

محمود الراعي

Final Version Of paper -V2.docx

 Midocean University

Document Details

Submission ID

trn:oid::2945:400374845

Submission Date

Jul 19, 2026, 10:20 PM GMT+3

Download Date

Jul 20, 2026, 3:00 PM GMT+3

File Name

Final Version Of paper -V2.docx

File Size

1.1 MB

18 Pages

6,054 Words

34,737 Characters

19% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

-  **124 Not Cited or Quoted 19%**
Matches with neither in-text citation nor quotation marks
-  **0 Missing Quotations 0%**
Matches that are still very similar to source material
-  **0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
-  **0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 10%  Internet sources
- 12%  Publications
- 15%  Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- 124 Not Cited or Quoted 19%**
Matches with neither in-text citation nor quotation marks
- 0 Missing Quotations 0%**
Matches that are still very similar to source material
- 0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 10% Internet sources
- 12% Publications
- 15% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

- 1** **Publication**
"Proceedings of 5th International Conference on Artificial Intelligence and Smart ... <1%
- 2** **Submitted works**
Queensland University of Technology on 2026-05-31 <1%
- 3** **Internet**
arxiv.org <1%
- 4** **Publication**
Gadhgadhi, Khalil. "Evaluation and Optimization of Machine Learning Techniques... <1%
- 5** **Publication**
Parvathaneni Naga Srinivasu, Vemuri Anirudh, Devarasetty Likhitha Kamali, Som... <1%
- 6** **Submitted works**
Librarian NITT on 2026-07-17 <1%
- 7** **Submitted works**
Midocean University on 2026-05-02 <1%
- 8** **Internet**
dev-journal-bucket.s3.amazonaws.com <1%
- 9** **Internet**
www.jetir.org <1%
- 10** **Submitted works**
University of Northumbria at Newcastle on 2023-08-24 <1%

11	Internet	ir.vistas.ac.in	<1%
12	Submitted works	Anna University on 2026-04-17	<1%
13	Internet	download.bibis.ir	<1%
14	Internet	iieta.org	<1%
15	Submitted works	Islington College,Nepal on 2026-05-10	<1%
16	Submitted works	Liverpool John Moores University on 2026-07-02	<1%
17	Submitted works	MIT-ADT University on 2026-06-24	<1%
18	Submitted works	University of Petroleum and Energy Studies on 2026-05-12	<1%
19	Internet	www.mdpi.com	<1%
20	Submitted works	Berlin School of Business and Innovation on 2026-05-14	<1%
21	Submitted works	Universiti Kuala Lumpur on 2026-06-08	<1%
22	Submitted works	Amrita Vishwa Vidyapeetham on 2026-06-08	<1%
23	Publication	Pushpa Choudhary, Sambit Satpathy, Arvind Dagur, Dharendra Kumar Shukla. "Re...	<1%
24	Submitted works	Glyndwr University on 2023-05-05	<1%

25	Publication	Niraj Pandey, Denish Oli. "Deep-Learning Based Tomato Leaf Disease Classificatio...	<1%
26	Internet	www.parasitestudy.com	<1%
27	Publication	"Proceedings of the Fifth International Conference on Trends in Computational a...	<1%
28	Internet	ijarcce.com	<1%
29	Publication	"Proceedings of Fourth International Conference on Computing and Communicat...	<1%
30	Publication	Cenk Temizel, Salih Tutun, Celal Hakan Canbaz, Ekrem Alagoz et al. "Artificial Inte...	<1%
31	Internet	www.irjweb.com	<1%
32	Publication	Thangaprakash Sengodan, Sanjay Misra, M Murugappan. "Advances in Electrical ...	<1%
33	Submitted works	University of Bradford on 2026-01-08	<1%
34	Internet	www.cancerbiosci.com	<1%
35	Publication	Lafrance, Samantha. "A Multi-Scale Fusion CNN-ViT Hybrid Architecture for Semic...	<1%
36	Internet	dokumen.pub	<1%
37	Internet	www.crawsec.com	<1%
38	Internet	www.ijacm.com	<1%

39	Submitted works	College of Public Health on 2026-03-09	<1%
40	Publication	Umeike, Chibuike Robinson. "An Interoperable Framework for Real-Time Infrastr...	<1%
41	Submitted works	University of East London on 2026-05-08	<1%
42	Internet	www.ijjsae.org	<1%
43	Internet	www.indjst.com	<1%
44	Internet	www.preprints.org	<1%
45	Publication	"AgriTech Revolution", Springer Science and Business Media LLC, 2025	<1%
46	Publication	"An Enhanced Plant Disease Classifier Model Based on Deep Learning Techniques...	<1%
47	Submitted works	Al-Hussein Bin Talal University on 2026-06-14	<1%
48	Submitted works	Buckinghamshire Chilterns University College on 2026-02-24	<1%
49	Submitted works	Cornell University on 2023-12-11	<1%
50	Submitted works	Haramaya University on 2026-05-11	<1%
51	Publication	Hernández, Valeria Martín. "Building Labeled Satellite Imagery Datasets and Ben...	<1%
52	Submitted works	Liverpool John Moores University on 2025-05-26	<1%

53	Submitted works	Liverpool John Moores University on 2026-06-21	<1%
54	Submitted works	MNS University of Engineering & Technology-Multan on 2026-07-18	<1%
55	Submitted works	University of Hertfordshire on 2025-04-28	<1%
56	Internet	ijcrt.org	<1%
57	Internet	ijesty.org	<1%
58	Internet	irjet.net	<1%
59	Internet	par.nsf.gov	<1%
60	Internet	prip.by	<1%
61	Internet	www.kaggie.com	<1%
62	Submitted works	Charotar University of Science And Technology on 2021-04-23	<1%
63	Submitted works	Griffth University on 2025-10-14	<1%
64	Submitted works	Liverpool John Moores University on 2020-11-30	<1%
65	Submitted works	Liverpool John Moores University on 2021-01-31	<1%
66	Submitted works	Liverpool John Moores University on 2025-07-20	<1%

