

Optimizing CNN Architecture Selection For Small-Scale Imbalanced Chili Leaf Disease Classification

Muhamad Fery Ardiyansah*, Iwan Setiawan Wibisono

Fakultas Komputer dan Pendidikan, Program Studi Teknik Informatika, Universitas Ngudi Waluyo, Semarang, Indonesia

Email: ^{1,*}muhammadferyyy@gmail.com, ²iwansetiawan@unw.ac.id

Correspondence Author Email: muhammadferyyy@gmail.com

Received: 17/07/2026, Revised: 28/07/2026, Accepted: 21/09/2026, Published: 22/09/2026

Abstract—Chili leaf disease reduces yield and farmer income in Indonesia, yet automated diagnosis is typically evaluated using a single architecture or artificially balanced datasets that do not reflect field conditions. This study compared eight CNN architectures (ResNet50, ResNet101, EfficientNetB0, EfficientNetB3, MobileNetV2, MobileNetV3Large, VGG16, and DenseNet121) for six-class chili leaf disease classification on an original dataset of 1,847 images, using on-the-fly augmentation only on the training subset. Each architecture was trained with a frozen feature-extraction transfer-learning protocol and class weighting across three random seeds (24 runs total) to assess robustness. MobileNetV3Large achieved the highest mean accuracy (98.80%) and was selected as the final model, reaching 98.92% accuracy, a macro-average F1-score of 98.09%, and a weighted-average F1-score of 98.90%. A Friedman test confirmed that performance differences among the eight architectures were statistically significant ($\chi^2=19.432$, $p=0.0069$), with a Kendall's W of 0.9253 indicating strong ranking agreement across seeds. Grad-CAM visualization showed that the final model's predictions were consistently driven by symptomatic leaf regions rather than background artifacts. These results indicate that a lightweight, lower-parameter architecture can match or exceed deeper models under frozen feature extraction, offering a favorable performance-to-size trade-off for agricultural extension services. This study contributes a statistically validated benchmarking protocol enabling fair comparison across CNN design philosophies on a single, non-synthetically-balanced dataset, a setting most prior chili leaf disease studies have not provided.

Keywords: Chili Leaf Disease Classification; Frozen Feature Extraction; MobileNetV3Large; Class Weighting; Convolutional Neural Network

1. INTRODUCTION

Red chili (*Capsicum annum* L.) is one of the important vegetable commodities in Indonesia, both for domestic consumption and for export. Chili plants can grow under various conditions, but are frequently attacked by various diseases that can reduce the quality and quantity of the harvest. Therefore, optimal plant care is an important factor in maintaining chili productivity. However, various types of chili disease cause economic losses for farmers[1].

The chili leaf disease identification method that is commonly used today is still carried out manually through visual observation by farmers. This approach is subjective, relatively time-consuming, and has low accuracy[2]. Therefore, an automatic system capable of detecting chili leaf disease is needed to support early disease management. A digital image-based approach is one of the effective solutions because it can identify disease based on the visual characteristics of the plant leaves.

One of the technologies widely used in image classification is the Convolutional Neural Network (CNN). CNN has the ability to automatically extract visual features and recognize visual patterns in data with high accuracy[3]. Various CNN architectures have been developed with different characteristics, such as ResNet[4], EfficientNet[5], MobileNet, VGG[1], and DenseNet[6], each offering a different balance between accuracy and computational efficiency. CNN belongs to the deep learning branch of artificial intelligence and machine learning, in which multiple layers of representation are learned directly from data rather than engineered manually [7], [8].

In practice, one of the main challenges in plant disease image classification is the limited amount of data and imbalanced class distribution. This condition can reduce the accuracy of a model in recognizing objects. Another impact of an imbalanced dataset is a decrease in other model evaluation metrics, such as recall, precision, and F1-score[9]. To overcome this problem, the transfer learning approach is an effective strategy because it allows the use of model weights that have already been trained on large-scale datasets such as ImageNet. This approach can accelerate the training process and improve accuracy even when the local dataset is limited[10]. In addition, a frozen feature extraction strategy was applied by retraining only the classification head, while a class weighting technique was used to handle the imbalanced class distribution in the original dataset.

The selection of Frozen Feature Extraction over full fine-tuning or partial fine-tuning was a deliberate methodological decision rather than a simplification of the transfer-learning process. Fine-tuning updates a large number of parameters in the pretrained convolutional layers, which may be more susceptible to overfitting or representation distortion when applied to small target datasets, particularly through catastrophic forgetting, a condition in which previously learned feature representations lose their generalization ability because they are overwritten by gradients computed from a small target dataset [11]. With only 1,847 chili leaf images distributed across six imbalanced classes, fine-tuning risks damaging basic ImageNet features, such as edges, textures, and color gradients, that remain relevant for recognizing leaf disease patterns. Therefore, the convolutional layers were frozen and only the classification head was trained. This approach significantly reduces the number of parameters that need to be updated, lowers the risk of overfitting, accelerates training time, and is more suitable for environments with limited

computational resources. These considerations formed the basis for the architecture selection and experimental design of this study.

Several prior studies have examined the application of various CNN architectures for plant disease classification, including chili leaf disease, providing important findings while also revealing limitations that motivate the present study. Khoirunnisa et al. [1] compared MobileNetV2 and VGG16 on a chili leaf disease dataset and found that MobileNetV2 outperformed VGG16; however, the study compared only two lightweight architectures and did not cover other CNN architecture families. Munir et al. [3] applied a MobileNet architecture to a local chili leaf disease dataset and obtained reasonably good accuracy, but did not provide a comparison model, making it difficult to determine whether the result stemmed from the architecture's advantage or from dataset characteristics. Setyadi et al. [4] applied transfer learning combined with data augmentation and obtained an accuracy of 97.18%, but this result depended on additional synthetic data and therefore does not reflect model performance on the original dataset without offline augmentation. Azhari and Husain [5] evaluated only the EfficientNet family without comparing it to other architecture families such as ResNet or DenseNet. Meanwhile, Khasanah and Fachrie [9] used DenseNet201 and obtained a training accuracy of 90%, but test accuracy dropped to 84%, indicating a model generalization problem. Ristanti [12] compared VGG16 and MobileNetV2 on a combined field and public dataset, showing that MobileNetV2 achieved 92% accuracy while VGG16 achieved only 74%, but again involved only two architectures. Anggiratih et al. [13] applied EfficientNetB3 to rice leaf disease and obtained an accuracy of 79.53% under a small-dataset condition, showing that even modern architectures do not necessarily achieve optimal performance without an appropriate data-handling strategy.

Although many studies have applied CNN for plant disease classification, most of them still use a single dataset and focus on only one or two specific CNN architectures. Research comparing various modern CNN architectures on an original dataset without offline augmentation or synthetic class balancing, under an imbalanced class distribution, is still relatively limited. Therefore, this study proposes a comparison of eight CNN architectures, namely ResNet50, ResNet101, EfficientNetB0, EfficientNetB3, MobileNetV2, MobileNetV3Large, VGG16, and DenseNet121, to classify six classes of chili leaf disease using a frozen feature extraction transfer learning approach on an original dataset of 1,847 images.

Based on prior research, three research gaps form the focus of this study. First, CNN architecture comparisons in previous studies are generally limited to one or two architectures, and therefore do not provide a comprehensive picture of the performance of different CNN design philosophies under the same conditions. Second, studies reporting high accuracy generally rely on data augmentation or dataset merging to overcome limited data, rather than directly addressing class imbalance. Third, model generalization issues, such as the accuracy drop from training to test data or low performance on small datasets, are rarely analyzed in relation to the depth or capacity characteristics of the CNN architecture. To address these gaps, this study makes three main contributions: evaluating eight CNN architectures representing different design philosophies under a single, uniform Frozen Feature Extraction protocol; using the original dataset without offline augmentation or synthetic class balancing, applying on-the-fly augmentation only to the training subset, and handling class imbalance through class weighting rather than synthetic data addition; and analyzing the causes of low performance in some architectures based on overfitting and underfitting patterns, rather than final accuracy values alone. To the authors' knowledge, the combination of the Frozen Feature Extraction protocol, eight diverse CNN architectures, and an original, imbalanced chili leaf disease dataset has not been previously reported.

