National College of Ireland

How Well Can MobileNetV3 Perform in Detecting Diseases in Tomato Plants

MSc Research Project
Data Analytics

Ziyi Yan
Student ID: x22198512

School of Computing
National College of Ireland

Supervisor: Prof. Jorge Basilio

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Ziyi Yan		
Student ID:	X22198512		
Programme:	MSc in Data Analytics	**Year:**	2024
Module:	Research Project		
Supervisor:	Prof. Jorge Basilio		
Submission Due Date:	12/08/2024		
Project Title:	How Well Can MobileNetV3 Perform in Detecting Diseases in Tomato Plants		
Word Count:	4718 **Page Count** 15		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Ziyi Yan

Date: 11/08/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

How Well Can MobileNetV3 Perform In Detecting Diseases In Tomato Plants

Ziyi Yan
x22198512

Abstract

In this study, we used the PlantVillage dataset to assess the performance of the MobileNetV3 model in identifying diseases in tomato plants. Our foremost objective is to develop a model that can not only achieve high accuracy but also operates systematically, making it practical for deployment in agricultural sectors with limited hardware resources. We're aiming to enhance the models diagnostic capabilities by employing transfer learning optimisation. We evaluated the performance of this model by using recall, accuracy, precision and F1 score. This research aims to highlight and prove the potential for integrating deep learning with agriculture technology to promote sustainable farming practices through better disease detection.

Keywords: MobileNetV3, Transfer learning, Plant disease Detection, Convolutional Neural Networks.

1 Introduction

Tomato plants are one of the most important corps but they are susceptible to numerous diseases. The annual production of tomatoes all over the world is about 180 million tons and the annual loss of tomato production due to pests and diseases accounts for more than 20% of the total production. Regular traditional techniques of disease detection take too much time and is labor-intensive. , also there performance are not high and most time have high error rate in large-scale agricultural production. New opportunities for automated plant disease diagnosis have appeared, offering the promise of increased accuracy and efficiency, due to the development of deep learning technology, particularly convolutional neural networks (CNNs).

In this study we use a public dataset namely PlantVillage. The goal of this research is to explore how well can MobileNetV3 perform in detecting diseases in tomato plants. MobileNetV3 is a simplified and effective type of convolutional neural network. Recent advancements in computer hardware and deep learning have made it possible to use these networks on mobile devices.

This study investigates the use of MobileNetV3 to identify diseases in tomato plants using a collection of pictures of healthy and sick plants known as the PlantVillage dataset.

The goal of this research is to create a tool that can accurately detect plant diseases and work well on mobile devices with limited computing power. This could significantly improve how we manage plant diseases in farming.

The integration of advanced technologies with traditional farming methods holds the potential to transform crop management, leading to enhanced productivity and reduced environmental impact. This work seeks to demonstrate that MobileNetV3 can play an important role in disease identification, contributing valuable insights to the fields of agricultural technology, machine learning, and plant pathology.

There are six sections in this paper, the next section is related work, the rearch methodology is discussed in section 3, the design specification and implementation are discussed in section 4 and section 5, then, in section 6, there are several experiments listed to evaluate the models, finally, conclusions and future work is discussed in section 7.

2 Related Work

2.1 MobileNetV3

Howard et al. (2019) introduced MobileNetV3, combining automated neural architecture search (NAS) and novel architectural improvements for mobile efficiency, making it ideal for real-time plant disease detection on mobile devices. Sladojevic et al. (2016) demonstrated the effectiveness of deep convolutional networks in recognizing plant diseases from leaf images, laying the groundwork for using CNNs in agriculture. Too et al. (2019) showed that transfer learning significantly improves the accuracy of plant disease detection models while reducing training time, highlighting its practical benefits.

The paper "Identification Method of Corn Leaf Disease Based on Improved MobileNetv3 Model" makes valuable contributions in the realm of agricultural image recognition. MobileNetV3 shows strong accuracy and efficiency in identifying corn leaf diseases compared to traditional methods. Specifically, the model achieved an 97.36% accuracy on test data. Additionally, the study underscores how lightweight networks enable real-time detection on mobile devices, essential for practical implementation in agricultural production.

Calma et al. (2023) developed a system to identify diseases in cassava plants using images of both leaves and stems. They used MobileNetV3 and improved the quality of their data by data augmentation. Their system was able to accurately detect five different leaf diseases and three stem diseases, achieving an accuracy of 93.20% for the leaf diseases detection and 90.80% for the stem diseases detection. This study shows that MobileNetV3 is effective at identifying plant diseases and that improving the data can significantly enhance model performance. While their research focused on cassava, the methods they used could also be applied to other crops, such as tomatoes.

2.2 Transfer Learning

Lakshmanarao et al. (2022) successfully used pre-trained models to identify plant diseases. They tested their approach on images of tomato, potato, and pepper plants. Their results showed that models like VGG16, ResNet50, and InceptionV3 were highly accurate in recognizing plant diseases, especially tomato leaf diseases. ResNet50 performed particularly well, achieving an accuracy rate of 98.6%. This study highlights the potential of transfer learning for plant disease detection and supports the use of pre-trained models as a strong foundation for my own research on tomato leaf diseases.