67	Submitted works	Munster Technological University (MTU) on 2025-06-12	<1%
68	Submitted works	St Ives High School on 2026-07-13	<1%
69	Submitted works	University of Stirling on 2026-04-29	<1%
70	Submitted works	University of Westminster on 2026-03-30	<1%
71	Publication	V. Sekhar, P. Ajay Kumar Reddy, K. Logesh, T. Raghunadha Reddy, Dara Raju. "Ne...	<1%
72	Internet	dspace.daffodilvarsity.edu.bd:8080	<1%
73	Internet	ebin.pub	<1%
74	Internet	mdpi-res.com	<1%
75	Internet	www.ijraset.com	<1%
76	Submitted works	Charotar University of Science And Technology on 2023-04-21	<1%
77	Submitted works	Dr.Mahalingam college of engineering and technology on 2026-04-10	<1%
78	Submitted works	Dublin City University on 2026-04-13	<1%
79	Publication	Amreen Batool, Yung-Cheol Byun. "Revolutionizing plant disease diagnosis throu...	<1%
80	Submitted works	DU Faculty Members on 2026-05-09	<1%

81	Submitted works	
East Delta university on 2025-07-08		<1%
82	Submitted works	
Islington College,Nepal on 2026-05-10		<1%
83	Submitted works	
LTI on 2025-07-03		<1%
84	Submitted works	
The Northcap University on 2026-07-18		<1%
85	Submitted works	
University Metropolitan Tirana on 2026-06-22		<1%

7

**Midocean Journal for Research and Studies**

8

ORCID ID (You can obtain it from this link: <https://orcid.org/signin>) Specialization – College – University Affiliation – Country (Font Size 12) Email (Font Size 12 in Black)

Accuracy and Efficiency Learning Models for Plant Leaf Disease Classification

Mahmoud Al Ra'e – Hussein Mudeheg

7

ORCID ID (You can obtain it from this link: <https://orcid.org/signin>)
Artificial Intelligence – College of Informatics – Midocean University

Supervisor: Dr. Hager Saleh (ORCID ID) Specialization, College, University, Country, Email

1. Introduction

Plant diseases reduce crop quality and early diagnosis remains difficult when disease symptoms are visually similar. Conventional diagnosis usually depends on human inspection and specialist experience. This approach can be slow, subjective and difficult to scale across large farms. Image based artificial intelligence offers a practical alternative by learning visual patterns associated with healthy and diseased leaves.

Deep learning has become a dominant approach for visual recognition because convolutional neural networks can learn hierarchical features directly from images. Transfer learning further reduces the data and computation required by reusing representations learned from large-scale image datasets. In plant pathology, transfer-learning architectures such as ResNet, EfficientNet, and MobileNet have frequently demonstrated strong classification performance. Nevertheless, model selection should not depend on accuracy alone. Large networks may provide higher predictive performance but require more memory and computation, whereas compact networks may be more suitable for mobile or edge deployment.

A further challenge is that many published studies use different datasets, class definitions, preprocessing procedures, training settings, and evaluation measures. Their reported accuracies are therefore not always directly comparable. A controlled comparison using the same dataset split, preprocessing pipeline, classification head, optimizer, and evaluation protocol is needed to distinguish architectural effects from experimental differences.

This study compares ResNet50, EfficientNetB0, and MobileNetV2 under a unified transfer-learning protocol using 20,638 PlantVillage images across 15 plant health and disease classes. The study makes three contributions. First, it provides a controlled comparison of three widely used architectures. Second, it evaluates predictive performance together with model size and training time. Third, it analyzes class-level errors to identify disease categories that remain difficult to separate visually.

2. Research Problem

Plants diseases are among the major concerns affecting the quality and production levels of crops. The farmers often determine the occurrence of diseases in the plant through experience which may be unreliable. Early symptoms of plant disease may be difficult to recognize leading to delay in diagnosis, thus cause more damage to the plant. Some diseases in plants look alike. Therefore, it could lead to a false conclusion in the diagnosis. Detecting the diseases of plants in their natural environment becomes very difficult owing to many factors including illumination, occlusion, and leaf orientation of the plant. Normally, checking of the plants requires manual labor. The detection of the diseases from each individual plant becomes time consuming particularly when dealing with many of them or a larger piece of farm. It is necessary to develop a system which will enable farmers to diagnose the diseases automatically. Artificial Intelligence and deep learning would help the farmers detect diseases faster and make informed decisions concerning the crops. Plant diseases account for about 40% annual losses globally. Plant diseases lead to huge financial losses annually, and they have been estimated to cost about 220 billion dollars annually globally.

3. Importance of the Study

Diseases of plants make up between 30 to 40 percent of crop loss in the world costing the economy about \$220 billion per year. Plants contribute to around 80 percent of the food that humans consume. Diagnosing plant diseases is largely dependent on the knowledge of the farmers and visual observations but this process is very inefficient, prone to errors, and difficult to scale due to the vast size of the farms. The problem gets even more complicated since many diseases share similar symptoms causing frequent misdiagnosis while the varying environmental factors such as lighting, backgrounds, and camera angles create additional difficulties for detecting diseases manually and using imagery. It is important to diagnose the diseases at their initial stage in order to reduce crop loss and minimize the unnecessary use of pesticides.

4. Study Objectives

The overall goal of this paper is to design an intelligent system that detect plant diseases based on the leaf images of the plants using AI and deep learning techniques. The study aims to improve the accuracy and speed of disease diagnosis to support early detection of the disease by applying and assessing different DL models for plant disease classification, Convolutional Neural Networks (CNNs), Vision Transformers (ViT), and Swin Transformers. This thesis also aims to compare these models in terms of classification accuracy, computational efficiency and applicability to real world agricultural environments. Another specific goal is to study the ability of these models to work under practical conditions such as different lighting, background complexity and leaf position.