In summary, the primary contribution of this study is not the introduction of a new CNN architecture, but rather a carefully controlled, multi-seed comparative benchmark that evaluates eight established CNN design philosophies under one consistent frozen feature-extraction protocol on a small, imbalanced, original chili leaf disease dataset. By combining this unified protocol with statistical significance testing and Grad-CAM-based qualitative verification, this study aims to provide a more reliable and reproducible reference point than prior single-architecture or single-comparison studies, while acknowledging that, as an initial benchmarking effort on a single-source dataset, its findings are intended as a starting point for further validation rather than a definitive conclusion.

2. RESEARCH METHODOLOGY

2.1 Research Design

This study employed a comparative experimental framework to evaluate the performance of eight convolutional neural network (CNN) architectures for six-class chili leaf disease classification. The research stages consisted of dataset preparation, stratified dataset splitting, image preprocessing, on-the-fly data augmentation, class imbalance handling, transfer learning, multi-seed comparative experiments, statistical analysis, final model selection, and interpretability analysis using Grad-CAM. The overall research workflow is presented in Figure 1. The dataset was first prepared and divided into training, validation, and independent test subsets. Data augmentation was applied only to the training subset, while class weighting was used to address the imbalanced class distribution. Eight CNN architectures were subsequently evaluated using three random seeds to reduce the influence of stochastic training factors. The architecture performances were then compared using quantitative evaluation metrics and statistical analysis. The highest accuracy

architecture was selected for final training and independent test evaluation, followed by Grad-CAM analysis to investigate the visual regions contributing to model predictions.

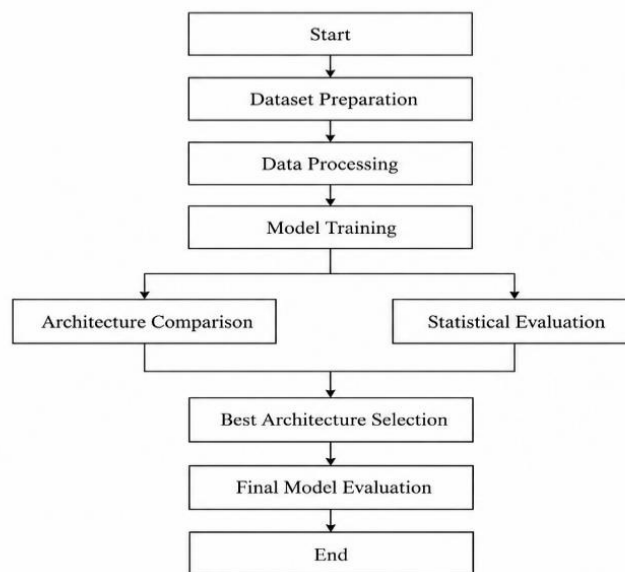


Figure 1. Research stages of the proposed CNN-based chili leaf disease classification framework

2.2 Dataset Preparation

2.2.1 Dataset Source and Verification

The dataset used in this study was obtained from the Mendeley Data repository and consisted of chili leaf images belonging to six classes: Bacterial Spot, Cercospora Leaf Spot, Curl Virus, Healthy Leaf, Nutrition Deficiency, and White spot. After dataset preparation and verification, 1,847 image files were included in the final experimental pipeline.

2.2.2 Stratified Train–Validation–Test Split

The dataset was divided into training, validation, and test subsets, consisting of 1,292, 277, and 278 images, respectively. The training subset represented approximately 69.95% of the dataset, while the validation and test subsets represented approximately 15.00% and 15.05%, respectively. The subsets were kept mutually exclusive to prevent data leakage between model optimization and final evaluation. The dataset was retained as the source dataset, while augmentation was performed dynamically during model training. Therefore, the reported dataset size refers to the number of original image files and does not include additional augmented images stored as separate files

2.2.3 Image Preprocessing

Each image was loaded and resized to 224×224 pixels with three color channels (RGB) using TensorFlow's `image_dataset_from_directory` utility, applied consistently to the training, validation, and test subsets. Pixel values were kept in their native 0–255 range and fed directly into the frozen pretrained backbone of each architecture without an additional explicit rescaling or architecture-specific normalization layer, consistent with the backbone operating in frozen inference mode (`training=False`) during feature extraction

2.2.4 On-the-Fly Data Augmentation

On-the-fly data augmentation was applied exclusively to the training subset using random rotation, random zoom, random brightness, and random contrast transformations. These transformations were used as a regularization strategy to improve model generalization. No augmentation was applied to the validation or test subsets to ensure that model performance was evaluated using unseen images under consistent conditions. The effectiveness of on-the-fly augmentation as a regularization strategy for CNN-based leaf/crack image classification with limited training data is consistent with recent findings [14].

2.3 Class Imbalance Handling

The dataset exhibited an imbalanced distribution among the six disease classes. To reduce the potential bias toward majority classes, class weighting was incorporated into the training process. The class weight for each class was calculated using the balanced weighting formulation:

$$w_i = \frac{N}{K \times n_i} \quad (1)$$

where w_i is the weight assigned to class i , N is the total number of training samples, K is the number of classes, and n_i is the number of training samples belonging to class i .

This inverse-frequency weighting formulation follows the standard balanced class-weight scheme commonly used to correct class-conditional bias in imbalanced multi-class classification, and its effectiveness relative to alternative resampling strategies (e.g., SMOTE) has been empirically demonstrated for imbalanced classification tasks [15]. The resulting class weights are presented in Table 1.

Table 1. Class Distribution and Class Weights

No	Class	Training Samples	Class Weight
1	Bacterial Spot	106	2.0314
2	Cercospora Leaf Spot	123	1.7507
3	Curl Virus	296	0.7275
4	Healthy Leaf	320	0.6729
5	Nutrition Deficiency	311	0.6924
6	White Spot	136	1.5833

Higher weights were assigned to minority classes, while lower weights were assigned to majority classes, confirming the intended inverse relationship between class frequency and assigned weight. The validity of the class-weight calculation was assessed mathematically by verifying that minority classes consistently received larger weights than majority classes; no experiment without class weighting was conducted in this study, so its practical effectiveness relative to an unweighted baseline is not established empirically. Class weighting and data augmentation served different purposes in the experimental framework: class weighting was used to address class imbalance during optimization, whereas on-the-fly augmentation was applied to improve generalization and reduce overfitting. The practical effectiveness of class weighting within this setup was instead evaluated through class-level precision, recall, F1-score, and confusion matrix analysis.

2.4 CNN Architectures and Experimental Protocol

2.4.1 Eight CNN Architectures

Eight representative CNN architectures were evaluated: ResNet50, ResNet101, EfficientNetB0, EfficientNetB3, MobileNetV2, MobileNetV3Large, VGG16, and DenseNet121. These architectures represent different CNN design approaches: VGG16 follows a plain sequential convolutional design and has been applied to Indonesian tomato-leaf disease classification [16]; ResNet50 and ResNet101 apply residual learning through identity skip connections and have been applied to rice-leaf disease classification in Indonesian studies [17]; DenseNet121 introduces dense connectivity, in which each layer receives the feature maps of all preceding layers, and has been applied to tomato leaf-mold identification [18]; EfficientNetB0 and EfficientNetB3 apply compound scaling, which jointly scales network depth, width, and input resolution, and have been applied to citrus leaf-disease classification [19]; MobileNetV2 uses depthwise separable convolutions with inverted residual bottlenecks, and together with MobileNetV3Large a lightweight, NAS-optimized CNN, has been benchmarked against ResNet50 for Indonesian batik motif classification [20].

