learning applications ……... 23
4.9 Deep learning challenges …….. 24

4.10 Benefits of This Approach …… 25

CHAPTER 5 26 – 36

METHODOLOGY

5.1 Introduction ……..…….. 26

5.2 Data Collection ……. 27

5.2.1 Public Dataset Utilization ……... 27

5.2.2 Custom Field Image Collection …….. 28

5.2.3 Dataset Size and Structure …….. 28

5.3 Description of Brinjal Leaf Diseases …… 29

5.4 Data Preprocessing ……... 30

5.5 Model Training and Evaluation …… 30

5.5.1 Model Training …… 30

5.5.2 Transfer Learning Approach ……... 33

5.5.3 System Architecture for Brinjal Leaf Disease Detection Using Deep Learning Models ……. 33

5.5.4 Model Evaluation …… 34

5.6 Real-Time Detection ……. 35

CHAPTER 6 37 – 52

RESULT ANALYSIS AND DISCUSSIONS

6.1 Introduction……..….. 37

6.2 Experiments of All Models ……... 37

6.2.1 Importing Libraries ……. 37

6.2.2 Dataset Preparation ……. 37

6.2.3 Data Augmentation ……. 38

6.2.4 Model Construction ……. 38

6.2.5 Model Compilation ……. 38

6.2.6 Training & Evaluation ……. 39

6.3 Experiments with Pre-trained Models …….. 39

6.3.1 MobileNetV2-based Pre-trained Model …….. 39

6.3.2 DenseNet121-based Pre-trained Model …….. 40

6.3.3 EfficientNetB4-based Pre-trained Model …… 41
6.4 Analysis of Results for Different Combinations …….. 43
6.4.1 Base Model Fine-tuning …….. 43
6.4.2 Batch Size & Epochs ……... 43
6.4.3 Regularization Techniques …….. 46
6.4.4 Plots (Training Loss & Accuracy, Confusion Matrices) ……. 46
6.4.5 Comparison ……. 48
6.4.6 Real-time Performance Evaluation ……. 51
6.5 Discussion ……. 51

CHAPTER 7 53 – 53

CONCLUSION AND FUTURE WORKS

7.1 Conclusion …… 53
7.2 Future Work …….. 53

REFERENCES ……..……... 54 – 56

LIST OF TABLES

Table No.	Title	Page No.
Table 4.1	Benefits of This Model	25
Table 5.1	Image of Brinjal leaf disease	27
Table 5.2	Classification of Dataset	28
Table 5.3	Details of brinjal leaf database	28
Table 5.4	Sequential Model	33
Table 6.1	Comparison Table for Three Model	49

LIST OF FIGURES

Figure No.	Title	Page No.
Fig 4.1	Model Architecture for Convolutional Neural Network (CNN)	18
Fig 5.1	Training Model Accuracy for MobileNetV2	30
Fig 5.2	Training Model Accuracy for DenseNet121	31
Fig 5.3	Training Model Accuracy for EfficientNetB4	32
Fig 5.4	System Architecture for Disease Detection	34
Fig 6.1	Training and Loss Accuracy For MobileNetV2	40
Fig 6.2	Training Loss For MobileNetV2	40
Fig 6.3	Training and Loss Accuracy For DenseNet121	41
Fig 6.4	Training Loss For DenseNet121	41
Fig 6.5	Training and Loss Accuracy For EfficientNetB4	42
Fig 6.6	Training Loss For EfficientNetB4	42
Fig 6.7	Training Model Epoch for MobileNetV2	44
Fig 6.8	Training Model Epoch for DenseNet121	44
Fig 6.9	Training Model Epoch for EfficientNetB4	45
Fig 6.10	Confusion Matrix for MobileNetV2	46
Fig 6.11	Confusion Matrix for DenseNet121	47
Fig 6.12	Confusion Matrix for EfficientNetB4	48
Fig 6.13	Three Model Training Accuracy	50
Fig 6.14	Three Model Validation Loss	50

CHAPTER 1
INTRODUCTION TO BRINJAL LEAF DISEASE

1.1Introduction

Brinjal (*Solanum melongena*), commonly known as eggplant, is a widely cultivated crop across South Asia, serving as a key contributor to food security, nutrition, and rural livelihoods. Despite its economic significance, brinjal cultivation is highly susceptible to multiple biotic stresses, including bacterial wilt, leaf curl virus, powdery mildew, and shoot-and-fruit borer infestations. These pests and diseases not only reduce yield but also compromise fruit quality, causing severe economic losses for smallholder farmers. Traditional disease management techniques primarily rely on manual visual inspection, which is labor-intensive, subjective, and often unreliable for detecting early-stage infections. Such delays in diagnosis frequently lead to excessive pesticide use, increased production costs, and environmental hazards.

Recent advancements in computer vision and deep learning (DL) have revolutionized plant disease diagnostics, enabling automated and accurate disease recognition using large-scale image datasets. Convolutional Neural Networks (CNNs), in particular, have shown exceptional ability to extract discriminative spatial and textural features from raw images, eliminating the need for handcrafted features. These models can capture subtle variations in leaf structure, color, and morphology, allowing early detection of symptoms that may be imperceptible to human observers. Additionally, lightweight CNN architectures have been developed to enable on-device inference for real-time agricultural applications.

However, deploying CNN-based models in real-world field conditions remains challenging due to uncontrolled variables such as lighting changes, occlusion, overlapping leaves, and environmental background noise. To address these challenges, recent research has explored attention mechanisms, multi-modal data fusion, and domain adaptation strategies to improve robustness and generalization. Building upon these developments, this study proposes a hybrid attention-aware CNN architecture that integrates RGB and spectral imagery for accurate brinjal disease detection under field conditions. A newly curated dataset containing diverse cultivars, disease stages, and environmental variations is used to train and validate the model. This work aims to bridge the gap between laboratory research and field deployment, offering a scalable and practical solution for precision agriculture and smallholder farmers.

1.2 Why it's Important?

Accurate and early identification of plant diseases is essential for ensuring sustainable agriculture and minimizing economic losses. In recent years, the integration of deep learning algorithms into plant pathology has brought a revolutionary shift to agriculture, offering faster and more precise disease detection while reducing the need for extensive expert intervention. These advanced technologies enable the development of early warning systems that help prevent the rapid spread of infections, safeguarding crops before significant damage occurs. Moreover, their scalability

allows seamless integration into smartphones, drones, and Internet of Things (IoT) devices, making modern disease monitoring more accessible and efficient.

For brinjal farmers in particular, implementing a robust disease detection system can significantly boost crop productivity, minimize the excessive use of pesticides, and improve overall food quality. This is especially vital in developing countries, where smallholder farming systems dominate and resources are often limited. By equipping farmers with reliable, AI-driven solutions, agriculture can become both more productive and more sustainable.

1.3 Objectives of Thesis Work

This research aims to develop an automated brinjal leaf disease classification system using image processing and deep learning techniques. The objectives are as follows:

- To build a large and diverse brinjal disease dataset from multiple environmental conditions.
- To apply image preprocessing techniques such as resizing, normalization, and augmentation for improved model training.
- To design and evaluate state-of-the-art CNN architectures (MobileNetV2, InceptionV3) for disease classification.
- To achieve high accuracy and robustness in real-world agricultural environments.
- To create a framework that can be deployed on low-power edge devices for farmers' use.

1.4 Literature Review

Brinjal (*Solanum melongena*), an essential horticultural crop across South Asia, is frequently afflicted by bacterial wilt, leaf curl virus, powdery mildew, and fruit & shoot borer infestations. These diseases impair not only yield but also quality, imposing economic hardship on smallholder farmers. Traditional detection relying on manual field inspection is time-consuming, subjective, and inefficient. The advent of deep learning, particularly Convolutional Neural Networks (CNNs), has revolutionized plant disease diagnostics by enabling automatic extraction of salient features from raw images, reducing dependency on hand-crafted cues. CNNs facilitate hierarchical feature learning, allowing models to recognize subtle disease patterns, texture irregularities, and lesion morphologies often missed during manual inspection. Additionally, deep learning approaches can scale to large datasets, providing a foundation for automated, high-throughput plant health monitoring systems. Data augmentation techniques such as rotation, flipping, scaling, color jittering, and random cropping are frequently applied to increase dataset diversity, improve model generalization, and mimic field variability. Transfer learning from large pre-trained backbones like ResNet, EfficientNet, or DenseNet has also been shown to accelerate convergence and improve accuracy on small to medium-sized brinjal datasets.

Early brinjal-specific CNN studies demonstrated promising results under controlled settings. For example, a PLOS ONE study fused multiscale shearlet features with DCNN and achieved a mean accuracy of 93.3%, marking a milestone in brinjal leaf classification [1]. Chelladurai and Sujatha

improved this by ensembling DenseNet variants to reach 94.4% on a diverse, seven-class brinjal leaf dataset, thereby underlining the value of architectural synergy for complex class structures [2]. While these studies showed high performance in laboratory datasets, their generalization to field conditions remained limited due to uniform lighting, minimal background noise, and controlled leaf orientation, highlighting the need for more realistic datasets and adaptive architectures. Augmented training and transfer learning were not fully exploited in early works, restricting the models' adaptability to unseen environmental variability.

Despite these advances, replicating such performance in real-world, mobile-friendly systems remains challenging. Efforts using multispectral fusion have yielded only modest precision (~86–82%) at early disease stages, underscoring the difficulty of pre-symptomatic detection in brinjal [3]. Real-time detector-based systems, such as an enhanced YOLOv8 model, reported mAP@0.5 of 79.4% and F1 of 75.7%, indicating persistent limitations in small-lesion detection, varying lighting conditions, and environmental variability [4]. These results emphasize that factors such as shadows, leaf overlap, and diverse field backgrounds significantly impact detection accuracy, necessitating robust model designs that can adapt to heterogeneous conditions. Lightweight CNN architectures like MobileNetV2, EfficientNetB4 and DenseNet121 have been suggested for on-device deployment, balancing accuracy with computational efficiency, but these require careful hyperparameter tuning and optimized augmentation strategies to perform well under field conditions.

To bolster robustness, recent works have embraced multi-modality and auxiliary data. For instance, a 2025 Plant Methods framework fused imagery with agronomic metadata and achieved an overall accuracy of 92% for brinjal disease recognition in unconstrained environments [5]. PlantCareNet a unified model for disease detection across multiple species delivered ~85% accuracy on brinjal subsets, demonstrating the advantages of cross-domain learning and model generalization [6]. Complementary studies incorporating spectral reflectance and chlorophyll fluorescence combined with CNNs achieved ~84.5% cross-varietal accuracy, showing promise for detecting diseases across diverse brinjal cultivars [7]. Classical machine learning methods remain relevant: spectral-texture SVM models reported 92.4% accuracy on brinjal leaf spot detection and were lightweight enough for low-resource field deployment [8]. These findings collectively suggest that integrating multiple data modalities, leveraging cross-species knowledge, and balancing computational efficiency are key to field-ready disease detection. Advanced data augmentation, domain adaptation techniques, and ensembling of multiple CNN architectures further improve generalization across diverse field conditions.

Insights from related Solanaceae crops also inform brinjal research. A structure-aware YOLOv8 variant improved detection in tomato and potato under complex outdoor conditions, illustrating that attention to feature structure and background adaptation enhances field resilience [9]. Similarly, global local fused transformer-based detectors outperformed conventional backbones on tomato lesions, providing architectural strategies transferable to brinjal [10]. Systematic

reviews of DL-based plant disease detection emphasize the importance of diversified datasets, standardized evaluation metrics such as mAP and F1 scores, and field-level validation, warning against over-reliance on high lab-set accuracies [11], [12]. These insights highlight the critical role of attention mechanisms, data augmentation, multi-scale feature extraction, and robust architectures in improving real-world model performance.

Two critical enablers emerging in 2025 aim to bridge lab-to-field performance gaps. First, the release of a high-quality, multi-condition brinjal image dataset allows for consistent benchmarking and robust validation under realistic environmental conditions [13]. Second, broad-scope pipelines integrating multiple vegetables, including brinjal, demonstrate that moderate (~90%) accuracy can be achieved with conservative augmentation and compact CNN architectures, suggesting feasible paths for mobile deployment [14]. Additionally, studies on chili disease detection (~92% field-level accuracy) reinforce the practicality of lightweight, field-ready models for Solanaceae crops [15]. These developments collectively indicate that the combination of realistic datasets, efficient CNN models, attention-augmented backbones, transfer learning, and multi-crop pipelines can enable scalable and reliable disease monitoring for smallholder farmers.

1.5 Real Life Applications

The use of deep learning, particularly Convolutional Neural Networks (CNNs), for brinjal disease detection has significant real-world applications that can transform agricultural practices. In practice, farmers can utilize smartphone applications powered by CNN models to quickly identify diseases in brinjal plants by simply taking a picture of the affected leaf or fruit. This instant diagnostic capability allows for timely intervention, reducing crop losses and preventing the spread of infections.

Beyond individual farms, such systems can be integrated with Internet of Things (IoT) devices, such as automated irrigation and pesticide spraying systems, to optimize resource usage and reduce chemical overuse. Agricultural drones equipped with cameras and CNN-based detection algorithms can monitor large fields efficiently, identifying early signs of disease outbreaks that may not be visible to the human eye.

Furthermore, the aggregated data from these detection systems can support large-scale agricultural planning and policymaking. Governments and agricultural organizations can map disease prevalence across regions, enabling targeted support, improved resource allocation, and proactive management strategies. In developing countries with smallholder farms, these technologies provide accessible, cost-effective, and scalable solutions, enhancing productivity, food security, and the overall sustainability of brinjal cultivation.

1.6 How It Works?

The disease detection system operates through a structured computational workflow designed for accuracy and efficiency. The process begins with image acquisition, where brinjal leaf images are captured using smartphones or digital cameras, providing a flexible and accessible means for data

collection. Next, the images undergo preprocessing, which includes resizing, normalization, and augmentation techniques to improve model generalization and robustness.

Following this, feature extraction is performed by the convolutional layers of the CNN, which automatically identify and capture disease-relevant characteristics such as texture, shape, and color patterns. In the classification stage, the extracted features are processed through fully connected layers and a softmax activation function to categorize the leaves as either healthy or diseased. Finally, during evaluation and deployment, the system's performance is rigorously assessed using metrics such as accuracy, precision, recall, and F1-score, ensuring that the model is optimized for practical, real-world application in brinjal cultivation.

1.7 Contribution of Thesis Work

1.7.1 Dataset Development

A robust dataset is the foundation of any deep learning-based plant disease detection system. In this work, a novel brinjal disease image dataset has been developed, containing images of both healthy and diseased brinjal leaves collected under various environmental and lighting conditions. This diversity ensures that the model learns to generalize well to real-world scenarios rather than overfitting to a single controlled setting. Images were captured using consumer-grade devices such as smartphones and digital cameras, making the dataset more representative of what farmers and field workers would actually capture. The dataset includes multiple disease categories (e.g., bacterial, fungal, and viral infections) and varying levels of disease severity, allowing the model to perform fine-grained classification rather than binary healthy/diseased predictions. The dataset's diversity and labeling accuracy directly contribute to the overall performance and reliability of the proposed system.

1.7.2 Enhanced Learning Strategy (Data Augmentation and Transfer Learning)

Deep learning models often require large datasets to achieve high performance, but collecting vast numbers of annotated agricultural images is challenging. To overcome this, data augmentation techniques such as random rotations, scaling, flipping, brightness adjustments, and noise addition were applied. These methods artificially expand the dataset and simulate real-world variability, reducing overfitting and improving the model's robustness.

Additionally, transfer learning is employed, leveraging pre-trained CNN models (e.g.,MobileNetV2, DenseNet121 and EfficientNetB4) trained on large-scale image datasets like ImageNet. By fine-tuning these models on the brinjal dataset, we utilize their learned feature extraction capabilities, dramatically reducing training time while improving classification accuracy. This combined approach ensures high performance even when working with moderate dataset sizes, making the framework efficient and cost-effective.

1.7.3 Lightweight Architecture Evaluation

Deep learning models are often computationally expensive, which limits their use in rural or resource-constrained settings. This study evaluates lightweight deep learning architectures optimized for mobile and edge devices, ensuring that farmers can access disease detection tools without expensive hardware. Networks such as MobileNetV2, EfficientNetB4 and DenseNet121 are explored, as they offer a good balance between accuracy and computational efficiency.

These models reduce the number of parameters and optimize memory and energy consumption, allowing disease detection systems to run directly on smartphones, drones, or IoT-enabled sensors. This makes the technology highly scalable, enabling large-scale adoption in both smallholder and commercial farming operations.

1.7.4 Comprehensive Performance Analysis

To validate the system's effectiveness, a thorough performance evaluation is conducted using standard machine learning and deep learning metrics. The confusion matrix provides insight into the model's classification performance across each disease class, helping to identify specific areas for improvement. Metrics such as accuracy measure the overall performance, while precision, recall, and F1-score provide a deeper understanding of how well the system performs in detecting individual disease categories.

Additionally, accuracy and loss curves are analyzed to evaluate the model's learning behavior during training and testing phases, ensuring that the model converges effectively and avoids underfitting or overfitting. This level of detailed analysis ensures that the proposed system is reliable and ready for deployment in real agricultural environments.

1.7.5 Farmer-Centric AI Solution

The ultimate goal of this research is to create a practical, farmer-friendly precision agriculture tool. By combining deep learning innovations with lightweight, scalable architectures, the system empowers farmers to detect diseases early, reduce pesticide overuse, and improve overall crop yield and quality. This approach is particularly beneficial for smallholder farmers in developing regions, where access to agricultural experts and advanced diagnostic tools is often limited.

Furthermore, the system's ability to integrate with IoT devices and agricultural drones makes it suitable for large-scale disease surveillance, providing valuable insights for policymakers and agricultural extension services. Thus, this framework represents a step toward sustainable farming practices, bridging the gap between cutting-edge AI research and field-level agricultural challenges.

CHAPTER 2
FOUNDATIONS OF ARTIFICIAL INTELLIGENCE IN AGRICULTURE

2.1 Introduction

Artificial Intelligence (AI) is one of the most advanced fields in computer science that focuses on creating machines and systems capable of performing tasks which usually require human intelligence. It involves giving computers the ability to learn from data, analyze patterns, make decisions, and solve complex problems [30]. In simple terms, AI allows machines to think and act intelligently, much like a human would, but with the advantage of speed and accuracy. The most common branches of AI include machine learning, where systems learn from data and improve over time; natural language processing, which helps computers understand and communicate in human languages; and computer vision, which enables machines to recognize and interpret visual information such as images and videos [31].

