Classification of Apple Leaf Diseases Using a Modified EfficientNet Model

1st Muhammad Abiya Makruf
School of Computing
Telkom University
Bandung, Indonesia
abiyamf@student.telkomuniversity.ac.id

2nd Febryanti Sthevanie
School of Computing
Telkom University
Bandung, Indonesia
sthevanie@telkomuniversity.ac.id

3rd Kurniawan Nur Ramadhani
School of Computing
Telkom University
Bandung, Indonesia
kurniawannr@telkomuniversity.ac.id

***Abstract*—Plant disease identification is crucial for maintaining agricultural productivity, as early detection can mitigate crop losses and improve overall yield quality. This study explores using EfficientNet, both in its original and modified forms, for classifying apple plant diseases based on leaf images. EfficientNet was chosen due to its scalability and efficiency, making it well-suited for agricultural applications requiring accurate yet feasible solutions. The architecture was modified by removing specific blocks to improve computational efficiency while maintaining accuracy. Experiments using datasets with complex and simple backgrounds evaluated model robustness under varied conditions, such as different lighting, background noise, and natural clutter. The modified EfficientNetB0 variant demonstrated an optimal balance of training time, accuracy, and efficiency, achieving a training accuracy of 99.10%, validation accuracy of 97.40% and a test accuracy of 84.50%, with up to 50% fewer parameters. These findings suggest that the modified EfficientNet is promising for real-world agricultural applications, especially in resource-constrained settings where computational power is limited. It offers an accessible solution for early disease detection, benefiting small-scale farmers without advanced computing infrastructure. Future work involves expanding the dataset to other crops, testing additional disease types, optimizing the model further for edge devices, and integrating it into decision-support systems for real-time monitoring and analysis.**

Keywords—EfficientNet, apple plant diseases, computational efficiency, modified EfficientNetB0

I. INTRODUCTION

Around 80% of the population in developing nations lives in rural areas, where agriculture serves as the primary source of livelihood [1]. As the demand for agricultural products continues to rise, ensuring crop health and productivity has become increasingly important. Apples are among the most widely consumed fruits globally, ranking as one of the top four produced fruits [2]. Not only are a major dietary staple consumed, but an economically valuable commodity is also recognized. However, apple plants are highly vulnerable to a variety of diseases that can significantly reduce yield and fruit quality. These infections can be classified into abiotic and biotic categories [3]. Abiotic infections are caused by non-living factors such as extreme weather and pollutants, while biotic infections are caused by living organisms such as fungi, bacteria, and viruses, all of which pose serious threats to the productivity and profitability of apple orchards. The traditional methods used to identify these diseases, such as Fluorescence In-Situ Hybridization (FISH), Polymer Chain Reaction (PCR), and Immunofluorescence (IF), require significant expertise and can only be performed by trained phytopathologists in laboratory environments [3]. This reliance on experts can delay disease identification and subsequent treatment, leading to severe economic losses for farmers.

Recent studies have demonstrated the advantages of deep learning techniques for plant disease classification, showcasing their superiority over traditional machine learning models. For instance, researchers showed that deep learning models like InceptionV3, VGG-16, and VGG-19 achieved the best accuracy of 89.5%, surpassing traditional approaches like Random Forest (RF), Stochastic Gradient Descent (SGD), and Support Vector Machine (SVM), which achieved an accuracy of only 87% [4]. Another study compared MobileNet, InceptionV3, and ResNet152, achieving accuracies of 73.50%, 75.59%, and 77.65% respectively [5]. Despite these promising results, a common limitation is shared by many studies: datasets with simplified environmental conditions, such as uniform or uncomplicated backgrounds, are used.