2.4.2 Architectural Heterogeneity Rationale

The architectures were intentionally heterogeneous in terms of network depth, parameter count, and internal feature extraction mechanisms. Therefore, this study did not aim to isolate the individual effect of network depth or width through a controlled architectural ablation experiment. Instead, the architectures were benchmarked as representative CNN models under a common experimental protocol.

2.4.3 Frozen Feature Extraction Strategy

To ensure a consistent comparison, all architectures were evaluated using the same dataset partitions, preprocessing strategy, augmentation procedure, class-weighting method, evaluation metrics, and experimental seed conditions. Transfer learning was implemented using a frozen feature extraction strategy. The convolutional backbone of each pretrained CNN architecture was initialized with pretrained weights and kept frozen during training, while only the newly added classification head was optimized for the six-class chili leaf disease classification task.

This strategy enabled the models to leverage general visual representations learned from large-scale image data while reducing the number of trainable parameters and computational requirements during training. This rationale for frozen feature extraction under limited labeled data is consistent with established transfer-learning literature for image classification with small datasets [11].

2.4.4 Multi-Seed Experimental Protocol

Each architecture was trained using three random seeds: 42, 123, and 2024. Thus, the comparative experiment consisted of 24 experimental runs, calculated as eight architectures multiplied by three random seeds. The results

obtained from these experiments were used to compare architecture performance and assess the robustness of the observed results across different random initializations.

2.5 Statistical Analysis

Model performance was evaluated using accuracy, precision, recall, F1-score, and confusion matrix analysis. Overall accuracy was used as the primary metric for comparing the eight CNN architectures, while precision, recall, and F1-score were used to assess class-specific performance under the imbalanced dataset distribution. The Friedman test was applied to determine whether statistically significant differences existed among the eight CNN architectures across the repeated experimental conditions. A significance level of $\alpha = 0.05$ was used. Kendall's W was additionally calculated to quantify the magnitude of agreement in the architecture rankings and complement the statistical significance analysis. The Friedman test statistic was computed as:

$$\chi_F^2 = \left[\frac{12N}{k(k+1)} \right] \times \sum_j R_j^2 - 3N(k+1) \quad (2)$$

where N is the number of experimental blocks (i.e., the three random seeds), k is the number of CNN architectures being compared ($k = 8$), and R_j is the sum of ranks obtained by architecture j across the N blocks. The resulting statistic was compared against the chi-square distribution with $(k - 1)$ degrees of freedom to determine statistical significance (Friedman, 1937, as cited in Demšar, 2006). Kendall's W (coefficient of concordance) was subsequently computed to quantify the strength of agreement among the architecture rankings, using:

$$W = \frac{12S}{[N^2(k^3 - k)]} \quad (3)$$

where S is the sum of squared deviations of each architecture's total rank from the mean total rank, N is the number of random seeds, and k is the number of architectures. W ranges from 0 (no agreement among rankings across seeds) to 1 (perfect agreement), and is related to the Friedman statistic through $W = \chi_F^2 / [N(k - 1)]$. The Friedman test with Kendall's W as an effect-size measure follows the standard non-parametric procedure recommended for comparing multiple classifiers across repeated experimental conditions.

2.6 Final Model Training and Evaluation

Based on the multi-seed comparative results, the highest accuracy architecture was selected for final model development. The selected model was subsequently evaluated on the independent test set using overall accuracy, loss, precision, recall, F1-score, and confusion matrix analysis.

2.7 Interpretability Analysis (Grad-CAM)

Finally, Gradient-weighted Class Activation Mapping (Grad-CAM) was employed to provide qualitative visual interpretation of the final model predictions. Grad-CAM was applied to the final convolutional layer of the selected model to visualize the image regions contributing most strongly to the predicted class. The analysis was conducted across the six disease categories to examine whether the model focused on visually relevant leaf regions when making its predictions. Grad-CAM was originally proposed as a gradient-based localization technique for producing visual explanations from CNN predictions [21], and has recently been applied specifically to lightweight CNNs for leaf-disease classification with results consistent with the interpretability analysis expected here [22].

3. RESULT AND DISCUSSION

3.1 Comparative Performance of Eight CNN Architectures

To address the reviewer's concern that a single training run is insufficient to demonstrate that performance differences between architectures are genuine rather than an artifact of stochastic training, each of the eight CNN architectures was retrained three times under identical data splits and hyperparameters, using three different random seeds (42, 123, and 2024). Table 2 reports the test accuracy obtained in each run together with the three-run mean and standard deviation.

Table 2. Test Accuracy of Eight CNN Architectures Across Three Random Seeds (sorted by mean accuracy)

Architecture	Seed 42	Seed 123	2024	Mean	SD
MobileNetV3Large	99.28%	98.92%	98.20%	98.80%	0.55
ResNet50	97.84%	98.92%	98.92%	98.56%	0.62
ResNet101	98.56%	98.56%	98.56%	98.56%	0.00
EfficientNetB0	98.20%	98.20%	97.84%	98.08%	0.21
EfficientNetB3	96.76%	97.12%	97.84%	97.24%	0.55
VGG16	92.81%	93.88%	94.96%	93.88%	1.08
DenseNet121	88.85%	91.73%	86.33%	88.97%	2.70
MobileNetV2	74.46%	76.98%	76.26%	75.90%	1.30

Among the eight architectures, the observed three-seed mean test accuracy ranged from 75.90% (MobileNetV2) to 98.80% (MobileNetV3Large). MobileNetV3Large, ResNet50, and ResNet101 obtained the highest observed means, clustering closely together at approximately 98.6–98.8%; EfficientNetB0 (98.08%) and EfficientNetB3 (97.24%) formed an intermediate tier; and VGG16 (93.88%), DenseNet121 (88.97%), and MobileNetV2 (75.90%) obtained comparatively lower observed means. DenseNet121 additionally showed the largest run-to-run variability among all eight architectures ($SD = 2.70$). As detailed in the following subsection, although the omnibus Friedman test indicated a statistically significant overall difference among the eight architectures, none of the pairwise post-hoc comparisons remained significant after Holm correction; these observed mean differences should therefore be read as a descriptive ranking rather than as evidence that any single architecture is definitively superior or inferior to another.

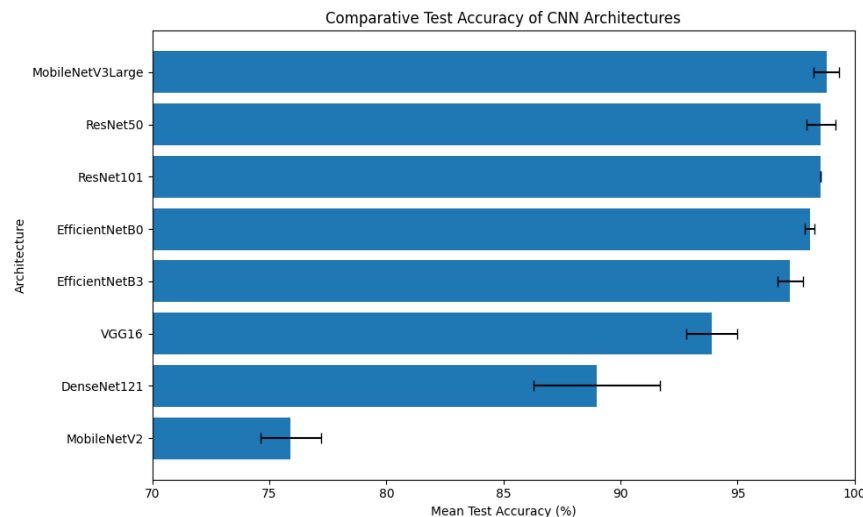


Figure 2. Mean Test Accuracy (%) of Eight CNN Architectures Across Three Random Seeds (Error Bars Indicate ± 1 SD)

This observed pattern differs from what raw parameter count alone would predict: MobileNetV3Large obtained its high observed mean accuracy with only 3.63 million parameters, while DenseNet121 (7.70M) and VGG16 (15.11M) obtained lower observed means than several both smaller and larger architectures. ResNet101, despite having the largest parameter count in this study (43.85M), performs on par with the much smaller ResNet50 (24.78M) rather than exceeding it. This pattern is discussed further below and in the Discussion section.