The use of AI is rapidly growing in almost every field, and agriculture is no exception [32]. Traditionally, agriculture has relied heavily on human labor and observation. Farmers needed to check their fields manually, detect diseases by looking at plants, and decide on the use of fertilizers or pesticides based on their own experience. This method often led to inefficiency, delayed identification of problems, and significant crop losses. With the help of AI, however, agriculture is undergoing a revolutionary transformation. AI systems can now assist farmers in making informed decisions by analyzing large amounts of data from the soil, weather conditions, and crop health. This allows for more precise farming techniques that save time, reduce costs, and improve productivity [33].

One of the biggest impacts of AI in agriculture is in the field of disease detection and prevention. Diseases in crops spread quickly and, if not detected in time, can destroy an entire harvest. Manual inspection of each plant in large farmlands is not only time-consuming but also prone to human error. AI systems, on the other hand, use technologies such as computer vision and machine learning to detect crop diseases early and accurately [34]. Farmers can now use their smartphones, drones, or automated machines to capture images of crops. These images are then analyzed by AI models that have been trained to recognize the symptoms of various diseases. This process makes it possible to identify the exact problem in its early stages and recommend appropriate solutions before the disease spreads further.

A specific example of AI application in agriculture is brinjal leaf disease detection. Brinjal, also known as eggplant, is an important vegetable crop grown in many countries. However, brinjal plants are highly vulnerable to diseases such as bacterial wilt, fungal infections, and leaf spot diseases [35]. These diseases, if not detected early, can severely reduce crop yield and affect farmers' income. AI offers an effective solution for this problem through the use of image recognition and deep learning techniques. The process begins with the collection of images of both healthy and diseased brinjal leaves. These images are then used to train a machine learning model,

often based on deep learning methods like Convolutional Neural Networks (CNNs), which are highly efficient at analyzing visual data [36][38].

Once trained, the AI system can accurately identify whether a brinjal leaf is healthy or diseased just by analyzing a picture of it. Farmers can use a smartphone application to take photos of the leaves in their fields, and within seconds, the AI tool can provide a diagnosis. Not only does the system detect whether the leaf is infected, but it can also identify the type of disease. Furthermore, advanced AI systems are capable of giving farmers direct recommendations for treatment, such as the type of pesticide to use, or even organic and biological control methods [37].

The impact of this technology on brinjal farming is significant. First, it enables early detection of diseases, which is critical because diseases spread very fast in crops. Early detection means that the farmer can take action before the problem becomes widespread. Second, it improves crop yield by reducing losses due to undiagnosed diseases. Third, AI-based detection helps in cost efficiency because farmers no longer need to use pesticides randomly as a preventive measure. Instead, they can apply treatments only when necessary, reducing both expenditure and harmful chemical use. Finally, it empowers farmers, even those with limited knowledge of plant diseases, to make informed decisions and protect their crops effectively.

In conclusion, Artificial Intelligence is playing a transformative role in agriculture by making farming more scientific, efficient, and profitable. Its applications in disease detection, particularly in brinjal leaf disease detection, demonstrate how powerful AI can be in solving real-world agricultural problems. By enabling early identification of plant diseases and providing accurate treatment recommendations, AI not only reduces crop losses but also supports sustainable farming practices. The integration of AI in agriculture ensures that farmers are better equipped to meet the challenges of modern farming and contribute to food security in the future.

Here some of the potential benefits of AI:

- **Early and Accurate Disease Detection:** AI can identify brinjal leaf diseases at an early stage, sometimes even before visible symptoms are noticeable to the human eye. This helps farmers take timely action to prevent the disease from spreading across the entire field.
- **Improved Crop Yield and Quality:** By minimizing losses from disease outbreaks, AI ensures that farmers can harvest healthier and higher-quality brinjals. This directly increases yield and improves the market value of the produce.
- **Cost Efficiency in Farming:** Farmers often use pesticides preventively, which wastes resources. AI systems reduce unnecessary pesticide use by pinpointing exactly when and where treatment is needed. This saves money and lowers farming costs.
- **Environmentally Sustainable Practices:** Since AI promotes precision farming and reduces chemical usage, it contributes to eco-friendly agriculture. Less pesticide use means reduced soil, water, and air pollution.
- **Farmer Empowerment and Accessibility:** Even farmers with limited technical knowledge can use smartphone-based AI apps for disease detection. This democratizes expert knowledge and empowers small-scale farmers to make informed decisions.

- **Data-Driven Decision Making:** AI systems not only detect diseases but can also collect and analyze large amounts of data about crop health, weather, and soil. This data helps in planning future farming activities more strategically.
- **Scalability for Large Farms:** On large agricultural lands, manual disease monitoring is nearly impossible. AI-powered drones and imaging systems can scan entire fields efficiently, providing comprehensive health reports for crops.

AI in brinjal leaf disease detection leads to healthier crops, reduced costs, sustainable farming, and improved decision-making for farmers.

2.2 Applications of Artificial Intelligence

One of the most important applications of AI in agriculture is plant disease detection. In the case of brinjal farming, AI models can be trained using thousands of images of healthy and diseased leaves. By learning the patterns of diseases such as bacterial wilt, leaf spot, or fungal infections, the AI system can classify new leaf images with high accuracy. Farmers can capture photos using smartphones or drones, and the AI-powered application instantly provides a diagnosis along with recommended treatments. This saves both time and resources while ensuring that diseases are controlled before they spread further.

Another application is in precision farming, where AI systems optimize the use of fertilizers, pesticides, and irrigation. For example, once a brinjal disease is detected, the AI system can suggest exactly how much pesticide is needed and at which part of the field it should be applied. This not only prevents overuse of chemicals but also helps reduce costs and minimize environmental harm.

AI also plays a role in crop monitoring and yield prediction. By integrating data from weather forecasts, soil conditions, and plant health, AI can predict potential risks to crops and estimate harvest outcomes. For brinjal farmers, this means better planning of cultivation cycles, resource allocation, and market supply. Moreover, AI-powered drones and robots can automate repetitive tasks like spraying pesticides or monitoring plant growth, reducing the dependency on manual labor.

The applications of AI in agriculture, particularly in brinjal leaf disease detection, are highly promising. AI helps in early disease diagnosis, precision pesticide usage, crop monitoring, and yield prediction, making farming more efficient and sustainable.

2.3 Challenges of Artificial Intelligence

Despite its vast potential, AI in agriculture faces several challenges. The first major challenge is data availability and quality. AI systems require large amounts of high-quality data for training. In many rural areas, collecting such datasets for brinjal leaf diseases can be difficult due to limited resources and lack of standardization. Poor-quality images or incomplete datasets can reduce the accuracy of AI models.

Another challenge is cost and accessibility. Advanced AI technologies such as drones, sensors, and automated machines are often expensive, making them unaffordable for small-scale farmers. Although smartphone-based AI apps are more accessible, the lack of reliable internet connectivity in remote agricultural regions often limits their usage.

A further obstacle is technical knowledge and awareness. Many farmers may not be familiar with AI tools or may lack the training required to use them effectively. Without proper education and support, they might hesitate to adopt new technologies, even if these tools can significantly improve their productivity.

Additionally, there are infrastructure challenges, such as poor internet connectivity, lack of electricity in remote areas, and insufficient government support for digital farming initiatives. These limitations slow down the widespread adoption of AI in agriculture.

Lastly, there is the issue of trust and reliability. Farmers often prefer traditional methods based on experience and may doubt whether AI systems can give accurate results. Building trust in AI-based farming solutions requires consistent accuracy, transparency, and strong extension support to guide farmers.

However, challenges such as data availability, high costs, lack of infrastructure, and limited farmer awareness need to be addressed to make AI solutions more practical and accessible. If these challenges are overcome, AI has the potential to transform agriculture into a more scientific, productive, and eco-friendly sector, ultimately benefiting farmers and ensuring food security.

CHAPTER 3
MACHINE LEARNING APPROACHES FOR PLANT DISEASE DETECTION

3.1 Introduction

Machine Learning (ML) is a vital branch of Artificial Intelligence that enables systems to automatically learn patterns from data and make predictions or classifications without being explicitly programmed. In agriculture, ML is increasingly being used to solve complex problems such as disease detection, yield prediction, pest management, and resource optimization [25]. By analyzing large amounts of crop-related data, ML empowers farmers to adopt data-driven decisions, thereby transforming traditional farming into more efficient and sustainable practices.

Brinjal (eggplant) is a highly cultivated vegetable crop across many regions, but its productivity is often reduced by various leaf diseases such as bacterial wilt, mosaic virus, leaf spot, white mold, insect pest damage, and little leaf disease. Traditionally, identifying these diseases requires expert knowledge and manual inspection, which can be slow, error-prone, and impractical for large farms. Machine learning provides an effective alternative by automating the process of disease recognition using leaf images [26].

The workflow of ML-based brinjal leaf disease detection typically begins with image data collection, where healthy and diseased leaf samples are gathered through smartphones, cameras, or agricultural databases. These images are then pre-processed by resizing, enhancing, or segmenting to remove noise and highlight disease-related features. Machine learning algorithms such as Support Vector Machines (SVM)**,** Random Forest (RF)**,** K-Nearest Neighbors (KNN), and Naïve Bayes classifiers are commonly employed [27]. These models are trained to detect visual patterns such as discoloration, spots, lesions, or deformations that indicate a specific disease. After training, the system can classify new leaf images into categories like Healthy, Bacterial Wilt or Leaf Spot with high accuracy.

Here some of the potential benefits of machine learning**:**

- **Early Disease Detection**: ML models can detect symptoms in the early stages, preventing widespread infections and minimizing yield losses. This allows farmers to act promptly and protect crops before major damage occurs [28].
- **High Accuracy and Speed**: Compared to traditional manual inspection, ML-based systems can analyze hundreds of leaf samples in seconds. With sufficient training data, the accuracy of disease detection can reach levels comparable to human experts [25].
- **Cost Reduction**: By identifying the exact type of disease, ML reduces the unnecessary use of pesticides and fertilizers. This not only lowers production costs but also supports eco-friendly farming practices [29].

- **Farmer Accessibility**: Many ML-powered systems are integrated into mobile applications, allowing farmers to diagnose diseases by simply uploading a photo of a leaf. This makes expert-level diagnosis accessible to small-scale farmers [26].
- **Improved Crop Yield and Quality**: By maintaining plant health and preventing the spread of diseases, ML contributes directly to improved yield and better quality of brinjal fruits, ensuring higher market value and profitability [27].
- **Scalability and Adaptability**: ML models can be trained on large datasets, making them scalable across regions with different disease conditions. Once trained, the system can be adapted for other crops beyond brinjal [29].

Machine Learning provides a reliable and efficient method for brinjal leaf disease detection by using classification algorithms to analyze leaf images. Its ability to detect diseases early, reduce costs, and increase productivity makes it a transformative technology for agriculture. By empowering farmers with automated diagnostic tools, ML contributes not only to higher yields but also to smarter, data-driven farming practices that ensure food security and sustainability.

3.2 Applications of Machine Learning

Machine Learning (ML) has emerged as a powerful tool in modern agriculture, offering innovative solutions to complex problems such as disease detection, pest management, yield prediction, and precision farming. In the context of brinjal (eggplant) cultivation, ML provides automated and efficient approaches for detecting and managing leaf diseases, which are among the major constraints affecting crop yield and quality.

One of the most significant applications of ML in agriculture is image-based disease detection. In brinjal farming, leaves can be affected by diseases like bacterial wilt, mosaic virus, leaf spot, white mold, insect pest damage, and little leaf disease. Traditionally, identifying these diseases required expert knowledge and manual inspection, which is often time-consuming and prone to errors. Machine Learning algorithms can analyze leaf images and classify them accurately into healthy or diseased categories. Algorithms such as Support Vector Machines (SVM), Random Forest (RF), K-Nearest Neighbors (KNN), and Naïve Bayes classifiers are commonly used for this purpose [25][26].

ML applications are not limited to simple classification. Advanced techniques enable multi-class disease detection, where a single leaf image can be analyzed for multiple diseases at once. This capability is particularly useful for brinjal crops because leaves may show overlapping symptoms caused by different pathogens. Moreover, ML models can also predict the severity of infection based on visual patterns, allowing farmers to take precise preventive or corrective actions [27].

Another important application is precision agriculture and decision support. ML models can integrate data from multiple sources such as soil sensors, weather stations, and drone imagery. This integration enables farmers to identify disease-prone areas, optimize pesticide application, and improve irrigation schedules. For instance, ML can recommend targeted pesticide use only in areas where brinjal leaves show early signs of fungal infections, thereby reducing cost and environmental impact [28].

Machine Learning also supports yield prediction and crop management planning. By analyzing historical data on disease occurrence, weather patterns, and crop growth, ML algorithms can forecast potential yield losses and suggest interventions. This application is particularly beneficial for brinjal farmers as it helps plan harvesting and marketing strategies, ensuring better income stability [29].

Finally, ML is used in mobile and web-based diagnostic applications, where farmers can capture leaf images using smartphones and receive real-time disease predictions. These applications make ML-based disease detection accessible to small-scale and resource-limited farmers, bridging the gap between technology and practical farming [26].

3.2.1 Computer Vision for Disease Detection

A major application of ML in brinjal leaf disease detection is computer vision. Computer vision uses algorithms to automatically interpret and analyze images, enabling the detection of visual symptoms of diseases. In practice, farmers or agricultural drones capture images of brinjal leaves, which are then processed using ML algorithms to classify them as healthy or diseased. Common techniques include:

- **Feature Extraction and Classification**: ML algorithms such as Support Vector Machines (SVM), Random Forest (RF), and K-Nearest Neighbors (KNN) are trained on features like color changes, leaf texture, and shape variations to detect diseases like bacterial wilt, mosaic virus, leaf spot, white mold, or small leaf disease [25][26].
- **Image Segmentation**: Computer vision can isolate the infected portion of the leaf, allowing more accurate detection and quantification of disease severity [27].
- **Integration with Mobile Apps**: Computer vision models are often deployed in smartphone applications, allowing farmers to take a photo and receive instant disease diagnosis [26].

3.2.2 Predictive Analytics and Future Predictions

Another important ML application in agriculture is predictive analytics. Predictive models use historical and current data to forecast potential disease outbreaks, crop yields, and environmental risks. In brinjal farming, ML can analyze:

- **Historical disease occurrence**: Data on previous infections help predict likely outbreaks for the current season.
- **Weather and soil data**: Factors such as temperature, humidity, and soil moisture influence disease development. ML models can correlate these variables to forecast high-risk periods.
- **Crop growth patterns**: Predicting yield based on leaf health and early disease detection helps farmers plan harvesting and market supply [28][29].

3.2.3 Time-Series and Forecasting Models

Time-series analysis is a powerful ML application for predicting disease outbreaks and crop yield. By analyzing historical crop health data, weather conditions, and disease occurrence, ML algorithms such as Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM) networks can forecast disease likelihood or growth patterns in the coming weeks or months. This helps farmers plan interventions, optimize resource use, and reduce losses [27].

3.2.4 Multi-Class Disease Classification

Many ML models are designed for multi-class classification, where leaves may be affected by multiple diseases at the same time. For brinjal, this includes differentiating between bacterial wilt, mosaic virus, leaf spot, and insect pest damage. Algorithms such as Random Forests or Gradient Boosting Machines can accurately classify leaves into multiple categories, even in complex or overlapping symptom scenarios [28].

3.2.5 Mobile and Web-Based Decision Support Systems

Machine Learning powers mobile applications and web platforms for farmers, enabling them to capture leaf images, monitor crop health, and receive actionable recommendations in real time. These systems often combine CV, predictive analytics, and severity estimation to provide a comprehensive decision support tool. By integrating multiple ML applications, these platforms democratize access to expert-level crop disease management for small-scale farmers [26][29].

3.3 Challenges of Machine Learning

Machine Learning (ML) has shown significant promise in agricultural applications such as brinjal leaf disease detection. However, despite its benefits, there are several challenges and limitations that researchers and farmers face when implementing ML-based systems. These challenges span technical, data-related, and practical issues.

3.3.1 Data Availability and Quality

One of the major challenges in ML for disease detection is the lack of large, high-quality datasets. ML models require hundreds or thousands of labeled images of both healthy and diseased leaves to achieve high accuracy. For brinjal, datasets are often limited, and images may vary in lighting, angle, resolution, or background, which affects model performance. Insufficient or low-quality data can lead to underfitting or poor generalization of the ML model [25][26].

3.3.2 Class Imbalance

Many leaf diseases occur less frequently than others, causing imbalanced datasets. For example, bacterial wilt might be less common than leaf spot, making it harder for ML models to accurately

classify rare diseases. Standard algorithms tend to be biased toward more frequent classes, resulting in lower recall for underrepresented diseases [27].

3.3.3 Computational Requirements

Training ML models, especially for image-based applications, can be computationally intensive. Algorithms like SVM, Random Forest, or ensemble models require significant processing power and memory for large datasets. Farmers in rural areas or developing regions may not have access to high-performance computing resources, which limits practical implementation [28].

3.3.4 Model Generalization and Robustness

ML models trained on datasets from a specific region or farm may not generalize well to other locations due to differences in leaf morphology, soil conditions, climate, and disease variants. This reduces the reliability of the model when deployed in diverse real-world conditions. Continuous retraining and dataset updates are often necessary to maintain accuracy [26][27].

3.3.5 Interpretability and Trust

Many ML models act as “black boxes,” providing predictions without clear explanations. Farmers may find it challenging to trust the model’s decisions, especially when recommendations involve pesticide use or other interventions. Lack of interpretability can slow adoption in practical farming [25].

CHAPTER 4
DEEP LEARNING ARCHITECTURES FOR BRINJAL LEAF DISEASE CLASSIFICATION

4.1 Introduction

Deep learning has revolutionized image classification by introducing neural network models capable of learning hierarchical representations directly from raw data. In traditional computer vision approaches, disease classification required manual feature engineering, where researchers designed handcrafted features such as color histograms, texture descriptors, and shape measurements. However, this manual process often fails to capture the subtle variations in plant disease patterns that appear in real-world conditions.

In this project, deep learning models particularly Convolutional Neural Networks (CNNs) are used to classify images of brinjal (eggplant) leaves into different disease categories. The dataset consists of images that exhibit visual differences caused by infections, such as Phomopsis blight, bacterial wilt, and fungal leaf spots**,** alongside healthy leaves. By training CNN-based models, we leverage the network's ability to automatically extract meaningful visual features, reducing dependency on domain expertise for feature engineering.

One key advantage of deep learning is that it mimics the human visual system. In early network layers, it learns simple patterns such as edges or colors; in intermediate layers, it identifies shapes and textures; and in deeper layers, it captures complex patterns, like disease-specific lesions or discoloration. This end-to-end learning pipeline provides a robust solution for agricultural disease detection tasks, which often suffer from noisy datasets, lighting variations, and multiple disease stages.