Various approaches have been undertaken to develop plant disease classification systems using deep learning techniques. In 2019, researchers in India employed a CNN-based model using the Plant Village dataset and attained an accuracy of 98.54% [6]. The Plant Village dataset is significant due to its large and diverse collection of labeled plant leaf images, but it lacks the environmental complexity of real-world agricultural settings. In 2021, another study used VGG16 to classify diseases affecting citrus plants, achieving an accuracy of 89.5%, but again the dataset lacked environmental complexity [7]. More recently, in 2022, researchers proposed a plant disease classification method based on an optimized lightweight YOLOv5 model, incorporating Ghostnet and WBF to reduce model weight and BiFPN for enhanced feature fusion, resulting in an accuracy of 92.57% [8]. In 2023, an updated version of MobileNet known as MobileNetV2 achieved 99.36% accuracy [9], but, similar to earlier studies, this high accuracy was obtained using a dataset with simple backgrounds.

Advances in image processing technology, artificial intelligence, and computational resources such as graphics processing units (GPUs) could revolutionize the process of plant disease detection [10]. This study aim to address this gap by using a dataset with complex backgrounds to better represent the practical challenges of apple disease classification [11]. By incorporating images with more natural, varied environments, the goal is to develop a model that is robust to different conditions. The model will be trained on a complex background dataset and subsequently tested on simpler datasets [12] to evaluate its generalizability and effectiveness across diverse environments, demonstrating its versatility and robustness.

This work utilize a modified version of EfficientNet as the backbone for apple disease classification. EfficientNet [13] has become well-known for its ability to scale effectively while achieving higher accuracy compared to basic CNN architectures [14]. Its scaling strategy uniformly adjusts depth, width, and resolution, providing an efficient balance between accuracy and computational requirements. This makes EfficientNet an excellent candidate for plant disease classification, particularly when resources are limited. In this research, EfficientNet is further modified by removing specific blocks to achieve faster computation while maintaining accuracy. This modification is intended to streamline the model, making it lighter and more suitable for practical deployment in settings where computational resources are constrained, such as rural farms or portable devices.

II. Proposed Method

This chapter presents the proposed method for classifying apple plant diseases using a modified version of EfficientNet. The approach aims to address the gaps identified in previous studies by using a dataset that captures real-world complexities, such as diverse and cluttered backgrounds, rather than the simplified conditions used in earlier works. The proposed method involves dataset preparation, modification of the EfficientNet architecture, and experimental setup.

A. Experimental Schemes

To comprehensively evaluate the model, two different experimental schemes were designed:

1) Original EfficientNet: In this schema there are three different scenarios to evaluate the performance of the original EfficientNet model under various conditions:

a) Freeze All Layers with Pre-Trained Weights: In this scenario, all layers were frozen during training. Pre-trained weights from ImageNet were utilized to evaluate the model's ability to perform on the datasets without further tuning.

b) Trainable with Pre-Trained Weights: In this scenario, 50% of the layers froze and 50% set as trainable. The model started with pre-trained weights from ImageNet, and was fine-tuned to better adapt to the dataset.

c) Trainable with No Pre-Trained Weights: In this scenario, the original EfficientNet was trained from scratch with no pre-trained weights. This configuration allowed observation of how well the model could learn directly from the dataset without relying on prior knowledge.

2) Modified EfficientNet: In this schema, a modified version of EfficientNet was used, where certain blocks were removed to decrease computational load. This modified architecture was then trained on the dataset to assess if a lighter model could still maintain high classification accuracy.

B. Datasets

The dataset used in this study consists of two parts: a dataset with complex backgrounds and a dataset with simplified backgrounds.

The use of two datasets, one with complex backgrounds (Plant Pathology Dataset) and one with simplified backgrounds (Plant Village Dataset), is crucial for evaluating the model's generalization capability. The complex dataset mimics real-world conditions with varied elements, testing the model's robustness in challenging scenarios, while the simplified dataset serves as a baseline with minimal background noise. Training on the complex dataset and testing on the simplified one helps determine if the model generalizes well.

