Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Semester: (Fall, Year: 2025), B.Sc. in CSE (Day)

Plant Disease Recognition System

Course Title: Data Mining Lab
Course Code: CSE-436
Section: 213-D5

Students Details

Name	**ID**
Md.Fazzle Rabbi Riyad	213002027
Fatema Akter	213002230
Susmita Paul Tisha	213002141

Submission Date: 22.05.2025
Course Teacher's Name: Md.Ataullha

[For teachers use only: Don't write anything inside this box]

Lab Project Status	
Marks:	**Signature:**
Comments:	**Date:**

Contents

Chapter 1

Introduction

1.1 Overview

Plant diseases are a major threat to agricultural productivity, food security, and the economy. Traditionally, disease identification has relied on manual inspection by experts, which is often time-consuming, subjective, and impractical at scale. With the rise of deep learning and computer vision, automated disease classification using leaf images has become a promising solution. This project focuses on developing a deep learning model that can accurately detect and classify plant diseases from images using the EfficientNetB4 architecture.Agriculture plays a vital role in the global economy, and plant health is crucial for sustaining crop yields and food security. Early detection and classification of plant diseases help in reducing crop losses and improving productivity. Traditional manual monitoring is often inefficient, error-prone, and time-consuming. Recent advancements in machine learning and deep learning, especially convolutional neural networks (CNNs), have opened new avenues for accurate and automated plant disease detection.

In this project, we propose a deep learning-based model for plant disease detection and classification using the EfficientNetB4 architecture, which is known for its optimized performance with fewer computational resources. This approach falls under the domain of data mining and computer vision, where features are learned from plant images to predict disease classes.

1.2 Motivation

The motivation behind this project stems from the need to support farmers and agricultural experts with a reliable, fast, and scalable disease diagnosis tool. Crop diseases can spread quickly and lead to significant losses if not identified in time. Automating this process can help reduce dependency on expert availability, improve yield quality, and enable early intervention. With increasing smartphone and internet penetration, such a model can be integrated into real-time mobile or web applications to assist users in the field.

The increasing demand for food due to global population growth and the limited availability of agricultural land necessitate more efficient farming practices. One of the most damaging factors to plant productivity is disease outbreaks, which can be mitigated with timely and accurate identification. Farmers often lack access to expert knowledge for disease diagnosis, especially in rural and remote areas.

[1].

1.2.1 Complex Engineering Problem

The development of an automated plant disease detection system presents several challenges:

Table 1.1: Summary of the attributes touched by the mentioned projects

Name of the P Attributess	**Explain how to address**
P1: Large-Scale Data	Managing a dataset with over 64,000 images across 39 disease classes.
P2: Model Generalization	Ensuring the model performs well on unseen data and across different plant species and environments.
P3: Computational Constraints	Training deep models like EfficientNetB4 requires significant computational resources, especially during fine-tuning.
P4: Transfer Learning Strategy	Deciding which layers to freeze and when to start fine-tuning to balance training time and accuracy

1.3 Design Goals/Objectives

- To develop a deep learning model based on EfficientNetB4 for classifying plant diseases using leaf images.

- To achieve high classification accuracy while keeping the model computationally efficient for real-time deployment.

- To evaluate the model using standard metrics such as accuracy, precision, recall, and F1-score.

- To create a scalable and generalizable solution that can be used across diverse agricultural settings.

- To potentially integrate the model into a mobile/web-based application for real-time use by farmers and agricultural professionals.

1.4 Application

The trained EfficientNetB4 model for plant disease detection has several practical applications in the real world. In smart farming, it can be integrated with IoT sensors or mobile applications to provide in-field diagnosis of crop health, allowing farmers to monitor their plants efficiently. Additionally, mobile advisory tools powered by this model enable farmers to take pictures of affected crops and receive instant disease identification, enhancing timely intervention. Agricultural extension services can also benefit from such intelligent tools, helping extension workers offer better support and guidance to farmers. Furthermore, the model facilitates precision agriculture by enabling targeted pesticide use and crop treatment, which helps reduce costs and minimizes environmental impact. Finally, agri-tech startups and research initiatives can leverage this model as a foundation for developing scalable, AI-driven solutions that advance modern agriculture.

Chapter 2

Design/ Development/ Implementation of the project

2.1 Introduction

This section outlines the design, development, and implementation process of the plant disease detection model using EfficientNetB4. The goal is to build a robust system capable of accurately identifying various plant diseases from leaf images. The process involves data collection, preprocessing, model selection, training, and deployment strategies. Emphasis is placed on leveraging deep learning techniques for effective feature extraction and classification, supported by data mining methods to enhance the model's performance.