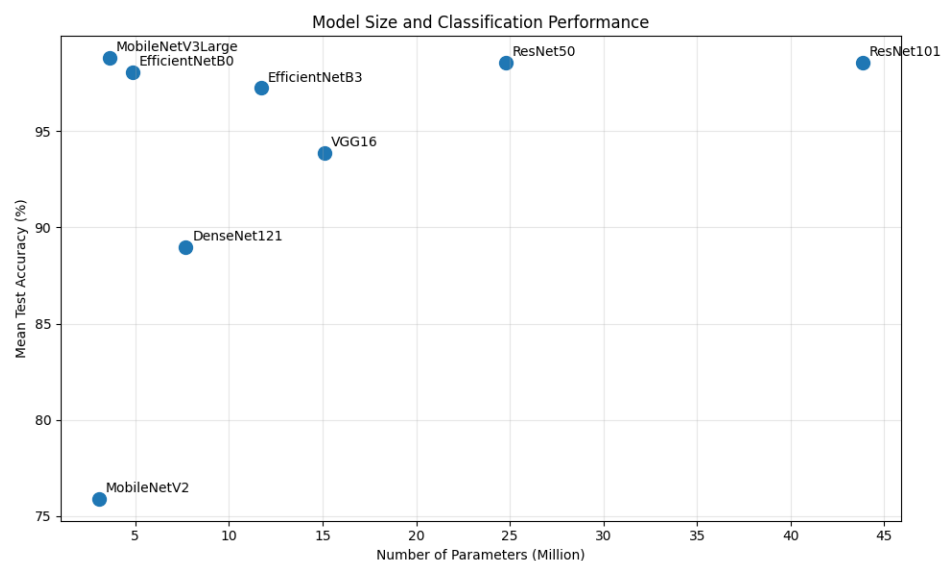


Figure 3. Relationship Between Model Size (Number of Parameters) and Mean Test Accuracy

As shown in Figure 3, parameter count did not reliably predict accuracy. MobileNetV3Large (3.63M parameters) achieved the highest mean accuracy (98.80%), while MobileNetV2 (3.05M, the smallest model) achieved the lowest (75.90%). Likewise, ResNet101 (43.85M) did not outperform the much smaller ResNet50 (24.78M). This indicates that model size alone does not determine classification performance under this protocol.

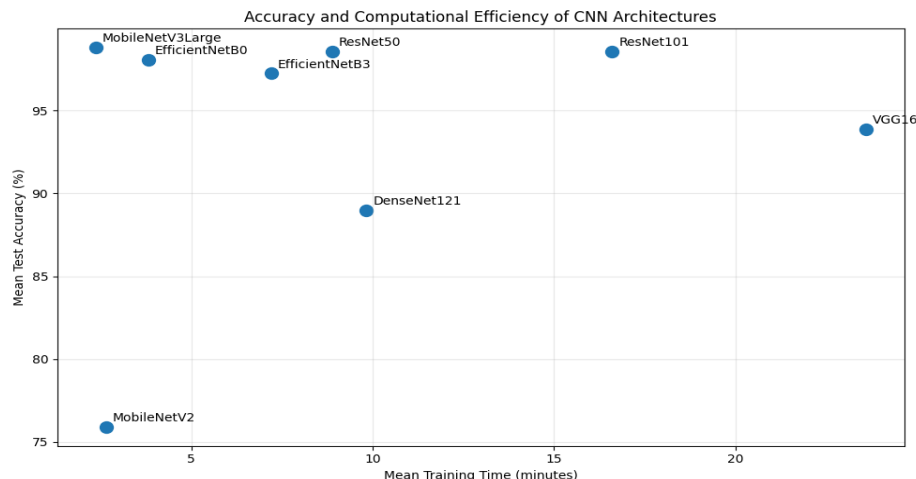


Figure 4. Mean Test Accuracy versus Mean Training Time of Eight CNN Architectures

As shown in Figure 4, training time also did not correspond to accuracy. VGG16 had the longest training time (23.59 minutes) yet only moderate accuracy (93.88%), and ResNet101 took twice as long as ResNet50 (16.61 vs. 8.89 minutes) without any accuracy gain. MobileNetV3Large and MobileNetV2 trained fastest (~2.4–2.7 minutes) but produced the highest and lowest accuracies, respectively. This confirms that architecture design, not training cost, was the more relevant factor.

3.2 Statistical Significance and Effect Size

A Friedman test was conducted on the three-seed accuracy values in Table 2 to test the null hypothesis that all eight architectures perform equally. The test indicated a statistically significant difference among the evaluated CNN architectures ($\chi^2 = 19.432$, $p = 0.0069 < 0.05$). The result provides evidence of systematic differences in architecture rankings across the three repeated training conditions, although the small number of seeds limits the statistical power of pairwise comparisons.

To quantify the strength of this result, Kendall's coefficient of concordance (W) was additionally computed as $W = \frac{12S}{[N^2(k^3 - k)]}$, where $n = 3$ seeds and $k = 8$ architectures, giving $W = 0.9253$. This value indicates a large effect and a high degree of agreement in architecture rankings across the three seeds that is, the relative ordering of the eight architectures in Table 3 was highly consistent from one random initialization to another, rather than being driven by a single favorable run.

A post-hoc pairwise Wilcoxon signed-rank test with Holm correction was subsequently applied to identify which specific architecture pairs differed significantly. After Holm correction, none of the 28 pairwise comparisons reached statistical significance at $\alpha = 0.05$ (all adjusted p -values = 1.00). This outcome, while at first appearing to be in tension with the significant omnibus Friedman result, is a well-documented consequence of the very small number of paired samples available for a pairwise post-hoc test: with only $n = 3$ seeds, the Wilcoxon signed-rank statistic can take only a few discrete values, so its raw p -values are bounded well above the level needed to survive a conservative Holm correction across 28 comparisons. The significant Friedman test together with the large Kendall's W provide evidence of systematic differences in architecture performance across the repeated experimental conditions; the non-significant post-hoc comparisons should be interpreted as a limitation of statistical power at $n = 3$ seeds rather than as evidence that the architectures are truly equivalent. This limitation, and the recommendation to increase the number of repeated runs in future work, is discussed further in Section 3.6.

3.3 Final Model Performance

Based on the comparison in Section 3.1, MobileNetV3Large was selected as the final model. It was retrained using the same frozen feature-extraction configuration and the same seed 42, which produced its highest accuracy among the three-seed run, then evaluated on the independent test set (278 images) that was held out from both training and validation. Table 3 summarizes the resulting training, validation, and test performance.

