
Comparative Performance Analysis of MobileNetV2, ResNet50V2, and EfficientNetB0 Models Using Transfer Learning for Medicinal Plant Leaf Image Classification

Mohammad Yoga Firnanda¹, Rohman Dijaya¹, Ika Ratna Indra Astutik¹, Yulian Findawati¹

¹ Muhammadiyah University of Sidoarjo, Faculty of Science and Technology, Department of Informatics, Jl. Mojopahit No.666 B, Sidowayah, Celep, Kec. Sidoarjo, Kabupaten Sidoarjo, Jawa Timur 61215, Indonesia

Article History

Submitted : 23-07-2026

Revised : 16-08-2026

Accepted : 24-08-2026

Published : 20-09-2026

DOI: 10.32664/smatika.v16i03.2520

Keyword

Convolutional Neural Network;
EfficientNetB0; Image Classification;
Medicinal Plant Leaf Classification;
MobileNetV2; ResNet50V2; Transfer
Learning

*Corresponding Author:

mohammadyoga259@gmail.com

Abstract

Accurate identification of medicinal plant leaves supports the correct and safe use of herbal remedies, yet manual identification is often unreliable because many species share closely related leaf shapes, textures, and colors. Transfer learning has emerged as a practical strategy for image classification tasks of this kind, since it reuses the feature representations of pre-trained convolutional neural networks (CNNs) and thereby lowers both training time and computational demand. This study aims to compare the performance of three transfer learning backbones—MobileNetV2, ResNet50V2, and EfficientNetB0—on a five-class medicinal plant leaf recognition task covering guava (*Psidium guajava*), basil (*Ocimum basilicum*), aloe vera (*Aloe vera*), betel (*Piper betle*), and pandan (*Pandanus amaryllifolius*), using a working dataset of 1,750 leaf images (350 images per class before deduplication), and to statistically verify whether the performance differences among the three architectures are significant. Data were partitioned 80:10:10 for training, validation, and testing, and each architecture was trained across five independent runs under an identical configuration; Accuracy, Precision, Recall, and F1-score were used to quantify performance, followed by the Shapiro-Wilk and Kruskal-Wallis tests. MobileNetV2 produced the strongest mean Accuracy ($98.27\% \pm 0.52$), while ResNet50V2 led on Recall. The Kruskal-Wallis results, however, showed no statistically significant gap between the three architectures on any metric ($p > 0.05$). In other words, MobileNetV2, ResNet50V2, and EfficientNetB0 can be regarded as functionally equivalent for this classification task, and the practical choice among them should therefore weigh factors such as inference speed, model size, and deployment constraints rather than average predictive performance alone.

1. Introduction

Correctly identifying medicinal plant leaves matters because misidentification can lead to the wrong plant being used for treatment, given that different species carry different therapeutic compounds and, sometimes, different toxicity profiles. The present work concentrates on five such species commonly found in Indonesia guava (*Psidium guajava*), basil (*Ocimum basilicum*), aloe vera (*Aloe vera*), betel (*Piper betle*), and pandan (*Pandanus amaryllifolius*) each of which differs noticeably in leaf outline, surface texture, coloration, and vein

pattern. That visual diversity is precisely what makes the group a suitable testbed for image classification research and a reasonable proxy for evaluating deep learning models intended for automated herbal-leaf recognition [1].

Relying on human judgment alone to tell these species apart is problematic, since several of them look deceptively alike in shape, hue, and venation, and correct identification tends to depend on the identifier's level of botanical training [2]. Convolutional Neural Networks (CNNs) have become the dominant tool for pulling discriminative visual features out of raw images automatically, and they consistently outperform earlier hand-crafted approaches on classification benchmarks [3]. Transfer learning adds a further practical advantage on top of that: instead of training a CNN from a random initialization, a network already trained on a large source dataset can be repurposed for a new, smaller target dataset, cutting down both the data and the compute required to reach strong accuracy [4]. This benefit is not confined to natural-image tasks: a broad review of 102 medical-imaging studies likewise found transfer learning to be a dependable route to strong performance even when labeled training examples were scarce, regardless of which specific backbone or fine-tuning strategy was chosen [5]. At the same time, no single CNN architecture works best across every problem; differences in depth, receptive field, and parameter count mean that one backbone may outperform another depending on the dataset at hand [6]. Picking the right architecture is therefore not a trivial decision and deserves empirical justification for each new application domain [7].

Several existing studies have already tried to make that kind of empirical comparison for plant-related image tasks. One study, for instance, pitted EfficientNetB7 against MobileNetV2 for herbal plant species recognition, while another benchmarked EfficientNetB0, MobileNetV2, and ResNet50 on a traffic-density classification problem using transfer learning; in both cases, the ranking among architectures shifted with the dataset, reinforcing the idea that architecture choice is dataset-dependent rather than universal [8], [9]. More recent, larger-scale benchmarking work points in the same direction: One such study trained 23 CNN architectures across 18 openly available plant-disease datasets and found no architecture that dominated across the board, while another compared MobileNetV2, EfficientNetB0, ResNet50, DenseNet121, and VGG16 specifically for *Jatropha* leaf disease classification and likewise reported that the best-performing model changed depending on the disease category being evaluated [10],[11]. None of this prior work, however, targeted medicinal plant leaves directly, and their conclusions cannot be transferred wholesale to the present problem because the leaf textures, colors, and shapes involved differ substantially from those in disease-detection or general herbal-plant datasets. A dedicated comparative study on medicinal plant leaf classification therefore remains warranted.