5. Theoretical Literature of the Study

Convolutional Neural Networks (CNN) use convolutional filters to automatically extract hierarchical image features edges and textures in early layers, and progressively complex class-discriminative patterns in deeper layers removing the need for hand crafted feature engineering. Transfer learning reuses a model pertained on a large dataset as the starting point for a related task. ResNet50 introduces connections so layers learn residual functions relative to their inputs and easing gradient flow in very deep networks and mitigating vanishing gradients.

EfficientNetB0 use compound scaling balancing network depth, width and input resolution together via a fixed ratio to achieve strong accuracy with a compact parameter count.

MobileNetV2 use inverted residuals with linear bottlenecks and depth wise convolutions, sharply reducing parameters and computation that making it suited to mobile evaluation Metrics. Accuracy, precision, recall and F1-score are derived from per-class true/false positive and negative counts.

6. Review of Previous Studies

Too et al. applied multiple types of pertained models such as VGG16, InceptionV4, ResNet50, and DenseNet121 which classify 38 different classes of plant diseases. The public dataset so called PlantVillage was used from KAGGLE, which has 54,306 images with 38 classes. The preprocessing steps included image resizing in order to match the required input size of the CNN architecture, and rotation, flipping, and scaling were applied to increase dataset variability and reduce overfitting. The result has shown that DenseNet121 achieved the highest classification accuracy among the evaluated models. Transfer learning significantly improved performance compared to training from scratch. The models demonstrated very high accuracy under laboratory conditions. [7]

The study examines the use of Vision Transformers (ViT) for plant disease classification to determine whether attention-based architectures can perform better than the traditional convolutional neural networks (CNNs) or not. A pre-trained ViT model on ImageNet was improved for plant disease detection and compared with CNN models such as ResNet50 and DenseNet121. The model utilizes five public plant datasets that include the famous PlantVillage dataset, which contains 54,306 images from 38 classes that were collected under controlled conditions. The preprocessing involved resizing the images to 244X244 pixels, patch embedding preparation, pixel normalization, and data augmentation such as rotation and flipping. The final results demonstrate that ViT achieved approximately 99% accuracy, which outperforms several models. [8]

The research proposed a lightweight deep learning model, so called Dise-Efficient which is built on EfficientNetV2 to identify plant diseases and pest identification. The study focused on improving model efficiency while ensuring strong recognition performance. The model was trained and evaluated using plant image datasets, using steps such as image resizing, normalization, and data augmentation for preprocessing, which were applied to enhance performance and generalization. The results show that the proposed model has achieved high accuracy of classification as well as reduced the computational complexity. [9]

The authors have proposed an integrated Internet of Things (IoT) solution that has combined deep learning techniques to detect plant diseases. Leaf images were captured through cameras along with environmental data collected from IoT sensors such as temperature and humidity. The purpose of the preprocessing techniques is to improve image quality prior to applying deep learning models for disease classification. The system includes training and evaluation stages that measure the performance of the model, which resulted in high accuracy. In addition, the framework supports real-time monitoring and feedback, allowing continuous tracking of crop conditions and early detection of diseases. [10]

Aldakheel et al. proposed a real time plant leaf disease detection system based on the YOLOv4 object detection framework. In contrast to the traditional classification models that only predict disease categories, YOLOv4 enables simultaneous detection of diseased regions within leaf images. The methodology involved training the YOLOv4 model to identify and classify multiple plant diseases directly from leaf images by bounding box coordinates and class probabilities in a single detection process. The model was trained on annotated plant leaf disease datasets that contain multiple disease categories in several environments. The preprocessing steps included image resizing to match the YOLOv4 input dimensions, image normalization, and data augmentation such as rotation, scaling, and flipping to improve robustness and at the same time

to reduce the overfitting. The result of the experimentation demonstrated that YOLOv4 achieved higher accuracy with fast inference speed which making it suitable for real time agricultural applications. [11]

Duhan et al. have designed a real-time plant disease diagnosis based on mobile and edge devices. The authors have combined the RTR_Lite framework with MobileNetV2, a compact convolutional neural network known for its use of separable convolutions, which significantly reduce computational cost and still maintain strong performance. The goal was to develop a model that balances between accuracy and speed, which makes it very practical for field deployment where the computational resources are always limited. The model was trained on available public plant disease datasets that include various crops and disease categories which were captured under different environmental conditions. Prior to commencing the training the images were resized to match the input size required by MobileNetV2, and pixel values were normalized. Data augmentation techniques such as rotation, flipping, and scaling were applied to improve the ability of the models to generalize. The results showed that RTR_Lite_MobileNetV2 achieved high classification accuracy and at the same time has maintained a small model size and fast inference speed. [12]

Bhola and Kumar have designed a CNN model that automatically learns the important visual features from leaf images which eliminates the need for manual feature extraction. The main objective was to support farmers in detecting diseases at an early stage so that timely treatment can reduce crop damage and yield loss. Pre-trained CNN models such as VGG16, ResNet-50, and Inception-v3 were used. The model was trained and tested on a dataset of vegetable leaf images that included several disease categories, collected under both controlled and natural environmental conditions. Before starting the training, the images were resized to fit the input requirements of the network and normalized to standardize pixel values. Data augmentation techniques such as rotation, flipping, and brightness adjustment were also applied to increase dataset diversity and accordingly improve the ability of the models to generalize. The results showed that the proposed model achieved high accuracy and performed well in recognizing early symptoms of the disease. [13]