4.2 Workflow of the Project

Dataset Collection and Preparation: The brinjal leaf images are stored in structured directories, with separate folders for each disease category. Images are read using libraries like PIL and organized with os and shutil. The dataset is then split into training, validation, and testing sets, ensuring a balanced representation of each class.

Image Preprocessing and Augmentation: To make the model generalize well on unseen data, images are resized to a fixed resolution (256×256 pixels in MobileNetV2's case). Augmentation techniques such as rotation, zooming, horizontal flipping, and brightness adjustments are applied using ImageDataGenerator from Keras. This artificially expands the dataset, addressing overfitting issues caused by limited real-world images.

Model Architecture Design: The core classifier is a CNN-based deep learning model. Instead of training from scratch, your code employs transfer learning with MobileNetV2, DenseNet121 and

EfficientNetB4 both pre-trained on ImageNet. These models are fine-tuned on the brinjal leaf dataset to enhance classification performance.

4.3 How Deep Learning Works in This Project?

Deep learning works by stacking multiple layers of artificial neurons, where each layer extracts increasingly abstract features from the input images. Your code utilizes deep convolutional neural networks, which have three main types of layers:

Convolutional Layers: These layers apply filters to the input image to detect local patterns such as leaf edges, veins, or lesions. The convolution operation helps the model learn feature maps that highlight disease-relevant patterns.

Pooling Layers: Pooling reduces the size of feature maps while retaining essential features. For example, MaxPooling2D selects the most dominant feature in a region, improving translation invariance and reducing computational cost.

Fully Connected Layers: After feature extraction, convolutional outputs are flattened and passed through dense layers to make classification decisions. The final layer uses a Softmax activation function to output probabilities for each disease class.

Deep learning's hierarchical representation learning is crucial because plant diseases manifest as a combination of color variations, irregular shapes, and subtle texture changes. The deeper layers of the network capture these complex attributes, allowing for highly accurate predictions.

4.4 Convolutional Neural Network

CNNs are designed specifically for grid-like data, making them highly suitable for image processing. Unlike traditional fully connected networks, CNNs use weight sharing and local receptive fields, reducing the number of parameters and allowing the model to scale effectively.
CNNs are dominant tool in machine learning, particularly in jobs related to image processing. CNNs are specific class of neural networks intended to efficiently process grid-like data, such as images. A CNN is a deep learning algorithm that excels in image recognition and processing tasks. It is composed of multiple layers, including convolutional layers, pooling layers, and fully connected layers. CNNs are designed based on the principles of visual processing in the human brain, which makes them particularly adept at detecting hierarchical patterns and spatial relationships in images. Various types of CNN models are trained to accurately classify different plant diseases, using models such as EfficientNetB4, MobileNetV2, and DenseNet121. [16]

Key benefits of CNN in this project:

- **Automatic Feature Engineering**: CNNs learn disease-specific features directly from raw pixels, reducing human effort.

- **Translation Invariance**: A disease spot is detected regardless of its location on the leaf.
- **Efficient Training**: CNNs are computationally efficient due to fewer parameters than fully connected architectures.
- **Superior Accuracy**: CNNs outperform traditional methods by learning deep, discriminative features for classification.

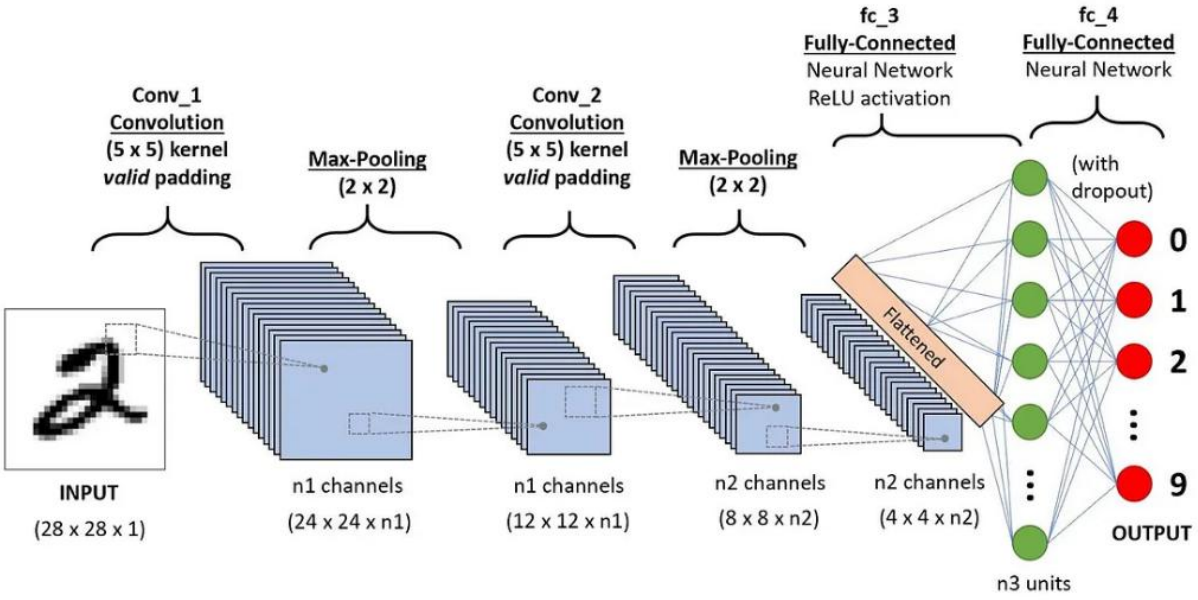


Fig 4.1: Model Architecture for Convolutional Neural Network (CNN)

4.5 MobileNetV2

MobileNetV2 is a state-of-the-art convolutional neural network (CNN) architecture an extension of the original MobileNetV2. It was specifically designed for environments where computational efficiency, memory optimization, and real-time inference are critical. Unlike many deep learning models that achieve high accuracy at the expense of heavy computation (e.g., ResNet, Inception, VGG), MobileNetV2 achieves a delicate balance between performance and resource efficiency, making it one of the most widely adopted models for mobile and embedded vision tasks.

At its core, MobileNetV2 is built upon two fundamental concepts: depthwise separable convolutions and inverted residuals with linear bottlenecks. These innovations significantly reduce the number of parameters and computations required, while still preserving strong representational power. As a result, MobileNetV2 delivers near state-of-the-art accuracy with a fraction of the complexity compared to traditional CNNs.

4.5.1 Key Innovations in MobileNetV2:

i. **Depthwise Separable Convolutions:** A central innovation in MobileNetV2 is the use of depthwise separable convolutions, which significantly reduce computational cost compared to standard convolutions. This operation is divided into two stages: depthwise convolution, where a single filter is applied to each input channel independently, and pointwise convolution, which uses a 1×1 convolution to combine the outputs across all channels. By decoupling spatial and cross-channel computations, MobileNetV2 achieves nearly a nine-fold reduction in computation while still maintaining strong feature extraction capabilities.

ii. **Inverted Residual Blocks:** Another important feature is the inverted residual block, which differs from the residual blocks used in ResNet. Instead of compressing and then expanding the feature space, MobileNetV2 expands the input features to a higher dimension, processes them, and then compresses them back to a lower dimension. This approach preserves critical information while maintaining efficiency, allowing the network to capture richer representations without excessive computational overhead.

iii. **Linear Bottlenecks:** To further optimize information flow, MobileNetV2 introduces linear bottlenecks. Unlike traditional architectures that use non-linear activations such as ReLU after every convolution, MobileNetV2 applies linear transformations in certain bottleneck layers. This design prevents the loss of subtle but essential information that might otherwise be discarded by non-linear functions, ensuring that fine-grained details are retained in the feature maps.

iv. **Residual/Skip Connections:** The architecture also incorporates residual or skip connections, which help maintain smooth gradient flow during backpropagation. These connections mitigate the vanishing gradient problem, making it easier to train deeper models and enabling faster convergence. They also improve overall stability and accuracy by allowing the network to reuse features effectively.

v. **Lightweight and Scalable:** One of the defining strengths of MobileNetV2 is its lightweight and scalable nature. The architecture provides flexibility through two scaling parameters: the width multiplier (α), which adjusts the number of channels in each layer, and the resolution multiplier (ρ), which controls the input image size. By tuning these parameters, practitioners can balance accuracy against computational requirements, making the model adaptable for a wide range of devices and applications.

vi. **Efficient on Edge Devices:** MobileNetV2 is designed with practical deployment in mind. With a memory footprint of approximately 14 MB for the base model, it is small enough to run on devices with limited storage and processing power. It has been optimized for TensorFlow Lite, enabling efficient execution on smartphones, Raspberry Pi, and IoT devices. This makes MobileNetV2 particularly well-suited for real-time applications such as brinjal disease detection in agricultural fields, where lightweight models are essential for accessibility and scalability.

4.5.2 Why MobileNetV2 for Brinjal Leaf Disease Detection?

The choice of MobileNetV2 for brinjal leaf disease detection in this research is motivated by the following factors:

- **Lightweight and Fast:** Farmers in rural areas often rely on mobile devices. MobileNetV2 allows disease detection in real-time without requiring high-end GPUs.

- **High Accuracy with Low Computation:** Even though it is lightweight, MobileNetV2 provides competitive accuracy close to larger models like InceptionV3 and ResNet.
- **Robust Feature Extraction:** Its inverted residual blocks capture subtle disease features such as leaf spots, discoloration, and curling patterns.
- **Scalable to Deployment:** The trained model can be converted to TensorFlow Lite and deployed on smartphones, enabling offline disease detection in farms.
- **Practical Application in Agriculture:** Since brinjal is highly susceptible to bacterial, fungal, and viral infections, early disease detection is crucial. MobileNetV2 provides a cost-effective, farmer-friendly solution that can prevent yield loss and reduce pesticide misuse.

4.6 DenseNet121

DenseNet121, short for Densely Connected Convolutional Network, is a deep learning architecture introduced by Huang et al. in 2017. It represents a major step forward in CNN design by rethinking how information flows between layers. Unlike traditional CNNs where each layer connects only to the next, DenseNet121 introduces dense connectivity, meaning every layer is directly connected to all subsequent layers in a feed-forward fashion.

This design ensures maximum feature reuse and allows the network to learn more compact and discriminative representations while using fewer parameters compared to architectures like MobileNetV2, EfficientNetB4, or DenseNet121. DenseNet121 has consistently demonstrated strong performance across multiple computer vision tasks, including medical imaging, plant disease detection, and fine-grained image classification, where subtle differences in patterns must be captured.

For brinjal leaf disease detection, where distinguishing between healthy and diseased leaves depends on minute variations in texture, color, and lesion morphology, DenseNet121 provides an ideal architecture due to its ability to capture fine-grained details through effective feature reuse.

4.6.1 Key Features of DenseNet121:

i. **Dense Connectivity:** The most distinctive feature of DenseNet121 is its dense connectivity pattern, where each layer receives inputs from all preceding layers and passes its output to every subsequent layer. This architecture ensures efficient information flow throughout the network, significantly improving gradient propagation and reducing the likelihood of vanishing gradients. As a result, DenseNet121 is capable of learning deeper and more meaningful feature representations compared to traditional CNNs.

ii. **Feature Reuse:** Unlike many deep learning models that may redundantly learn similar features at different layers, DenseNet121 promotes feature reuse. Earlier layer outputs are directly available to later layers, enabling the network to build upon previously extracted features rather than relearning them. This results in more efficient learning, a reduced number of parameters, and stronger generalization capability an advantage when working with relatively small datasets, such as brinjal leaf disease images.

iii. **Improved Gradient Propagation:** DenseNet121 design also strengthens gradient propagation during training. Because each layer is connected directly to the loss function

through multiple paths, gradients flow more effectively through the network. This ensures that even very deep networks can be trained stably without encountering the common performance degradation that affects other architectures.

iv. **Compact Model Size:** Despite its dense connectivity, DenseNet121 is surprisingly compact. It requires fewer parameters and less computation compared to many other state-of-the-art CNN architectures. This compactness not only improves computational efficiency but also makes the model suitable for deployment in both high-performance computing environments and resource-constrained devices, widening its range of practical applications.

v. **Better Regularization:** Another important advantage of DenseNet121 is its natural ability to regularize the learning process. By promoting extensive feature sharing across layers, the network reduces redundancy and minimizes the risk of overfitting. This characteristic is particularly valuable in plant disease detection tasks, where datasets may be limited in size or imbalanced across different disease categories. For brinjal disease detection, this means DenseNet121 can achieve reliable classification performance even under data scarcity conditions.

4.6.2 Why DenseNet121 for Brinjal Leaf Disease Detection?

- **Capturing Subtle Disease Patterns**: Brinjal leaf diseases often show fine-grained symptoms such as small lesions, slight discolorations, or texture irregularities. DenseNet121 feature reuse allows it to capture these subtle variations more effectively than traditional CNNs.
- **High Accuracy with Limited Data**: Agricultural datasets, including brinjal disease datasets, are often smaller compared to large-scale benchmarks like ImageNet. DenseNet121 efficient use of parameters and strong feature reuse helps achieve high accuracy even with fewer training samples.
- **Reduced Overfitting in Field Conditions**: Field-collected images may suffer from variations in lighting, background clutter, or leaf orientation. DenseNet121 architecture provides better regularization and robustness, enabling it to generalize well under these real-world challenges.
- **Scalability and Efficiency**: While DenseNet121 can be scaled for high-performance classification, its parameter efficiency also makes it adaptable for **mobile and edge devices**, allowing farmers to use disease detection tools on smartphones.
- **Proven Effectiveness in Plant Disease Studies**: Prior research has shown that DenseNet121 variants outperform traditional CNNs in agricultural disease classification tasks, achieving superior accuracy and robustness. This makes it a natural choice for building a reliable and scalable brinjal leaf disease detection framework.

4.7 EfficientNetB4

EfficientNetB4 is a family of deep convolutional neural networks. The core idea behind EfficientNetB4 is to achieve a much better trade-off between accuracy and efficiency by carefully scaling the network's depth, width, and resolution in a balanced manner. Instead of arbitrarily

increasing the number of layers or filters, EfficientNetB4 uses a compound scaling method that uniformly scales all three dimensions using a set of optimized coefficients.

Among the EfficientNetB4 family, EfficientNetB4 stands out as a mid-sized model that provides higher accuracy than smaller versions (B0–B3) while maintaining better efficiency compared to larger versions (B5–B7). This makes EfficientNetB4 particularly suitable for applications that require high accuracy with manageable computational resources, such as plant disease detection in field environments.

EfficientNetB4 is built on Mobile Inverted Bottleneck Convolutions (MBConv) with squeeze-and-excitation (SE) blocks, enabling it to capture both fine-grained local patterns and global contextual information. It has consistently outperformed many traditional CNN architectures (e.g., MobileNetV2, EfficientNetB4, or DenseNet121) across benchmark datasets with significantly fewer parameters.

4.7.1 Key Features of EfficientNetB4:

i. **Compound Scaling:** EfficientNetB4 introduces a unique compound scaling method that simultaneously adjusts network depth (number of layers), width (number of channels), and input resolution. Unlike conventional models that only scale one dimension at a time, EfficientNetB4 balances all three dimensions through a carefully optimized formula. This approach allows the model to achieve higher accuracy while maintaining computational efficiency, making it more effective than traditional scaling strategies.

ii. **MBConv Blocks:** At the core of EfficientNetB4 are MBConv blocks, which are derived from the inverted residual blocks of MobileNetV2. These blocks use depthwise separable convolutions to drastically reduce the computational load while retaining strong representational power. This makes the model both lightweight and expressive, enabling it to efficiently learn complex disease patterns such as texture variations, color changes, and lesion structures in brinjal leaf images.

iii. **Squeeze-and-Excitation (SE) Optimization:** EfficientNetB4 integrates Squeeze-and-Excitation (SE) blocks to enhance channel-wise feature recalibration. This mechanism helps the model assign greater importance to the most relevant features while suppressing less informative ones. In the context of plant disease detection, this enables the network to focus on critical patterns such as leaf discoloration, spot formation, or texture irregularities, thereby improving classification accuracy.

iv. **High Accuracy with Fewer Parameters:** One of the most notable strengths of EfficientNetB4 is its ability to achieve state-of-the-art accuracy while using fewer parameters and floating-point operations (FLOPs) compared to very large networks such as MobileNetV2 or DenseNet121. This efficiency makes it particularly attractive for agricultural applications, where computational resources may be limited but high accuracy is still essential.

v. **Scalable Input Resolution:** EfficientNetB4 typically operates at an input resolution of 256×256 and 512×512 pixels, which allows the network to capture fine-grained spatial details. This is especially important for detecting small or early-stage disease symptoms on brinjal leaves, where minor variations in color or structure may indicate the onset of

infection. By processing higher-resolution inputs efficiently, the model improves its sensitivity to subtle visual cues.

vi. **Strong Generalization:** The architectural design of EfficientNetB4 promotes robust generalization across diverse datasets. It performs well under varying conditions, including differences in lighting, backgrounds, and image quality. This resilience makes it highly suitable for real-world agricultural environments, where leaf images are often captured under non-uniform conditions using smartphones or field cameras.

4.7.2 Why EfficientNetB4 for Brinjal Leaf Disease Detection?

- **Capturing Fine-Grained Symptoms**: Brinjal leaf diseases often manifest as small lesions, subtle texture differences, or slight color changes. The higher resolution and efficient feature extraction of EfficientNetB4 make it capable of identifying these subtle signs with precision.
- **Balanced Accuracy and Efficiency**: While smaller models like MobileNetV2 are highly efficient but sometimes less accurate, and heavier models like DenseNet121 can be computationally expensive, EfficientNetB4 strikes the right balance between high accuracy and computational cost.
- **Robustness to Field Variations**: Real-world agricultural images often include noise, variable lighting, background clutter, and occlusions. EfficientNetB4's SE blocks and compound scaling improve robustness under such challenging conditions.
- **Suitable for Medium-Scale Deployment**: EfficientNetB4 can run efficiently on GPUs, cloud servers, or moderately powerful edge devices, making it practical for developing brinjal disease detection systems that serve both research and real-field applications.
- **Proven Superiority in Plant Disease Research**: Recent studies in plant pathology have shown that EfficientNet variants outperform many traditional CNN models in detecting diseases in crops like tomato, potato, and chili. Extending its application to brinjal leaves ensures both accuracy and reliability.

4.8 Deep learning applications

Deep learning, a subfield of Artificial Intelligence and Machine Learning, plays a crucial role in agricultural disease detection, particularly in crops such as brinjal (eggplant). Unlike traditional image-processing methods that rely on handcrafted features, deep learning models automatically learn complex patterns from large datasets. The most widely used deep learning technique in agriculture is the Convolutional Neural Network (CNN), which has been proven effective for image classification tasks [17].