Table 3. Final Training, Validation, and Test Performance of MobileNetV3Large (Seed 42)

Class	Precision
Training Accuracy	98.22%
Training Loss	0.1269
Validation Accuracy	99.64%
Validation Loss	0.0879
Test Accuracy	98.92%
Test Loss	0.1164

The final test accuracy (98.92%) is closely consistent with the three-seed mean reported in Table 2 (98.80%), consistent with the selected run being representative of the model's typical performance rather than an outlier. The training and validation results do not indicate an obvious overfitting pattern. The validation accuracy (99.64%) remained slightly higher than the training accuracy (98.22%), while the validation loss (0.0879) was lower than the training loss (0.1269). This pattern is plausible under the applied training configuration, where data augmentation and dropout were active during training but not during validation. The final MobileNetV3Large model (seed 42), as reported by its `model.summary()` output, contained 3,002,118 total parameters, of which 5,766 parameters (0.19%) belonged to the classification head and were trainable under the frozen feature-extraction protocol, while 2,996,352 parameters (99.81%) in the pretrained backbone remained frozen at their ImageNet-pretrained values. This is the exact parameter count of the trained model; it differs slightly from the 3.63 million figure listed for MobileNetV3Large in Tables 3 and 4, which reflects the architecture's standard published parameter count and is used there only for cross-architecture size comparison. This indicates that the strong final performance was achieved by adapting a very small number of task-specific parameters rather than by re-learning the underlying visual feature extractor, consistent with the frozen feature-extraction rationale established in the Introduction.

3.4 Classification Class Level Performance and Error Analysis

In Table 4 presents the class-level classification report of the final MobileNetV3Large model on the 278-image test set.

Table 4. Classification Report of the Final MobileNetV3Large Model

Class	Precision	Recall	F1-Score	Support
Bacterial Spot	1.0000	0.8696	0.9302	23
Cercospora Leaf Spot	0.9286	1.0000	0.9630	26
Curl Virus	0.9846	1.0000	0.9922	64
Healthy Leaf	1.0000	1.0000	1.0000	69
Nutrition Deficiency	1.0000	1.0000	1.0000	67
White Spot	1.0000	1.0000	1.0000	29
Accuracy			0.9892	278
Macro Average	0.9855	0.9783	0.9809	278
Weighted Average	0.9898	0.9892	0.9890	278

Three of the six classes Healthy Leaf, Nutrition Deficiency, and White Spot were classified with perfect precision, recall, and F1-score, indicating that MobileNetV3Large learned highly discriminative features for these categories. Curl Virus was classified almost as well, with a near-perfect F1-score of 0.9922 (one false positive out of 65 predictions, recall 1.0000). Bacterial Spot was the most challenging class, with the lowest recall (0.8696) of all six classes, meaning 3 of its 23 true test images were misclassified into another class despite perfect precision (every image the model did label Bacterial Spot was correct). Cercospora Leaf Spot showed the inverse pattern: perfect recall (1.0000) but the lowest precision (0.9286), indicating that all true Cercospora Leaf Spot images were correctly identified, but two images from another class were incorrectly predicted as Cercospora Leaf Spot.

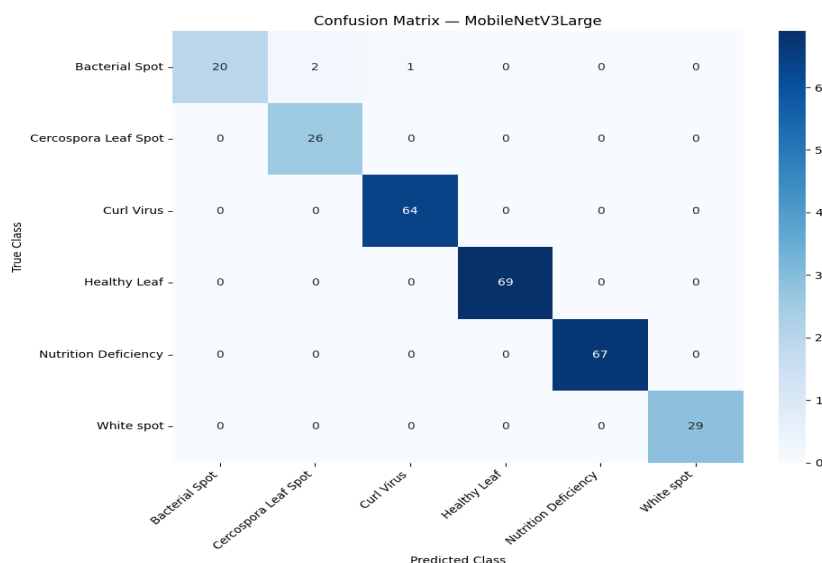


Figure 5. Confusion Matrix of the Final MobileNetV3Large Model on the Independent Test Set

As shown in Figure 2, the confusion matrix confirms that all three misclassified test images originated from a single source class, Bacterial Spot, and are distributed between two destination classes: two images were misclassified

as Cercospora Leaf Spot and one image was misclassified as Curl Virus. No other class pair exhibited any confusion. This is consistent with the precision/recall pattern in Table 4 (Cercospora Leaf Spot's precision loss accounts for exactly two false positives, and Curl Virus's precision loss accounts for exactly one) and indicates that the model's error mode is narrow and specific driven by visual similarity between early-stage Bacterial Spot lesions and the other two disease classes rather than reflecting a general weakness in the model's ability to separate all six classes.