Kheir et al. have proposed a hybrid model combined of Vision Transformer (ViT) and MobilePlantViT, the lightweight design generalized plant disease image classification and balance both accuracy and efficiency for mobile and edge deployment. Traditional deep learning models for plant disease recognition often struggle on resource limited devices caused by high computational costs. MobilePlantViT combines efficient convolutional operations with a Vision Transformer backbone to capture local and global features while maintain the architecture compact. The model was evaluated across multiple diverse plant disease datasets that has shown a strong generalization and test accuracies ranging from around 80% to over 99% depending on the dataset. Preprocessing consisted of the usual image resizing and normalization, along with data augmentation to improve its reliability and prevent overfitting. The outcomes from the experiments highlight the potential that MobilePlantViT holds for practical disease detection in plants. [14]

Thanjaivadivel developed EnConv which is an improved CNN, meant to accurately classify plant leaves and their diseases using more efficient feature extraction techniques. The objective is to use ML and DL to detect diseases in plants which does not need extensive knowledge in the field. EnConv uses deep convolutional layers, which optimize calculations and retain crucial properties such as leaf colors and shapes. Instead of having a flattening layer as in most CNNs, EnConv has global pooling layers that can reduce overfitting and improve performance. After

testing EnConv on data from a total of 39 diseases from different crops, it had a maximum accuracy rate of 99.87%. This clearly indicates that EnConv is more accurate than other machine learning and deep learning models. [15]

In this research, we have proposed a deep learning and IoT-based framework for detecting plant leaf diseases in real-time. The proposed framework relies on a CNN algorithm to analyze leaf images and determine whether the plant is healthy or unhealthy. For improving the efficiency of pre-trained models such as MobileNetV2, ResNet50, and VGG16, transfer learning technology is used, along with datasets including PlantVillage. An IoT camera is used for taking images from the leaves and performing operations such as removing noise, adjusting size, and applying data augmentation techniques to enhance reliability irrespective of the weather conditions. Furthermore, the developed model is capable of executing at edge computing devices including Raspberry Pi or Jetson Nano. The model has an accuracy rate of more than 90%. [16]

The paper explains a framework that utilizes DL techniques to identify and classify plant leaf diseases is introduced. A custom model named CropLeafNet is constructed using reliable backbone architectures ResNet50, MobileNetV2, and VGG16 to enhance feature detection and classification processes. Using the concept of transfer learning, the proposed framework identifies not only the local characteristics such as the leaf textures but also the global features including the leaf colors and patterns found all around the world. This makes it possible to diagnose a variety of diseases through one single model. For training purposes, datasets such as PlantVillage as well as photographs from the field are employed to guarantee a wide range of leaf environments. Image preprocessing consists of resizing and normalizing operations along with some augmentation operations including rotation, flipping, and contrast changing. From the experimental results, it is clear that the developed model named CropLeafNet (CNN architecture for agricultural problems) is able to classify the diseases with high accuracy and performs better than other standard CNN models. [17]

In this study, an ML approach is proposed to detect plant disease in pigeon pea leaves, intending to provide assistance for early diagnosis for farmers. This paper explicitly uses no deep neural network nor other recent large-scale models called black boxes. The researchers propose a novel ML solution to the problem by combining classical computer vision techniques and classical classifiers. Dense SIFT (DSIFT) and Bag of Visual Words (BoVW) techniques are used as feature extractors that have proven themselves to be robust and invariant to noise and rotations in previous works. This framework tests several classifiers, including SVM, KNN, Random Forest, Decision Tree, Gradient Boosting, and a CNN classifier, in order to classify the pigeon pea leaf disease categories using their extracted descriptors. The dataset consists of 900 wild images of pigeon pea leaves (diseased and healthy). The preprocessing stage involved image size adjustment and enhancement of quality. It was discovered that the CNN classifier and the LightGBM classifier were efficient in classifying pigeon pea leaves into categories. [18]

PlantDoc was designed as a real-life dataset, which is supposed to aid in improving the abilities of artificial intelligence to recognize plant diseases under non-laboratory circumstances. As such, all the images used in PlantDoc come from a variety of farms, thus being quite messy, with lots of background information, inconsistent illumination, and pictures shot from a wide range of perspectives. A series of deep learning models were chosen by the authors to see how successfully they would be able to recognize plant diseases. Prior to the training phase, all the images were processed, having been resized, cropped using bounding boxes for disease localization, normalized, and subjected to various augmentations, like images' flip and adjustment of brightness levels. This yielded some positive results, as PlantDoc-trained models demonstrated

higher robustness when applied to real farm conditions than other models, which were trained exclusively based on laboratory data. At the same time, it resulted in somewhat worse performance overall due to the nature of real-life data. [19]

In this work, it was examined how EfficientNet, which is known for scaling depth, width, and resolution together, performs regarding recognizing plant diseases. Researchers used different versions of EfficientNet, from its smaller version B0 to a larger one, B7. Transfer learning was performed based on existing plant disease recognition datasets such as PlantVillage. The data was processed thoroughly before entering the training phase. Specifically, the images were resized according to the requirements of the model, normalized, and augmented using techniques like rotation, flipping, zooming, and others. It easily beat the performance of the previous versions of CNNs, like ResNet and Inception, while being even more efficient due to using less number of parameters. The classification accuracy exceeded 99%. It should be noted that these tests were conducted using clean datasets. [20]

This paper describes ResNet, which is a deep convolutional neural network using residual learning to solve the problem of degradation in extremely deep models. Designed for generalized image recognition, ResNet has shown impressive performance in plant disease classification due to its efficient image feature extraction capabilities. One of the main concepts covered in the paper is the skip connection, which helps retain the continuity of gradient flow during training and enables ultra-deep neural nets, including ResNet50, ResNet101, and ResNet152.

ResNet is fine-tuned using transfer learning on platforms like PlantVillage. This process involves preprocessing the images which includes resizing, normalizing the data and using data augmentation to increase the size of the dataset. In various research studies, the model has consistently achieved impressive accuracy rates, as seen in reference [21].