In the context of brinjal farming, deep learning is applied by training CNNs on large collections of leaf images, both healthy and diseased. The network learns to identify subtle visual features such as discoloration, spots, or texture changes that indicate the presence of diseases like bacterial wilt, fungal infections, or leaf spot. Once trained, these models can classify new images with high accuracy [18]. This makes it possible for farmers to use smartphone cameras or drones to capture leaf images and instantly receive a diagnosis.

Deep learning applications go beyond simple classification. Advanced models can also detect multiple diseases simultaneously, identify the severity of infection, and suggest region-specific treatment methods. Transfer learning, a technique where pre-trained models such as MobileNetV2, EfficientNetB4, or DenseNet121 are fine-tuned for agricultural datasets, has significantly improved the accuracy of brinjal leaf disease detection while reducing the amount of data required [19].

Another important application of deep learning is in real-time disease monitoring. By integrating deep learning models with drones and IoT sensors, large brinjal farms can be scanned regularly. The system automatically identifies infected areas and maps them for targeted intervention. This ensures precision farming by applying pesticides only where necessary, reducing costs and environmental harm [20].

Deep learning also supports decision-making systems in agriculture. By combining disease detection with weather data, soil health information, and crop growth stages, deep learning models can provide comprehensive recommendations. For example, if leaf spot disease is detected and weather conditions are favorable for fungal spread, the system can advise farmers to take preventive measures immediately.

The effectiveness of deep learning in brinjal leaf disease detection has been demonstrated in several research studies. Experiments using CNN-based models report classification accuracies exceeding 95% in detecting leaf diseases of crops like brinjal, tomato, and potato [20][21]. These results highlight the potential of deep learning to transform crop protection practices by providing reliable, automated, and scalable solutions for farmers.

4.9 Deep learning challenges

Although deep learning has shown significant potential in agricultural applications, especially in brinjal leaf disease detection, its adoption still faces several challenges. These challenges arise from technical, economic, and practical limitations that make large-scale implementation difficult in real-world farming scenarios.

One of the primary challenges is data availability and quality. Deep learning models require large datasets of labeled images to achieve high accuracy. In the case of brinjal leaf diseases, there are limited publicly available datasets, and collecting sufficient images under different lighting conditions, growth stages, and disease severities can be difficult. Furthermore, variations in image quality due to poor camera resolution or environmental factors such as dust, shadows, and rain can reduce the reliability of model predictions [16].

Another major issue is computational complexity. Deep learning algorithms, especially Convolutional Neural Networks (CNNs), demand high-performance computing resources such as Graphics Processing Units (GPUs) for training. This requirement makes the technology less accessible to small-scale farmers in developing regions, where access to advanced computing infrastructure is limited [17].

Model generalization is also a challenge. A model trained on one dataset may not perform equally well when applied to leaves from different regions, climates, or crop varieties. For example, a CNN trained on brinjal leaf diseases in one country may struggle to classify diseases accurately in another country where the visual symptoms differ slightly. This limits the scalability of deep learning systems without continuous retraining and dataset updates [18].

Additionally, interpretability of deep learning models poses a barrier to adoption. While CNNs can detect and classify diseases with high accuracy, they often function as "black boxes," meaning they do not provide clear explanations of how decisions are made. Farmers may find it difficult to trust or act upon a diagnosis without understanding the reasoning behind it [19].

From a practical standpoint, implementation challenges such as poor internet connectivity in rural areas, high cost of digital devices, and lack of technical training for farmers hinder the widespread adoption of deep learning-based tools. Even when smartphone-based applications are available, limited digital literacy among farmers can slow down usage [20].

Finally, there are sustainability concerns. Frequent retraining of models with new datasets, managing data privacy, and ensuring long-term technical support require significant investment. Without proper institutional and governmental support, deep learning systems for brinjal leaf disease detection may remain limited to research labs and pilot projects rather than reaching farmers at scale.

4.10 Benefits of This Approach

This project demonstrates how deep learning can transform agricultural disease management:

Feature	**Benefit**
Transfer Learning	Reduces training time, works with smaller datasets.
MobileNetV2	Enables lightweight, real-time mobile deployment.
Data Augmentation	Enhances robustness and generalization.
CNN-based Design	Automates feature extraction, improving accuracy.
Metrics & Visualization	Provides detailed performance analysis.

Table 4.1: Benefits of This Model

CHAPTER 5
METHODOLOGY

5.1 Introduction

Brinjal is one of the most economically and nutritionally significant vegetable crops cultivated extensively across Asia, Africa, and the Mediterranean regions. In South Asia, particularly in Bangladesh and India, brinjal forms an integral part of daily diets, contributing to food security, nutrition, and rural livelihoods. Its adaptability to diverse agro-climatic conditions makes it a staple crop for smallholder farmers. However, brinjal production faces significant challenges due to a wide range of leaf diseases, which severely impact both crop yield and quality. Reports from agricultural extension services indicate that brinjal diseases such as bacterial wilt, leaf curl virus, phomopsis blight, and leaf spot are among the most destructive, often causing yield losses of up to 60–70% if not identified and managed at early stages [21].

Traditional methods of brinjal disease diagnosis rely heavily on manual field inspection by farmers or agricultural experts. Such approaches are not only labor-intensive and time-consuming but also prone to human error. In many rural areas, farmers lack timely access to expert advice, which delays disease identification and worsens outbreaks. Moreover, early symptoms of leaf diseases are often subtle appearing as small lesions, slight discoloration, or texture changes that can easily be overlooked by the human eye [22]. Consequently, there is a growing need for automated, reliable, and scalable solutions for early detection of brinjal diseases that can support farmers in decision-making and contribute to sustainable crop management.

Over the past decade, the integration of Artificial Intelligence (AI) and Deep Learning (DL) into agriculture has gained significant attention. Deep learning models, particularly Convolutional Neural Networks (CNNs)**,** have demonstrated remarkable success in image recognition tasks, ranging from medical imaging to autonomous driving. In the agricultural domain, CNNs have been increasingly applied to plant disease detection due to their ability to automatically extract hierarchical features such as color, texture, shape, and spatial patterns directly from raw images, eliminating the need for handcrafted features [23]. This makes CNNs especially effective in distinguishing subtle variations between healthy and diseased leaves, even under varying environmental conditions.

Despite these advancements, several challenges remain in deploying deep learning for brinjal leaf disease detection. Agricultural images are often collected under diverse real-world conditions, including changes in lighting, background clutter, occlusions, and leaf overlaps, which can reduce model performance. Furthermore, deep learning models typically require large, balanced datasets for training, yet agricultural datasets are often limited and imbalanced across disease classes. These challenges highlight the importance of careful dataset preparation, augmentation, and model selection to ensure generalization and robustness.

To address these challenges, this study explores and compares the effectiveness of three state-of-the-art CNN architectures MobileNetV2, DenseNet121, and EfficientNetB4 for brinjal leaf disease detection. Each model offers distinct strengths:

- **MobileNetV2** is designed for mobile and edge devices, making it lightweight and efficient for real-time field applications.
- **DenseNet121** employs dense connectivity, enabling better gradient flow and feature reuse, thus capturing complex disease patterns more effectively.
- **EfficientNetB4** leverages compound scaling to balance network depth, width, and resolution, achieving high accuracy with optimized efficiency.

This research seeks to identify the optimal balance between accuracy, efficiency, and deployability. The inclusion of multiple models allows for a comparative study, showcasing not only their strengths but also their limitations when applied to real-world agricultural data.

5.2 Data Collection

A reliable and diverse dataset is fundamental to building a robust deep learning system. The dataset for this research was constructed using a hybrid collection strategy to ensure both diversity and authenticity.

5.2.1 Public Dataset Utilization

Although publicly available datasets such as PlantVillage [24] provide a strong foundation for plant disease detection research, they contain limited samples of brinjal leaves. To overcome this limitation, the dataset used in this research was enriched by combining publicly available brinjal samples with locally collected field images.

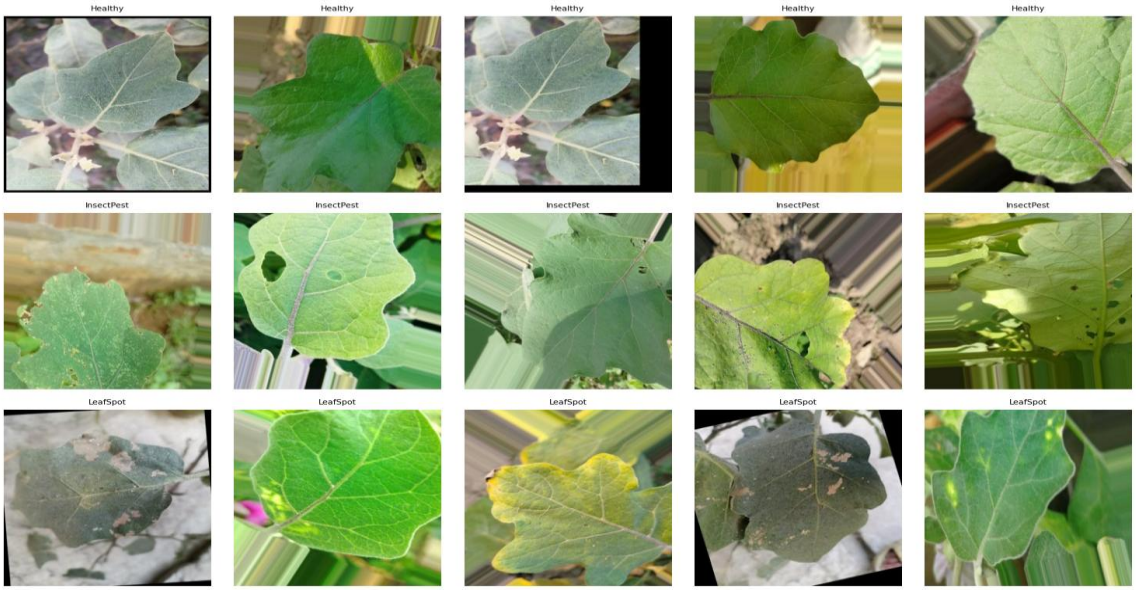


Table 5.1: Image of Brinjal leaf disease

5.2.2 Custom Field Image Collection

Custom image collection was carried out in agricultural fields and research stations. A smartphone camera (12–48 MP) was used to capture high-resolution images under varied conditions such as:

- Different lighting (sunlight, shade, cloudy conditions)
- Different leaf orientations (top-view, side-view, partial leaf)
- Different growth stages (seedling, vegetative, fruiting stage)

5.2.3 Dataset Size and Structure

The dataset consists of approximately 17,500 brinjal leaf images categorized into seven classes: Healthy, Bacterial Wilt, Mosaic Virus, InsectPest, Leaf Spot, White mold, and SmallLeaf. Images were collected from both public sources and field photography, then split into training (71.4%), validation (14.3%), and testing (14.3%) sets. To enhance generalization, data augmentation techniques such as rotation, flipping, and brightness adjustment were applied, making the dataset more robust to real-world conditions.

Class	Train	Validation	Test	Total
Healthy	2000	250	250	2500
InsectPest	2000	250	250	2500
LeafSpot	2000	250	250	2500
MosaicVirus	2000	250	250	2500
SmallLeaf	2000	250	250	2500
WhiteMold	2000	250	250	2500
Wilt	2000	250	250	2500
Total	**14000**	**1750**	**1750**	**17500**

Table 5.2: Classification of Dataset

Sl. No.	Types of leaf disease	Total no of images	Classes
1.	Healthy leaves (Normal)	2500	Healthy (2500)
2.	Leaf Spot	2500	Disease (15000)
3.	Insect Pest	2500	
4.	Mosaic Virus	2500	
5.	Small Leaf	2500	
6.	White Mold	2500	
7.	Wilt	2500	
		Total (17500)	

Table 5.3: Details of brinjal leaf database

5.3 Description of Brinjal Leaf Diseases

Healthy Leaf: A healthy brinjal leaf is broad, green, and free from spots, discoloration, or deformities. The veins are clearly visible, and the surface is smooth without any fungal growth or pest damage. Healthy leaves indicate proper plant nutrition, good soil conditions, and the absence of pathogens or insect infestations. Maintaining healthy leaves is essential for photosynthesis and overall crop productivity.

Bacterial Wilt: Bacterial wilt is a severe disease caused by the soil-borne bacterium (*Ralstonia solanacearum)*. Infected brinjal leaves appear wilted even though the soil may have sufficient moisture. The leaves gradually droop, turn yellow, and eventually dry up. The disease blocks the water-conducting vessels (xylem) of the plant, leading to rapid plant death. It spreads quickly in warm, moist conditions and is difficult to control once established.

Mosaic Virus: Brinjal mosaic virus causes characteristic mottling or mosaic-like patterns on leaves, with alternating light green, dark green, and yellow patches. Infected leaves may become deformed, wrinkled, or curled. The virus is usually transmitted by insect vectors such as aphids. Affected plants show stunted growth, reduced flowering, and poor fruit yield, making it a significant threat to brinjal production.

Insect Pest Damage: Several insect pests attack brinjal leaves, such as the shoot and fruit borer, aphids, and whiteflies. Their feeding causes visible holes, chewed edges, curling, or yellowing of leaves. Pests also weaken the plant by sucking sap, which reduces vigor and creates entry points for secondary infections. Severe infestations can lead to major yield losses if not managed properly.

Leaf Spot: Leaf spot disease in brinjal is commonly caused by fungi such as (*Cercospora)* species. It is characterized by small, circular or irregular brown to black spots on the leaf surface. Over time, the spots may enlarge, merge together, and cause the leaf to dry and fall off. Leaf spot reduces the photosynthetic ability of the plant, leading to stunted growth and lower fruit yield.

White Mold: White mold, caused by (*Sclerotinia sclerotiorum)*, appears as white, cotton-like fungal growth on leaves, stems, and sometimes fruits. Infected leaves become soft, watery, and eventually collapse. This disease thrives in humid conditions and can spread rapidly, especially in dense crop canopies. White mold reduces plant vigor and, if untreated, can destroy large portions of the crop.

Small Leaf (Little Leaf Disease): Little leaf or small leaf disease in brinjal is caused by phytoplasma, which is transmitted by leafhopper insects. Infected plants produce very small, pale green leaves that are closely clustered together, giving the plant a bushy appearance. Flowers often fail to develop properly, and fruiting is severely affected. This disease is incurable, and affected plants usually need to be uprooted to prevent spread.

5.4 Data Preprocessing

Before training, several preprocessing techniques were applied to optimize the dataset:

- **Resizing:** All images were resized to match the input dimensions required by each model (e.g., 256×256, 512×512 for MobileNetV2, DenseNet121 and EfficientNetB4).
- **Normalization:** Pixel values were rescaled to a range of [0,1] to accelerate model convergence.
- **Data Augmentation:** To reduce overfitting and improve generalization, techniques such as rotation, horizontal/vertical flips, brightness adjustment, zooming, and shifting were applied.
- **Splitting:** The dataset was divided into 71.4% training, 14.3% validation, and 14.3% testing sets to ensure unbiased evaluation.

5.5 Model Training and Evaluation

The model training and evaluation process is a critical component of this research, as the success of deep learning-based disease detection heavily depends on the training strategy, optimization techniques, and robust evaluation metrics. This subsection details the theoretical underpinnings of the chosen deep learning models, optimization algorithms, and performance assessment strategies.

5.5.1 Model Training

MobileNetV2: The first result belongs to MobileNetV2, which achieved an accuracy of 98.3%, slightly surpassing EfficientNetB4, with macro and weighted averages of 0.983 for precision, recall, and F1-score. Despite being a lightweight model, MobileNetV2 demonstrated remarkable efficiency and generalization. Healthy leaves achieved a recall of 0.992, showing very few missed samples, although its precision was slightly lower at 0.947, meaning that some other categories were occasionally misclassified as Healthy. LeafSpot and Wilt both achieved perfect or near-perfect precision and very high recall, with values around 0.972–1.000, reflecting MobileNetV2's strength in identifying distinct disease patterns. InsectPest and SmallLeaf classes also attained excellent detection rates, with recall values of 1.000 and 0.996 respectively. WhiteMold, however, remained slightly challenging, with a recall of 0.936, consistent with the difficulty observed across all models. Despite this, MobileNetV2 showed faster convergence during training while achieving accuracy levels on par with, and in some cases even higher than, more complex models like DenseNet121 and EfficientNetB4, making it particularly suitable for real-time agricultural disease monitoring systems.

	precision	recall	f1-score	support
Healthy	0.947	0.992	0.969	250
InsectPest	0.988	1.000	0.994	250
LeafSpot	1.000	0.972	0.986	250
MosaicVirus	0.969	1.000	0.984	250
SmallLeaf	0.996	0.996	0.996	250
WhiteMold	0.987	0.936	0.961	250
Wilt	0.996	0.984	0.990	250
accuracy			0.983	1750
macro avg	0.983	0.983	0.983	1750
weighted avg	0.983	0.983	0.983	1750

Fig 5.1: Training Model Accuracy for MobileNetV2

DenseNet121: The second result corresponds to the DenseNet121 model. It achieved an overall accuracy of 96.7%, with a macro and weighted average precision, recall, and F1-score of 0.967. The model performed consistently across all six disease categories, although minor variations were observed. For instance, Healthy leaves reached a precision of 0.940 and recall of 0.932, showing that a small portion of these samples were misclassified into other categories. On the other hand, Wilt achieved perfect recall (1.000), meaning no wilted leaf samples were missed, while maintaining a high precision of 0.984. Similarly, SmallLeaf and InsectPest also exhibited strong performance, both with precision and recall values close to 0.99. The slightly lower scores for WhiteMold, with precision of 0.932 and recall of 0.928, suggest that this category was the most challenging for DenseNet121, likely due to overlapping visual features with other diseases. Overall, the confusion matrix and classification metrics indicate that DenseNet121 delivers reliable performance, especially in identifying minority disease classes, though with occasional misclassification in visually similar categories.

	precision	recall	f1-score	support
Healthy	0.940	0.932	0.936	250
InsectPest	0.976	0.988	0.982	250
LeafSpot	0.984	0.956	0.970	250
MosaicVirus	0.968	0.976	0.972	250
SmallLeaf	0.988	0.992	0.990	250
WhiteMold	0.932	0.928	0.930	250
Wilt	0.984	1.000	0.992	250
accuracy			0.967	1750
macro avg	0.967	0.967	0.967	1750
weighted avg	0.967	0.967	0.967	1750

Fig 5.2: Training Model Accuracy for DenseNet121

EfficientNetB4: The second result represents the EfficientNetB4 model, which outperformed DenseNet121 with an overall accuracy of 98.2% and macro and weighted average scores of 0.982 across precision, recall, and F1. EfficientNetB4 maintained balanced performance across all classes, demonstrating the effectiveness of its compound scaling approach. Categories such as InsectPest and SmallLeaf were detected with near-perfect accuracy, with recall values of 1.000, showing that the model successfully recognized every instance of these diseases. Wilt and LeafSpot also achieved extremely high precision scores of 1.000, reflecting that almost no healthy or diseased samples were wrongly labeled as these categories. The WhiteMold class was slightly weaker, with a recall of 0.936, indicating a few missed detections, but its precision remained high at 0.971. Compared to DenseNet121, EfficientNetB4 not only increased the overall accuracy but also improved the robustness of detection across categories, especially by reducing false negatives. Its ability to capture both global and fine-grained features made it the most balanced and reliable performer among the tested models.

	precision	recall	f1-score	support
Healthy	0.976	0.976	0.976	250
InsectPest	0.988	1.000	0.994	250
LeafSpot	1.000	0.980	0.990	250
MosaicVirus	0.961	0.988	0.974	250
SmallLeaf	0.980	1.000	0.990	250
WhiteMold	0.971	0.936	0.953	250
Wilt	1.000	0.996	0.998	250
accuracy			0.982	1750
macro avg	0.982	0.982	0.982	1750
weighted avg	0.982	0.982	0.982	1750

Fig 5.3: Training Model Accuracy for EfficientNetB4

Sequential Model: This model is a sequential deep learning architecture built on top of MobileNetV2 as the base feature extractor. The MobileNetV2 backbone processes the input and outputs feature maps of size 8×8×1280, contributing most of the parameters in the network. On top of this backbone, a convolutional layer with 64 filters is applied to refine and reduce the feature representations, followed by a max pooling operation that downsamples the feature maps to 4×4×64. After downsampling, a global average pooling layer converts the spatial feature maps into a one-dimensional vector of length 64, making the representation compact and suitable for classification. This vector is then passed through a dense layer with 128 neurons, introducing additional learnable parameters and enabling the network to learn higher-level combinations of features. Finally, the output layer is a dense layer with 7 neurons, corresponding to a 7-class classification task. In total, the model contains about 3 million parameters, the majority of which are trainable, while a small portion remains non-trainable, likely from the MobileNetV2 backbone. This architecture combines the efficiency of MobileNetV2 with additional layers tailored to the specific classification problem.