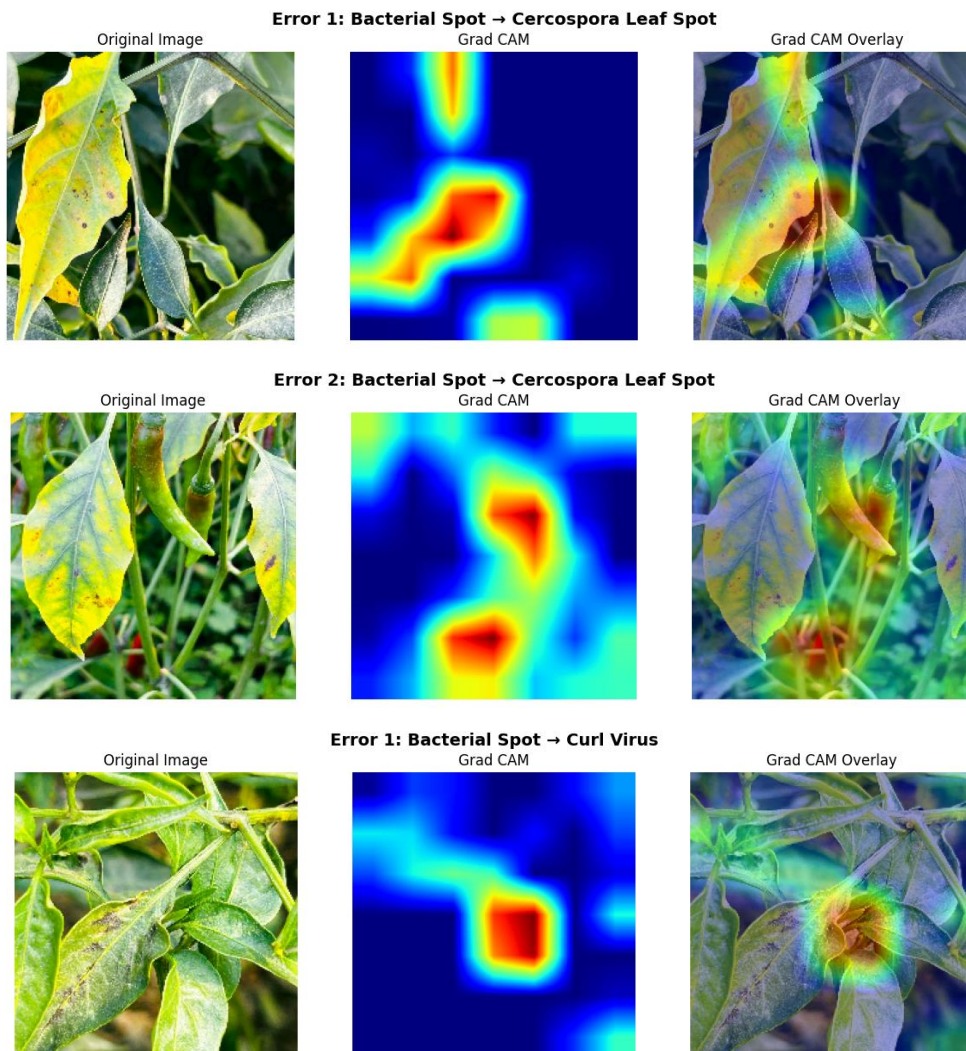


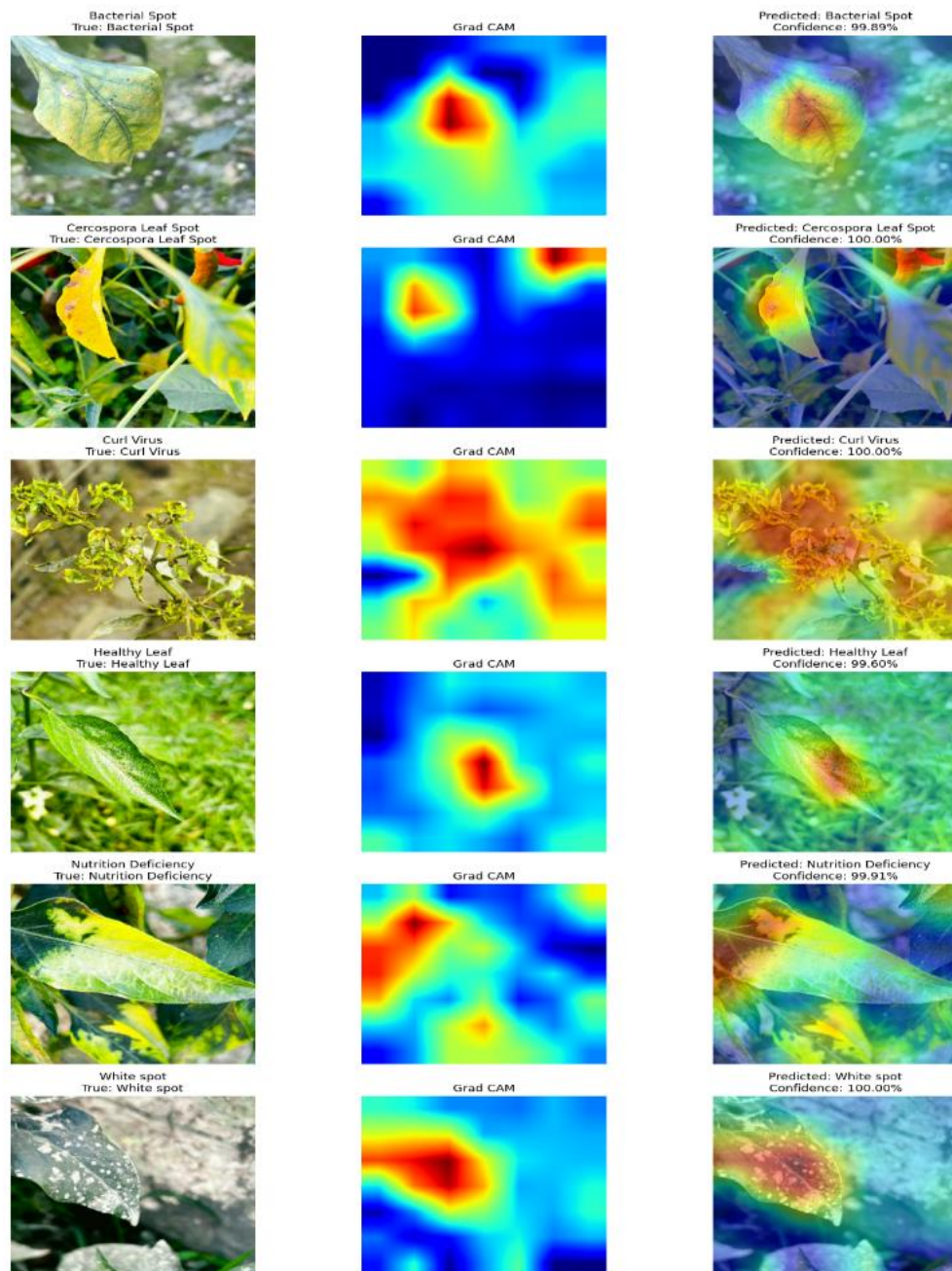
Figure 6. Grad-CAM Visualization of the Three Misclassified Bacterial Spot Test Images: Predicted as Cercospora Leaf Spot, (c) Predicted as Curl Virus

As shown in Figure 3, Grad-CAM was applied to a correctly classified Bacterial Spot sample and to each of the three misclassified samples to examine whether the errors were associated with a shift in the region of the leaf the model attended to. In all three misclassified cases, the Grad-CAM activation continued to localize to the lesion area on the leaf surface rather than shifting to an irrelevant background region, indicating that the activation maps remained concentrated around visually symptomatic regions. This suggests the errors are best explained by visual similarity or ambiguity between early-stage Bacterial Spot lesions and the lesion patterns of Cercospora Leaf Spot and Curl Virus, rather than by the model attending to an uninformative image region, and is consistent with Bacterial Spot retaining perfect precision (1.0000) even though its recall was the lowest of the six classes.

3.5 Computational Model Interpretability Using Grad-CAM

To provide the direct visual evidence requested by the reviewer, Grad-CAM was applied to the final MobileNetV3Large model using the last convolutional layer of the backbone, conv_1, as the target layer. This layer was selected automatically as the deepest Conv2D layer in the trained model graph, following standard Grad-CAM practice of using the final convolutional feature map, which retains the highest-level spatial features before the classification head. One correctly classified test image was selected per class (six classes in total), and the resulting heatmaps were overlaid on the original leaf images to visualize the regions that most strongly influenced each prediction.

Grad CAM Visualization of MobileNetV3Large Across Six Chili Leaf Disease Classes

**Figure 7.** Grad-CAM Visualization for One Correctly Classified Test Image per Class (Six Classes)

As shown in Figure 7, across all six classes, the Grad-CAM activation is concentrated on the diseased or symptomatic region of the leaf, lesions, discoloration, or curling patterns — rather than on the image background, suggesting that the model's predictions were primarily influenced by visually relevant symptomatic regions rather than obvious background artifacts. This provides direct, figure-based support for the model's high overall accuracy (98.92%) and complements the error-specific Grad-CAM evidence presented for the Bacterial Spot misclassifications in Section 3.4, together addressing the reviewer's earlier concern that architecture-performance explanations in this study were not previously supported by activation-level visualization.

3.6 Discussion

The findings of this study both confirm and diverge from prior chili leaf disease classification research in ways that help clarify the research gap motivating this work. Khoirunnisa et al. [1] compared MobileNetV2 and VGG16 on a chili leaf disease dataset and reported that MobileNetV2 outperformed VGG16. In the present study, however, VGG16 (93.88%) obtained a substantially higher observed mean accuracy than MobileNetV2 (75.90%) under the frozen feature-extraction protocol. This reversal is unlikely to reflect a general superiority of one architecture over the other; rather, it illustrates that architecture ranking is highly sensitive to the specific dataset, preprocessing pipeline, and

transfer-learning strategy used, a dependency that single-comparison studies such as [1] cannot reveal because they evaluate only two architectures under one fixed configuration. A similar reversal is observed relative to Ristanti [12], who reported MobileNetV2 (92%) substantially outperforming VGG16 (74%) on a combined field-and-public dataset; again, the opposite ranking emerged here. Taken together, these contrasting results across three independent studies, including the present one, demonstrate the exact research gap identified in the Introduction: prior work compares too few architectures under too narrow a set of conditions to determine whether a reported ranking reflects a genuine architectural property or an artifact of the specific dataset and protocol used. By benchmarking eight architectures under one uniform, multi-seed protocol with statistical validation, this study is able to show that such rankings are not stable across studies and should therefore be interpreted with caution unless obtained under a controlled, like-for-like comparison.