7. Methodology

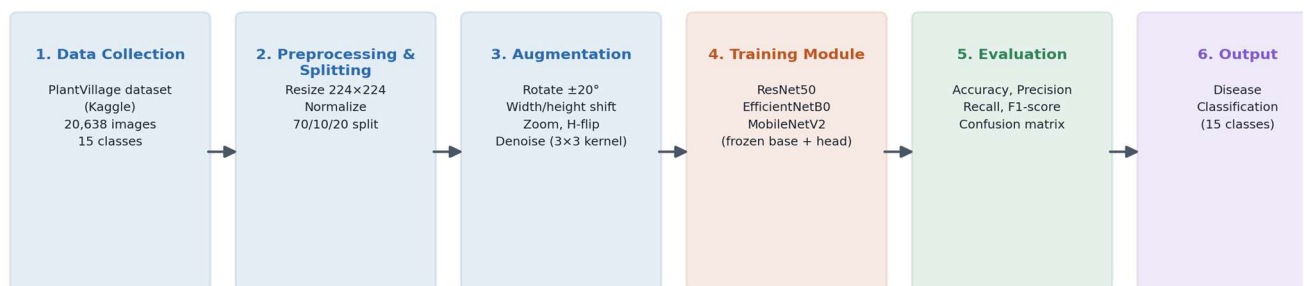
The methodology used to develop and evaluate the plant leaf disease classification system proposal. Three CNN models are used (ResNet50, EfficientNetB0 and MobileNetV2). The three models trained under the same experimental protocol that the results can be compared fairly in chapter four. Each model is implemented in its own notebook and follows the same procedural pipelines that include dataset preparation and splitting, data quality control, data augmentation and preprocessing, model construction, training, and evaluation.

The evaluation step saves the metrics and confusion matrix for each model so they can be compared later. This chapter does not present accuracy, precision, recall, or other performance results as they are reported in chapter four

Transfer learning is used for all three models instead of training each network from scratch. The PlantVillage dataset used in this study contains 20,638 images across 15 classes which is a relatively sufficient dataset for training a deep CNN from random initialization. Reusing ImageNet pretrained features as shown in the transfer learning studies reviewed in chapter two provide high accuracy on this dataset with less training data and less training time.

7.1 Proposed Framework

This section presents the framework used for plant leaf disease classification start from image collection to the final prediction. Figure 3.1 shows the six steps of the pipelines as follow 1.Data collection from the PlantVillage dataset 2.Preprocessing, which includes resizing, normalization, and splitting the dataset 3.Augmentation, which increases the size of the training set and includes the noise removal step described in Section 3.5 4.the training step, where each of the three models ResNet50, EfficientNetB0, and MobileNetV2) is trained with the same classification head 5. Evaluation of each model on the test set 6.Final disease classification output across the 15 classes.



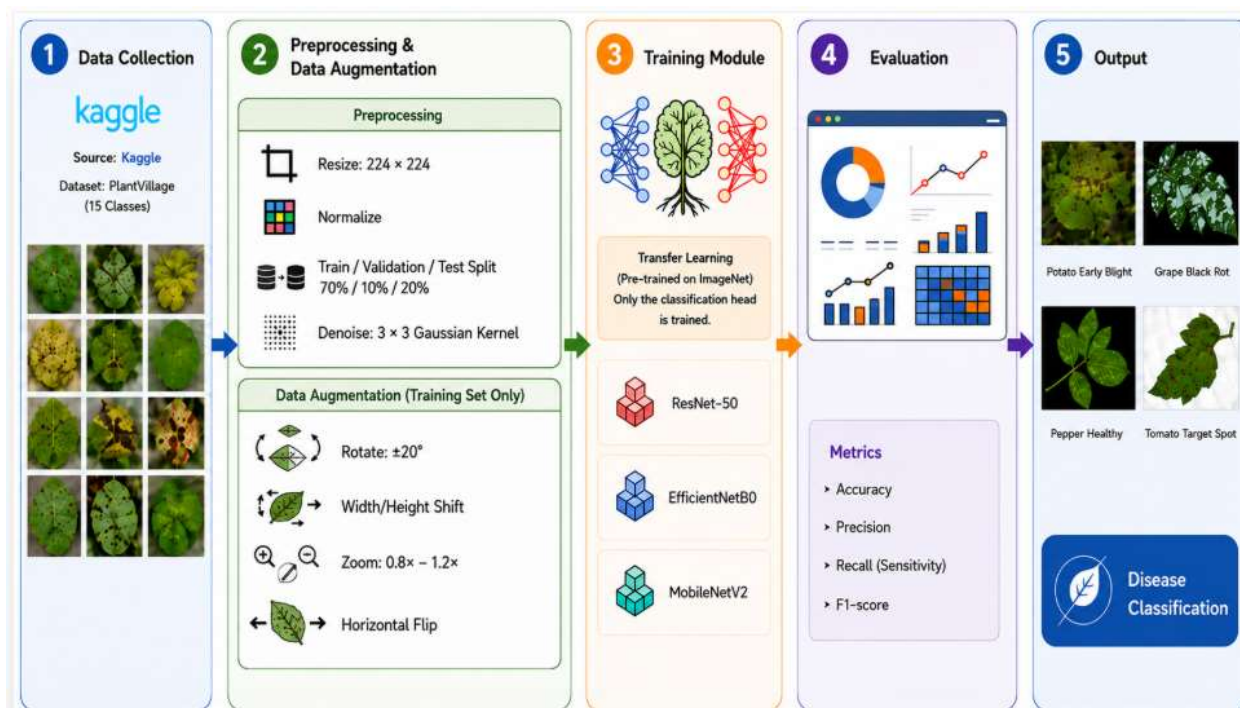


Figure 3.1. Proposed end-to-end framework for plant leaf disease classification, from data collection through to disease-class output.