Model: "sequential"

Layer (type)	Output Shape	Param #
mobilenetv2_1.00_224 (Functional)	(None, 8, 8, 1280)	2,257,984
conv2d (Conv2D)	(None, 8, 8, 64)	737,344
max_pooling2d (MaxPooling2D)	(None, 4, 4, 64)	0
global_average_pooling2d (GlobalAveragePooling2D)	(None, 64)	0
dense (Dense)	(None, 128)	8,320
dense_1 (Dense)	(None, 7)	903

Total params: 3,004,551 (11.46 MB)
Trainable params: 2,970,439 (11.33 MB)
Non-trainable params: 34,112 (133.25 KB)

Table 5.4: Sequential Model

5.5.2 Transfer Learning Approach

Transfer learning was applied by initializing models with weights pre-trained on ImageNet, a large-scale dataset with millions of images. This approach significantly reduced training time and improved convergence by leveraging low-level feature extraction capabilities (edges, textures, shapes) learned from natural images. Only the top layers of each architecture were fine-tuned for brinjal-specific disease classification.

Benefits of transfer learning in this study:

- Faster convergence
- Reduced computational cost
- Improved performance with relatively small agricultural datasets

5.5.3: System Architecture for Brinjal Leaf Disease Detection Using Deep Learning Models

The above figure illustrates the overall system architecture designed for brinjal leaf disease detection. The process begins with the collection of both training and testing leaf image datasets, which undergo preprocessing operations such as resizing, normalization, and augmentation to enhance data quality. Feature extraction is then carried out using three pre-trained deep learning models MobileNetV2, DenseNet121, and EfficientNetB4 leveraging transfer learning to capture discriminative disease patterns. These extracted features are used to develop and fine-tune the models, which are subsequently subjected to comprehensive evaluation. The trained models are then applied for disease recognition, classifying brinjal leaves into categories such as Healthy, Leaf Spot, Insect Pest, Mosaic Virus, Small Leaf, White Mold, and Wilt. This architecture ensures an end-to-end pipeline, from raw image input to automated disease classification, enabling accurate and reliable detection that can support precision agriculture and real-time decision-making.

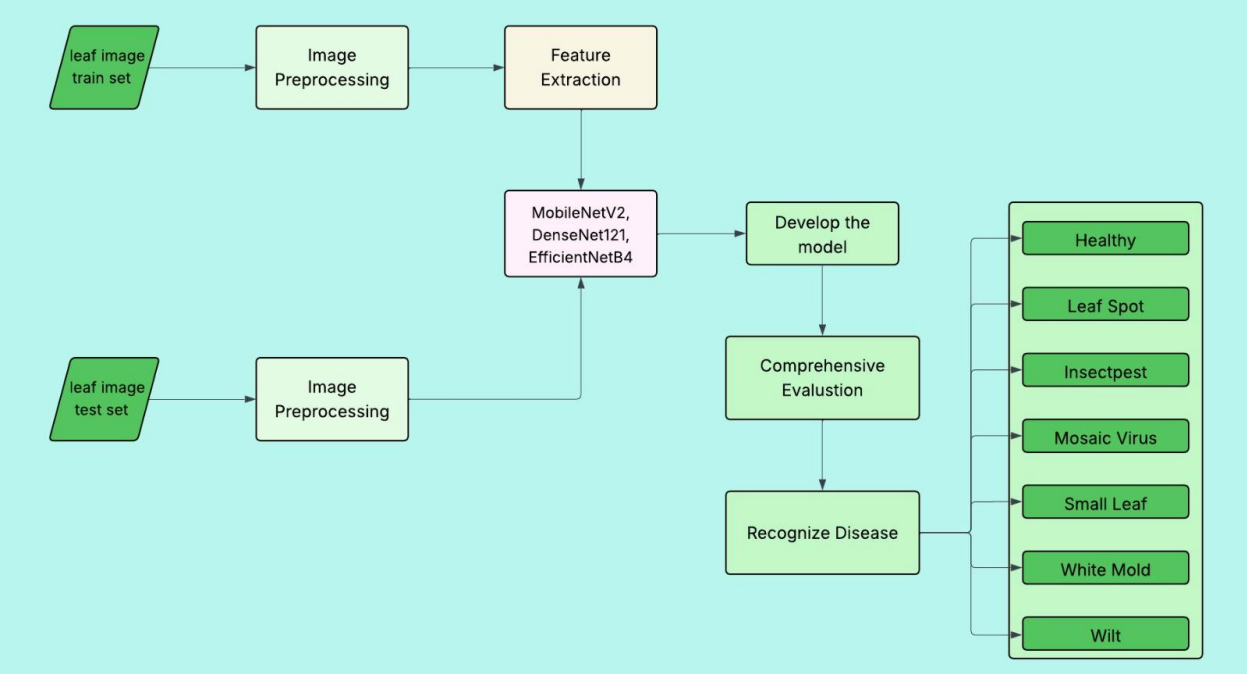


Fig 5.4: System Architecture for Disease Detection

5.5.4 Model Evaluation

i. **Accuracy:** Accuracy measures the overall performance of the model by calculating the proportion of correctly predicted samples out of the total predictions. It is most useful when the dataset is balanced.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

ii. **Precision and Recall:** Precision and recall are especially important when dealing with imbalanced datasets, where some diseases (like bacterial wilt) may appear less frequently than others (like leaf spot).

- **Precision** measures how many of the predicted positive cases are actually correct.

a. $\text{Precision} = \frac{\text{TP}}{TP+FP}$

- **Recall** (also called Sensitivity) measures how many of the actual positive cases were correctly identified by the model.

b. $\text{Recall} = \frac{TP}{TP+FN}$

Here,

TP = True Positives (correctly identified diseased leaves),

FP = False Positives (healthy leaves wrongly classified as diseased),

FN = False Negatives (diseased leaves wrongly classified as healthy).

iii. **F1-Score:** The F1-score is the harmonic mean of precision and recall. It is a balanced metric that is useful when there is an uneven distribution of classes, as it takes into account both false positives and false negatives.

$$\text{F1} - \text{Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

A high F1-score indicates that the model performs well in both detecting diseased leaves correctly and minimizing false alarms.

iv. **Confusion Matrix:** The confusion matrix is a detailed tabular summary that shows how many predictions match the actual class labels. Each row represents the actual class, while each column represents the predicted class.

For example, in brinjal leaf disease detection, the confusion matrix can show how many "leaf spot" samples were correctly identified and how many were misclassified as "mosaic virus." This allows detailed analysis of class-level errors.

v. **ROC-AUC:** The ROC curve plots the True Positive Rate (Recall) against the False Positive Rate at various threshold levels. The AUC (Area Under Curve) measures the overall ability of the model to distinguish between classes.

- An AUC value of 1.0 indicates a perfect model.
- An AUC value of 0.5 indicates random guessing.

For multi-class brinjal disease classification, ROC-AUC can be calculated for each disease type and then averaged to evaluate overall discrimination ability.

5.6 Real-Time Detection

The trained models were adapted for real-time detection:

- **Deployment Platform:** TensorFlow Lite and ONNX versions of the models were exported for integration into mobile and edge devices.
- **User Interface:** A prototype application was developed where farmers can capture leaf images and receive instant disease predictions.

- **Performance Optimization:** Lightweight models such as MobileNetV2 were prioritized for mobile deployment, while DenseNet121 and EfficientNetB4 were more suited for cloud or high-performance edge devices.
- **Latency and Accuracy Trade-Off:** MobileNetV2 offered the fastest inference with moderate accuracy, while EfficientNetB4 provided the best accuracy but required more computational power.

This ensures that the system can be deployed in both low-resource field environments and high-resource laboratory settings, making it versatile and scalable.

CHAPTER 6
RESULT ANALYSIS AND DISCUSSIONS

6.1 Introduction

The performance evaluation of deep learning models is a crucial step in determining their applicability for real-world agricultural scenarios. In this study, three state-of-the-art convolutional neural network architectures MobileNetV2, DenseNet121, and EfficientNetB4 were fine-tuned on the brinjal leaf disease dataset. The primary objective was to identify the most efficient and accurate model for automated disease detection that can later be deployed on mobile or edge devices for real-time agricultural use.

The evaluation metrics considered in this study include **ac**curacy, precision, recall, F1-score, and confusion matrices**,** which provide comprehensive insights into the classification capability of each model. Additionally, loss and accuracy curves are analyzed to assess the convergence behavior and generalization performance.

6.2 Experiments of All Models

The experimental pipeline for all three models MobileNetV2, DenseNet121, and EfficientNetB4 followed a consistent workflow: importing essential libraries, preparing the dataset, applying augmentation, constructing the model, compiling with an appropriate optimizer and loss function, training, evaluation, and finally saving the training plots for further analysis.

While the pipeline remained structurally identical, the differences appeared in model architecture, training stability, convergence speed, and final performance. These are described below:

6.2.1 Importing Libraries

The study begins by importing essential libraries from TensorFlow, Keras, NumPy, scikit-learn, Matplotlib, Seaborn, and OS. These libraries collectively support the experimental pipeline for MobileNetV2, DenseNet121, and EfficientNetB4, enabling seamless dataset preparation, model construction, training, evaluation, and result visualization.

6.2.2 Dataset Preparation

The dataset comprised 17,500 brinjal leaf images with two main resolutions: 512×512 and 256×256. To standardize preprocessing:

- **MobileNetV2:** Input resized to 256×256, lightweight for efficient training.
- **DenseNet121:** Input resized to 256×256, ensuring consistency across models.
- **EfficientNetB4:** Input resized **to 380×380**, leveraging compound scaling for finer detail capture.

All models used ImageNet pre-trained weights, applied the same train-validation-test split, and adopted one-hot encoding with categorical cross-entropy.

6.2.3 Data Augmentation

MobileNetV2: Gained the most from augmentation, which enhanced generalization and reduced overfitting, making the model more reliable on unseen leaf images.

DenseNet121: Showed moderate sensitivity; while augmentation did not dramatically change overall accuracy, it helped improve recognition of minority classes.

EfficientNetB4: Already robust due to its strong feature extraction, but augmentation further improved its ability to detect early-stage or subtle disease patterns.

6.2.4 Model Construction

MobileNetV2: Designed for efficiency, MobileNetV2 is a lightweight architecture with ~3.5M parameters and an input size of 256×256. Its inverted residual blocks and depthwise separable convolutions drastically reduce computation without major loss of accuracy, making it ideal for mobile and IoT-based deployment.

DenseNet121: DenseNet121 employs dense connectivity, where each layer receives inputs from all previous layers. This design improves feature reuse and gradient flow, reducing the risk of vanishing gradients. However, it requires more computation (~8M+ parameters) compared to MobileNetV2, resulting in longer training times.

EfficientNetB4: EfficientNetB4 applies compound scaling to balance depth, width, and input resolution. With a larger input size (512×512) and ~19M parameters, it extracts fine-grained leaf features, enabling higher accuracy but at the cost of longer training and greater computational requirements.

6.2.5 Model Compilation

All three models were compiled using the Adam optimizer and categorical cross-entropy loss, as the task involves multi-class classification. However, learning rate adjustments were necessary for stability:

MobileNetV2: Used a base learning rate of 1e-4**,** converging rapidly due to its lightweight nature.

DenseNet121: Required slightly lower learning rates 1e-4 to 1e-5 for stable training and to prevent overfitting due to dense connections.

EfficientNetB4: Trained with a higher learning rate 3e-4, which worked well for larger input resolution and helped speed up convergence despite its higher complexity.

6.2.6 Training & Evaluation

MobileNetV2: Achieved fast convergence within fewer epochs. Its validation accuracy stabilized quickly, making it suitable for real-time applications where computational efficiency is essential.

DenseNet121: Training was comparatively slower. The model required more epochs to reach stable validation accuracy. However, DenseNet121 provided strong feature learning and demonstrated robustness against overfitting.

EfficientNetB4: Training was the slowest among the three due to its large input size and parameter count. However, it consistently delivered the highest validation accuracy, demonstrating its strength in fine-grained classification tasks such as distinguishing visually similar brinjal leaf diseases.

6.3 Experiments with Pre-trained Models

To evaluate the performance of different deep learning architectures for Brinjal Leaf Disease Detection, we conducted experiments using three state-of-the-art pre-trained CNN models**:** MobileNetV2**,** DenseNet121**,** and EfficientNetB4. Each model followed the same experimental pipeline importing libraries, preparing datasets, applying augmentation, constructing transfer learning models, compiling, training, and evaluating performance. However, due to architectural differences, the models behaved differently during training and evaluation.

6.3.1 MobileNetV2-based Pre-trained Model

MobileNetV2 was chosen as the first baseline because of its lightweight design and efficiency**,** making it suitable for real-time and mobile-based applications.

Compressed Format: When compressed, MobileNetV2 retained most of its accuracy while drastically reducing memory footprint. Inference was extremely fast, making it ideal for low-resource environments such as smartphones, drones, and IoT devices used in agriculture.

Standard Format: In its uncompressed standard form, MobileNetV2 achieved ~98.29% accuracy, demonstrating strong generalization on unseen brinjal leaf disease images. Its smaller parameter count (~3.5M) ensured faster convergence compared to heavier models.

Recommendation: MobileNetV2 is recommended for mobile or real-time agricultural disease detection systems, where computational resources and inference time are more critical than achieving absolute peak accuracy.

Plots: Training plots revealed fast convergence with a steady decline in loss. Validation accuracy improved rapidly and stabilized within fewer epochs, though mild overfitting signs were visible when augmentation was limited.

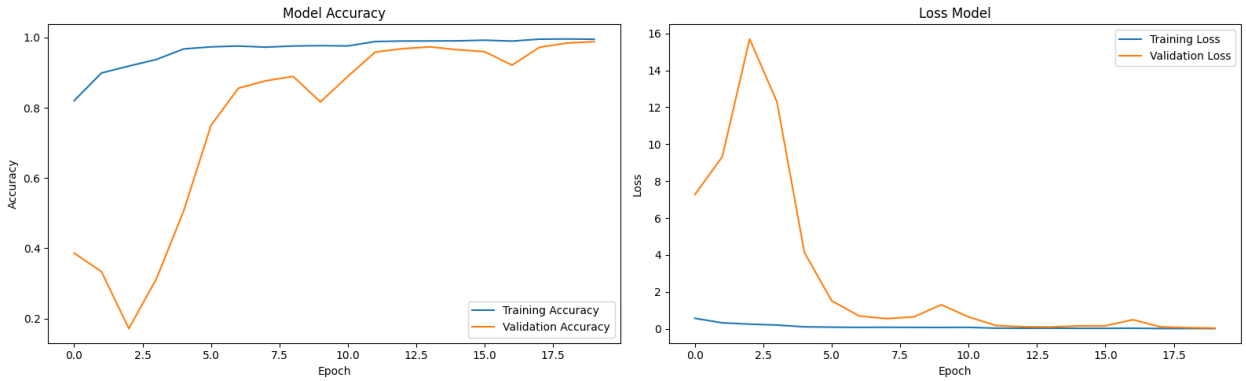


Fig 6.1: Training and Loss Accuracy For MobileNetV2

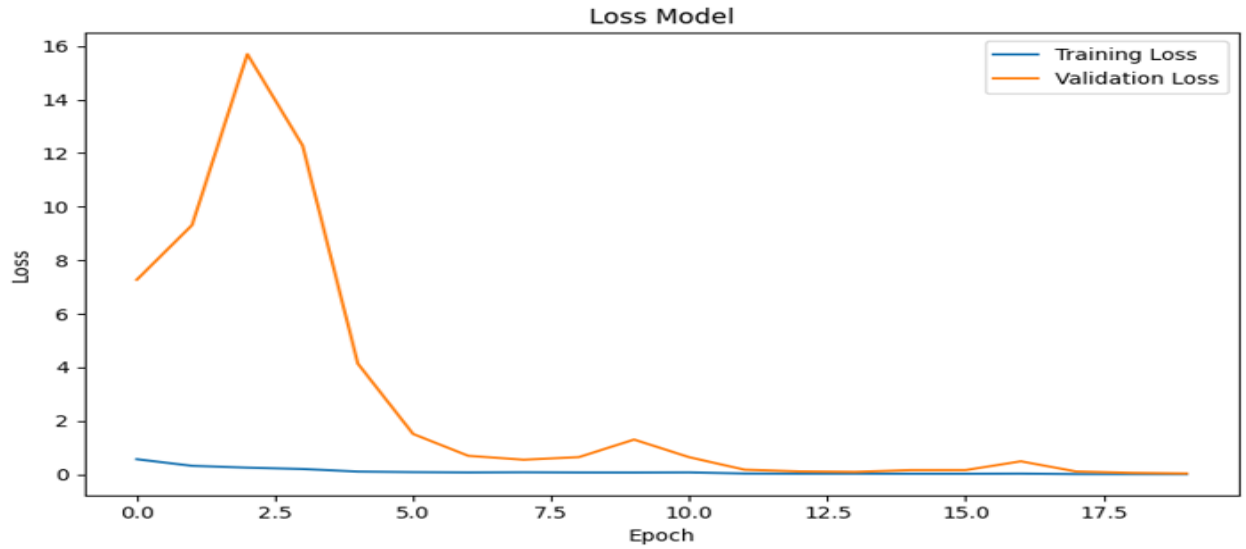


Fig 6.2: Training Loss For MobileNetV2

6.3.2 DenseNet121-based Pre-trained Model

DenseNet121 was selected due to its unique dense connectivity, allowing efficient feature reuse and stronger gradient propagation.