To the best of our knowledge, no earlier study has directly compared EfficientNetB0, MobileNetV2, and ResNet50V2 under transfer learning on a five-class medicinal plant leaf dataset of the kind used here. This study fills that gap by running a controlled comparative experiment across the three architectures, assessing them with standard classification metrics and following up with statistical testing to determine whether any observed performance gaps are meaningful or merely noise. Unlike most prior comparative studies in this area, which typically report results from a single training run, the present study explicitly repeats each architecture's training five times and subjects the resulting metrics to formal normality and significance testing (Shapiro–Wilk and Kruskal–Wallis) this repeated-run statistical validation, rather than the architecture comparison alone, constitutes the primary methodological contribution of this work. The intended contribution is twofold: to point toward the architecture best suited to medicinal plant leaf recognition under these conditions, and to give future researchers in this specific sub-area a reference point grounded in repeated, statistically validated experiments.

2. Research Method

2.1 Research Workflow

An experimental design was adopted to compare the three transfer learning models under matched conditions the same dataset, the same preprocessing pipeline, the same training configuration, and the same evaluation protocol were applied to each architecture so that any performance gap could be attributed to the network itself rather than to inconsistencies in the experimental setup.

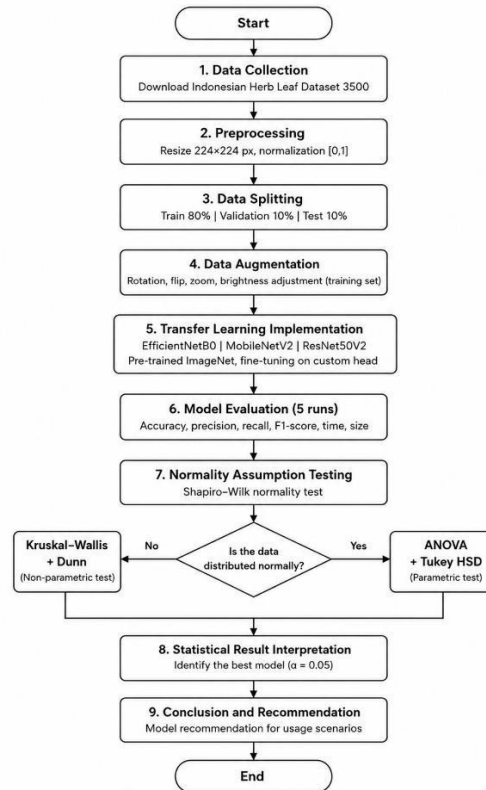


Figure 1. Research Workflow of the Proposed Study

Figure 1 illustrates the nine stages this study went through. Stage 1 is data collection, namely downloading the Indonesian Herb Leaf Dataset 3500. Stage 2 is preprocessing, namely resizing all images to 224×224 pixels and normalizing pixel values to the [0,1] range. Stage 3 is data splitting, with a ratio of 80% training data, 10% validation data, and 10% testing data. Stage 4 is data augmentation rotation, flipping, zoom, and brightness variation applied only to the training data. Stage 5 is the implementation of transfer learning using three ImageNet pre-trained architectures EfficientNetB0, MobileNetV2, and ResNet50V2 with the backbone frozen and training carried out only on the classification head added on top of it. Stage 6 is model evaluation, carried out across five repetitions (5 runs) for each architecture using the Accuracy, Precision, Recall, and F1-score metrics. Stage 7 is testing the normality assumption using the Shapiro-Wilk test on the results of the five repetitions; when the data is normally distributed, testing continues with ANOVA followed by the Tukey HSD post-hoc test (parametric test), whereas when the data is not normally distributed, the Kruskal-Wallis test and Dunn's post-hoc test (non-parametric test) are used instead. Stage 8 is the interpretation of the statistical test results to identify the best-performing model at a significance level of $\alpha = 0.05$. Stage 9 is drawing conclusions and formulating model recommendations according to the intended usage scenario. Each model was trained and evaluated only after the shared dataset and preprocessing pipeline had been finalized, and the results feeding into the statistical tests were drawn from that same evaluation stage.

2.2 Dataset and Image Preprocessing

The images came from the Indonesian Herb Leaf Dataset 3500, an openly available collection on Mendeley Data comprising 3,500 leaf photographs spanning ten Indonesian medicinal plant species (350 images per class). Only five of those ten classes were retained for this study guava (*Psidium guajava*), basil (*Ocimum basilicum*), aloe vera (*Aloe vera*), betel (*Piper betle*), and pandan (*Pandanus amaryllifolius*) giving a working set of 1,750 images [12].