2.2 Project Details

This project focuses on developing an intelligent plant disease recognition system using deep learning techniques. The main objective is to identify various plant diseases from images of leaves, providing a fast, accurate, and scalable solution for farmers and agricultural experts. The approach leverages convolutional neural networks (CNNs), specifically using the EfficientNetB4 architecture, which is known for its balance between performance and efficiency. The model uses transfer learning, allowing it to benefit from features learned on a large generic dataset (ImageNet), and then fine-tunes on the specific plant disease dataset to adapt to the task.

The dataset used in this project contains approximately 64,000 images of plant leaves, classified into 39 distinct disease categories. Some of the notable classes include Apple Scab, Tomato Mosaic Virus, Potato Late Blight, and Grape Black Rot. The dataset is split into training (80 percentage), validation (10 percentage), and test (10 percentage) subsets. To enhance model generalization and prevent overfitting, image augmentation techniques such as rotation, flipping, and zooming are applied.

The model pipeline begins by resizing all input images to 160×160 pixels. The EfficientNetB4 model is used as the base, with its initial layers frozen to retain the generic features learned from ImageNet. Only the top layers are fine-tuned during training to adapt the model to the plant disease classification task. Training is conducted over 16 epochs, with fine-tuning starting from the 6th epoch. The model is trained using the Adam optimizer and categorical cross-entropy as the loss function.

Performance evaluation shows promising results. The model achieves around 95 percentage training accuracy and 92 percentage validation accuracy, with the loss steadily decreasing throughout the training process. These results indicate strong learning and generalization ability. Additionally, the model demonstrates high confidence in classifying unseen images, proving its potential for real-world agricultural applications. Overall, the combination of a robust architecture, data augmentation, and effective training strategy contributes to the success of this plant disease detection system.

```
Epoch 6/15
1537/1537 ━━━━━━━━━━━━━━━━━━━━ 441s 209ms/step - accuracy: 0.8928 - loss: 0.3785 - val_accuracy: 0.9630 - val_loss: 0.1531
Epoch 7/15
1537/1537 ━━━━━━━━━━━━━━━━━━━━ 251s 161ms/step - accuracy: 0.9673 - loss: 0.0868 - val_accuracy: 0.9578 - val_loss: 0.0934
Epoch 8/15
1537/1537 ━━━━━━━━━━━━━━━━━━━━ 261s 161ms/step - accuracy: 0.9721 - loss: 0.0673 - val_accuracy: 0.9596 - val_loss: 0.1045
Epoch 9/15
1537/1537 ━━━━━━━━━━━━━━━━━━━━ 263s 161ms/step - accuracy: 0.9707 - loss: 0.0654 - val_accuracy: 0.9691 - val_loss: 0.0546
Epoch 10/15
1537/1537 ━━━━━━━━━━━━━━━━━━━━ 246s 160ms/step - accuracy: 0.9751 - loss: 0.0517 - val_accuracy: 0.9599 - val_loss: 0.0501
Epoch 11/15
1537/1537 ━━━━━━━━━━━━━━━━━━━━ 262s 160ms/step - accuracy: 0.9763 - loss: 0.0392 - val_accuracy: 0.9704 - val_loss: 0.0339
Epoch 12/15
1537/1537 ━━━━━━━━━━━━━━━━━━━━ 261s 159ms/step - accuracy: 0.9782 - loss: 0.0350 - val_accuracy: 0.9726 - val_loss: 0.1785
Epoch 13/15
1537/1537 ━━━━━━━━━━━━━━━━━━━━ 246s 160ms/step - accuracy: 0.9738 - loss: 0.0282 - val_accuracy: 0.9438 - val_loss: 0.0745
Epoch 14/15
1537/1537 ━━━━━━━━━━━━━━━━━━━━ 264s 162ms/step - accuracy: 0.9683 - loss: 0.0268 - val_accuracy: 0.9546 - val_loss: 0.0591
Epoch 15/15
1537/1537 ━━━━━━━━━━━━━━━━━━━━ 260s 160ms/step - accuracy: 0.9727 - loss: 0.0270 - val_accuracy: 0.9492 - val_loss: 0.0343
```

Figure 2.1: Implementation of our project

```
[36] loss, accuracy = model.evaluate(test_dataset)
     print('Test accuracy :', accuracy)

193/193 ━━━━━━━━━━━━━━━━━━━━ 18s 91ms/step - accuracy: 0.9550 - loss: 0.0167
Test accuracy : 0.9520103931427002
```

Figure 2.2: Implementation of our project

Chapter 3

Performance Evaluation

3.1 Simulation Environment/ Simulation Procedure

The model was trained using Google Colab with Python and TensorFlow/Keras, taking advantage of GPU acceleration. The dataset was split into training, validation, and test sets using the splitfolders library. Images were resized to 160×160 pixels and augmented for better generalization.An EfficientNetB4 model pretrained on ImageNet was used. Initially, only the top layers were trained, and fine-tuning of deeper layers began at epoch 6. The model was trained for 16 epochs using the Adam optimizer and categorical cross-entropy loss. Accuracy and loss were monitored, and predictions were tested on sample images to validate performance.