The dataset with complex backgrounds contains 1730 images of apple leaves affected by various diseases, collected under realistic agricultural conditions. The Plant Pathology Dataset consists of three different classes: two disease classes, rust and scab, and one healthy class. The number of images for each class in the dataset is shown in TABLE I. as a training set, and example images of each class can be seen in Fig. 1

TABLE I. DATASET DISTRIBUTION

Class Name	*Number of Images*	
	Training Set	***Test Set***
Healthy	516	1645
Rust	622	275
Scab	592	630
Total Images	1730	2550

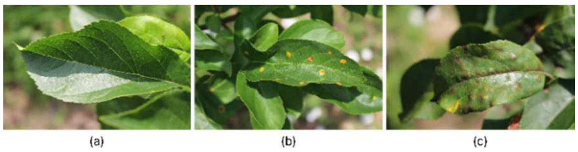

Fig. 1. Examples of apple leaf images from plant pathology dataset, showing various diseases categories: (a) healthy, (b) rust, (c) scab.

The dataset with simplified backgrounds, referred to as the Plant Village Dataset, was used for testing purposes and contains 2550 images of apple leaves with plain backgrounds. This dataset includes the same three classes i.e. rust, scab, and healthy. The number of images for each class in the simplified background dataset is shown in TABLE I. as a test set, and examples of each class can be seen in Fig. 2

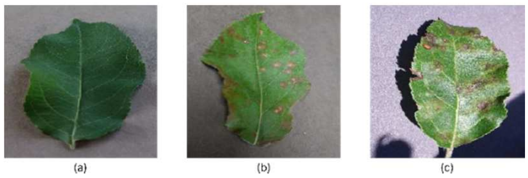

Fig. 2. Examples of apple leaf images from plant village dataset, showing various diseases categories: (a) healthy, (b) rust, (c) scab.

C. Modified EfficientNet

The main focus of the method is a modified version of the EfficientNet architecture. To make EfficientNet more suitable for apple disease classification, particularly in complex environments, several modifications were made to improve its computational efficiency.

To achieve this, certain blocks were removed from the original EfficientNet architecture. The purpose of this modification was to minimize the number of parameters and cut down computation time, resulting in a lighter model better suited for real-time applications. The selection of which blocks to remove was determined based on experimental analysis, which will be detailed in Chapter III and the architecture of the modified EfficientNet is illustrated in Fig. 3. This analysis considered the contribution of each block to overall model complexity and accuracy. Through an iterative evaluation process, it was determined that Blocks 5 and 6, which utilize larger kernel configurations, contributed disproportionately to computational demands relative to their impact on accuracy. Removing these blocks resulted in a reduction of the parameter count by 50%

The goal is to achieve a balance between model accuracy and computational efficiency, ensuring that the modified EfficientNet performs effectively without demanding excessive resources. Its lightweight nature enables it to run directly on smartphones, using machine learning frameworks like TensorFlow Lite allowing farmers to capture leaf images and receive instant disease diagnoses even in remote locations with limited connectivity. This offline capability, combined with the model's low latency, facilitates real-time disease detection in the field. Additionally, edge devices like Raspberry Pi can support automated and continuous crop monitoring, making this model an accessible, practical solution for plant health assessment.

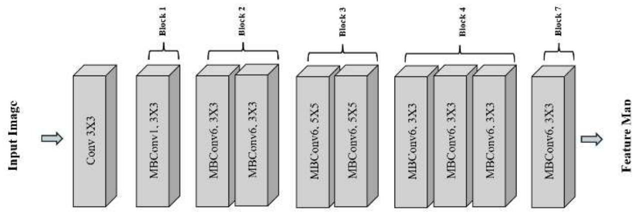

Fig. 3. Architecture of the modified EfficientNet with block 5 and 6 removed to enhance computational efficiency.

D. Experiment Setup

The experiments were performed on a cloud server featuring an NVIDIA RTX 4090 GPU with 24GB VRAM, 100GB RAM, and using TensorFlow 2.17.0 along with Python 3.12.4. These hardware and software choices were suitable for the experiments due to their ability to handle computationally intensive tasks, such as deep learning model training, with high efficiency and stability. As outlined in the experimental schemes, four different scenarios were implemented to determine if the modified EfficientNet could provide improved results and to identify the best variation of EfficientNet. The input size for the model varied depending on the specific EfficientNet variation used, ranging from 224x224 pixels to 600x600 pixels.