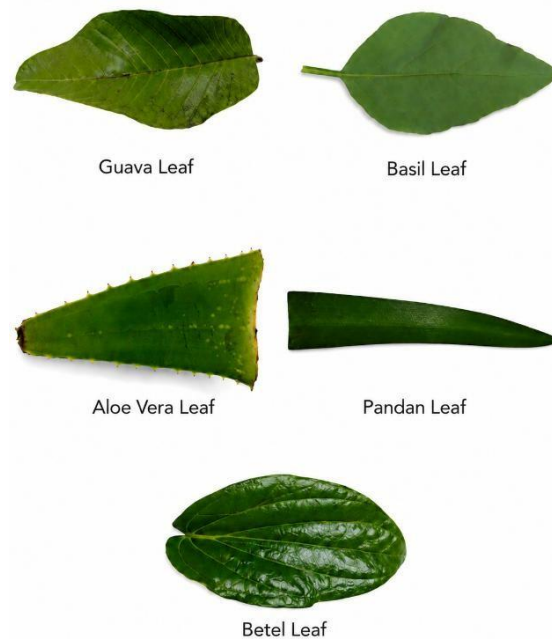


Figure 2. Sample Images of the Five Medicinal Plant Leaf Classes

Because visually near-duplicate images such as multiple photographs of the same leaf taken in quick succession can leak between the training, validation, and test partitions and thereby inflate reported test performance, an explicit deduplication step was carried out prior to splitting. Perceptual hashing (pHash) was computed for every image, and images within each class whose hash distance fell below a similarity threshold were grouped into duplicate clusters. The dataset was then partitioned using a group-aware stratified procedure (StratifiedGroupKFold) that kept every cluster entirely within a single partition, so that near-duplicate images could never appear in more than one of the training, validation, or test sets. The absence of leakage was verified programmatically: after splitting, the cluster identifiers of the three partitions were checked for pairwise overlap, and the workflow asserts that this overlap equals zero before proceeding.

Every image went through the same preprocessing pipeline prior to training so that all three architectures received identical inputs. Images were resized to 224×224 pixels to match the expected input dimensions of the chosen networks, and pixel intensities were rescaled to a 0–1 range using a $1/255$ scaling factor. The dataset was then split 80:10:10 into training, validation, and testing subsets, with class proportions preserved across each partition. Only the training subset was subjected to data augmentation horizontal flips, random rotation, zoom, and brightness jitter a set of transformations broadly consistent with common practice in the image data augmentation literature for mitigating overfitting and improving generalization under limited data [13].

2.3 Transfer Learning Implementation

Three pre-trained CNN backbones MobileNetV2, ResNet50V2, and EfficientNetB0 were used to perform the classification task [14]. These three were deliberately chosen because they sit at different points along the design trade-off between depth, feature-extraction capacity, and computational cost. MobileNetV2 relies on depthwise separable convolutions combined with inverted residual blocks to stay lightweight and efficient [15]. ResNet50V2 instead leans on identity shortcut connections to enable deeper training while keeping the vanishing-gradient problem in check [16]. EfficientNetB0, meanwhile, uses compound scaling to jointly balance depth, width, and input resolution, aiming for strong accuracy without a heavy computational footprint [17].

Table 1. Experimental parameter configuration used in this study

Parameter	Value
Input Image Size	224 × 224 pixels
Batch Size	32
Learning Rate	0.001
Number of Epochs	15
Early Stopping Patience	5 Epochs
Optimizer	Adam
Loss Function	Categorical Crossentropy
Transfer Learning Strategy	Frozen Feature Extractor
Output Classes	5

All of the parameters in Table 1 were chosen with particular attention to the relatively small size of the working dataset 1,750 images across five classes (an average of 350 images per class, with only about 1,400 images available for training after the 80:10:10 split). Under such data-limited conditions, the main risk is not insufficient model capacity but overfitting, so nearly every setting in Table 1 was directed at containing that risk. The 224×224 input size was kept to match the native dimensions of the ImageNet pre-trained weights across all three architectures, so that the transferred features could be exploited to their fullest without having to retrain the early convolutional layers from scratch an approach that matters greatly when training data is limited. The Frozen Feature Extractor strategy was the most decisive choice: by freezing the entire backbone, only the classification head (Dense 256 and Dense 5, with a total number of trainable parameters far smaller than the millions of parameters in the backbone) was updated during training, keeping the ratio between the amount of training data and the number of optimized parameters within a reasonable range so as to avoid the model memorizing the training data. A batch size of 32 was chosen so that each epoch produced a sufficient number of weight-update iterations (around 44 iterations per epoch on ±1,400 training images) for stable gradient estimation, without straining computational memory requirements. A learning rate of 0.001 is the Adam optimizer's default value, retained because only the small head was being trained, so a very low learning rate was not required to reach stable convergence; this combination of the Adam optimizer with a 0.001 learning rate is also consistent with earlier CNN-based image classification research by Sahera et al., who found that this combination produced the highest accuracy (98.50%) for a food-object image classification task, making this choice not merely a default value but a setting already proven effective in a similar image classification domain [18]. The number of epochs was capped at 15 with an early stopping patience of 5, not to speed up computation but as the primary overfitting-prevention mechanism, since on a dataset this small the classification head tends to reach its minimum validation loss within the first few epochs, and further training would only increase the risk of the model overfitting to the training data. In other words, the combination of a frozen backbone, a capped epoch budget, and early stopping forms a single, coherent regularization strategy consistently tailored to the characteristics of this study's small dataset.

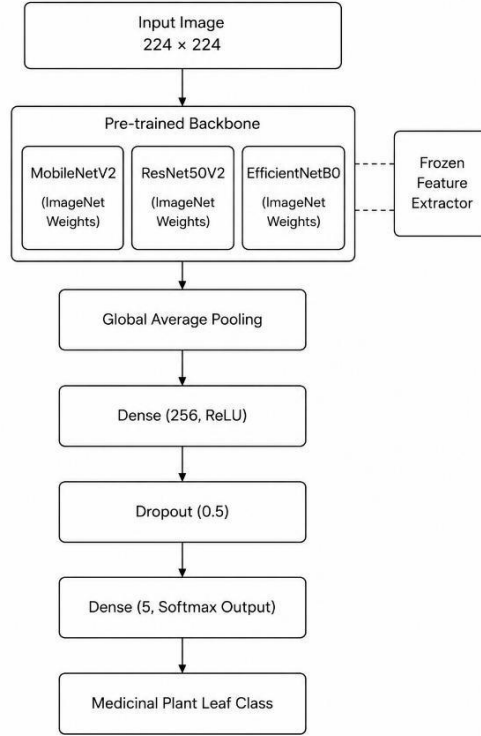


Figure 3. Proposed Transfer Learning Architecture

An early stopping rule was used to keep training efficient and avoid overfitting: training was halted once validation loss failed to improve for five consecutive epochs, and the weights corresponding to the lowest validation loss were restored for the final evaluation.