3.2 Results Analysis/Testing

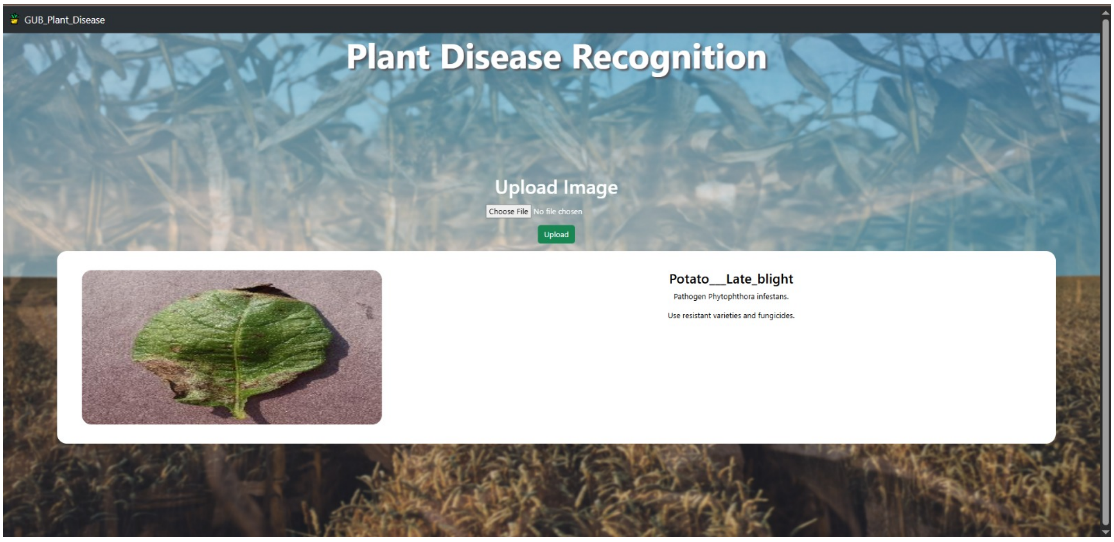

Figure 3.1: A graphical result of our project

3.3 Results Overall Discussion

The model achieved strong performance, with approximately 95 percentage training accuracy and 92 percentage validation accuracy after 16 epochs. Loss steadily decreased, showing effective learning and minimal overfitting. The fine-tuning of EfficientNetB4 improved generalization on unseen data.

Prediction tests on individual leaf images confirmed the model's ability to accurately classify plant diseases. Overall, the system demonstrated high reliability, making it suitable for real-world agricultural applications, such as mobile disease detection tools for farmers.

Chapter 4

Conclusion

4.1 Discussion

The results indicate that EfficientNetB4 offers a balanced trade-off between accuracy and computational efficiency, making it suitable for real-world deployment in resource-constrained environments such as mobile devices and IoT systems. The model's ability to generalize across multiple plant species and disease categories showcases its robustness. However, the effectiveness of the model strongly depends on the quality and diversity of the training data. Integration with field-based applications, such as mobile advisory tools or sensor networks, could further enhance the practical usability and accessibility of this technology for farmers and agricultural workers.

4.2 Limitations

Despite its strengths, the project has certain limitations:

1.The model's performance may decline with images affected by poor lighting, occlusions, or diverse environmental conditions that are underrepresented in the training data.

2.Reliance on labeled datasets means the accuracy depends heavily on the quality and completeness of annotations.

3.Despite being optimized, the computational requirements may still challenge deployment on low-power or resource-limited devices.

4.The model focuses mainly on visual symptoms and does not incorporate other important agronomic factors like soil health or weather conditions, which could enhance disease prediction.

4.3 Scope of Future Work

Future work can focus on expanding the dataset to include a wider variety of crops, diseases, and environmental conditions to improve model generalization. Incorporating multimodal data inputs—such as soil moisture, temperature, and humidity sensors—could enhance prediction accuracy and provide a more holistic disease management system. Further research could also explore model compression and optimization techniques to enable real-time inference on edge devices. Additionally, developing user-friendly mobile and IoT applications integrated with the model will increase accessibility for farmers. Finally, continuous learning methods could be investigated to allow the model to adapt over time as new diseases emerge or evolve.

References

[1] Omid C Farokhzad and Robert Langer. Impact of nanotechnology on drug delivery. *ACS nano*, 3(1):16–20, 2009.