All three models use the same classification head on top of a frozen, ImageNet pretrained convolutional base. The base network uses include_top=False and an input shape of 224×224×3, and its layers are frozen (trainable = False) so that only the head is trained. The head consists of a GlobalAveragePooling2D layer, a Dense layer of 256 units with ReLU activation, a Dropout layer with a rate of 0.5, and a final Dense layer of 15 units with softmax activation for the 15 target classes.

We freeze the convolutional base instead of fine-tuning it because the ImageNet pretrained filters already contain general features such as edges, textures, and color gradients that transfer well to leaf images. Freezing the base also keeps the number of trainable parameters and the training time similar across the three models, which keeps the comparison fair. We use 256 units in the dense layer and a dropout rate of 0.5 because these are standard values for a classification head of this size, and the dropout helps reduce overfitting.

Table 3.1 shows that the three models are very different in size. This is a property of each network, and the performance results of these models are presented in Chapter 4.

Table 3.1. Total parameter counts of the three transfer-learning models (frozen ImageNet base + trainable classification head).

Backbone	Trainable parameters (head only)	Total parameters	Family
ResNet50	frozen base + head	24,116,111	Residual CNN
EfficientNetB0	frozen base + head	4,381,362	Compound-scaled CNN
MobileNetV2	frozen base + head	2,589,775	Depthwise-separable CNN

7.2 Dataset Description

The PlantVillage dataset from Kaggle is being used that contains 20,638 images of 15 classes from three crops that include pepper, potato, and tomato. The classes include both healthy and diseased leaves. The 15 class names used in the classification pipeline are as follows:

Pepper bell Bacterial spot, Pepper bell healthy, Potato Early blight, Potato Late blight, Potato healthy, Tomato Bacterial spot, Tomato Early blight, Tomato Late blight, Tomato Leaf Mold, Tomato Septoria leaf spot, Tomato Spider mites Two spotted spider mite, Tomato Target Spot, Tomato Tomato YellowLeaf Curl Virus, Tomato Tomato mosaic virus, Tomato healthy.

The dataset is imbalanced across classes. Table 3.2 shows the number of images per class in the test subset, which is 20% of the data as explained in Section 3.4. The smallest class is Potato healthy with 31 test images, and the largest class is Tomato Yellow Leaf Curl Virus with 643 test images, a ratio of about 1:21. We discuss this imbalance in Chapter 4 when we look at the per-class results.

7.3 Data Preparation and Splitting

The PlantVillage was archive on Google Drive folder under a fixed project path. Every notebook was unzipped to the locally since reading images directly from Drive is slower than reading them locally. The images then split into training, validation, and test sets with a fixed seed equal 42 and a ratio of 70% training, 10% validation, and 20% test.

7.4 Data Augmentation and Preprocessing

Before augmentation, we check every image in the split dataset for corruption using PIL's `verify()` method. One corrupted file was found and removed. We then apply light Gaussian denoising with a 3×3 kernel to the remaining images to reduce noise while keeping the leaf texture and lesion boundaries that are important for diagnosis. This step is applied only to the local copy, so the original archive on Drive is not changed.

We build the image generators using Keras' `ImageDataGenerator`. For the training set, we apply augmentation to increase variability and reduce overfitting: rotation up to 20 degrees, width and height shift up to 20%, zoom up to 20%, and random horizontal flip, with the fill mode set to 'nearest'. We do not apply augmentation to the validation and test sets, since these sets are only used to measure the real performance of the model.

We choose these augmentation ranges to represent realistic variation in how a leaf photo might be taken, such as changes in camera angle, distance, and zoom. We apply horizontal flip because a leaf disease does not depend on the direction the leaf is facing. We keep the ranges small enough that the lesion patterns used to tell the diseases apart are not distorted. Each backbone uses its own Keras `preprocess_input` function instead of a general $1/255$ rescale, for the training, validation, and test sets. Each pretrained ImageNet backbone expects its input to be normalized in a specific way, based on how it was pretrained. Using a fixed $1/255$ rescale together with, or instead of, this function would give the network input values it was not trained to expect, which would reduce its ability to use the pretrained features. Using the correct preprocessing function for each backbone means all three models receive input in the range they expect, which is needed for a fair comparison.

7.5 Training Configuration and Evaluation Protocol

The three models are compiled with the Adam optimizer at a learning rate of 0.001, categorical cross-entropy as the loss function, and accuracy as the training metric. We train each model with a batch size of 32 for a maximum of 20 epochs. We use two callbacks for all three models: EarlyStopping, which monitors validation loss with a patience of 5 epochs and restores the best weights, and ModelCheckpoint, which saves only the best model, by validation accuracy, to Google Drive. We use the same optimizer, loss function, batch size, number of epochs, and callbacks for all three models, so that the differences reported in Chapter 4 come from the architectures themselves and not from the training setup. After training, we evaluate each model on the test set using a generator with shuffle turned off, so the predicted labels line up correctly with the true labels. We convert the predicted probabilities to class labels using arg-max. From these predictions, we compute accuracy, macro-averaged and weighted-averaged precision, recall, and F1-score using scikit-learn's precision_recall_fscore_support, along with a full per-class classification report and a confusion matrix. We also save the confusion matrix as a heatmap image. We compute the four main metrics from the per-class counts of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) as follows:

$$Accuracy = (TP + TN) / (TP + TN + FP + FN)$$

$$Precision = TP / (TP + FP)$$

$$Recall = TP / (TP + FN)$$

$$F1-score = 2 \times (Precision \times Recall) / (Precision + Recall)$$

Macro-averaged values give each of the 15 classes the same weight, no matter how many test images it has. Weighted average values give more weight to classes with more test images. We report both in Chapter 4 to check whether the results are mainly driven by the larger classes.