2.3 Tomato Leaf Diseases Detection

There is one notable approach for tomato plant disease detection through the application of deep learning and transfer learning techniques discussed by Bahrami et al. (2024), evaluated the performance of different transfer learning models, including VGG19, ResNet-101, and MobileNet-v2, on the PlantVillage and CCMT datasets. Their study demonstrated that VGG19 achieved the highest accuracy of 99.48% on the PlantVillage dataset, outperforming the other models. However, it was noted that the performance dropped when tested on the CCMT dataset, which contained more complex backgrounds, highlighting the challenge of generalizing models across different data conditions (Bahrami et al., 2024).

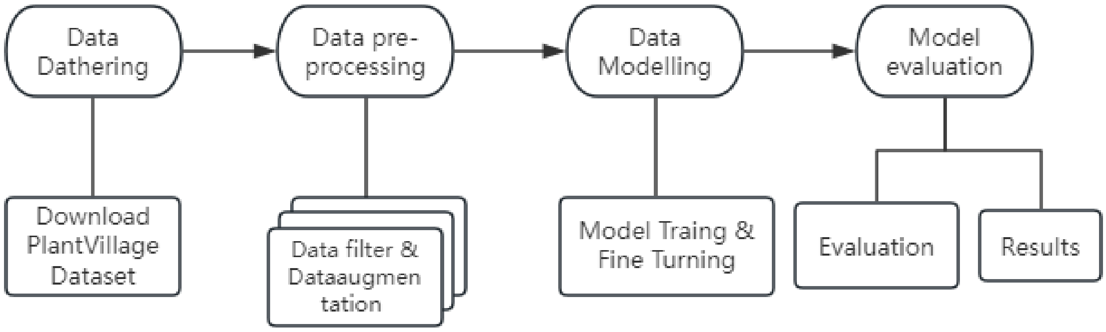
Similarly, Srivastava et al. (2023) proposed a hybrid model combining MobileNetV2 for feature extraction and Extreme Learning Machine (ELM) for classification. This approach was tested on the PlantVillage dataset and achieved an accuracy of 99.0% with a low loss of 0.06, suggesting its robustness in classifying various tomato leaf diseases. The use of ELM was particularly effective in reducing computational costs while maintaining high accuracy, presenting a viable solution for real-time applications (Srivastava et al., 2023).

Previous research has shown that using pre-trained models like MobileNetV2 can be highly effective in identifying diseases in tomato plants. These studies combined powerful image recognition techniques with these pre-trained models to achieve impressive results. Building on these findings, my research aims to investigate the potential of MobileNetV3 for tomato disease detection. I will compare its performance with other models and assess its suitability for use in real-world applications, such as on smart devices.

3 Research Methodology

There are many related studies on the application of deep learning and transfer learning in the field of plant disease detection, these researches inspired me. For instance, Bahrami et al. The works of 2024 showed the effectiveness of transfer learning models like VGG19 and ResNet-101 on tomato leaf disease detection, with this knowledge in my study I chose to use MobilnetV3 due to it's light architecture and good performs in similar tasks as mine. The success pushed me to adapt MobileNetV3 for my project. Data augmentation guided my data processing in these studies as well. These references help to understand the technical background and development direction involved in this study. Combined with the specific problem of tomato leaf disease detection in this study, they help me construct a rough technical framework.

In this study, we use PlantVillage dataset to explore the preformance of MobileNetV3 model in detecting tomato plant desiese, The research methodology mainly consists of 4 steps: data gathering, data pre-processing, data modeling, and model evaluation.



Firure 1: Research Methodology

3.1 Data gathering

We can get the dataset on Kaggle's website, Plantvillage is a publicly available dataset that anyone can use for learning or research. The dataset contains 54,304 images. The images span 14 crop species: Apple, Blueberry, Cherry, Corn, Grape, Orange, Peach, Bell Pepper, Potato, Raspberry, Soybean, Squash, Strawberry, Tomato. In containes images of 17 fungal diseases, 4 bacterial diseases, 2 mold (oomycete) diseases, 2 viral disease, and 1 disease caused by a mite. In this study, we just going to focus on tomato plants, there are 18160 images of tomato leaves grouped by disease type, including the healthy leaves(Bacterial spot, Early blight, Healthy, Late blight, Leaf mold, Septoria leaf spot, Spider mites Two-spotted spider mite, Target Spot, Tomato mosaic virus, Tomato Yellow Leaf Curl Virus)

3.2 Data pre-processing

In this step, we used some data augmentation operations such as random cropping and horizontal flipping. We also uniformly adjusted the images to 224x224 pixels and carried out normalisation processing to make the pixel value between 0 and 1. This ensures a consistent image and compliance with the input requirements of MoblieNetV3. We splited the dataset into a training set and a validation set in an 8:2 ratio.

3.3 Modelling

MobileNetV3 model is a lightweight model that can run effectively on limited computing resources and performs well. Specifically, we explored three different training strategies on the model for vertical comparison: 1, training from scratch, 2, using pre-training weights, 3, transfer learning. In the scratch training, the weights of the model are initialized from zero. When using the pre-trained weights, the model utilizes the pre-trained weights on the ImageNet dataset. In transfer learning, we only fine-tune the last few layers of the model. Additionally, we trained the EfficientNet and ResNet models to compare with this MobileNetV3.