Each backbone was initialized with ImageNet weights and kept entirely frozen, so that it functioned purely as a fixed feature extractor an approach that remains common in transfer learning practice, particularly when the target dataset is comparatively small [5]. On top of each frozen backbone, a shared classification head was attached, consisting of Global Average Pooling, a Dense layer (256 units, ReLU), Dropout (0.5), and a final Dense layer (5 units, Softmax) to produce the five-class prediction. The complete architecture is shown in Figure 3.

2.4 Performance Evaluation

Four standard classification metrics Accuracy, Precision, Recall, and F1-score were used to judge model quality, since together they capture both overall correctness and how well each model handled individual medicinal plant classes [19].

Accuracy represents the proportion of correctly classified samples to the total number of testing samples and is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

where TP denotes True Positive, TN denotes True Negative, FP denotes False Positive, and FN denotes False Negative.

Precision measures the proportion of correctly predicted positive samples among all samples predicted as positive and is expressed as:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Recall indicates the proportion of correctly predicted positive samples among all actual positive samples and is formulated as:

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

The F1-score represents the harmonic mean of Precision and Recall, providing a balanced evaluation of the classification performance, particularly when class distributions are imbalanced. The F1-score is calculated as:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

As a concrete illustration, the four formulas above are computed on a per-class (one-vs-rest) basis for Accuracy, Precision, Recall, and F1-score one class is treated as "positive" while the remaining four classes are combined as "negative" after which the values for the five classes are macro-averaged for each run, and the average across the five runs is what is reported in Table 3. As an example, consider MobileNetV2's classification result for the Guava class, based on the average confusion matrix across five repetitions in Figure 6, with a total of 162 images in the test set:

Table 2. Worked Example of the Evaluation Metrics

Component	Value	Meaning
TP (True Positive)	28.6	Guava images correctly predicted as Guava
FN (False Negative)	1.4	Guava images incorrectly predicted as Basil
FP (False Positive)	1.0	Betel images incorrectly predicted as Guava
TN (True Negative)	131.0	Non-Guava images correctly predicted as not Guava

Substituting these four values into Equations (1)–(4) gives:

$$Accuracy = \frac{(28.6 + 131.0)}{162} = \frac{159.6}{162} = 0.9852 \rightarrow 98.52\% \quad (5)$$

$$Precision = \frac{28.6}{(28.6 + 1.0)} = \frac{28.6}{29.6} = 0.9662 \rightarrow 96.62\% \quad (6)$$

$$Recall = \frac{28.6}{(28.6 + 1.4)} = \frac{28.6}{30.0} = 0.9533 \rightarrow 95.33\% \quad (7)$$

$$F1 - Score = \frac{(2 \times 28.6)}{(2 \times 28.6 + 1.0 + 1.4)} = \frac{57.2}{59.6} = 0.9597 \rightarrow 95.97\% \quad (8)$$

This calculation shows directly how classification errors in the confusion matrix pull down Precision (due to the FP contributed by Betel images being confused with Guava) and Recall (due to the FN contributed by Guava images being predicted as Basil). Per-class values such as the Guava example above are then averaged together with the other four classes (Aloe Vera, Basil, Betel, Pandan) to produce a single value per metric for each run, and the average across the five runs is what is reported as MobileNetV2's final result in Table 3 (Accuracy 98.27% ± 0.52). The identical calculation procedure was applied to ResNet50V2 and EfficientNetB0 using their respective confusion matrices.

2.5 Statistical Analysis

Because a single training run can be misleading due to random initialization and data shuffling, each architecture was trained and evaluated five separate times, and the resulting accuracy values were then subjected to formal hypothesis testing. Importantly, the training, validation, and test partitions described in the Dataset and Image Preprocessing subsection were fixed once using the leakage-aware, group-stratified procedure described above and reused identically across all five runs; what varied between runs was the random seed governing the classification head's weight initialization and the order in which the training data was shuffled and augmented (seeds 42, 123, 2024, 777, and 555 for runs 1–5, respectively), so that the resulting variability reflects stochasticity in training rather than differences in the data seen by the model.

The Shapiro–Wilk test was run first to check whether the five-run results for each architecture followed a normal distribution [20]. Where normality held ($p > 0.05$), a One-Way ANOVA was used to test for significant differences among the three architectures' mean performance [21], with the Tukey Honestly Significant Difference (Tukey HSD) test applied afterward for pairwise comparisons whenever ANOVA flagged a significant effect ($p < 0.05$) [22].

When the normality assumption did not hold ($p \leq 0.05$), the non-parametric Kruskal Wallis test was used in place of ANOVA. Any significant result from that test would have been followed by Dunn's post-hoc test with Bonferroni correction for pairwise comparison among the three architectures.

3. Result and Discussions

3.1 Overview of Experimental Results

The three architectures MobileNetV2, ResNet50V2, and EfficientNetB0 were each run five separate times under identical conditions, for a combined total of fifteen training-and-evaluation trials ($3 \text{ architectures} \times 5 \text{ runs}$). Keeping the dataset split, preprocessing steps, transfer learning strategy, optimizer, learning rate, batch size, and epoch budget fixed across all trials meant that any performance gap observed between architectures could reasonably be attributed to the architecture itself rather than to inconsistent experimental conditions.