All models used the same hyperparameter settings, which are detailed in TABLE II. To strengthen the training dataset and boost model robustness, data augmentation techniques were employed. These included preprocessing, zoom variations of 20%, width and height shifts up to 20%, shear transformations of 20%, rotations up to 40 degrees, horizontal flipping, and nearest neighbor interpolation for filling. These methods simulated diverse environmental conditions and mitigated overfitting.

TABLE II. HYPERPARAMETERS SETTING

Hyperparameter	*Value*
Optimizer	Adam
Batch Size	32
Epoch	100
Dropout Rate	0.2

After training, the model was tested on both the Plant Pathology Dataset and the Plant Village Dataset to evaluate its generalizability. This contrast allows for the assessment of the model's robustness in both challenging and controlled scenarios. Once the best-performing modified EfficientNet variation was identified based on test accuracy, its performance was further validated using k-fold cross-validation [15] with k=5. This two-step evaluation approach helped confirm that the modifications to the EfficientNet architecture successfully enhanced its ability.

III. RESULT AND ANALYSIS

This chapter presents and analyzes the results obtained from the experimental setups discussed in Chapter II. The primary objective is to evaluate the effectiveness of the original and modified versions of EfficientNet for apple leaf disease classification. A detailed comparison of the four scenarios tested is provided, focusing on accuracy, computational efficiency, and overall robustness.

The models were assessed using multiple metrics such as accuracy, precision, recall, and F1-score, offering a comprehensive evaluation of performance across various classification aspects [16]. Accuracy reflects the overall percentage of correct classifications, while precision and recall offer insights into the model's behavior in classifying specific classes such as rust, scab, and healthy leaves.

A. Original EfficientNet Results

From the original EfficientNet with all layers frozen scenarios, the EfficientNetB6 variant provided the highest training and validation accuracy, reaching 95.40%, but only achieved a test accuracy of 76.30%, with a total training time of 1740 seconds. On the other hand, the variant with the highest test accuracy was EfficientNetB3, achieving 80.06% accuracy with a total training time of 854 seconds. The results indicate that although the pre-trained features were useful, the frozen architecture restricted the model's ability to extract more features from complex backgrounds. More detailed information on each variation can be found in TABLE III.

TABLE III. PERFORMANCE SUMMARY OF ORIGINAL EFFICIENTNET WITH ALL LAYERS FROZEN AND PRE-TRAINED WEIGHTS

Model	*Train Accuracy*	*Val Accuracy*	*Test Accuracy*	*Training Duration*
EfficientNetB0	0.899	0.922	0.756	**339**
EfficientNetB1	0.913	0.928	0.793	662
EfficientNetB2	0.916	0.908	0.771	674
EfficientNetB3	0.930	0.925	**0.806**	854
EfficientNetB4	0.932	0.945	0.772	832
EfficientNetB5	0.942	0.951	0.785	1036
EfficientNetB6	**0.954**	**0.968**	0.763	1740
EfficientNetB7	0.937	0.931	0.762	1995

From the original EfficientNet with 50% of layers trainable, 50% of layers freeze, and initialized with ImageNet scenarios, the EfficientNetB7 variant achieved the highest training, validation, and test accuracy compared to the other models, with values of 98.80%, 97.70%, and 84.0%, respectively. Despite its high training accuracy, the training time was significantly long, totaling 5615 seconds. The partially trainable model demonstrated improved adaptability, resulting in better performance. Fine-tuning allowed the model to learn specific disease patterns, enhancing its accuracy particularly in real-world scenarios. More detailed information on each variation can be found in TABLE IV.