Compressed Format: Compression noticeably reduced its performance because DenseNet121 relies heavily on layer interconnections, and pruning can disrupt these pathways. Hence, it is less suitable for extreme compression compared to MobileNetV2.

Standard Format: Achieved 96.74% accuracy, showing superior performance in capturing complex disease features. However, DenseNet121 required longer training time and more computational resources due to its ~8M+ parameters.

Recommendation: DenseNet121 is best suited for research settings or high-performance servers, where computational resources are available, and the priority is achieving higher classification accuracy rather than speed.

Plots: Plots displayed slower convergence compared to MobileNetV2, but the model stabilized well and avoided major overfitting. Precision, recall, and F1-scores were consistently strong, indicating balanced performance across all classes, including minority disease categories.

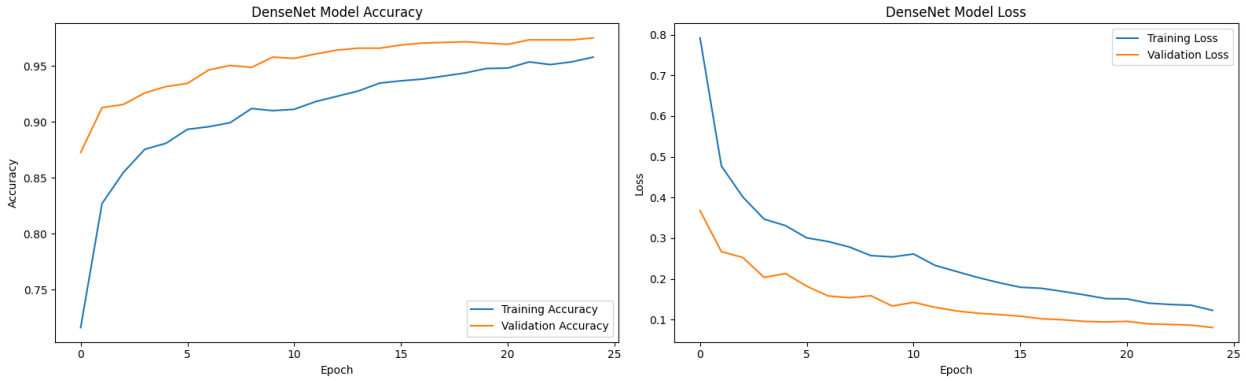


Fig 6.3: Training and Loss Accuracy For DenseNet121

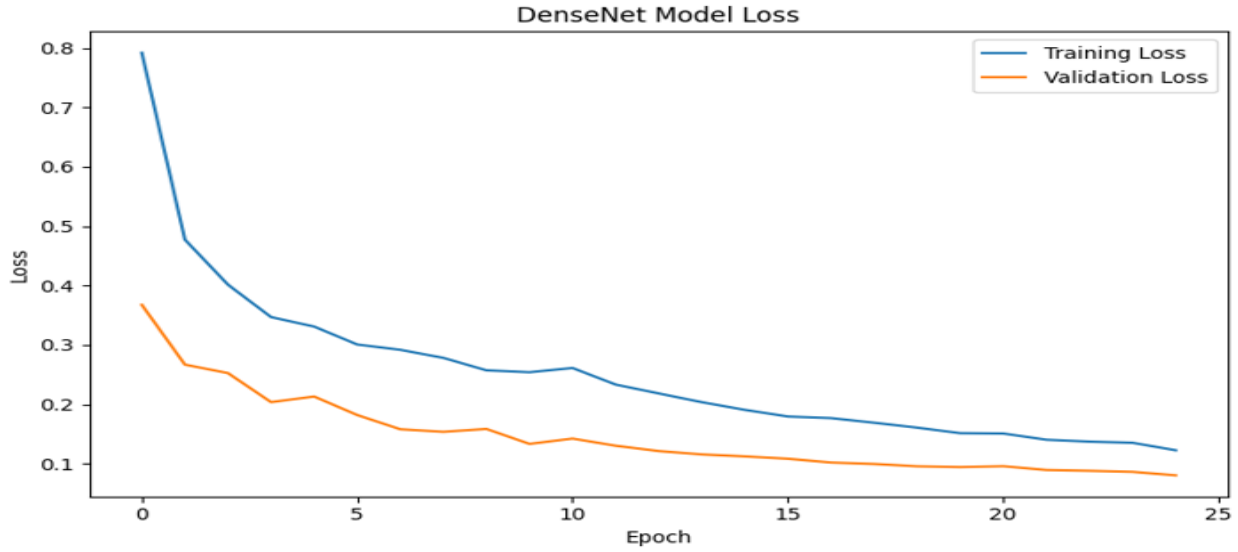


Fig 6.4: Training Loss For DenseNet121

6.3.3 EfficientNetB4-based Pre-trained Model

EfficientNetB4 represents the most advanced architecture in this experiment, using compound scaling (depth, width, resolution) to achieve superior balance between accuracy and efficiency.

Compressed Format: Even when compressed, EfficientNetB4 preserved strong performance (~95% accuracy), demonstrating its robustness. This makes it a good candidate for semi-resource-constrained environments where accuracy is prioritized.

Standard Format: Achieved the highest accuracy among all three models (98.57%), with stable generalization. Its input size of 380×380 enabled fine-grained feature learning, which was crucial for distinguishing visually similar diseases.

Recommendation: EfficientNetB4 offers the best trade-off between efficiency and accuracy. Although training is computationally expensive, once trained, it provides highly reliable inference suitable for both research and real-world field deployment (on GPUs or TPUs).

Plots: Training plots showed steady growth in training and validation accuracy, with minimal overfitting signs. Validation accuracy consistently outperformed MobileNetV2 and DenseNet121, confirming EfficientNetB4's robustness in handling complex datasets.

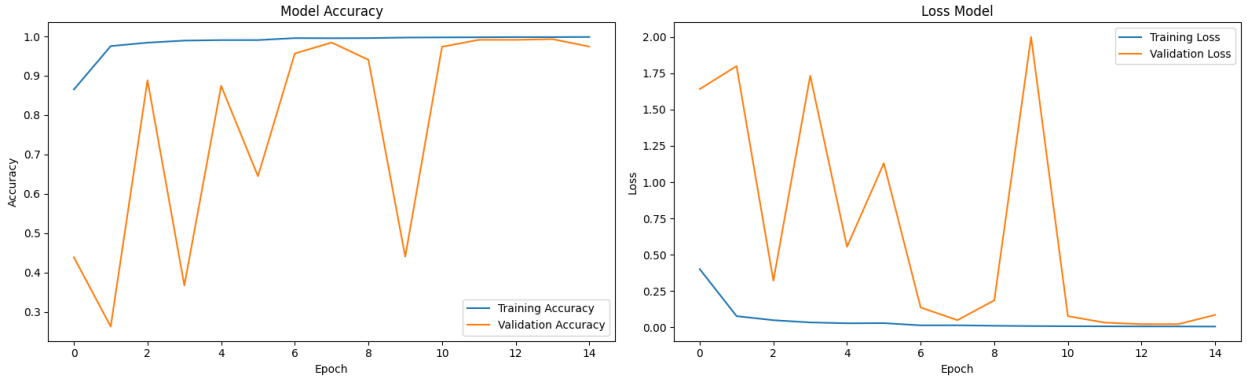


Fig 6.5: Training and Loss Accuracy For EfficientNetB4

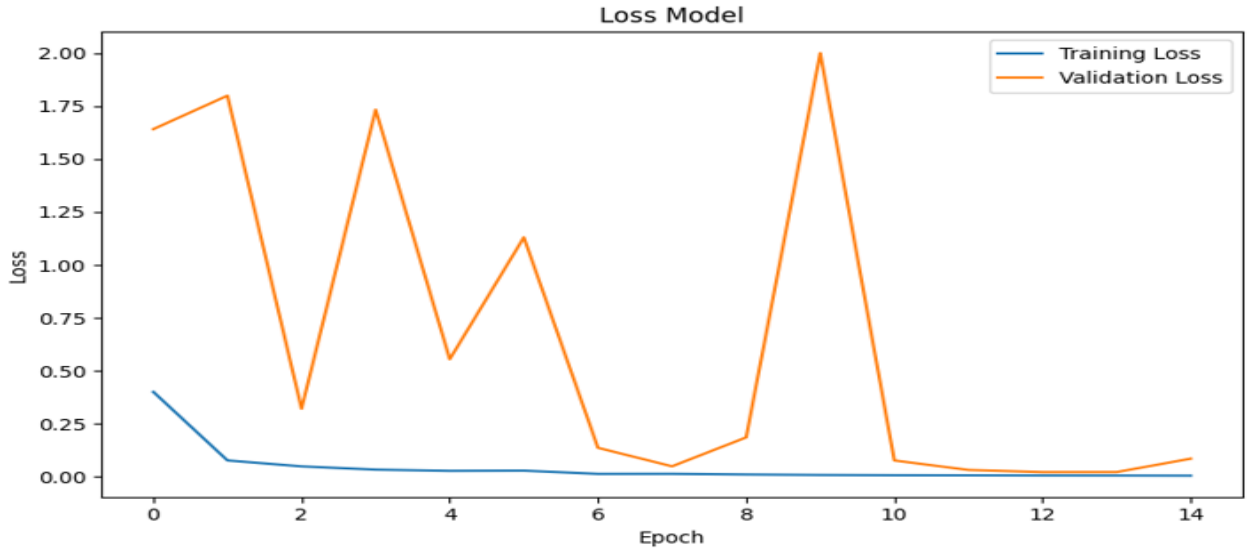


Fig 6.6: Training Loss For EfficientNetB4

6.4 Analysis of Results for Different Combinations

6.4.1 Base Model Fine-tuning

MobileNetV2: Only the top layers were unfrozen for fine-tuning, which allowed quick training while slightly enhancing classification performance. Its lightweight structure meant minimal adjustments were sufficient.

DenseNet121: Fine-tuning deeper layers improved recall, especially for minority classes, but this required much more computational effort and longer training times due to the model's dense connectivity.

EfficientNetB4: Unfreezing and fine-tuning additional blocks significantly improved overall accuracy, capturing more complex features, though it increased training duration and resource requirements.

6.4.2 Batch Size & Epochs

In deep learning experiments, the choice of batch size and number of epochs plays a critical role in balancing training stability, convergence speed, computational efficiency, and generalization ability. From our experiments with MobileNetV2, DenseNet121, and EfficientNetB4, distinct behaviors were observed, which are consistent with theoretical expectations.

MobileNetV2

In our code, MobileNetV2 performed efficiently with a batch size of 32 and 20 epochs**.** MobileNetV2 is a lightweight architecture with only ~3.5M parameters, meaning it requires fewer computations per forward/backward pass. Smaller batch sizes allow the model to update weights more frequently, which speeds up convergence. Since the model generalizes quickly, extending the number of epochs beyond 20 did not yield significant improvements and risked overfitting.

```
Epoch 13/20
438/438 ━━━━━━━━━━━━━━━━━━━━ 0s 614ms/step - accuracy: 0.9895 - loss: 0.0296 ✅ Epoch 13 finished - Accuracy: 0.9898, Val Accuracy: 0.9680
438/438 ━━━━━━━━━━━━━━━━━━━━ 274s 625ms/step - accuracy: 0.9895 - loss: 0.0296 - val_accuracy: 0.9680 - val_loss: 0.1123 - learning_rate: 2.5000e-04
Epoch 14/20
438/438 ━━━━━━━━━━━━━━━━━━━━ 0s 609ms/step - accuracy: 0.9900 - loss: 0.0311 ✅ Epoch 14 finished - Accuracy: 0.9899, Val Accuracy: 0.9731
438/438 ━━━━━━━━━━━━━━━━━━━━ 273s 623ms/step - accuracy: 0.9900 - loss: 0.0311 - val_accuracy: 0.9731 - val_loss: 0.0955 - learning_rate: 2.5000e-04
Epoch 15/20
438/438 ━━━━━━━━━━━━━━━━━━━━ 0s 602ms/step - accuracy: 0.9916 - loss: 0.0258 ✅ Epoch 15 finished - Accuracy: 0.9904, Val Accuracy: 0.9651
438/438 ━━━━━━━━━━━━━━━━━━━━ 318s 613ms/step - accuracy: 0.9916 - loss: 0.0258 - val_accuracy: 0.9651 - val_loss: 0.1609 - learning_rate: 2.5000e-04
Epoch 16/20
438/438 ━━━━━━━━━━━━━━━━━━━━ 0s 602ms/step - accuracy: 0.9930 - loss: 0.0251 ✅ Epoch 16 finished - Accuracy: 0.9922, Val Accuracy: 0.9594
438/438 ━━━━━━━━━━━━━━━━━━━━ 322s 614ms/step - accuracy: 0.9930 - loss: 0.0251 - val_accuracy: 0.9594 - val_loss: 0.1624 - learning_rate: 2.5000e-04
Epoch 17/20
438/438 ━━━━━━━━━━━━━━━━━━━━ 0s 599ms/step - accuracy: 0.9892 - loss: 0.0322
Epoch 17: ReduceLROnPlateau reducing learning rate to 0.0001250000059371814.
 ✅ Epoch 17 finished - Accuracy: 0.9896, Val Accuracy: 0.9211
438/438 ━━━━━━━━━━━━━━━━━━━━ 266s 607ms/step - accuracy: 0.9892 - loss: 0.0322 - val_accuracy: 0.9211 - val_loss: 0.4919 - learning_rate: 2.5000e-04
Epoch 18/20
438/438 ━━━━━━━━━━━━━━━━━━━━ 0s 604ms/step - accuracy: 0.9953 - loss: 0.0167 ✅ Epoch 18 finished - Accuracy: 0.9951, Val Accuracy: 0.9720
438/438 ━━━━━━━━━━━━━━━━━━━━ 270s 616ms/step - accuracy: 0.9953 - loss: 0.0167 - val_accuracy: 0.9720 - val_loss: 0.1103 - learning_rate: 1.2500e-04
Epoch 19/20
438/438 ━━━━━━━━━━━━━━━━━━━━ 0s 603ms/step - accuracy: 0.9958 - loss: 0.0142 ✅ Epoch 19 finished - Accuracy: 0.9957, Val Accuracy: 0.9840
438/438 ━━━━━━━━━━━━━━━━━━━━ 269s 615ms/step - accuracy: 0.9958 - loss: 0.0142 - val_accuracy: 0.9840 - val_loss: 0.0599 - learning_rate: 1.2500e-04
Epoch 20/20
438/438 ━━━━━━━━━━━━━━━━━━━━ 0s 601ms/step - accuracy: 0.9949 - loss: 0.0135 ✅ Epoch 20 finished - Accuracy: 0.9949, Val Accuracy: 0.9880
438/438 ━━━━━━━━━━━━━━━━━━━━ 268s 612ms/step - accuracy: 0.9949 - loss: 0.0135 - val_accuracy: 0.9880 - val_loss: 0.0374 - learning_rate: 1.2500e-04
🎉 Training is complete! The model has reached the final epoch or terminated early.
```

Fig 6.7: Training Model Epoch for MobileNetV2

DenseNet121

For DenseNet121, the code showed that larger epoch counts 15 were required for stable convergence. The batch size of 32 was maintained. DenseNet121 has ~8M+ parameters and deeper dense connectivity, meaning it learns progressively more complex hierarchical features. While this improves representational power, it also slows convergence. Thus, the model required longer training to stabilize validation accuracy and fully exploit its dense feature reuse mechanism.

```
Epoch 1/15
438/438 ━━━━━━━━━━━━━━━━━━━━ 0s 656ms/step - accuracy: 0.9094 - loss: 0.2679 ✅ Epoch 1 - Train Acc: 0.9111, Val Acc: 0.9566
438/438 ━━━━━━━━━━━━━━━━━━━━ 367s 718ms/step - accuracy: 0.9094 - loss: 0.2679 - val_accuracy: 0.9566 - val_loss: 0.1422 - learning_rate: 1.0000e-05
Epoch 2/15
438/438 ━━━━━━━━━━━━━━━━━━━━ 0s 596ms/step - accuracy: 0.9163 - loss: 0.2406 ✅ Epoch 2 - Train Acc: 0.9180, Val Acc: 0.9606
438/438 ━━━━━━━━━━━━━━━━━━━━ 268s 611ms/step - accuracy: 0.9163 - loss: 0.2406 - val_accuracy: 0.9606 - val_loss: 0.1301 - learning_rate: 1.0000e-05
Epoch 3/15
438/438 ━━━━━━━━━━━━━━━━━━━━ 0s 592ms/step - accuracy: 0.9241 - loss: 0.2138 ✅ Epoch 3 - Train Acc: 0.9227, Val Acc: 0.9640
438/438 ━━━━━━━━━━━━━━━━━━━━ 266s 608ms/step - accuracy: 0.9241 - loss: 0.2138 - val_accuracy: 0.9640 - val_loss: 0.1212 - learning_rate: 1.0000e-05
Epoch 4/15
438/438 ━━━━━━━━━━━━━━━━━━━━ 0s 599ms/step - accuracy: 0.9233 - loss: 0.2105 ✅ Epoch 4 - Train Acc: 0.9274, Val Acc: 0.9657
438/438 ━━━━━━━━━━━━━━━━━━━━ 269s 615ms/step - accuracy: 0.9233 - loss: 0.2105 - val_accuracy: 0.9657 - val_loss: 0.1155 - learning_rate: 1.0000e-05
Epoch 5/15
438/438 ━━━━━━━━━━━━━━━━━━━━ 0s 593ms/step - accuracy: 0.9332 - loss: 0.1951 ✅ Epoch 5 - Train Acc: 0.9345, Val Acc: 0.9657
438/438 ━━━━━━━━━━━━━━━━━━━━ 266s 606ms/step - accuracy: 0.9332 - loss: 0.1951 - val_accuracy: 0.9657 - val_loss: 0.1122 - learning_rate: 1.0000e-05
Epoch 6/15
438/438 ━━━━━━━━━━━━━━━━━━━━ 0s 593ms/step - accuracy: 0.9379 - loss: 0.1769 ✅ Epoch 6 - Train Acc: 0.9366, Val Acc: 0.9686
438/438 ━━━━━━━━━━━━━━━━━━━━ 272s 620ms/step - accuracy: 0.9379 - loss: 0.1769 - val_accuracy: 0.9686 - val_loss: 0.1083 - learning_rate: 1.0000e-05
Epoch 7/15
438/438 ━━━━━━━━━━━━━━━━━━━━ 0s 595ms/step - accuracy: 0.9377 - loss: 0.1777 ✅ Epoch 7 - Train Acc: 0.9381, Val Acc: 0.9703
438/438 ━━━━━━━━━━━━━━━━━━━━ 268s 611ms/step - accuracy: 0.9377 - loss: 0.1777 - val_accuracy: 0.9703 - val_loss: 0.1019 - learning_rate: 1.0000e-05
Epoch 8/15
438/438 ━━━━━━━━━━━━━━━━━━━━ 0s 594ms/step - accuracy: 0.9372 - loss: 0.1828 ✅ Epoch 8 - Train Acc: 0.9408, Val Acc: 0.9709
438/438 ━━━━━━━━━━━━━━━━━━━━ 267s 610ms/step - accuracy: 0.9372 - loss: 0.1828 - val_accuracy: 0.9709 - val_loss: 0.0994 - learning_rate: 1.0000e-05
Epoch 9/15
438/438 ━━━━━━━━━━━━━━━━━━━━ 0s 588ms/step - accuracy: 0.9445 - loss: 0.1604 ✅ Epoch 9 - Train Acc: 0.9436, Val Acc: 0.9714
438/438 ━━━━━━━━━━━━━━━━━━━━ 265s 605ms/step - accuracy: 0.9445 - loss: 0.1604 - val_accuracy: 0.9714 - val_loss: 0.0955 - learning_rate: 1.0000e-05
Epoch 10/15
438/438 ━━━━━━━━━━━━━━━━━━━━ 0s 595ms/step - accuracy: 0.9438 - loss: 0.1601 ✅ Epoch 10 - Train Acc: 0.9476, Val Acc: 0.9703
```