Use the same training settings for all of the models, including the optimizer(SDG, Adam), the loss fuction(CrossEntropyLoss), the epoch(25), the batch size(32). In the training process, the last layers of the pre-training model are fine-tuned, while the early layers are kept frozen to adapt to the task of tomato disease detection.

3.4 Model evaluation

Use multiple metrics to make a comprehensive understanding of the model's performance. The tools we used is Pytorch's build-in evaluation functions and a confusion matrix. We evaluated the MobileNetV3 model and analyzed its performance under different training strategies in detail.

		True Class	
		Positive	*Negative*
Predicted Class	*Positive*	True Positive Count (TP)	False Positive Count (FP)
	Negative	False Negative Count (FN)	True Negative Count (TN)

Figure 2: Classification / Co-incidence Matrix

- **Accuracy**: The most intuitive metric of the model, by measuring the proportion of total predictions that were correct.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Precision**: Precision measures the accuracy of positive predictions. It quantifies the number of true positive predictions in relation to the number of total positive predictions (true positives and false positives).

$$Precision = \frac{TP}{TP + FP}$$

- **Recall**: A higher recall indicates a stronger ability of the model to identify positive samples correctly.

$$Recall = \frac{TP}{TP + FN}$$

- **F1 Score**: This metric is particularly valuable because it provides a single measure that shows how well the model accurately identifies diseased plants (precision) while also ensuring it captures as many of the actual disease samples as possible (recall).

$$F1\ Score = \frac{2 * TP}{2 * TP + FP + FN}$$

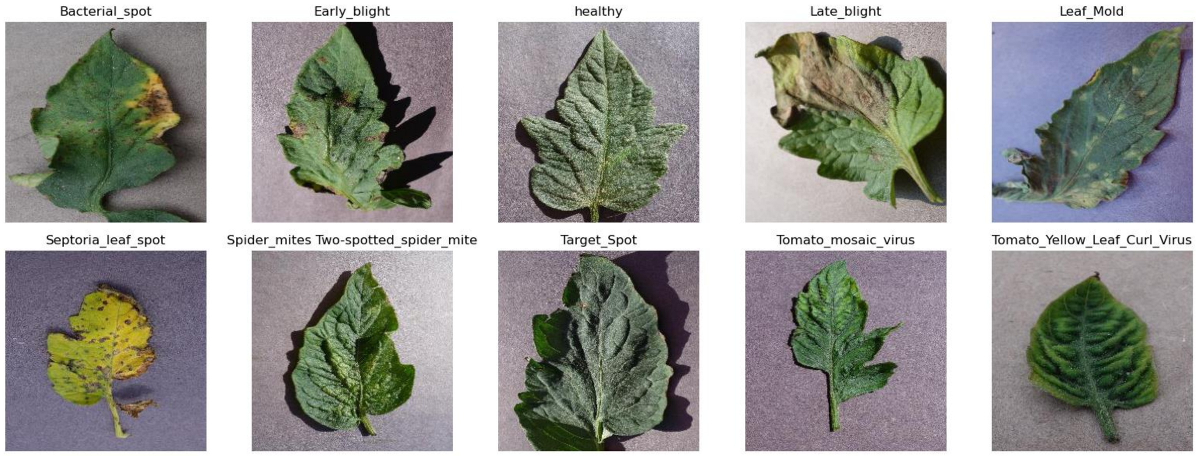
We will evaluate the models on the validation set , and compare the results under different training strategies to assess the performance of MobileNetV3 model. Trying to get a better understanding of the advantages and disadvantages of each training strategy. In addition, we are planning to compare the result of MobileNetV3 to EfficientNet and ResNet.

4 Design Specification

This section focuses on the implementation details and the architectural choices made to optimize the performance and efficiency of different models for detecting diseases in tomato plants. The design is divided into several key components: dataset handling, model architecture, training configuration, and model evaluation.

4.1 Dataset Handling

The PlantVillage dataset contains images of tomato leaves, categorized into various disease classes and a healthy class. To effectively handle this dataset, the following steps were implemented:



Firure 3: Leaf samples

Figure 3 shows some samples selected randomly from the dataset, it is obvious that each disease has different visual characteristics.