TABLE IV. PERFORMANCE SUMMARY OF ORIGINAL EFFICIENTNET WITH 50% TRAINABLE LAYERS AND PRE-TRAINED WEIGHTS

Model	*Train Accuracy*	*Val Accuracy*	*Test Accuracy*	*Training Duration*
EfficientNetB0	0.875	0.917	0.804	**1316**
EfficientNetB1	0.896	0.914	0.775	1410
EfficientNetB2	0.915	0.934	0.827	1514
EfficientNetB3	0.935	0.951	0.816	1766
EfficientNetB4	0.961	0.954	0.836	2320
EfficientNetB5	0.966	0.968	0.824	2958
EfficientNetB6	0.981	0.968	0.816	4128
EfficientNetB7	**0.988**	**0.977**	**0.840**	5615

From the original EfficientNet with all the layers trainable and no pre-trained weights scenarios, the EfficientNetB4 variant achieved the highest training, validation, and test accuracy compared to the other models, with values of 98.40%, 96.80%, and 82.20%, respectively, with a total training time of 5462 seconds. Additionally, the EfficientNetB7 variant required a training time of 10074 seconds, which is nearly twice the time compared to the scenario with 50% of layers frozen and 50% trainable, which took 5615 seconds.

This scenario was particularly important to evaluate because the modified EfficientNet cannot use pre-trained ImageNet weights, thus it was necessary to establish a baseline for comparison between the modified and non-modified models. The model struggled without any pre-existing knowledge, especially when dealing with the complex dataset. This demonstrates the importance of pre-trained weights especially when the dataset size is relatively limited. More detailed information on each variation can be found in TABLE V.

TABLE V. PERFORMANCE SUMMARY OF ORIGINAL EFFICIENTNET WITH ALL LAYERS TRAINABLE AND NO PRE-TRAINED WEIGHTS

Model	*Train Accuracy*	*Val Accuracy*	*Test Accuracy*	*Training Duration*
EfficientNetB0	0.869	0.842	0.742	**2876**
EfficientNetB1	0.810	0.819	0.115	3139
EfficientNetB2	0.981	0.966	0.792	3417
EfficientNetB3	0.593	0.632	0.177	3999
EfficientNetB4	**0.984**	**0.968**	**0.822**	5462
EfficientNetB5	0.373	0.394	0.108	6692
EfficientNetB6	0.377	0.371	0.108	8218
EfficientNetB7	0.389	0.451	0.108	10074

B. *Modified EfficientNet Results*

Based on experimental results regarding the removal of blocks in various EfficientNetB0 configurations, as shown in TABLE VI. It was observed that removing Blocks 5, 6, and 7 results in a lower total parameter count compared to the removal of other blocks. Further analysis was conducted to compare the removal of Blocks 5 and 6 with the removal of Blocks 6 and 7.

TABLE VI. SHOWING THE EFFECT OF REMOVING DIFFERENT BLOCKS IN EFFICIENTNETB0

Block Removed	*Total Parameter*	*Accuracy*
None	5330571	0.247
Block 1	5330563	0.721
Block 2	5311649	0.829
Block 3	5278395	0.829
Block 4	5063161	0.850
Block 5	4757951	0.797
Block 6	3194015	0.745
Block 7	4444251	0.793
Block 5 and 6	2606035	**0.857**
Block 6 and 7	**2297455**	0.756

The number of parameters before and after the modification can be seen in TABLE VII. and the architecture of the modified EfficientNet is illustrated in Fig. 3.

TABLE VII. PARAMETER COUNT COMPARISON FOR ORIGINAL AND MODIFIED EFFICIENTNET

Model	*Total of Parameters (In Million)*	
	Original	*Modified*
EfficientNetB0	5.3M	2.6M
EfficientNetB1	7.9M	4.3M
EfficientNetB2	9.2M	5M
EfficientNetB3	12.3M	6M
EfficientNetB4	19.5M	8M
EfficientNetB5	30.6M	14.7M
EfficientNetB6	43.3M	18.7M
EfficientNetB7	66.7M	30M

The modified version of EfficientNet achieved the highest test accuracy and lowest training time among several other scenarios before. The modified EfficientNetB0 variant provided the highest training, validation, and testing accuracy among all modified EfficientNet, reaching 99.10%, 97.40%, and 84.50% respectively. This variant also had the shortest training time among models with high accuracy, totaling 2760 seconds. More detailed information on each variation can be found in TABLE VIII.