8. Results

8.1 Experimental setup

This chapter presents the results obtained by applying the methodology to the three transfer learning models ResNet50, EfficientNetB0, and MobileNetV2. The experiments implemented using Python, Google Colab with GPU acceleration for PlantVillage dataset contains 20,638 leaf images across 15 healthy and diseased classes from pepper, potato, and tomato plants. The dataset split into 70% training, 10% validation, and 20% for testing purpose of each model. The overall test set performance was presented followed by the classification results and confusion matrix analysis. Finally, the comparative evaluation to find the best performing model based on accuracy and the other evaluation metrics.

All three backbones ResNet50, EfficientNetB0, and MobileNetV2 were compiled with the Adam optimizer for training with learning rate equal to 0.001 and categorical cross-entropy loss, and trained with a batch size of 32 upto 20 epochs. An EarlyStopping callback monitored validation loss with a patience of 5 epochs and restored the best weights while a ModelCheckpoint callback saved only the best weights by validation accuracy to Google Drive. The same protocol described in Section 3.6 is applied to every model, so that the differences reported below come from the architectures themselves and not from the training procedure.

8.2 Performance Evaluation and Comparative Analysis

This section compares the three transfer learning models (ResNet50, EfficientNetB0 and MobileNetV2) using the protocol described in Section 3.6. The models compared per class performance, confusion matrix and computational efficiency.

8.3 Overall Performance Comparison

The overall performance comparison is elaborated in the table 4.1 that shows ResNet50 records the highest accuracy of the three models at 96.47% followed by EfficientNetB0 at 95.97% and then MobileNetV2 at 92.13%. EfficientNetB0 achieves accuracy within half a percentage point of ResNet50 (95.97% vs. 96.47%) and stays close across most other metrics while using less than one fifth of its parameters (4,381,362 compared to 24,116,111). This makes EfficientNetB0 the best trade off between accuracy and size among the three models. MobileNetV2 is the smallest model with under 2.6 million parameters, but records the lowest accuracy and more than four points behind ResNet50. Training time is similar for all three models between 63 and 66 minutes, because only the head is trained while the convolutional base stays frozen in every case. This shows that training time does not depend on parameter count in the same way accuracy.

Model	Accuracy	Precision	Recall (macro)	F1-score (	Parameters	Training time
ResNet50	96.47%	96.55%	96.20%	96.33%	24,116,111	63.5 min
EfficientNetB0	95.97%	95.80%	95.74%	95.67%	4,381,362	66.0 min
MobileNetV2	92.13%	92.08%	91.75%	91.69%	2,589,775	63.2 min

Table 4.1 shows the accuracy, precision, recall, F1-score, parameter count and training time for all three models.

ResNet50 records the highest accuracy of the three models at 96.47%, followed by EfficientNetB0 at 95.97% and then MobileNetV2 at 92.13%. EfficientNetB0 achieves almost the same accuracy as ResNet50 within half a percentage point on every metric, while using less than one-fifth of its parameters (4,381,362 compared to 24,116,111). This makes EfficientNetB0 the best trade-off between accuracy and size among the three models. MobileNetV2 is the smallest model, with under 2.6 million parameters, but records the lowest accuracy, more than four points behind ResNet50. Training time is similar for all three models, between 63 and 66 minutes, because only the head is trained while the convolutional base stays frozen in every case. This shows that training time does not depend on parameter count in the same way accuracy does.

8.4 Per-Class Performance

All the three models record their highest F1-score on the pepper and potato classes and their lowest F1-score on the Tomato Target Spot or Tomato Early blight classes. ResNet50 records the highest F1-score of 0.993 for Pepper bell healthy and the lowest F1-score of 0.894 for Tomato Target Spot. EfficientNetB0 records the highest F1-score of 0.991 for Tomato Yellow Leaf Curl Virus and the lowest F1-score of 0.886 for Tomato Target Spot.

MobileNetV2 records the highest F1-score of 0.985 for Pepper bell healthy and the lowest F1-score of 0.791 for Tomato Early blight which is the lowest F1-score recorded by any model on any class. This shows that MobileNetV2 has more difficulty than the other two models in separating the visually similar tomato diseases.

Model	Best class (F1-score)	Worst class (F1-score)
ResNet50	Pepper bell healthy (0.993)	Tomato Target Spot (0.894)
EfficientNetB0	Tomato Yellow Leaf Curl Virus (0.991)	Tomato Target Spot (0.886)
MobileNetV2	Pepper bell healthy (0.985)	Tomato Early blight (0.791)

Table 4.2. Best and worst performing class (by F1-score) for each model on the test set.

8.5 Confusion Matrix Analysis

The confusion matrices of ResNet50, EfficientNetB0 and MobileNetV2 are presented in the Figure 4.2. All three models record high overall accuracy, but the confusion matrices show consistent misclassification patterns among visually similar tomato diseases.

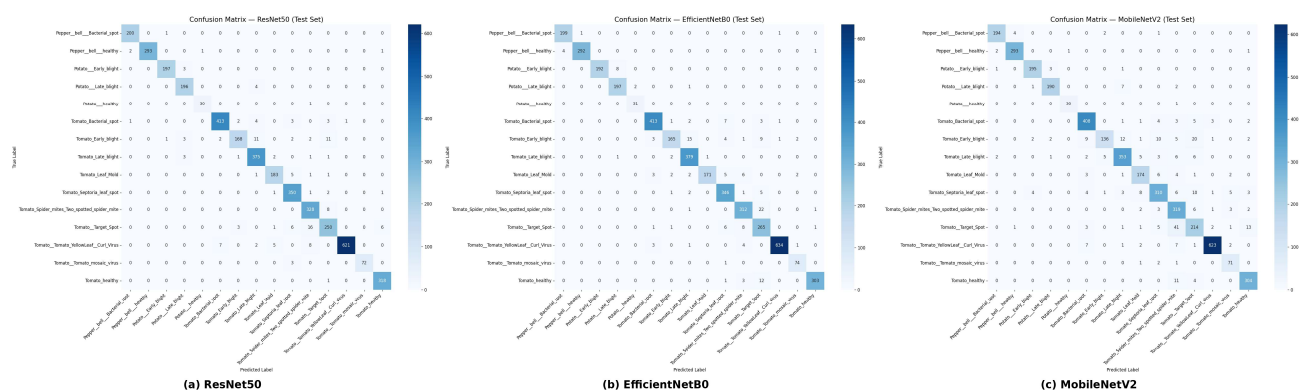


Figure 4.2. Confusion matrices on the test set for (a) ResNet50, (b) EfficientNetB0, and (c) MobileNetV2.