Accuracy, Precision, Recall, and F1-score were computed for every run, and the mean and standard deviation across the five repetitions were used to summarize both the typical performance level and its run-to-run stability for each architecture. These summary statistics appear in Table 3.

Table 3. Performance Comparison of Transfer Learning Models (Mean $\pm$ Standard Deviation)

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
MLobilleNetV2	98.27 $\pm$ 0.52	98.05 $\pm$ 0.58	98.25 $\pm$ 0.55	98.12 $\pm$ 0.57
ResNet50V2	98.02 $\pm$ 1.01	97.41 $\pm$ 1.16	98.66 $\pm$ 0.79	97.93 $\pm$ 1.05
EffilcilentNetB0	97.65 $\pm$ 0.28	96.96 $\pm$ 0.30	98.42 $\pm$ 0.28	97.56 $\pm$ 0.29

All three architectures cleared 96% on every metric on average, which suggests that any of them would be a workable choice for this classification task under the tested conditions. The gaps between them, while present, are modest the next subsection breaks those differences down metric by metric.

3.2 Performance Comparison Across Architectures

Table 3 and Figure 4 together summarize the five-run averages for each architecture across all four metrics. Every architecture cleared 97% average accuracy, and although the spread between them is not large, each still displays a somewhat distinct performance profile.

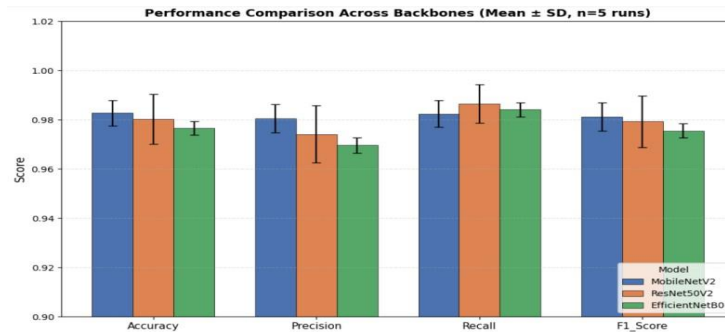


Figure 4. Average classification performance comparison of MobileNetV2, ResNet50V2, and EfficientNetB0 across Accuracy, Precision, Recall, and F1-score

MobileNetV2 came out ahead on Accuracy (98.27%), Precision (98.05%), and F1-score (98.12%), suggesting it struck the best overall balance between correct predictions and error minimization among the three architectures tested. ResNet50V2, by contrast, edged out the others on Recall (98.66%), meaning it was somewhat better at catching true positives across the five plant classes. EfficientNetB0 trailed slightly on most metrics, though the margin separating it from the other two remained small throughout.

Beyond the average scores, run-to-run consistency was also examined by looking at how each architecture's results were distributed across its five repetitions, as shown in Figure 5.

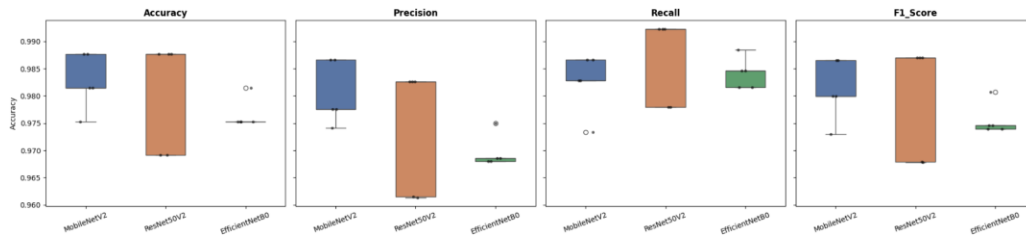


Figure 5. Distribution of classification performance across five independent runs for MobileNetV2, ResNet50V2, and EfficientNetB0 using boxplot visualization

EfficientNetB0's results were the tightest of the three, indicating the most stable behavior across repeated training runs. ResNet50V2 showed noticeably wider spread, especially for Precision and F1-score, whereas MobileNetV2 combined strong average performance with comparatively low variability. Taken together, these patterns show that while overall performance across the three architectures was similar, each still carries a distinct signature in terms of predictive strength versus run-to-run consistency.

3.3 Confusion Matrix Analysis

To get a class-level view of performance beyond the aggregate metrics above, confusion matrices were examined for all three architectures. Unlike Accuracy, Precision, Recall, and F1-score, which summarize performance across the whole test set, a confusion matrix shows exactly how predicted labels map onto true labels, making it possible to spot which specific plant classes were harder to distinguish.

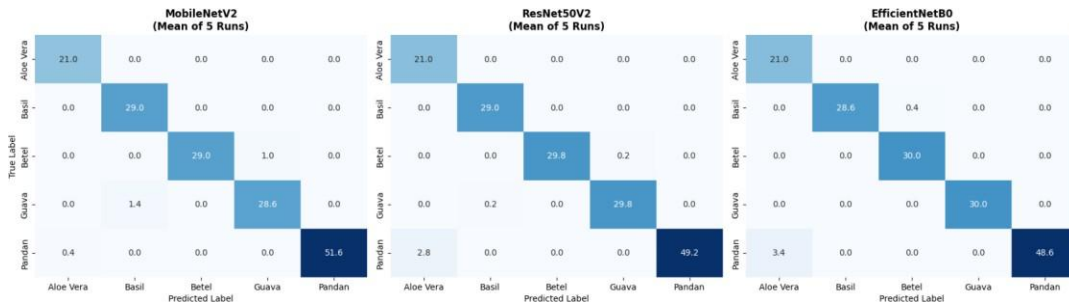


Figure 6. Average confusion matrices of MobileNetV2, ResNet50V2, and EfficientNetB0 obtained from five independent experimental runs

As Figure 6 shows, all three architectures produced confusion matrices dominated by correct predictions along the diagonal, consistent with the strong aggregate metrics reported in Table 2 the large majority of leaf images were assigned to the correct species.