The modified EfficientNetB0 model demonstrates strong performance in classifying healthy leaves, as evidenced by the confusion matrix (Fig. 4), training history (Fig. 5), and classification report (TABLE IX.). The model achieves high precision (0.96), recall (0.99), and F1-score (0.97) for the

'Healthy' class, with 1623 correct predictions and minimal misclassifications. During training, accuracy rapidly increased to 95-99% within the first 20 epochs, and the training loss remained consistently low, suggesting effective learning and robust error minimization.

The confusion matrix and classification report indicate that the 'Rust' class shows lower precision, recall, and F1-score compared to the 'Healthy' and 'Scab' classes. This performance gap is likely due to the visual similarity between 'Rust' and 'Scab', which often exhibit similar colors and textures, making them difficult for the model to distinguish. Furthermore, the complex backgrounds present in the training images may have compounded the issue, as 'Rust' symptoms tend to blend with the natural variations in leaf color.

To enhance the classification performance for the 'Rust' class, employing image segmentation techniques to isolate leaf regions could be beneficial. Additionally, preprocessing methods that emphasize unique features of 'Rust' may help improve differentiation. Increasing the number of high-quality labeled images for the 'Rust' class could further assist the model in learning the subtle visual differences more effectively.

TABLE VIII. PERFORMANCE SUMMARY OF MODIFIED EFFICIENTNET WITH ALL LAYERS TRAINABLE AND NO PRE-TRAINED WEIGHTS

Model	*Train Accuracy*	*Val Accuracy*	*Test Accuracy*	*Training Duration*
EfficientNetB0	**0.991**	**0.974**	**0.845**	**2760**
EfficientNetB1	0.990	0.972	0.816	2983
EfficientNetB2	0.988	0.971	0.798	3260
EfficientNetB3	0.988	0.966	0.717	3894
EfficientNetB4	0.984	0.973	0.647	5386
EfficientNetB5	0.703	0.693	0.647	6439
EfficientNetB6	0.414	0.497	0.108	8034
EfficientNetB7	0.407	0.445	0.496	9324

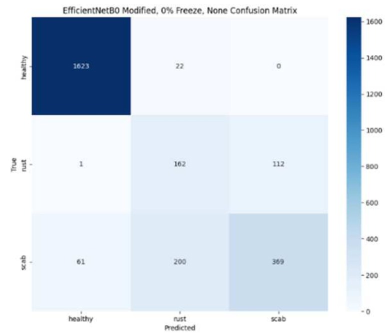

Fig. 4. Confusion matrix for modified EfficientNetB0.

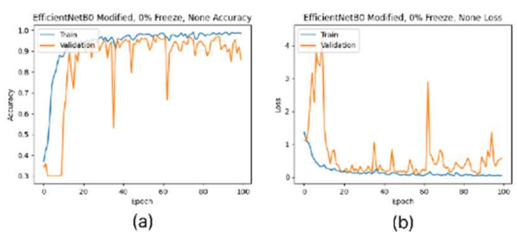

Fig. 5. Training history of modified EfficientNetB0: (a) training and validation accuracy, (b) training and validation loss.

TABLE IX. CLASSIFICATION REPORT FOR MODIFIED EFFICIENTNETB0

	Precision	*Recall*	*F1-Score*	*Support*
Healthy	0.96	0.99	0.97	1645
Rust	0.42	0.59	0.49	275
Scab	0.77	0.59	0.66	630
Accuracy			0.84	2550
Macro Avg	0.72	0.72	0.71	2550
Weighted Avg	0.86	0.84	0.85	2550

To evaluate the model's generalization ability, a retraining experiment was conducted with reversed dataset usage. Initially, the model was trained on the complex-background dataset and tested on the simple-background dataset. In the revised setup, the roles were swapped. Unlike the initial scenario that required 20 epochs to achieve over 95% accuracy, this time it only took 10 epochs, indicating effective learning. However, the model's accuracy on test data dropped to 39%, revealing some generalization issues.

The lower accuracy is likely due to the simple-background datasets have lower resolution (256x256 pixels), which results in fewer features for the model to learn from. In contrast, the complex-background dataset, with its higher original resolution (2048x1365 pixels), retains more features even after resizing both datasets to 224x224 pixels.