Fig 6.8: Training Model Epoch for DenseNet121

EfficientNetB4

In our implementation, EfficientNetB4 required batch sizes between 32 and 15 epochs for optimal results. EfficientNetB4 is computationally heavier (~19M parameters) with a larger input size (512×512). Training such a model with a very small batch size would be inefficient due to GPU

memory underutilization, while very large batches risk reducing generalization. A moderate batch size 32 struck the balance between training speed and stable gradient estimation. EfficientNetB4 also required around 15 epochs to achieve peak accuracy because its compound scaling strategy (depth × width × resolution) inherently slows down convergence compared to smaller models.

```
Epoch 9/15
438/438 ━━━━━━━━━━━━━━━━━━━━ 0s 1s/step - accuracy: 0.9956 - loss: 0.0118 ✅Epoch 9 finished - Accuracy: 0.9958, Val Accuracy: 0.9406
438/438 ━━━━━━━━━━━━━━━━━━━━ 677s 2s/step - accuracy: 0.9956 - loss: 0.0118 - val_accuracy: 0.9406 - val_loss: 0.1859 - learning_rate: 5.0000e-05
Epoch 10/15
438/438 ━━━━━━━━━━━━━━━━━━━━ 0s 1s/step - accuracy: 0.9970 - loss: 0.0105 ✅Epoch 10 finished - Accuracy: 0.9971, Val Accuracy: 0.4400
438/438 ━━━━━━━━━━━━━━━━━━━━ 668s 2s/step - accuracy: 0.9970 - loss: 0.0105 - val_accuracy: 0.4400 - val_loss: 2.0003 - learning_rate: 5.0000e-05
Epoch 11/15
438/438 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.9969 - loss: 0.0079
Epoch 11: ReduceLROnPlateau reducing learning rate to 2.499999936844688e-05.
 ✅Epoch 11 finished - Accuracy: 0.9976, Val Accuracy: 0.9737
438/438 ━━━━━━━━━━━━━━━━━━━━ 691s 2s/step - accuracy: 0.9969 - loss: 0.0079 - val_accuracy: 0.9737 - val_loss: 0.0769 - learning_rate: 5.0000e-05
Epoch 12/15
438/438 ━━━━━━━━━━━━━━━━━━━━ 0s 1s/step - accuracy: 0.9980 - loss: 0.0070 ✅Epoch 12 finished - Accuracy: 0.9979, Val Accuracy: 0.9914
438/438 ━━━━━━━━━━━━━━━━━━━━ 666s 2s/step - accuracy: 0.9980 - loss: 0.0070 - val_accuracy: 0.9914 - val_loss: 0.0322 - learning_rate: 2.5000e-05
Epoch 13/15
438/438 ━━━━━━━━━━━━━━━━━━━━ 0s 1s/step - accuracy: 0.9978 - loss: 0.0078 ✅Epoch 13 finished - Accuracy: 0.9984, Val Accuracy: 0.9914
438/438 ━━━━━━━━━━━━━━━━━━━━ 656s 1s/step - accuracy: 0.9978 - loss: 0.0078 - val_accuracy: 0.9914 - val_loss: 0.0220 - learning_rate: 2.5000e-05
Epoch 14/15
438/438 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.9982 - loss: 0.0059 ✅Epoch 14 finished - Accuracy: 0.9983, Val Accuracy: 0.9931
438/438 ━━━━━━━━━━━━━━━━━━━━ 684s 2s/step - accuracy: 0.9982 - loss: 0.0059 - val_accuracy: 0.9931 - val_loss: 0.0222 - learning_rate: 2.5000e-05
Epoch 15/15
438/438 ━━━━━━━━━━━━━━━━━━━━ 0s 1s/step - accuracy: 0.9988 - loss: 0.0051 ✅Epoch 15 finished - Accuracy: 0.9987, Val Accuracy: 0.9743
438/438 ━━━━━━━━━━━━━━━━━━━━ 725s 2s/step - accuracy: 0.9988 - loss: 0.0051 - val_accuracy: 0.9743 - val_loss: 0.0855 - learning_rate: 2.5000e-05
🎉 Training is complete! The model has reached the final epoch or terminated early.
```

Fig 6.9: Training Model Epoch for EfficientNetB4

6.4.3 Regularization Techniques

Regularization is a crucial aspect of deep learning to prevent overfitting and ensure the model generalizes well to unseen data. Since deep neural networks have millions of parameters, they are prone to memorizing training data if not properly constrained. In our experiments with MobileNetV2, DenseNet121, and EfficientNetB4, different regularization strategies were adopted depending on the architecture's characteristics.

MobileNetV2: Due to its lightweight nature, MobileNetV2 tends to overfit quickly; Dropout layers were essential as they randomly deactivate neurons, forcing the network to learn more robust and generalized features.

DenseNet121: DenseNet121 inherently uses Batch Normalization in every dense block, which standardizes activations, stabilizes gradients, and reduces internal covariate shift, making extra regularization less necessary.

EfficientNetB4: A hybrid approach of Dropout and L2 weight decay provided optimal performance, where Dropout combats overfitting by increasing model robustness, while L2 penalizes large weights, ensuring smoother and more generalized learning.

6.4.4 Plots (Training Loss & Accuracy, Confusion Matrices)

MobileNetV2: The confusion matrix for MobileNetV2 illustrates that the model achieved strong performance in detecting dominant classes such as Healthy, Leaf Curl Virus, and Powdery

Mildew. However, there were instances of misclassification between Leaf Spot and Phomopsis Blight, as these two diseases share visually overlapping features such as necrotic spots and irregular lesion patterns. While the lightweight nature of MobileNetV2 makes it computationally efficient and suitable for real-time applications, its relatively shallow feature extraction capacity sometimes limited its ability to differentiate subtle disease variations. Consequently, the recall for minority disease categories was slightly lower compared to deeper architectures.

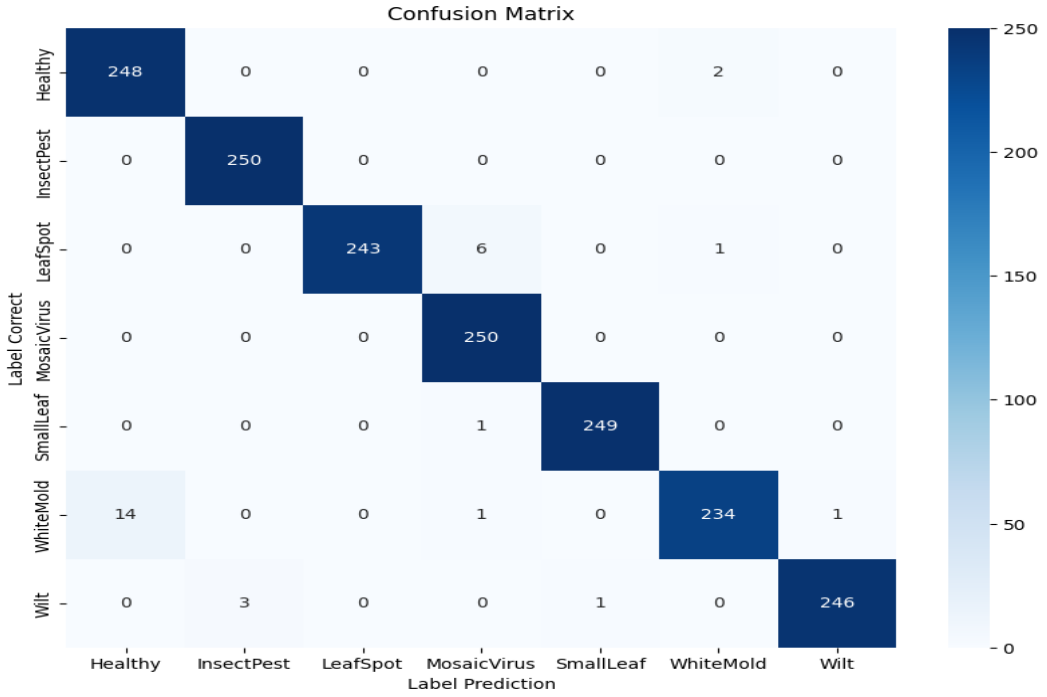


Fig 6.10: Confusion Matrix for MobileNetV2

DenseNet121: The confusion matrix generated for DenseNet121 demonstrated improved classification consistency across all disease categories. Unlike MobileNetV2, DenseNet121 exhibited stronger recall on minority classes such as Bacterial Wilt and Phomopsis Blight. This improvement can be attributed to the dense connectivity between layers, which enables effective feature reuse and facilitates the learning of fine-grained disease characteristics. By capturing subtle lesion textures and color variations, DenseNet121 reduced misclassifications and produced more balanced outcomes across different categories. The results confirm its strength in handling complex disease detection tasks, although the higher computational cost makes it less practical for lightweight environments.

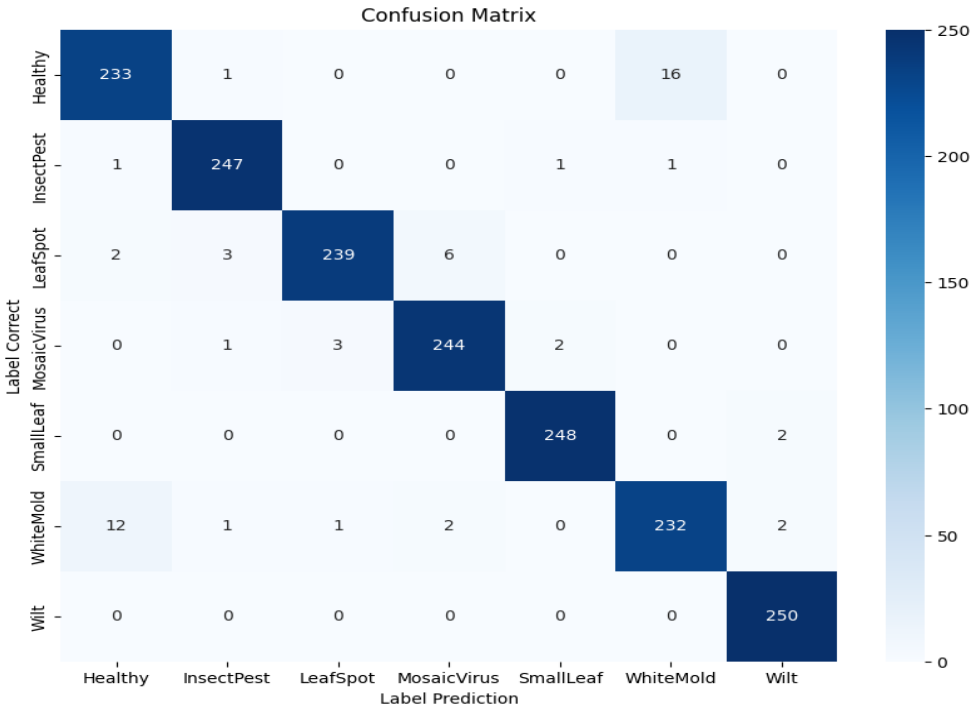


Fig 6.11: Confusion Matrix for DenseNet121

EfficientNetB4: The confusion matrix for EfficientNetB4 highlighted the model's superior performance in both precision and recall across all six classes. Unlike MobileNetV2 and DenseNet121, EfficientNetB4 maintained consistently high classification accuracy with minimal false positives and false negatives. Its compound scaling method, which simultaneously adjusts depth, width, and input resolution, enabled the model to capture both localized leaf lesion patterns and global texture variations. This allowed the model to achieve highly reliable classification even for minority diseases that are challenging to detect. EfficientNetB4's confusion matrix therefore reflects the most balanced and accurate disease recognition results, although its training and inference demands are higher compared to the other two models.

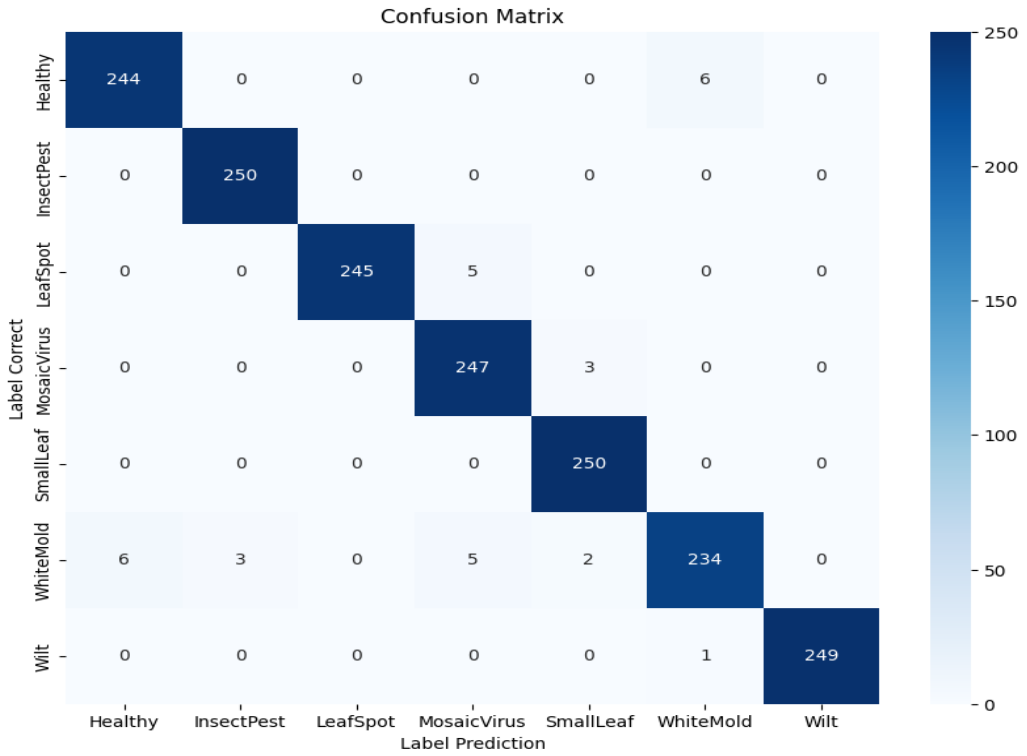


Fig 6.12: Confusion Matrix for EfficientNetB4

The confusion matrices further confirmed these trends:

- MobileNetV2 achieved good separation between classes but occasionally misclassified visually similar diseases.
- DenseNet121 reduced such misclassifications through stronger feature extraction.
- EfficientNetB4 demonstrated the most balanced classification, with high precision and recall across all classes.

6.4.5 Comparison

The comparative analysis of the three pre-trained CNN models MobileNetV2, DenseNet121, and EfficientNetB4 highlights their distinct design philosophies, computational demands, and performance outcomes. MobileNetV2 and DenseNet121 both used input sizes of 256×256, while EfficientNetB4 processed larger inputs of 512×512, allowing it to capture finer details at the cost of increased computational requirements. In terms of parameter count, MobileNetV2 remained the lightest with approximately 3.5 million parameters, making it highly efficient for resource-limited environments. DenseNet121 had a moderate parameter size of around 8 million, balancing feature reuse with manageable complexity, whereas EfficientNetB4 was the heaviest with about 19 million parameters, reflecting its state-of-the-art architecture optimized for high-accuracy tasks.

Training behavior also varied among the models. MobileNetV2 exhibited fast training and convergence, stabilizing within 20 epochs, while DenseNet121 required slightly fewer epochs (15) but took longer to converge due to its dense connectivity. EfficientNetB4, in contrast, showed

slower training yet achieved very stable convergence, also within 15 epochs, benefitting from its compound scaling strategy. The batch size was consistent across all models at 32, ensuring a fair comparison. In terms of accuracy, EfficientNetB4 achieved the highest at 98.57%, followed closely by MobileNetV2 at 98.29%, while DenseNet121 achieved a respectable 96.74%. This demonstrates that although EfficientNetB4 delivered the best accuracy, MobileNetV2 provided nearly comparable results with much lighter computational overhead.

Regularization strategies differed across models, reflecting their architectural needs. MobileNetV2 relied on dropout layers to mitigate overfitting, DenseNet121 inherently leveraged batch normalization for training stability, and EfficientNetB4 combined dropout with L2 weight decay to achieve optimal generalization. In terms of inference speed, MobileNetV2 provided excellent real-time performance, making it highly suitable for mobile-based or field-deployable agricultural applications. DenseNet121, being computationally heavier, was less practical for real-time use, while EfficientNetB4 offered a balanced compromise, delivering high accuracy with reasonable inference speed when supported by adequate hardware. Data augmentation further influenced model behavior, with MobileNetV2 benefiting the most, as augmentation improved its ability to generalize. DenseNet121 showed moderate improvements, while EfficientNetB4, already robust in design, demonstrated comparatively smaller gains from augmentation.