A handful of misclassifications did occur, but they remained a small fraction relative to correct classifications across all three models. The error patterns were also broadly similar between MobileNetV2, ResNet50V2, and EfficientNetB0, hinting that the three networks converged on comparable discriminative features when learning from the medicinal plant leaf data. Overall, the confusion matrix results reinforce the conclusion that all three architectures classified reliably and behaved consistently across the different plant classes.

3.4 Statistical Significance Testing

To move beyond visually comparing averages, formal hypothesis testing was carried out to check whether the observed differences among architectures were statistically meaningful. The Shapiro Wilk test was applied first to check the normality of the five-run results for each metric; since normality was rejected for all of them, the non-parametric Kruskal–Wallis test was used instead of ANOVA to compare MobileNetV2, ResNet50V2, and EfficientNetB0, as it does not assume normally distributed data and is suited to comparing more than two independent groups.

Table 4. Summary of Statistical Significance Testing Results

Evaluation Metric	Shapiro–Wilk Result	Statistical Test	p-value	Decision
Accuracy	Non-normal	Kruskal–Wallis	0.3263	Not Significant
Precision	Non-normal	Kruskal–Wallis	0.1176	Not Significant
Recall	Non-normal	Kruskal–Wallis	0.6728	Not Significant
F1-score	Non-normal	Kruskal–Wallis	0.6083	Not Significant

Every metric in Table 3 came back with a p-value above 0.05, meaning the gaps observed in Accuracy, Precision, Recall, and F1-score among the three architectures are not statistically distinguishable from random variation. So while MobileNetV2 topped Accuracy, Precision, and F1-score, and ResNet50V2 topped Recall, none of those leads were large enough to reject the null hypothesis.

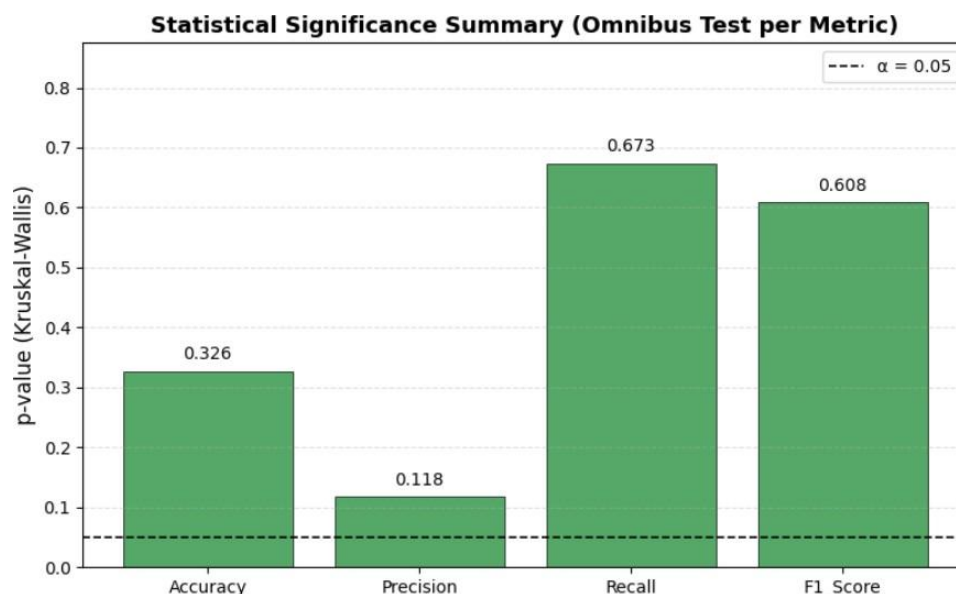


Figure 7. Summary of p-values obtained from the Kruskal–Wallis test for all evaluation metrics

Figure 7 illustrates that every p-value sat above the $\alpha = 0.05$ threshold, so the null hypothesis was retained across the board statistically speaking, MobileNetV2, ResNet50V2, and EfficientNetB0 performed comparably under the experimental conditions used here. Because no significant differences turned up, post-hoc pairwise testing was not carried out.

3.5 Discussion

Taken together, the results show that MobileNetV2, ResNet50V2, and EfficientNetB0 are all fully capable of classifying medicinal plant leaves to a high standard through transfer learning. MobileNetV2 led on Accuracy, Precision, and F1-score, while ResNet50V2 led on Recall differences that hint at distinct model-specific tendencies, yet the statistical tests confirmed those differences do not rise above chance level. In practice, this means the choice of architecture for this task should be guided by considerations beyond raw average metrics, such as deployment constraints and computational budget.

These findings line up with what earlier comparative work in the field has reported. It has been observed that CNN architectures can differ modestly in predictive performance depending on their internal design and feature-extraction capacity [6]. MobileNetV2 has similarly been found to be competitive with the considerably larger EfficientNetB7 on herbal plant classification, underscoring that lightweight networks can hold their own despite having far fewer parameters [7]. A comparison of MobileNetV2, ResNet50, and EfficientNetB0 under transfer learning likewise reported broadly comparable performance across the three, with only minor gaps in the averaged metrics a pattern this study reproduces in a different application domain [8].