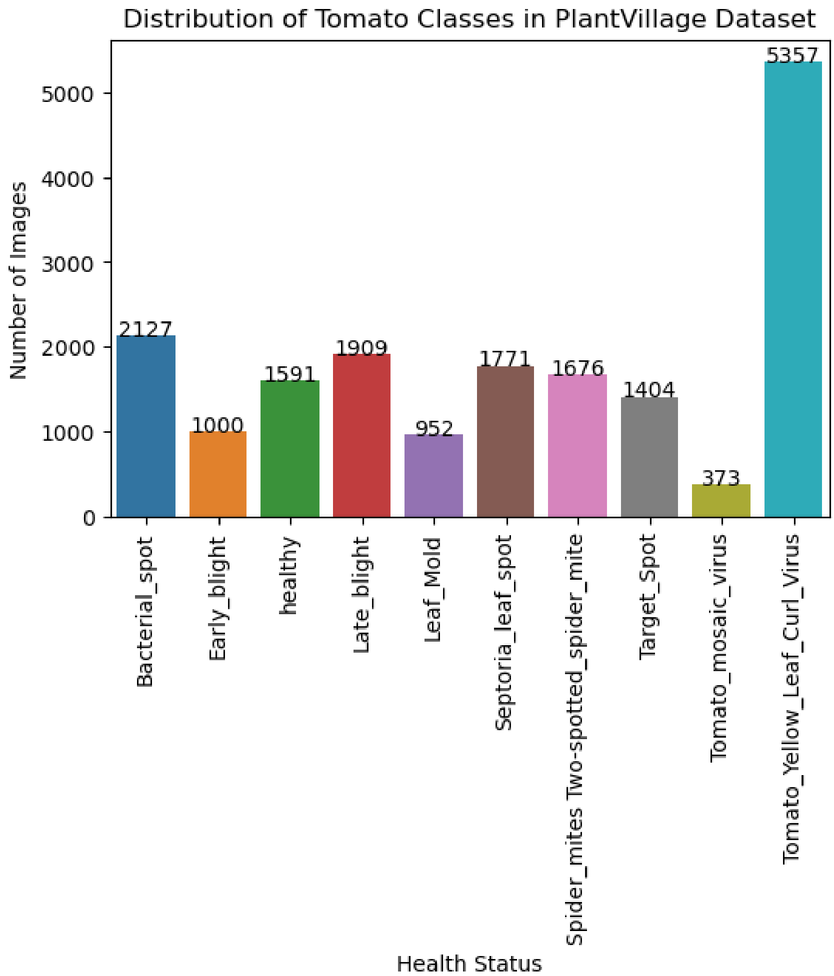


Figure 4: Distribution of Tomato Classes in PlantVillage Dataset

4.1.1 Data Augmentation

We could obtain a more diverse training set by using a series of data augmentation techniques to the dataset, such as random cropping, horizontal flipping, rotation, etc., this operation can improve the robustness and generalization ability of the model.

4.1.2 Dataset Splitting

The dataset was split into a ratio of 8:2 for training and testing.

4.2 Model Architecture

We explored MobileNetV3, EfficientNet, and ResNet models to detect diseases in tomato plants. The following subsections details the architectural choices and configurations for each model.

4.2.1 MobileNetV3

MobileNetV3 is a lightweight, efficient convolutional neural network (CNN) architecture designed for mobile and edge devices. It builds upon the previous versions (MobileNetV1 and V2) by introducing new techniques such as the use of squeeze-and-excitation modules, hard-swish activation functions, and a combination of depthwise separable convolutions. There are two versions of MobileNetV3: MobileNetV3-Large, which is optimized for higher accuracy, and MobileNetV3-Small, designed for environments with limited Limited hardware computing resources. Its architecture strikes a balance between performance and efficiency, making it a ideal choice for real-time applications on resource-constrained devices.

We explored three different implementations of MobileNetV3:

- **Pretrained MobileNetV3:** Utilized a pre-trained MobileNetV3 model on ImageNet, fine-tuning it for our specific classification task by modifying the final fully connected layer to match the number of classes in the PlantVillage dataset.
- **Scratch MobileNetV3:** Initialized MobileNetV3 with random weights and trained the entire network on the PlantVillage dataset from scratch.
- **Transfer Learning MobileNetV3:** Froze the initial layers of the pre-trained MobileNetV3 model and fine-tuned only the final layers for our specific task.

4.2.2 EfficientNet

EfficientNet is a family of CNN architectures that scale model dimensions—depth, width, and resolution—in a balanced manner. The EfficientNet models, proposed by Google AI, use a compound scaling method that uniformly scales all dimensions of the network to achieve better accuracy and efficiency. EfficientNet starts with a base network (EfficientNet-B0) and scales it up to larger versions (EfficientNet-B1 to B7), each offering improved performance and accuracy. EfficientNet models are known for their state-of-the-art performance on several benchmarks while maintaining lower computational costs compared to other architectures of similar accuracy levels.

4.2.3 ResNet

ResNet, or Residual Network, is a groundbreaking deep learning architecture that introduced the concept of residual learning. Instead of forcing the network to learn complex mapping functions directly, ResNet focuses on learning the difference between the input and desired output. This approach effectively mitigates the problem of vanishing gradients in deep networks, enabling the construction of very deep networks (e.g., ResNet50, ResNet101) without performance degradation. ResNet models are known for their robustness and have set new benchmarks in various computer vision tasks, including image classification and object detection.

5 Implementation

Tools and Languages:
Programming Language: Python
Deep Learning Framework: PyTorch
Data Handling: Pandas, Torchvision.
Visualization: Matplotlib.
Others: VS Code, Jupyter Notebook.

5.1 Data Preprocessing

The first step is data preprocessing, use a series of transformations including random resizing and cropping to 224x224 pixels, random horizontal flipping, conversion to tensor, and normalization with mean [0.485, 0.456, 0.406] and standard deviation [0.229, 0.224, 0.225] on the train set. The validation set is resized to 256x256 pixels, center-cropped to 224x224 pixels, converted to tensor, and normalized using the same mean and standard deviation values. Finally, split the dataset into training set and validation set.

5.2 Modeling

This study used the MobileNetV3 model as well as ResNet50 and EfficientNet-B0 models. These models were constructed based on the PyTorch framework and pretrained on ImageNet. The final few layers of the pretrained models were fine-tuned to adapt them to the task of tomato disease detection. These models are implemented follow these steps:

5.2.1 Loss Function and Optimizer Definition

Ues the cross-entropy loss function, and we trained the model using two optimizers separately: the SGD and Adam.