ResNet50 produces the fewest misclassifications across nearly all disease pairs, EfficientNetB0 shows slightly more confusion between Spider Mites and Target Spot and MobileNetV2 produces considerably more errors, particularly among the tomato diseases. The largest confusion is MobileNetV2 classifying 41 Target Spot samples as Spider Mites compared with only 16 for ResNet50 and 22 for EfficientNetB0. This suggest that the higher capacity architectures may capture more discriminative features for these visually similar diseases. The pepper and potato classes show almost no cross-class confusion for any of the three models, which confirms that this difficulty is specific to a cluster of visually similar tomato diseases rather than a general limitation of any architecture.

Disease Pair	ResNet50	EfficientNetB0	MobileNetV2
Target Spot → Spider Mites	16	22	41
Early Blight → Late Blight	11	15	12
Early Blight → Target Spot	11	8	20

Table 4.3. Number of test-set misclassifications for the three most frequent confusion pairs, compared across models. (Additional, smaller confusions are visible in the full matrices of Figure 4.2.)

8.6 Model Efficiency Comparison

ResNet50 gives the highest overall accuracy but the lowest accuracy per parameter due to its much larger convolutional base. EfficientNetB0 gives the best overall trade-off as it recovers almost all of ResNet50's accuracy while using less than one-fifth of the parameters by giving more than five times ResNet50's accuracy-per-million-parameters value. MobileNetV2 is the most efficient model by this measure and with under 2.6 million parameters remains a good candidate for edge or mobile deployment where memory and inference speed matter more than the last few points of accuracy.

Model	Parameters	Accuracy	Accuracy / Million Parameters
ResNet50	24,116,111	96.47%	4.00
EfficientNetB0	4,381,362	95.97%	21.90
MobileNetV2	2,589,775	92.13%	35.58

Table 4.4. Accuracy-per-parameter efficiency comparison of the three transfer-learning models.

8.7 Discussion

Net50 gives the highest classification performance across all evaluation metrics and produces the lowest number of misclassifications in the confusion matrix. EfficientNetB0 gives comparable performance while needing about one fifth of the parameters which makes it a practical choice for deployment in resource constrained environments. MobileNetV2 gives lower classification accuracy and more confusion among visually similar tomato diseases. its small size remains attractive for edge computing applications where computing resources are limited.

A consistent pattern appears across all three confusion matrices. The same group of tomato diseases, Target Spot, Spider mites, Early blight and Late blight accounts for most of the misclassifications in every model and Tomato Target Spot and Tomato Early blight are among the two lowest F1-score classes for all three models. This matches the underlying plant pathology rather than a weakness of one architecture: Target Spot, Early blight, Late blight, and Septoria leaf spot are all necrotic-lesion diseases that produce similar brown to dark, roughly circular spotting on the leaf surface, and Spider mite damage can produce similar leaf discoloration and stippling at the resolution of the PlantVillage images. Since even the highest-capacity model in this comparison (ResNet50) could not fully resolve this confusion, the difficulty appears to come from the visual similarity between these disease categories rather than from model capacity.

References

- [1] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Pearson, 2021.
- [2] S. Mohanty, D. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, p. 1419, 2016
- [3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [4] A. Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *International Conference on Learning Representations (ICLR)*, 2021.
- [5] X. Zhang et al., "A Multitask Learning-Based Vision Transformer for Plant Disease Classification," *Springer*, 2024.
- [6] K. Rose, S. Eldridge, and L. Chapin, "The Internet of Things: An Overview – Understanding the Issues and Challenges of a More Connected World," *Internet Society*, 2015.
- [7] S. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," 2019.
- [8] X. Chen et al., "Vision Transformer for Plant Disease Classification in the Wild: A Multimodal Approach," 2022.
- [9] X. Xie et al., "A lightweight model for efficient identification of plant diseases and pests based on deep learning," 2023.
- [10] F. M. Talaat and M. Ibrahim, "IoT-Integrated Robotic System for Automated Plant Disease Detection and Environmental Monitoring," 2023.
- [11] M. Rahman et al., "Detection and Identification of Plant Leaf Diseases Using YOLOv4," 2024.
- [12] A. Duhan et al., "RTR_Lite_MobileNetV2 for Plant Disease Detection," 2025.
- [13] R. Bhola and A. Kumar, "Deep Learning Model for Early Vegetable Disease Detection," 2025.
- [14] M. Kheir et al., "MobilePlantViT: A Mobile-Friendly Hybrid Vision Transformer for Generalized Plant Disease Image Classification," 2025.
- [15] M. Thanjaivadivel, "Enhanced CNN for Leaf Disease Classification," 2025.
- [16] "A Real-Time Monitoring System for Accurate Plant Leaves Disease Detection Using Deep Learning," 2024.
- [17] "Conditional Mutual Information Constrained Deep Learning for Classification," 2023.
- [18] S. Rajput and S. Doddamani, "Pigeon Pea Leaf Disease Classification Using BoVW and DSIFT," 2024.

	Midocean Journal for Research and Studies	
--	--	--

- [19] D. Singh et al., “PlantDoc: A Dataset for Visual Plant Disease Detection,” 2020.
- [20] M. Tan and Q. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” 2019.
- [21] K. He et al., “Deep Residual Learning for Image Recognition,” 2016.