What this study adds on top of those earlier efforts is the use of repeated trials paired with formal statistical testing, rather than relying on single-run comparisons. Even though MobileNetV2 posted the highest average Accuracy, Precision, and F1-score, and ResNet50V2 the highest Recall, the Kruskal–Wallis results confirmed that none of these gaps were statistically significant reinforcing that the three architectures are, for practical purposes, comparable classifiers under the conditions tested here.

A methodological point worth flagging for future work in this area is dataset partitioning quality. Studies elsewhere have shown that near-duplicate or otherwise leaked samples slipping between training and test sets can noticeably inflate reported accuracy, sometimes by a wide margin, which is a reminder that careful, leakage-aware data splitting is essential for trustworthy transfer learning comparisons of this kind [23].

The present study also has some acknowledged limitations. It focused on classification performance alone and did not measure computational efficiency training time, inference latency, memory footprint, or model size all of which matter for real-world deployment, particularly on resource-constrained hardware. A further limitation concerns the size of the test set: after the 80:10:10 split, only 162 images an average of roughly 32 images per class were available for testing. This is a modest sample for per-class evaluation and limits the statistical power of the reported metrics; although results were consistent across five independent runs, they should be interpreted with appropriate caution until validated on a larger, independently collected test set. Future work is therefore encouraged to pair classification benchmarking of this kind with an efficiency analysis and a larger held-out test set, so that architecture recommendations for medicinal plant image classification account for both accuracy and practical deployability.

4. Conclusions and Future Works

This study set out to compare three transfer learning architectures MobileNetV2, ResNet50V2, and EfficientNetB0 for classifying medicinal plant leaves across five species. Each architecture was trained and evaluated across five independent runs under matched conditions, with performance assessed through Accuracy, Precision, Recall, and F1-score, followed by Shapiro–Wilk normality testing and Kruskal–Wallis significance testing.

MobileNetV2 achieved the highest average Accuracy ($98.27\% \pm 0.52$), Precision ($98.05\% \pm 0.58$), and F1-score ($98.12\% \pm 0.57$), while ResNet50V2 posted the highest average Recall ($98.66\% \pm 0.79$). EfficientNetB0 also performed competitively, averaging $97.65\% \pm 0.28$ Accuracy. None of these differences, however, reached statistical significance under the Kruskal–Wallis test Accuracy ($p = 0.3263$), Precision ($p = 0.1176$), Recall ($p = 0.6728$), and F1-score ($p = 0.6083$) — so the null hypothesis of comparable performance across architectures could not be rejected.

This points to a practical takeaway: all three architectures are reliable options for medicinal plant leaf classification, and although MobileNetV2 posted the strongest averages across most metrics, the lack of statistically significant separation means architecture selection for similar tasks shouldn't rest on average performance alone. Computational cost, deployment context, and application-specific requirements deserve equal weight in that decision.

Since this study concentrated on predictive performance and left computational efficiency unexamined, that remains a natural direction for follow-up work particularly given how relevant training time, inference latency, memory usage, and model size are to mobile or edge deployment. Beyond that, future studies could explore fine-tuning strategies that go past frozen feature extraction, and could extend the dataset to include a wider range of medicinal plant species to test how well these findings generalize.

5. Declarations

Author Contributions

Mohammad Yoga Firnanda: Conceptualization, Methodology, Software, Data Curation, Formal Analysis, Investigation, Visualization, Writing Original Draft, and Writing – Review & Editing. Rohman Dijaya, Ika Ratna Indra Astutik, and Yulian Findawati: Supervision, Validation, and Writing – Review & Editing. All authors reviewed and approved the final version of the manuscript.

Funding

This research received no specific funding from any funding agency in the public, commercial, or not-for-profit sectors.

Conflict of Interest

The authors declare that they have no conflict of interest related to the research, authorship, or publication of this article.

Data Availability Statement

The dataset used in this study is publicly available from Mendeley Data as the Indonesian Herb Leaf Dataset 3500. The processed data and experimental results supporting the findings of this study are available from the corresponding author upon reasonable request.

Use of Artificial Intelligence (AI)

The authors used artificial intelligence (AI) tools to assist with language editing, manuscript preparation, and improving manuscript clarity. AI was not used to generate, analyze, or interpret the research data. All AI-assisted content was reviewed and verified by the authors, who take full responsibility for the final content of the manuscript.

Ethical Clearance

This study did not involve human participants, animals, or collection of personal or sensitive data. Therefore, ethical clearance was not applicable.