5.2.2 Training

This study involved three models—MobileNetV3, ResNet, and EfficientNet.
We used three training strategies on the MobileNetV3 model to compare their performance:
Pretrained MobileNetV3: Modifying the final fully connected layer to fit our classification task. The convolutional layers' weights were frozen, and only the final layer was fine-tuned.
Scratch MobileNetV3: We trained a MobileNetV3 model from scratch, initializing all weights randomly and training it with our dataset.
Transfer Learning MobileNetV3: Load a MobileNetV3 model pretrained on ImageNet, freeze the convolutional layers, and retrain the final fully connected layer with our dataset.

For ResNet and EfficientNet models, we used pretrained models, modifying the final fully connected layer to suit our disease detection task and freezing all convolutional layers. Only the final fully connected layer was trained.
We saved the trained models as .pth file. The models can be deployed on smart devices for tomato leaf disease detection in the future.
The training configuration includes several key aspects:
Optimizer: We used the SGD optimizer with a learning rate of 0.001 and momentum of 0.9 to train the models. This choice balances the trade-off between convergence speed and stability. Additionally, we use Adam optimizer with a learning rate of 0.001 to train the models. The batch size is 32 and the epochs is 25 for the training process.
Loss Function:The We choose Cross-Entropy as loss function because it effectively measures and minimizes the difference between predicted and true labels, making it ideal for the disease detecttion tasks.

5.2.3 Validation

We evaluate the models on validation set, The process involved monitoring performance metrics such as Accuracy, Precision, Recall, and F1 Score to assess the model's ability to generalize to unseen data. Additionally, we use a confusion matrix to analyze the types of errors made by the model, which provides a more intuitive view of the model's performance.

6 Evaluation

6.1 Experiment 1: Comparison of Different CNN Architectures

In this experiment, we compared three CNN architectures models, MobileNetV3, ResNet, and EfficientNet, the performance metrics are summarized in Table 2:

Model Architecture	Accuracy	F1 Score	Precision	Recall
pretrained_mobilenetsv3(SGD)	0.9838	0.9837	0.9839	0.9837
scratch_mobilenetv3(SGD)	0.9383	0.9386	0.9408	0.9383

transfer_mobilenetv3(SGD)	0.9871	0.9871	0.9872	0.9871
ResNet(SGD)	0.9904	0.9904	0.9904	0.9904
Efficientnet(SGD)	0.9884	0. 9885	0. 9885	0. 9884

Table 2: Different Models Architectures

The ResNet model, which uses SGD optimizer and applies data augmentation technology, has an accuracy rate of up to 99.04%, outperforming all other models. However, the performance of other models is also good, with accuracy rates above 93%. For MobileNetV3, the transfer learning strategy performs slightly better than the others.

6.2 **Experiment 2**: Impact of Data Augmentation on Model Performance

This experiment investigates the impact of data augmentation on the performance of CNN models. By comparing models trained with augmentation and without augmentation. Table 3 experimental data shows that using data augmentation can greatly enhance the performance.

Model Architecture	Accuracy (With Augmentation)	Accuracy (Without Augmentation)
pretrained_mobilenetv3_Adam	0.9785	0.7616
transfer_mobilenetv3_Adam	0.9769	0.7987
efficientnet_Adam	0.9741	0.8150
resnet_Adam	0.9705	0.7591