To address this, it is recommended to segment the complex-background dataset to focus on leaves while removing irrelevant elements such as soil.

C. Comparative Analysis

The comparison of all four scenarios reveals that the modified EfficientNet performed best, both in terms of accuracy and computational efficiency. The modified EfficientNet achieved the highest accuracy on test datasets as detailed in TABLE X. , highlighting its robustness in handling varying environmental conditions.

TABLE X. COMPARING THE PERFORMANCE OF VARIOUS EFFICIENTNET MODELS

Model	*Accuracy*	*Weighted F1-Score*	*Parameter*	*Training Time*
Original EN B3	0.806	0.80	12.3	**854**
Original EN B7 50% Freeze	0.840	0.82	66.7	5615
Original EN B4 0% Freeze	0.822	0.79	19.5	5462
Modified EN B0	**0.845**	**0.85**	**2.6**	2760

[a] Paremeter in million

[b] Training time in seconds

The additional testing conducted using the best model, EfficientNetB0 Modified, involved evaluations on various datasets, as outlined in TABLE XI. For this testing phase, the input shape was adjusted to match the original size of the test dataset i.e. 256x256 pixels. The model achieved an accuracy of 87.60% on the original test set, indicating robust performance in a standard test setting. For a down sampled version of the test set, the accuracy slightly decreased to 72.0%, on a segmented test set, where non-leaf portions of images were removed, the model achieved an accuracy of

81.50%, showing that segmentation improves performance slightly but remains lower than the original test set.

TABLE XI. RESULTS OF BEST MODEL TESTED ON DIFFERENT TEST SET

Model	*Dataset*	*Test Accuracy*
Modified EfficientNet B0	Test Set	0.876
	Test Set Downsampled	0.720
	Segmented Test Set	0.815

D. Cross-validation Results

A 5-fold cross-validation was conducted on the best-performing model, the modified EfficientNetB0, to ensure its robustness and consistency. As presented in TABLE XII. , the results demonstrate stable and reliable performance across various metrics. The model attained a strong mean training accuracy of 97.80% and a validation accuracy of 96.70%, demonstrating effective learning and minimal signs of overfitting. The test accuracy averaged 82.0%, supporting the model's generalization capabilities. The mean precision, recall, and F1-score values were 79.90%, 60.70%, and 59.80%, respectively, with low standard deviations, showcasing consistent performance across different folds.

TABLE XII. SUMMARY OF 5-FOLD CROSS-VALIDATION RESULTS FOR THE MODIFIED EFFICIENTNETB0 MODEL

	Train Accuracy	*Val Accuracy*	*Test Accuracy*	*Precision*	*Recall*	*F1-Score*
Mean	0.978	0.967	0.820	0.799	0.607	0.598
Std	0.004	0.012	0.046	0.059	0.083	0.101

IV. CONCLUSION

This study evaluated both the original and modified versions of EfficientNet to classify apple plant diseases using leaf images. Four different training configurations were experimented and the modified EfficientNet with certain blocks removed, achieved an optimal balance between accuracy and computational efficiency. Specifically, the modified EfficientNetB0 attained highest training, validation, and testing accuracy reaching 99.10%, 97.40%, and 84.50% respectively, while significantly reducing training time compared to other high-accuracy models. This makes it particularly suitable for practical, real-world applications in agriculture.

The main contribution of this research is the development of a computationally efficient yet highly accurate model for apple plant disease classification. The modifications made to the architecture allowed it to generalize effectively across datasets with varying levels of complexity, demonstrating its versatility for different agricultural environments.

Despite these promising results, there are some limitations. The datasets, while diverse, were relatively small, which could limit the model's ability to generalize to other plant diseases or crop types. Future work should focus on expanding the dataset to include more plant species and disease types to improve robustness. Additionally, the modified EfficientNet should be tested across different crops to validate its adaptability. Exploring further architectural enhancements, such as incorporating attention mechanisms, could help boost accuracy without significantly increasing computational costs. Finally, integrating this model into a decision-support system for farmers, with real-time alert capabilities, could enhance its practical impact, helping to reduce crop losses more effectively.