The comparison with Setyadi et al. [4] further clarifies the role of data augmentation in previously reported high-accuracy results achieved 97.18% accuracy using transfer learning combined with data augmentation, but this result depends on synthetic training samples and therefore does not directly indicate how a model would perform on the original data distribution without offline augmentation. The final model in this study, MobileNetV3Large, achieved a comparable and in fact slightly higher test accuracy (98.92%), relying on class weighting and frozen feature extraction rather than synthetic oversampling to address class imbalance; on-the-fly augmentation was still applied to the training subset, but only as a regularization strategy and not to artificially balance class counts. This suggests that, at least for the chili leaf disease task and dataset scale considered here, careful architecture selection combined with class weighting provides an alternative strategy for addressing class imbalance without altering the original class distribution through offline synthetic oversampling, while preserving a more faithful representation of the original, field-realistic class distribution, addressing the second research gap noted in the Introduction, namely the field's reliance on synthetic data expansion rather than direct class-imbalance handling. However, because this comparison spans different studies, datasets, and preprocessing pipelines, it does not by itself establish that this combination is generally superior to augmentation-based approaches.

The generalization concern raised by Khasanah and Fachrie [6], who reported a drop from 90% training accuracy to 84% test accuracy using DenseNet201, is echoed by the DenseNet121 results obtained here. Although the architectures and exact configurations differ, DenseNet121 was among the weaker performers in this study (88.97% mean test accuracy) and additionally showed the largest run-to-run variability of all eight architectures ($SD = 2.70$), indicating unstable generalization across different random initializations under the frozen feature-extraction protocol. This convergence of evidence across two independent studies, using different DenseNet variants and different protocols, strengthens the inference that dense-connectivity architectures may benefit from further investigation using either a larger training set or a less restrictive transfer-learning strategy (e.g., partial fine-tuning) to reach their expected performance on small, imbalanced agricultural datasets, an architecture-level generalization pattern that prior single-architecture studies were not positioned to identify.

Overall, this comparison against four prior studies [1], [5], [9], [12] indicates that the research gap motivating this work is genuine rather than incremental: reported architecture rankings for chili leaf disease classification have not been consistent across studies, prior high-accuracy results have often depended on data augmentation rather than directly resolved class imbalance, and generalization weaknesses in specific architecture families have been observed only in isolation rather than confirmed across independent studies. By evaluating eight architectures under one uniform, multi-seed, statistically validated protocol, this study provides evidence that helps reconcile these inconsistencies, while also making clear that a single dataset and a maximum of three seeds per architecture remain insufficient to resolve the question definitively; consistent with the limitations acknowledged in Section 4, broader confirmation would require replicating this protocol on independently collected chili leaf datasets.

4. CONCLUSION

This study compared eight CNN architectures, namely ResNet50, ResNet101, EfficientNetB0, EfficientNetB3, MobileNetV2, MobileNetV3Large, VGG16, and DenseNet121, for six-class chili leaf disease classification using a frozen feature-extraction transfer-learning approach with on-the-fly augmentation applied only to the training subset of an imbalanced dataset containing 1,847 original images, thereby addressing the study's three stated objectives. Across 24 experimental runs spanning three random seeds, MobileNetV3Large achieved the highest mean test accuracy (98.80%) and, when retrained as the final model, reached 98.92% accuracy, a macro-average F1-score of 98.09%, and a weighted-average F1-score of 98.90% on an independent test set, with the Friedman test ($\chi^2 = 19.432$, $p = 0.0069$) and a Kendall's W of 0.9253 providing evidence that the performance differences among architectures were statistically significant, providing evidence of systematic differences in architecture rankings across the three repeated training conditions, although the small number of seeds limits the statistical power of pairwise comparisons. Notably, MobileNetV3Large's strong performance despite its small parameter count (3.63 million) compared with deeper architectures such as ResNet101 (43.85 million) indicates that classification accuracy and model compactness are not necessarily contradictory for this task; this relatively compact model size may be favorable for farmers and agricultural extension workers who require deployment on resource-constrained mobile or field devices, although this study did not directly benchmark inference latency, memory usage, or on-device throughput, so a computational-

efficiency claim in the strict sense is not established here. Grad-CAM visualization indicated that the final model's predictions were generally focused on symptomatic leaf regions, and error analysis traced all three test-set misclassifications to visually ambiguous, early-stage Bacterial Spot lesions rather than to an irrelevant or unstable decision process. Nevertheless, these findings should be interpreted with several limitations in mind: the eight evaluated architectures were intentionally heterogeneous in depth and parameter count rather than matched for a controlled ablation study, so the results represent a practical benchmarking comparison rather than isolated evidence of any single architectural factor; the post-hoc pairwise comparisons did not reach significance after Holm correction, a known consequence of the limited statistical power available with only three seeds; and although the final model showed no obvious overfitting pattern between training and validation performance, the possibility of some degree of overfitting to the specific visual conditions of this single-source dataset cannot be fully excluded and would benefit from external validation on field-collected images from different locations, seasons, and camera conditions. Future work should therefore prioritize increasing the number of repeated training runs to strengthen post-hoc statistical comparisons, validating the selected model on independent, externally collected chili leaf datasets, and extending the frozen feature-extraction protocol to additional crop diseases to assess its broader applicability for resource-constrained agricultural deployment. While the eight evaluated architectures were not matched for depth or parameter count and the statistical comparison was constrained by only three random seeds, the central contribution of this study is a reusable, statistically grounded benchmarking protocol, rather than a single fixed-architecture prescription, that future work can extend with additional seeds, external datasets, and controlled architectural ablation to build increasingly reliable guidance for real-world chili leaf disease diagnosis.

5. DECLARATIONS

5.1 Author Contributions

Conceptualization: M.F.A., and I.S.W.;

Methodology: M.F.A., and I.S.W.;

Software: M.F.A.;

Validation: M.F.A.;

Formal Analysis: M.F.A.;

Investigation: M.F.A.;

Resources: M.F.A.;

Data Curation: M.F.A.;

Writing Original Draft Preparation: M.F.A.;

Writing Review and Editing: M.F.A.;

Visualization: M.F.A.;

All authors have read and agreed to the published version of the manuscript.

5.2 Data Availability

The data presented in this study are available on request from the corresponding author.

5.3 Funding

This research and the publication of this manuscript were fully funded by the author's personal fund. No external funding was received for the research, preparation, or publication of the manuscript.

5.4 Institutional Review Board Statement

Not applicable.

5.5 Informed Consent Statement

Not applicable.

5.6 Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

5.7 Generative AI and AI-assisted Technologies in The Writing Process

The authors used Generative AI to improve the writing clarity of this paper. They reviewed and edited the AI-assisted content and took full responsibility for the final publication.