6. References

- [1] B. Setiyono and others, "Identifikasi Tanaman Obat Indonesia Melalui Citra Daun Menggunakan Metode Convolutional Neural Network (CNN)," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 10, no. 2, pp. 385–392, 2023, doi: 10.25126/jtiik.20236809.
- [2] R. Pujiati and N. Rochmawati, "Identifikasi Citra Daun Tanaman Herbal Menggunakan Metode Convolutional Neural Network (CNN)," *J. Informatics Comput. Sci.*, vol. 3, 2022, doi: 10.26740/jinacs.v3n03.p351-357.
- [3] S. Salsabila and A. Suharso, "Comparison of Deep Learning Architectures in Identifying Types of Medicinal Plant Leaf Images," *J. Appl. Informatics Comput.*, vol. 8, no. 1, pp. 39–46, 2024, doi: 10.30871/jaic.v8i1.6289.
- [4] Y. Gulzar, "Fruit Image Classification Model Based on MobileNetV2 with Deep Transfer Learning Technique," *Sustainability*, vol. 15, no. 3, 2023, doi: 10.3390/su15031906.
- [5] M. A. Morid, A. Borjali, and G. Del Fiore, "A scoping review of transfer learning research on medical image analysis using ImageNet," *Comput. Biol. Med.*, vol. 128, no. 408, 2021, doi: 10.1016/j.combiomed.2020.104115.
- [6] H. J. Mustaqim and O. Suria, "Comparative Analysis of ResNet50V2, ResNet152V2, and MobileNetV2 Architectures in Monkeypox Classification," *J. Appl. Informatics Comput.*, vol. 9, no. 4, pp. 1326–1331, 2025, doi: 10.30871/jaic.v9i4.9756.
- [7] S. Arnandito and T. B. Sasongko, "Comparison of EfficientNetB7 and MobileNetV2 in Herbal Plant Species Classification Using Convolutional Neural Networks," vol. 8, no. 1, pp. 176–185, 2024, doi: 10.30871/jaic.v8i1.7927.
- [8] D. R. Fauzi and G. A. Haqdu D, "Comparison of CNN Models Using EfficientNetB0, MobileNetV2, and ResNet50 for Traffic Density with Transfer Learning," *J. Intell. Syst. Technol. Informatics*, vol. 1, no. 1, pp. 22–30, 2025, doi: 10.64878/jistics.v1i1.6.
- [9] H. Kaur, R. Sharma, and J. Kaur, "Comparison of deep transfer learning models for classification of cervical cancer from pap smear images," *Sci. Rep.*, vol. 15, no. 1, pp. 1–18, 2025, doi: 10.1038/s41598-024-74531-0.
- [10] D. J. Richter and K. Kim, "Assessing the performance of domain-specific models for plant leaf disease classification: a comprehensive benchmark of transfer-learning on open datasets," *Sci. Rep.*, vol. 15, no. 1, pp. 1–31, 2025, doi: 10.1038/s41598-025-03235-w.
- [11] A. R. T. T. and A. Sudarshan, "Comparative Analysis of Deep Learning Models for Tomato Leaf Disease Classification Using Transfer Learning," *Int. J. Multidiscip. Res.*, vol. 8, no. 3, pp. 287–304, 2026, doi: 10.36948/ijfmr.2026.v08i03.80069.

- [12] A. Sonita and others, "Herbal Plant Classification Using EfficientNetV2B0 Model and CRISP-DM Approach," *J. Tek. Inform.*, vol. 6, no. 4, pp. 2253–2262, 2025, doi: 10.52436/1.jutif.2025.6.4.5141.
- [13] S. R. Yang, H. C. Yang, F. R. Shen, and J. Zhao, "Image Data Augmentation for Deep Learning: A Survey," *Ruan Jian Xue Bao/Journal Softw.*, vol. 36, no. 3, pp. 1390–1412, 2025, doi: 10.13328/j.cnki.jos.007263.
- [14] W. Zheng, "Comparison of transfer-learning for lightweight pre-trained model on image classification," *Appl. Comput. Eng.*, vol. 53, no. 1, pp. 56–63, 2024, doi: 10.54254/2755-2721/53/20241244.
- [15] M. Sandler, M. Zhu, A. Zhmoginov, and C. V Mar, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. doi: 10.1109/CVPR.2018.00474.
- [16] K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks," 2016, pp. 1–15. doi: 10.1007/978-3-319-46493-0_38.
- [17] M. Tan and Q. V Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," 2019.
- [18] R. D. C. N. Sahera and Y. Rahmawati, "Optimasi Penerapan Algoritma Convolution Neural Network dalam Klasifikasi Tingkat Kesegaran Daging Sapi," *SNTEI*, pp. 207–212, 2022, doi: 10.37600/tekinkom.v7i1.1122.
- [19] T. H. Pinem and Z. P. Putra, "Evaluasi Kinerja Algoritma Klasifikasi Deep Learning dalam Prediksi Diabetes," *J. Ilm. FIFO*, vol. 17, no. 1, p. 17, 2025, doi: 10.22441/fifo.2025.v17i1.003.
- [20] R. Sianturi, "Test Normality As a Condition of Hypothesis Testing," *J. Pembelajaran dan Mat. Sigma*, vol. 11, no. 1, pp. 1–14, 2025, doi: 10.36987/jpms.v11i1.7091.
- [21] A. Gunasti, R. Ariany, S. Duwi, and S. Ahmad, "Perbandingan Jumlah Kendaraan Roda 4 Pada Jalan yang Berbeda di Hari yang Sama Menggunakan Analisa One Way Anova," *J. Smart Teknol.*, vol. 5, no. 3, pp. 422–429, 2024.
- [22] Z. R. Mardhotillah, Bunga, and S. Rozi, "Tukey HSD Post Hoc Test untuk Perbandingan Karakteristik Lingkungan dan Sumber Daya Provinsi Provinsi di Indonesia," *J. Eng.*, vol. 3, no. 2, 2021, doi: 10.22437/jurnalengineering.v3i2.14445.
- [23] I. E. Tampu, A. Eklund, and N. Haj-Hosseini, "Inflation of test accuracy due to data leakage in deep learning-based classification of OCT images," *Sci. Data*, vol. 9, no. 1, pp. 1–8, 2022, doi: 10.1038/s41597-022-01618-6.