Table 3: Impact of Data Augmentation

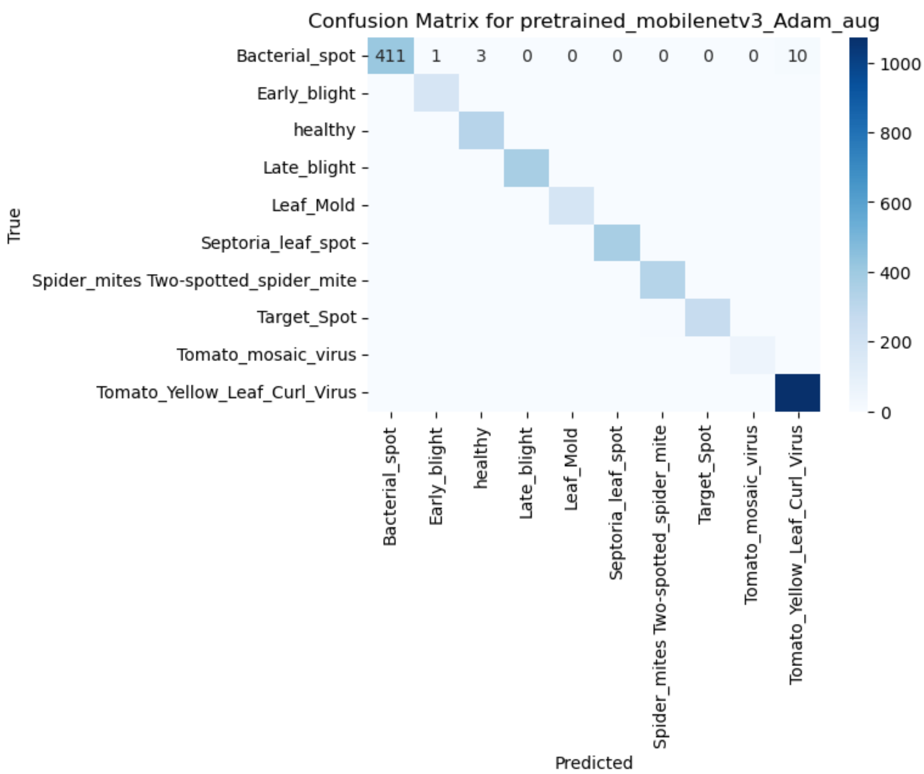


Figure 5: Confusion matrix for pretrained MobileNetV3 with data augmentation

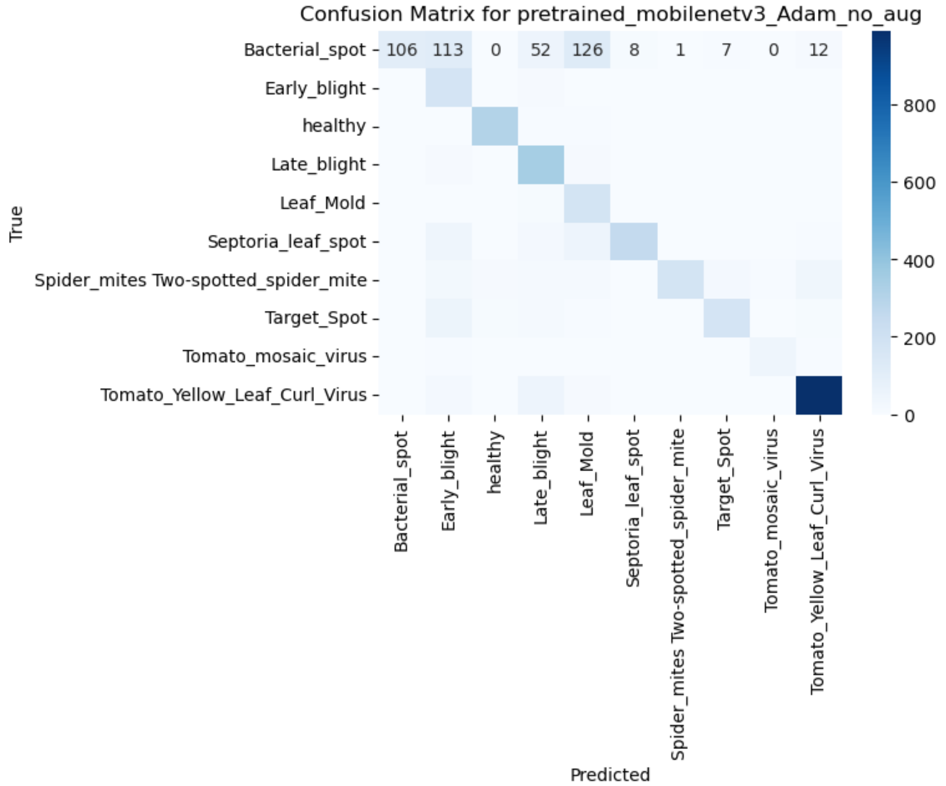


Figure 6: Confusion matrix for pretrained MobileNetV3 without data augmentation

The model architecture and optimizer in Figure 5 and 6 (confusion matrix) are the same: Transfer learning MobileNetV3 and Adam optimizer, Figure 5 uses data augmentation while Figure 6 does not. There is high classification performance with most of the classes being correctly predicted in Figure 5, such as "Bacterial Spot" with 411 accurate predictions. The model misclassification is minimal, indicating high performance accuracy. Without data augmentation as in Figure 6, there is significantly low classification and high errors, such as 113 "Bacterial Spot" were misclassified as "Early blight".

Figure 7: Training history for transfer learning MobileNetV3 with data augmentation

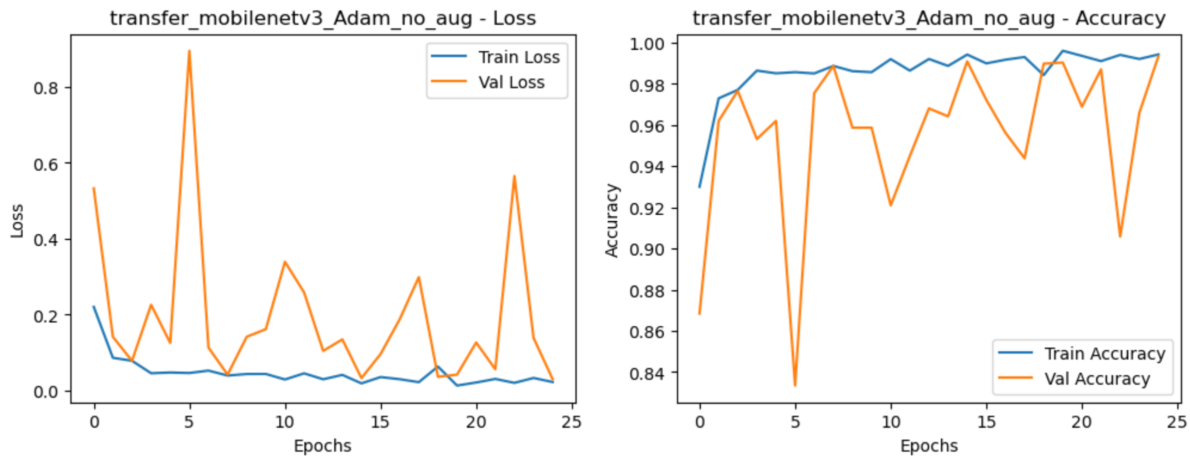


Figure 8: Training history for transfer learning MobileNetV3 without data augmentation

