Classification of Apple Leaf Diseases Using a Modified EfficientNet Model

Muhammad Abiya Makruf, Febryanti Sthevanie, Kurniawan Nur Ramadhani

Telkom University
Bandung, Indonesia

Camera view

Introduction

Agriculture is the main source of livelihood for approximately **80% of rural populations in developing nations**.

Traditional methods like Fluorescence In-Situ Hybridization (FISH) and Polymer Chain Reaction (PCR) require Laboratory-based resources.

Develop a **computationally efficient model** for identifying apple plant diseases.

Camera view

Related Work

References	Model	Result
R. Sujatha et al. [1]	SVM	87%
C. Bi et al. [2]	MobileNet	73.50%
S. Baranwal et al. [3]	GoogLeNet	98.42%
S. Dahiya et al. [4]	VGG16	89.50%
H. Wang et al [5]	YOLOv5	92.57%

Camera view

Gap Analysis

- A major limitation in current research is the **use of datasets with simple backgrounds.**

Objective and Contribution

Objective
- Enhance model efficiency while maintaining high accuracy.

Contributions
- Modification of EfficientNet Architecture
- Evaluation on Complex Background Datasets

Camera view

Dataset Overview


(a) (b) (c)

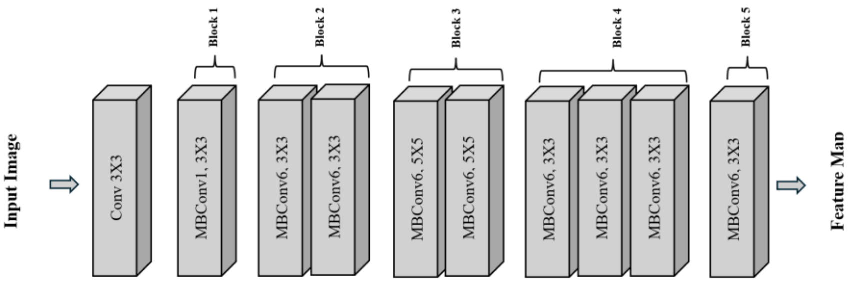

(a) (b) (c)

- Combined dataset from 2 sources (4280 images).
- Divided into three different classes, namely (a) healthy, (b) rust, and (c) scab.

Camera view

Proposed Method

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



Camera view

Experimental Schemes

1. **Original EfficientNet with Pre-trained Weights**
    - All layers of the original EfficientNet model were **frozen** during training.

2. **EfficientNet with Partially Trainable Layers**
    - **50% of the layers were trainable**, while the rest were frozen.

3. **EfficientNet Without Pre-trained Weights**
    - All layers were **trainable**, and the model was trained **from scratch** without any pre-trained weights.

4. **Modified EfficientNet**
    - The modified EfficientNet architecture with **Blocks 5 and 6 removed.** Trained with all layers trainable and no pre-trained weights.

Camera view

Experiment Setup

- **Input Size**: Varied depending on the EfficientNet variant, ranging from **224x224** to **600x600** pixels.
- NVIDIA RTX 4090
- 100GB RAM
- TensorFlow 2.17.0
- Python 3.12.4

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

Camera view

Results – Original EfficientNet

All layers of the original EfficientNet model were **frozen** during training

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

- EfficientNetB6 variant provided the highest training and validation accuracy, reaching 95.40% and 96.80%
- But only achieved a test accuracy of 76.30%
- Total training time of 1740 seconds

Camera view

Results – Original EfficientNet

50% of the layers were trainable, while the rest were frozen.