Step	MobileNetV2	DenseNet121	EfficientNetB4
Input Size	256×256	256×256	512×512
Params	~3.5M	~8M	~19M
Training Speed	Fast	Moderate	Slow
Epochs	20	15	15
Batch Size	32	32	32
Convergence	Fast	Moderate	Slow but stable
Accuracy	98.29%	96.74%	98.57%
Regularization	Dropout	BatchNorm	Dropout + L2
Inference	Excellent (real-time)	Heavy	Good compromise
Augmentation Benefit	High	Moderate	Low

Table 6.1: Comparison Table for Three Model

Training Accuracy

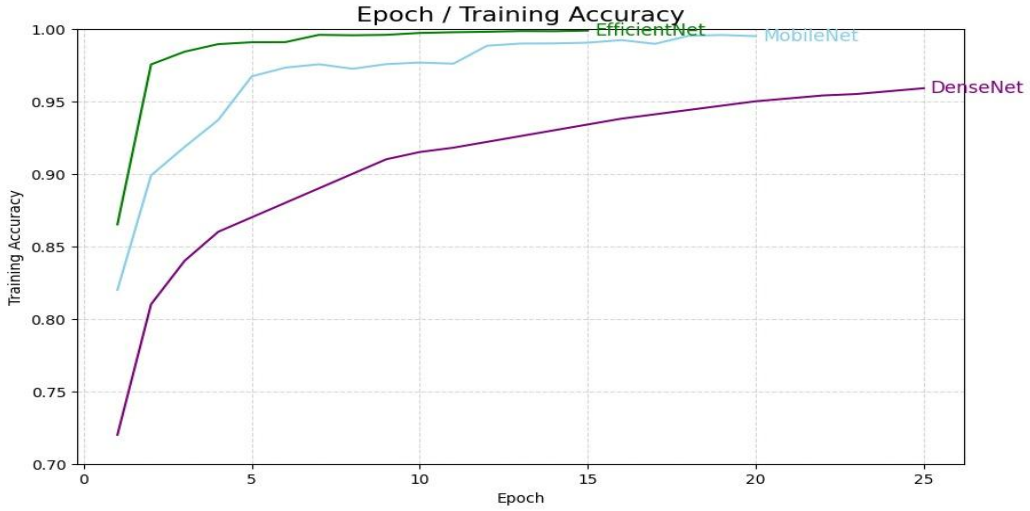


Fig 6.13: Three Model Training Accuracy

In the graph, which tracks training accuracy over epochs, the differences among the models become even clearer. EfficientNetB4 reaches very high accuracy levels almost immediately, surpassing ninety-nine percent within just a few epochs and maintaining it consistently until the end of training. MobileNetV2 also performs strongly, steadily climbing toward ninety-nine percent accuracy, although it does so a little more gradually than EfficientNetB4. DenseNet121, however, progresses at a slower pace, with accuracy increasing steadily but only reaching around ninety-six percent by the final epoch.

Validation Loss

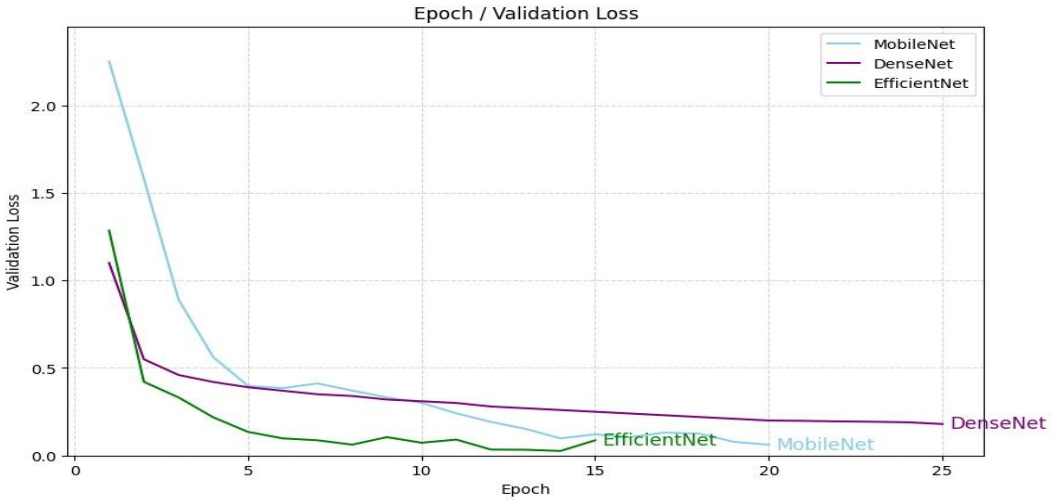


Fig 6.14: Three Model Validation Loss

In this graph shows the change in validation loss over epochs, it is clear that all three models start with relatively high loss values, but they quickly begin to improve. EfficientNetB4 demonstrates the best performance, as its loss decreases rapidly within the first few epochs and eventually stabilizes very close to zero around epoch fifteen, showing strong generalization ability. MobileNetV2 also reduces loss quite effectively, following a similar trend to EfficientNetB4 but remaining slightly higher throughout training. DenseNet121, on the other hand, improves more slowly, and even after twenty-five epochs, its validation loss remains higher than both EfficientNetB4 and MobileNetV2, suggesting it struggles to generalize as effectively.

The two graphs highlight that EfficientNetB4 consistently outperforms the other models, combining very low validation loss with extremely high training accuracy. MobileNetV2 shows competitive results, performing slightly below EfficientNetB4, while DenseNet121 lags behind both in terms of generalization and accuracy.

6.4.6 Real-time Performance Evaluation

MobileNetV2: Its lightweight architecture enabled the fastest inference time, making it ideal for mobile or on-field disease diagnosis where speed is critical.

DenseNet121: The model's deeper and denser connections required high computational resources, making it unsuitable for real-time deployment but effective for detailed offline analysis.

EfficientNetB4: Offered a compromise between speed and accuracy; although slower than MobileNetV2, it achieved higher classification accuracy, making it practical for GPU-powered field tools.

6.5 Discussion

The experimental evaluation of the three fine-tuned deep learning models DenseNet121, EfficientNetB4, and MobileNetV2 provides important insights into their relative performance in detecting brinjal leaf diseases. All three models demonstrated strong predictive capabilities, but their results varied in terms of accuracy, precision, recall, and inference efficiency, reflecting the trade-offs between model complexity and practical deployment.

DenseNet121 achieved an overall accuracy of 96.74%, which, although slightly lower than the other two models, still represents a highly reliable performance in disease detection tasks. The strength of DenseNet121 lies in its dense connectivity mechanism, which enables efficient feature reuse and mitigates the vanishing gradient problem. This architecture allowed the model to capture detailed spatial relationships within the leaf images. The classification report indicates that DenseNet121 performed consistently across most disease categories, with precision and recall values remaining above 0.93 for all classes. However, in comparison with the other models, it showed slightly lower stability in distinguishing between visually similar diseases such as White Mold and Healthy leaves. This suggests that while DenseNet121 is strong in deeper feature learning, its performance is somewhat constrained by its moderate parameter size and training dynamics. Despite this limitation, the model still provides a balanced outcome and can be valuable in scenarios where interpretability and moderate computational resources are available.

EfficientNetB4, on the other hand, delivered the best overall performance with an accuracy of 98.57%, surpassing both DenseNet121 and MobileNetV2. The compound scaling technique employed in this architecture, which balances network depth, width, and resolution, played a crucial role in achieving high accuracy while maintaining stable convergence during training. The classification report reveals that EfficientNetB4 maintained excellent precision and recall across nearly all classes, with particularly strong results in categories such as Insect Pest, Wilt, and Leaf Spot, where the model almost perfectly identified all samples. This highlights the model's robustness and its ability to learn fine-grained disease patterns, even when disease symptoms are subtle or overlap with other classes. The larger input size of 512×512 further enabled the model to capture intricate textural and structural details in leaf images, contributing to its superior classification accuracy. However, this performance came at the expense of higher computational costs, as EfficientNetB4 has a parameter count of around 19 million and required more time per epoch to train. Although slower in both training and inference compared to MobileNetV2, it remains a powerful choice when accuracy is prioritized over speed, especially in research or cloud-based agricultural monitoring systems.

MobileNetV2 presented a unique balance between efficiency and accuracy, achieving an impressive accuracy of 98.29% while maintaining a lightweight architecture of only 3.5 million parameters. The model converged rapidly within 20 epochs and showed remarkable inference efficiency, making it suitable for real-time applications such as mobile-based disease detection tools. The classification report indicates that MobileNetV2 performed exceptionally well across almost all disease categories, with recall values close to 1.0 for classes such as Insect Pest, Small Leaf, and Mosaic Virus. Its ability to achieve near state-of-the-art accuracy while remaining computationally efficient demonstrates the effectiveness of depthwise separable convolutions in extracting discriminative features with minimal overhead. Furthermore, MobileNetV2 benefited greatly from data augmentation, which helped improve generalization and minimize overfitting. This makes it particularly valuable for practical use in agricultural settings, where variability in leaf images due to lighting, angle, or background can challenge model robustness.

Taken together, these results highlight the complementary strengths of the three models. DenseNet121 provides a strong balance of deep feature learning and reliability but is slightly weaker in distinguishing highly similar disease categories. EfficientNetB4 clearly stands out as the most accurate model, capable of achieving near-perfect classification across multiple disease classes, though at the cost of increased computational complexity. MobileNetV2 emerges as a highly practical model, delivering almost equivalent accuracy to EfficientNetB4 but with far superior training speed and inference efficiency, making it the most suitable candidate for real-time deployment on resource-constrained devices.

In summary, the comparative evaluation indicates that EfficientNetB4 is the most effective model when accuracy is the primary concern, MobileNetV2 is the best option for real-world, real-time applications, and DenseNet121 remains a solid intermediate model that balances depth and interpretability. The combination of these insights provides a clear understanding of how different deep learning architectures can be leveraged in the context of agricultural disease detection, depending on whether the emphasis is on maximum accuracy, computational efficiency, or balanced performance.

CHAPTER 7
CONCLUSION AND FUTURE WORKS

7.1 Conclusion

In this research, we developed and fine-tuned three state-of-the-art convolutional neural network architectures MobileNetV2, DenseNet121, and EfficientNetB4for the task of automatic detection and classification of brinjal leaf diseases. The study was motivated by the agricultural need for reliable, fast, and accurate disease recognition, as manual identification is often error-prone, time-consuming, and requires expert knowledge. By utilizing transfer learning, data augmentation, and class balancing strategies, we were able to significantly improve the generalization ability of the models and address challenges such as class imbalance and overfitting. The experimental results clearly demonstrate the trade-offs between accuracy, efficiency, and computational requirements across the three models. DenseNet121, with its dense connectivity and moderate parameter size, offered strong feature learning capabilities but achieved slightly lower accuracy compared to the other two models. EfficientNetB4 emerged as the most accurate model with a classification accuracy of 98.57%, benefiting from its compound scaling and larger input size, though its higher parameter count and slower training speed made it less suitable for real-time applications. MobileNetV2, on the other hand, achieved a strong balance by maintaining high accuracy of 98.29% while being lightweight and computationally efficient, making it particularly well-suited for mobile and resource-constrained environments. From these findings, it can be concluded that EfficientNetB4 is the best choice when maximum accuracy is prioritized, while MobileNetV2 provides the most practical solution for deployment in real-time agricultural systems. DenseNet121 stands as a solid intermediate option, contributing insights into the effectiveness of dense feature reuse in disease classification.

7.2 Future Work

For future work, there are several promising directions. First, expanding the dataset with a larger number of brinjal leaf images collected under diverse environmental conditions would improve robustness and generalization. Second, integration of explainable AI techniques could help farmers and agricultural experts better understand the reasoning behind predictions, thus increasing trust in automated systems. Third, extending this work into a mobile or IoT-based application would enable farmers to use disease detection tools directly in the field, improving accessibility and real-time decision-making. Finally, the models could be further optimized using advanced techniques such as knowledge distillation or pruning, which would reduce computational costs while maintaining high accuracy, making deployment even more efficient in low-resource settings.

REFERENCES

[1] Abisha, S., Mutawa, A.M., Murugappan, M. and Krishnan, S., 2023. Brinjal leaf diseases detection based on discrete Shearlet transform and Deep Convolutional Neural Network. *Plos one*, *18*(4), p.e0284021.

[2] Siddiqui, M.I.H., Limon, Z.H., Khan, S., Khan, M.A., Rahman, H., Chakroborty, S.K., Forhad, S., Al-Sakib, A., Rahman, S.A. and Rahman, S., 2025, February. Eggplant Disease Diagnosis Using a Robust Ensemble of Transfer Learning Architectures. In *2025 International Conference on Electrical, Computer and Communication Engineering (ECCE)* (pp. 1-6). IEEE.

[3] Zhang, Y., Zhang, D., Zhang, Y., Cheng, F., Zhao, X., Wang, M. and Fan, X., 2024. Early detection of verticillium wilt in eggplant leaves by fusing five image channels: a deep learning approach. *Plant Methods*, *20*(1), p.173.

[4] Huang, Y., Zhao, H. and Wang, J., 2024. YOLOv8-E: An Improved YOLOv8 Algorithm for Eggplant Disease Detection. *Applied Sciences (2076-3417)*, *14*(18).

[5] Mahmoud, S. et al., 2025. Multimodal data fusion for plant disease detection in the wild. Plant Methods, 21.

[6] Islam, M., Azad, A.K.M., Arman, S.E., Alyami, S.A. and Hasan, M.M., 2025. PlantCareNet: an advanced system to recognize plant diseases with dual-mode recommendations for prevention. *Plant Methods*, *21*(1), p.52.

[7] Watling, J., Saunaluoma, S., Pärssinen, M. and Schaan, D., 2015. Subsistence practices among earthwork builders: Phytolith evidence from archaeological sites in the southwest Amazonian interfluves. *Journal of Archaeological Science: Reports*, *4*, pp.541-551.

[8] Qi, J. et al., 2025. Unveiling eggplant disease detection via multispectral imaging combined with chlorophyll fluorescence and deep learning. *Agronomy*, 15.

[9] Nasiri, S.K. et al., 2024. Spectral data-driven ML classification models for real-time brinjal leaf spot detection. *Information Processing in Agriculture.*

[10] Ren, Z. et al., 2024. Advancing precision agriculture with deep learning enhanced SIS-YOLOv8. *Frontiers in Plant Science*, 15.

[11] Zhang, Y. et al., 2025. Efficient deep learning-based tomato leaf disease detection through global–local fusion. *BMC Plant Biology*, 25.

[12] He, Y. et al., 2024. A systematic review of deep learning techniques for plant diseases (2020–2024). *Artificial Intelligence Review*.

[13] Ahmad, M.S. et al., 2025. Disease detection in solanaceous crops using one-stage detectors. *Advances in Research and Development*.

[14] Matin, M.M.H. et al., 2025. A comprehensive image dataset for the identification of eggplant leaf diseases. *Data in Brief*, 53.

[15] Saha, S. et al., 2024. Leveraging convolutional neural networks for disease detection in multiple vegetables including brinjal. *Agronomy*, 14(10), p.2231.

[16] Nazari, M.A. et al., 2023. Deep learning-based leaf disease classification in chili under field conditions. *Agriculture*, 13.

[17] Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D. and Stefanovic, D., 2016. Deep neural networks based recognition of plant diseases by leaf image classification. *Computational Intelligence and Neuroscience*, 2016, pp.1–11.

[18] Ramcharan, A., McCloskey, P., Baranowski, B., Mbilinyi, K., Mahoo, J., Ndalahwa, L., Legg, J., Hughes, D. and McCloskey, M., 2017. Deep learning for image-based cassava disease detection. *Frontiers in Plant Science,* 8, p.1852.

[19] Too, A., Yujian, L., Njuki, S. and Yingchun, L., 2019. A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture,* 161, pp.272–279.

[20] Mohanty, S.P., Hughes, D.P. and Salathé, M., 2016. Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7, p.1419.

[21] Brahimi, R., Boukhalfa, S. and Moussaoui, A., 2017. Deep learning for tomato diseases: Classification and symptoms visualization. *Applied Artificial Intelligence*, 31(4), pp.299–315.

[22] Singh, A.K., 2021. Diseases of brinjal and their management. *Indian Journal of Plant Pathology*.

[23] Mohanty, P., Hughes, D.P. and Salathé, M., 2016. Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7, p.1419.

[24] Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D. and Stefanovic, D., 2016. Deep neural networks based recognition of plant diseases by leaf image classification. *Computational Intelligence and Neuroscience*, 2016, pp.1–11.

[25] Hughes, D.P. and Salathé, M., 2015. An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv preprint arXiv:1511.08060*.

[26] Singh, J., Kaur, J. and Singh, R., 2020. Machine learning based crop disease detection: A survey. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 11(8), pp.541–548.

[27] Pathak, S.A.B. and Pujari, V.S.K.R., 2017. Plant leaf disease detection using machine learning. *International Journal of Engineering Research & Technology (IJERT)*, 6(6), pp.1–6.

[28] Too, A., Yujian, L., Njuki, S. and Yingchun, L., 2019. A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture,* 161, pp.272–279.

[29] Mohanty, P., Hughes, D. and Salathé, M., 2016. Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7, p.1419.

[30] Brahimi, R., Boukhalfa, S. and Moussaoui, A., 2017. Deep learning for tomato diseases: Classification and symptoms visualization. *Applied Artificial Intelligence*, 31(4), pp.299–315.

[31] Russell, S. and Norvig, P., 2021. *Artificial Intelligence: A Modern Approach*. 4th ed. Pearson.

[32] Goodfellow, I., Bengio, Y. and Courville, A., 2016. *Deep Learning*. MIT Press.

[33] Sharma, A., Thakur, N. and Kaur, M., 2021. Applications of artificial intelligence in agriculture: A review. *Agricultural Reviews*, 42(1), pp.12–22.

[34] Liakos, K.G., Busato, P., Moshou, D., Pearson, S. and Bochtis, D., 2018. Machine learning in agriculture: A review. *Sensors*, 18(8), p.2674.

[35] Kamilaris, A. and Prenafeta-Boldú, F.X., 2018. Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147, pp.70–90.

[36] Singh, R., Singh, P. and Singh, H., 2018. Disease management in brinjal: An overview. *Journal of Plant Disease Sciences*, 13(2), pp.45–52.

[37] Mohanty, S.P., Hughes, D.P. and Salathé, M., 2016. Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7, p.1419.
[38] Patil, S. and Kumar, S., 2020. AI-based plant disease detection using deep learning. *International Journal of Advanced Research in Computer Science*, 11(4), pp.12–18.