The loss and accuracy curves in Figure 7 and Figure 8 further demonstrate that models trained with augmentation not only achieve higher accuracy but also display more stable learning curves. Figure 7 shows the training process of a transfer learning model using MobileNetV3 with Adam optimizer and data augmentation. The left graph shows the loss values over epochs, while the right graph displays the accuracy progression. Both graphs demonstrate ideal training trends: the loss decreases rapidly, and the accuracy increases steadily, ultimately reaching about 98%. The close alignment of training and validation curves indicates good generalization without significant overfitting or underfitting. Overall, the model achieves a stable high-performance state after approximately 15 epochs.

Data augmentation generally improves the performance across various models, as indicated by the high accuracy, F1 score, precision, and recall. Data augmentation is a particularly crucial factor in the model training process.

6.3 Experiment 3: Evaluation of Different Optimizers

In this experiment, we evaluate the performance of models trained with two different optimizers: Adam and SGD. The objective is to explore how the choice of optimizer influences the accuracy, F1 Score, precision, and recall across various CNN architectures and training strategies, specifically focusing on MobileNetV3, EfficientNet, and ResNet.

The following table summarizes the results across different models and training strategies using both Adam and SGD optimizers:

Model	Training Strategy	Optimizer	Accuracy	F1 Score	Precision	Recall
MobileNetV3 (Pretrained)	Pretrained	Adam	0.9785	0.9785	0.9786	0.9785
MobileNetV3 (Pretrained)	Pretrained	SGD	0.9838	0.9837	0.9840	0.9838
MobileNetV3 (Scratch)	Scratch	Adam	0.9598	0.9597	0.9601	0.9598
MobileNetV3 (Scratch)	Scratch	SGD	0.9383	0.9386	0.9407	0.9383
MobileNetV3 (Transfer)	Transfer Learning	Adam	0.9769	0.9768	0.9773	0.9769
MobileNetV3 (Transfer)	Transfer Learning	SGD	0.9871	0.9871	0.9872	0.9871

EfficientNet	-	Adam	0.9741	0.9741	0.9747	0.9741
EfficientNet	-	SGD	0.9884	0.9885	0.9885	0.9884
ResNet	-	Adam	0.9705	0.9705	0.9712	0.9705
ResNet	-	SGD	0.9904	0.9904	0.9904	0.9904

Table 4: Impact of Different Optimizers

The results show that optimiser has a certain influence on the performance.

• **MobileNetV3 (Pre-trained):** The SGD optimiser slightly performed better than Adam, particularly in accuracy, where the model with SGD achieved 98.38%, compared to 97.85% of the one with Adam.

This suggests that SGD may be more suited with fine-tunned pre-trained models.

MobileNetV3 (Scratch): Adam provided a better performance than SGD for the model that is trained from scratch. Scratch-trained model with Adam reached 95.98% in accuracy, while the one with SGD achieved a lower accuracy as 93.83%.

• **MobileNetV3 (Transfer Learning):** The result is similar to the pre-trained strategy. The transfer learning strategy also benefited more from the SGD optimiser, achieving 98.71% in accuracy, compared to 97.69% of the one with Adam.

• **EfficientNet and ResNet:** When SGD optimiser being applied, both EfficientNet and ResNet models show excellent performance. The high-efficiency network achieved an accuracy of 98.84%, and ResNet achieved the highest accuracy of 99.04% in all experiments.

These results show that although Adam may provide better performance for models trained from scratch, SGD usually provides better results for fine-tuning retention models or applying transfer learning. Therefore, the selection of optimiser should be carefully considered according to the training strategy and the specific model architecture.

6.4 Discussion

Through the above experiments, we have a preliminary understanding of the performance of different model architectures. In addition, we have also seen the impact of different optimizer choices and whether to use data augmentation techniques during the retraining process on model performance. This study mainly explores MobileNetV3, the model trained from scratch using PlantVillage dataset, with significantly lower metrics than the pre trained MobileNetV3 model. There are only 18160 images of tomato leaves in the dataset, which is difficult to match the massive images of the pre trained model. Therefore, it is not recommended to train from scratch. The model fine tuned using transfer learning performs the best and there should be further room for improvement. We can try to test various parameters and do more tests to improve its performance.

7 Conclusion and Future Work

The objective of this study is to evaluate MobileNetV3's performance in detecting diseases in tomatoes plants, its performance is been compared against ResNet and EfficentNet models. From the results in this study, MobileNetV3 is highly effective architecture for the task, particularly when it's been trained in some strategies, with the various training configurations. Highest accuracy of 98.71%, with an F 1 score of 98.71%, precision of 98.72% and a recall of 98.71% is achieved by the tranfer learning MobileNetV3 with the SGD optimizer and data augmentation. This demonstrates that leveraging a transfer learning model with data augmentation and a suitable optimiser can significantly improve the performance of MobileNetV 3 in agricultural disease detection fields.