REFERENCES

[1] A. Wijerathna-Yapa and R. Pathirana, "Sustainable Agro-Food Systems for Addressing Climate Change and Food Security," *Agriculture*, vol. 12, no. 10, p. 1554, Sep. 2022, doi: 10.3390/agriculture12101554.

[2] V. K. Vishnoi, K. Kumar, B. Kumar, S. Mohan, and A. A. Khan, "Detection of Apple Plant Diseases Using Leaf Images Through Convolutional Neural Network," *IEEE Access*, vol. 11, pp. 6594–6609, 2023, doi: 10.1109/ACCESS.2022.3232917.

[3] V. K. Vishnoi, K. Kumar, and B. Kumar, "Plant disease detection using computational intelligence and image processing," *Journal of Plant Diseases and Protection*, vol. 128, no. 1, pp. 19–53, Feb. 2021, doi: 10.1007/s41348-020-00368-0.

[4] R. Sujatha, J. M. Chatterjee, N. Jhanjhi, and S. N. Brohi, "Performance of deep learning vs machine learning in plant leaf disease detection," *Microprocess Microsyst*, vol. 80, p. 103615, Feb. 2021, doi: 10.1016/j.micpro.2020.103615.

[5] C. Bi, J. Wang, Y. Duan, B. Fu, J.-R. Kang, and Y. Shi, "MobileNet Based Apple Leaf Diseases Identification," *Mobile Networks and Applications*, vol. 27, no. 1, pp. 172–180, Feb. 2022, doi: 10.1007/s11036-020-01640-1.

[6] S. Baranwal, S. Khandelwal, and A. Arora, "Deep Learning Convolutional Neural Network for Apple Leaves Disease Detection," *SSRN Electronic Journal*, 2019, doi: 10.2139/ssrn.3351641.

[7] S. Dahiya, T. Gulati, and D. Gupta, "Performance analysis of deep learning architectures for plant leaves disease detection," *Measurement: Sensors*, vol. 24, p. 100581, Dec. 2022, doi: 10.1016/j.measen.2022.100581.

[8] H. Wang, S. Shang, D. Wang, X. He, K. Feng, and H. Zhu, "Plant Disease Detection and Classification Method Based on the Optimized Lightweight YOLOv5 Model," *Agriculture*, vol. 12, no. 7, p. 931, Jun. 2022, doi: 10.3390/agriculture12070931.

[9] S. Banarase and S. Shirbahadurkar, "The Orchard Guard: Deep Learning powered apple leaf disease detection with MobileNetV2 model," *Journal of Integrated Science and Technology*, vol. 12, no. 4, Feb. 2024, doi: 10.62110/sciencein.jist.2024.v12.799.

[10] S. Kaur, S. Pandey, and S. Goel, "Plants Disease Identification and Classification Through Leaf Images: A Survey," *Archives of Computational Methods in Engineering*, vol. 26, no. 2, pp. 507–530, Apr. 2019, doi: 10.1007/s11831-018-9255-6.

[11] R. Thapa, K. Zhang, N. Snavely, S. Belongie, and A. Khan, "The Plant Pathology Challenge 2020 data set to classify foliar disease of apples," *Appl Plant Sci*, vol. 8, no. 9, Sep. 2020, doi: 10.1002/aps3.11390.

[12] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection," *Front Plant Sci*, vol. 7, Sep. 2016, doi: 10.3389/fpls.2016.01419.

[13] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," May 2019, [Online]. Available: http://arxiv.org/abs/1905.11946.

[14] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks," Nov. 2015.

[15] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-Validation," in *Encyclopedia of Database Systems*, Boston, MA: Springer US, 2009, pp. 532–538. doi: 10.1007/978-0-387-39940-9_565.

[16] S. S. Harakannanavar, J. M. Rudagi, V. I. Puranikmath, A. Siddiqua, and R. Pramodhini, "Plant leaf disease detection using computer vision and machine learning algorithms," *Global Transitions Proceedings*, vol. 3, no. 1, pp. 305–310, Jun. 2022, doi: 10.1016/j.gltp.2022.03.016.