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

- EfficientNetB7 variant provided the highest training, validation and test accuracy, reaching 98.80%, 97.70% and 84.0%
- Training time was significantly long, totaling 5615 seconds

Camera view

Results – Original EfficientNet

All layers were trainable, and the model was trained from scratch without any 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

- EfficientNetB4 variant provided the highest training, validation and test accuracy, reaching 98.40%, 96.80% and 82.20%
- Training time was significantly long, totaling 5462 seconds

Camera view

Results – Modified EfficientNet

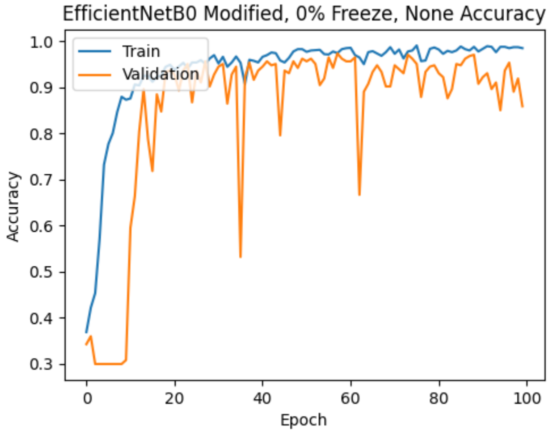
The modified EfficientNet architecture with Blocks 5 and 6 removed. Trained with all layers trainable and no pre-trained weights.

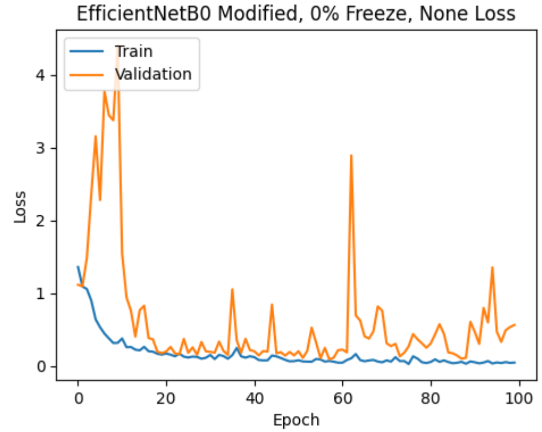
Model	*Train Accuracy*	*Val Accuracy*	*Test Accuracy*	*Training Duration*
EfficientNetB0	**0.991**	**0.974**	**0.845**	**2760**
EfficientNetB1	0.990	0.972	0.816	2983
EfficientNetB2	0.988	0.971	0.798	3260
EfficientNetB3	0.988	0.966	0.717	3894
EfficientNetB4	0.984	0.973	0.647	5386
EfficientNetB5	0.703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

- EfficientNetB0 variant provided the highest training, validation and test accuracy, reaching 99.10%, 97.40% and 84.50%
- This variant also had the shortest training time among models with high accuracy, totaling 2760 seconds

Camera view

Results – Best Model





	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

Camera view

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 EfficientNetB0** outperforms the others, achieving the highest accuracy while maintaining the lowest parameter count (**2.6 million**)

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

Camera view

Cross – Validation Result

A 5-fold cross-validation was conducted on the bestperforming model, the modified EfficientNetB0, to ensure its robustness and consistency.

The model attained a strong mean training and a validation accuracy

	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

Camera view

Discussion

- The confusion matrix and classification report indicate that the 'Rust' class shows lower precision, recall, and F1- score.
- This performance gap is likely due to the visual similarity between 'Rust' and 'Scab'.
- To enhance the classification performance for the 'Rust' class, employing image segmentation techniques to isolate leaf regions could be beneficial.

Camera view

Conclussion

Model Efficiency and Accuracy:

- The modified EfficientNetB0 achieved a balance between **high accuracy** (training: 99.10%, validation: 97.40%, testing: 84.50%) and **computational efficiency**, making it ideal for real-world agricultural applications.

Main Contribution:

- Developed a **computationally efficient and versatile model** capable of generalizing across datasets with varying complexities, demonstrating its adaptability for diverse agricultural environments.

Future Work:

- Expand datasets to include more plant species and disease types.
- Test adaptability across different crops and explore **attention mechanisms** to enhance accuracy.

Camera view

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

Camera view

Thank You

Camera view