Future study could expand the application of MobileNetV3 to many types of plant diseases, thereby making it universal. Additionally, actually deploying the model in agricultural environment will provide more valuable insights of the performance been in different conditions. Also in the future researchers could focus on the application of the optimize model in resource-constrained environments, making it more suitable for mobile devices or edge computing platforms. We can also have a further study on the influence of different data augmentation technologies and optimisers on the model performance, so as to fine-tune the method for specific agricultural applications and achieve the best results.

References

Agarwal, M., Singh, A., Arjaria, S., Sinha, A., & Gupta, S. (2020). ToLeD: Tomato Leaf Disease Detection using Convolution Neural Network. *Procedia Computer Science*, *167*. https://doi.org/10.1016/j.procs.2020.03.225

Bahrami, M., Pourhatami, A., & Maboodi, M. (2024). Tomato leaf disease detection using transfer learning: A comparative study. 2024 13th Iranian/3rd International Machine Vision and Image Processing Conference (MVIP). https://doi.org/10.1109/MVIP62238.2024.10491178

Bi, C., Xu, S., Hu, N., Zhang, S., Zhu, Z., & Yu, H. (2023). Identification Method of Corn Leaf Disease Based on Improved Mobilenetv3 Model. *Agronomy*, *13*(2). https://doi.org/10.3390/agronomy13020300

Calma, A. C. P., Guillermo, J. D. M., & Paglinawan, E. C. C. (2023). Cassava Disease Detection using MobileNetV3 Algorithm through Augmented Stem and Leaf Images. *Proceedings of the 2023 17th International Conference on Ubiquitous Information Management and Communication, IMCOM 2023*. https://doi.org/10.1109/IMCOM56909.2023.10035648

Gonzalez-Huitron, V., León-Borges, J. A., Rodriguez-Mata, A. E., Amabilis-Sosa, L. E., Ramírez-Pereda, B., & Rodriguez, H. (2021). Disease detection in tomato leaves via CNN with lightweight architectures implemented in Raspberry Pi 4. *Computers and Electronics in Agriculture*, *181*. https://doi.org/10.1016/j.compag.2020.105951

Hammad Saleem, M., Khanchi, S., Potgieter, J., & Mahmood Arif, K. (2020). Image-based plant disease identification by deep learning meta-architectures. *Plants*, *9*(11). https://doi.org/10.3390/plants9111451

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861. https://arxiv.org/abs/1704.04861

Howard, A., Sandler, M., Chen, B., Wang, W., Chen, L. C., Tan, M., Chu, G., Vasudevan, V., Zhu, Y., Pang, R., Le, Q., & Adam, H. (2019). Searching for mobileNetV3. *Proceedings of the IEEE International Conference on Computer Vision*, *2019-October*. https://doi.org/10.1109/ICCV.2019.00140

Iqbal, M. A., & Talukder, K. H. (2020). Detection of Potato Disease Using Image Segmentation and Machine Learning. *2020 International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2020*. https://doi.org/10.1109/WiSPNET48689.2020.9198563

Jayswal, H. S., & Chaudhari, J. P. (2020). Plant Leaf Disease Detection and Classification using Conventional Machine Learning and Deep Learning. *Jayswal & Chaudhari International Journal on Emerging Technologies*, *11*(3).

Joshi, B. K., & Panthee, Dr. D. R. (2011). Molecular Tagging of Resistance Genes to Septoria Leaf Spot and Late Blight in Tomato (Solanum lycopersicum L.). In *Horticultural Science*.

Lakshmanarao, A., Supriya, N., & Arulmurugan, A. (2022). Plant Disease Prediction using Transfer Learning Techniques. *2022 2nd International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies, ICAECT 2022*. https://doi.org/10.1109/ICAECT54875.2022.9807956

Liu, J., & Wang, X. (2021). Plant diseases and pests detection based on deep learning: a review. In *Plant Methods* (Vol. 17, Issue 1). https://doi.org/10.1186/s13007-021-00722-9

Oyewola, D. O., Dada, E. G., Misra, S., & Damaševičius, R. (2021). Detecting cassava mosaic disease using a deep residual convolutional neural network with distinct block processing. *PeerJ Computer Science*, *7*. https://doi.org/10.7717/peerj-cs.352

Prathusha, P., Murthy, K. and Srinivas, K. (2020) 'Plant Disease Detection Using Machine Learning Algorithms', Advances in Computational and Bio-Engineering, pp. 103-112. DOI: 10.1007/978-3-030-46954-3_9

Ramcharan, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J., & Hughes, D. P. (2017). Deep learning for image-based cassava disease detection. *Frontiers in Plant Science*, *8*. https://doi.org/10.3389/fpls.2017.01852

Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767. https://arxiv.org/abs/1804.02767

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR.2018.00474

Srivastava, M., Sisaudia, V., & Meena, J. (2023). Tomato Leaf Disease Detection and Classification using MobileNetV2 and Extreme Learning Method: A Hybrid Approach. *2023 10th International Conference on Soft Computing and Machine Intelligence, ISCMI 2023*. https://doi.org/10.1109/ISCMI59957.2023.10458596

Too, E. C., Yujian, L., Njuki, S., & Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture*, *161*. https://doi.org/10.1016/j.compag.2018.03.032