REFERENCES

- [1] I. I. Khoirunnisa, A. Fadlil, and H. Yuliansyah, "Perbandingan Kinerja MobileNetV2 dan VGG16 dalam Klasifikasi Penyakit

- pada Citra Daun Tanaman Cabai,” *JISKA (Jurnal Inform. Sunan Kalijaga)*, vol. 11, no. 1, pp. 70–82, 2026, doi: 10.14421/jiska.5075.
- [2] Y. Mulyana Prayoga, D. Wahyuni, and Elisawati, “Deteksi Penyakit Tanaman Cabai Berbasis Deep Learning Menggunakan Convolutional Neural Network Secara Realtime,” *J. Unitek*, vol. 18, no. 2, pp. 292–302, 2025, doi: <https://doi.org/10.52072/unitek.v18i2.1467>.
 - [3] M. Misbahul Munir, P. Kasih, and A. Sanjaya, “Penerapan Arsitektur Mobilenet Dalam Cnn Pada Klasifikasi Penyakit Daun Cabai,” *Inotek*, vol. 8, pp. 2549–7952, 2024, doi: <https://doi.org/10.29407/5m496763>.
 - [4] R. A. Setyadi, S. Rahman, D. Manurung, M. Hasanah, and A. Indrawati, “Implementasi Transfer Learning Untuk Klasifikasi Pada Daun Cabai Menggunakan CNN,” *Djtechno J. Teknol. Inf.*, vol. 5, no. 2, pp. 304–315, Aug. 2024, doi: 10.46576/djtechno.v5i2.4642.
 - [5] M. Gugus Azhari, M. Ajid Husain, and M. Lutfi, “Transfer Learning Convolutional Neural Network dengan Arsitektur EfficientNet Untuk Identifikasi Penyakit Daun Cabai,” *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 7, no. 5, pp. 3662–3666, Jan. 2023, doi: 10.36040/jati.v7i5.7791.
 - [6] N. U. Khasanah and M. Fachrie, “Klasifikasi Jenis Penyakit Tanaman Cabai Menggunakan Arsitektur DenseNet201,” *SINTECH (Science Inf. Technol. J.)*, vol. 7, no. 3, pp. 148–158, Dec. 2024, doi: 10.31598/sintechjournal.v7i3.1708.
 - [7] C. C. Aggarwal, *Artificial Intelligence*. Cham: Springer International Publishing, 2021. doi: 10.1007/978-3-030-72357-6.
 - [8] C. C. Aggarwal, *Neural Networks and Deep Learning*. Cham: Springer International Publishing, 2023. doi: 10.1007/978-3-031-29642-0.
 - [9] M. Resa Arif Yudianto, P. Sukmasetya, R. Abul Hasani, and D. Sasongko, “Pengaruh Data Preprocessing terhadap Imbalanced Dataset Pada Klasifikasi Citra Sampah menggunakan Algoritma Convolutional Neural Network,” *Build. Informatics, Technol. Sci.*, vol. 4, no. 3, pp. 1367–1375, Dec. 2022, doi: 10.47065/bits.v4i3.2575.
 - [10] Mariana Purba, “Klasifikasi Jenis Daun Tanaman Tropis Menggunakan Model ResNet50 Berbasis Transfer Learning dengan Dataset Tropical Plant Leaf,” *JCOSIS (Journal Comput. Sci. Inf. Syst.)*, vol. 2, no. 2, pp. 51–57, Oct. 2025, doi: 10.61567/jcosis.v2i2.255.
 - [11] A. Kumar, A. Raghunathan, R. Jones, T. Ma, and P. Liang, “Fine-Tuning Can Distort Pretrained Features and Underperform Out-of-Distribution,” *ICLR 2022 - 10th Int. Conf. Learn. Represent.*, vol. 10, no. Id, pp. 1–54, 2022, doi: <https://doi.org/10.48550/arXiv.2202.10054>.
 - [12] E. L. P. Ristanti, “Analisis dan Perbandingan Arsitektur VGG16 dan MobileNetV2 Untuk Klasifikasi Penyakit Daun Pada Tanaman Cabai Menggunakan CNN,” vol. 2, pp. 216–226, 2024, doi: 10.572349/scientica.v2i9.2381.
 - [13] E. Anggiratih, S. Siswanti, S. K. Octaviani, and A. Sari, “Klasifikasi Penyakit Tanaman Padi Menggunakan Model Deep Learning Efficientnet B3 dengan Transfer Learning,” *J. Ilm. SINUS*, vol. 19, no. 1, p. 75, 2021, doi: 10.30646/sinus.v19i1.526.
 - [14] Ricky Putra Sardika and W. Widhiarso, “Klasifikasi Otomatis Tingkat Kerusakan Retak Bangunan pada Citra Digital Menggunakan MobileNetV2 dan Augmentasi Data,” *Arcitech J. Comput. Sci. Artif. Intell.*, vol. 5, no. 1, pp. 108–124, 2025, doi: 10.29240/arcitech.v5i1.13938.
 - [15] Q. A. Fitroh, M. S. A. K. Mardlian, I. Kurniawan, and M. K. Namariq, “Evaluasi Strategi Penanganan Ketidakseimbangan Kelas pada Klasifikasi Sentimen Multikelas Ulasan Aplikasi ChatGPT,” *JUSIFOR J. Sist. Inf. dan Inform.*, vol. 5, no. 1, pp. 287–294, Jun. 2026, doi: 10.70609/jusifor.v5i1.9714.
 - [16] M. I. Fathur Rozi, N. O. Adiwijaya, and D. I. Swasono, “Identifikasi Kinerja Arsitektur Transfer Learning Vgg16, Resnet-50, Dan Inception-V3 Dalam Pengklasifikasian Citra Penyakit Daun Tomat,” *J. Ris. Rekayasa Elektro*, vol. 5, no. 2, p. 145, 2023, doi: 10.30595/jrre.v5i2.18050.
 - [17] U. Oktaviana Nur, R. Hendra, A. Dwi Khoirul Annas, and G. Wasis Wicaksono, “Klasifikasi Penyakit Padi berdasarkan Citra Daun Menggunakan Model Terlatih Resnet101,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 6, pp. 1216–1222, 2021, [Online]. Available: <https://eprints.umm.ac.id/id/eprint/4780>
 - [18] N. Awalia, “Identifikasi Penyakit Leaf Mold Pada Daun Tomat Menggunakan Model Densenet121 Berbasis Transfer Learning,” *J. Ilm. Ilmu Komput.*, vol. 8, no. 1, pp. 49–52, 2022, doi: 10.35329/jiik.v8i1.212.
 - [19] B. S. Acarya, A. Muhaimin, and K. M. Hindrayani, “Identifikasi Penyakit Daun Jeruk Siam Menggunakan Convolutional Neural Network (CNN) dengan Arsitektur EfficientNet,” *G-Tech J. Teknol. Terap.*, vol. 8, no. 2, pp. 1040–1048, Apr. 2024, doi: 10.33379/gtech.v8i2.4120.
 - [20] W. Amurrohman and N. Nafi’iyah, “Perbandingan ResNet50, MobileNetV2, dan MobileNet V3-Large dalam Klasifikasi Motif Batik Indonesia Berbasis Transfer Learning,” *Rabit J. Teknol. dan Sist. Inf. Univrab*, vol. 11, no. 1, pp. 1457–1467, Jan. 2026, doi: 10.36341/rabit.v11i1.7372.
 - [21] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization,” *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 336–359, 2020, doi: 10.1007/s11263-019-01228-7.
 - [22] S. Rahman, A. Sembiring, M. K. Zuhanda, and B. A. Pratama, “Analisis Komparatif CNN Ringan untuk Klasifikasi Penyakit Daun Tomat Menggunakan Visualisasi Grad-CAM,” *Explor. J. Comput. Sci. Inf. Technol. E-ISSN*, vol. 6, no. 1, pp. 73–86, 2026, doi: <https://doi.org/10.47065/explorer.v6i1.2601>.