**Journal of Artificial Intelligence and Technology Information**
**Website : https://ejournal.techcart-press.com/index.php/jaiti**
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 4, December 2026
**https://doi.org/10.58602/jaiti.v4i4.454**
Page 717-729

# Disentangling the Effects of Data Augmentation and Fine-Tuning in Herbal Plant Image Classification

**Titik Misriati[1*]✉, Riska Aryanti[2]✉, Siti Marlina[1]✉, Sarifah Agustiani[3]✉, Henny Leidiyana[3]✉**
[1]Information System, Universitas Bina Sarana Informatika, Indonesia
[2]Information Technology, Universitas Bina Sarana Informatika, Indonesia
[3]Informatics, Universitas Bina Sarana Informatika, Indonesia

**Abstract**
Image-based herbal plant identification requires reliable evaluation to distinguish improvements attributable to model architecture, data augmentation, and fine-tuning. This study investigated these factors through a controlled comparison of EfficientNetV2-B0, ResNet-50, and MobileNetV3-Small using 3,500 images from ten herbal plant classes. An image audit combined exact-content hashing, perceptual-similarity screening, and recorded pair decisions to construct image groups before partitioning. The resulting training, validation, and test sets contained 2,450, 530, and 520 images, respectively, with no detected overlap between recorded groups or identical image content. Each architecture was evaluated under four scenarios combining augmentation and backbone fine-tuning, with three random seeds, yielding 36 experimental runs. Performance was assessed using classification metrics, paired ablation contrasts, confusion matrices, parameter counts, and inference latency. Fine-tuning consistently improved macro-F1, whereas augmentation reduced performance with frozen backbones and produced modest improvements after fine-tuning. ResNet-50 with augmentation and fine-tuning achieved the highest mean accuracy of 99.29% and macro-F1 of 99.30%, although its numerical advantage over EfficientNetV2-B0 did not establish statistical superiority. MobileNetV3-Small achieved 98.47% macro-F1 with approximately 1.53 million parameters, supporting further investigation for resource-constrained applications. These findings demonstrate that augmentation effectiveness depends on the adaptation strategy and provide guidance for selecting herbal-image classifiers.
**Keywords:** Herbal plant identification; Transfer learning; Data augmentation; Fine-tuning; Group-based partitioning

***Corresponding Author:**
Titik Misriati, Universitas Bina Sarana Informatika, Indonesia
Email: titik.tmi@bsi.ac.id

## 1. INTRODUCTION

Reliable identification of medicinal plants supports botanical documentation, biodiversity conservation, and the organization of herbal resources. Image-based recognition offers a practical means of assisting this task because leaf appearance provides observable information about shape, venation, texture, and colour. Recent reviews document substantial interest in deep learning for medicinal plant identification, while emphasizing persistent limitations in dataset coverage and model generalization [1], [2] . Research on Indonesian medicinal plants has further demonstrated the relevance of transfer learning to locally important species [3]. Public resources such as DIMPSAR and BDMediLeaves also illustrate the growing availability of images for reproducible research across different geographical settings [4], [5]. These developments motivate the investigation of accessible recognition systems whose performance is supported by transparent experimental evidence.

Nevertheless, accurate recognition remains difficult when different species exhibit similar leaf morphology or when images of the same species vary in orientation, illumination, background, and

**Copyright © 2026, The Author(s).**
This is an open access article under the CC BY-SA license
**https://creativecommons.org/licenses/by-sa/4.0/**

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 4, December 2026
https://doi.org/10.58602/jaiti.v4i4.454
Page 717-729

growth stage. Studies of Ayurvedic plant misidentification and leaf-image standardization highlight the importance of discriminative visual characteristics and their preservation during preprocessing [6], [7]. Hierarchical classification research likewise addresses the combined challenges of interclass similarity and intraclass variation [8]. A further concern arises when duplicate or closely related images occur in different data partitions. In this situation, a classifier may be evaluated on content that is insufficiently independent of its training observations. Research on machine-learning reproducibility and biological prediction identifies such information leakage as a threat to the validity of reported performance [9], [10]. Consequently, evaluation quality depends on both model design and the construction of the experimental dataset.

Recent investigations have explored several complementary approaches. Ayur-PlantNet introduced a lightweight convolutional architecture for Ayurvedic plant classification [11], whereas AELGNet combined attention with local and global features for medicinal leaf and plant recognition [12]. HybNet investigated hybrid convolutional models [13], and Plant-CNN-ViT combined convolutional networks with a vision transformer [14]. Herbify subsequently developed an ensemble framework integrating these two architectural families for herb identification [15]. Beyond leaf-only recognition, automatic multimodal fusion has been investigated to integrate information from multiple plant organs [16]. Research on houseplant leaves has also examined pretrained networks and augmentation within a smaller classification setting [17]. Collectively, these studies establish several viable modelling strategies; however, their results are conditional on their respective datasets, preprocessing procedures, and evaluation protocols, and therefore do not constitute directly interchangeable benchmarks.

An unresolved methodological question is how much improvement originates from adapting pretrained representations, how much originates from data augmentation, and whether these effects depend on one another. Transfer learning provides several ways to reuse previously learned representations [18], while augmentation surveys describe substantial variation in transformation policies and their suitability for particular tasks [19], [20]. When augmentation and fine-tuning are introduced simultaneously, a comparison with a single baseline cannot separate their individual effects. Similarly, an architecture ranking obtained from one training seed does not characterize sensitivity to stochastic training. The present study therefore addresses a specific empirical gap: the absence of a controlled decomposition of these effects for the selected ten-class Indonesian herbal dataset under one fixed, audited partition. This scope avoids assuming that the same research gap applies to every existing plant-recognition study.

To address this question, the study employs a reproducible evaluation framework combining image-integrity checks, duplicate-aware grouping, and a two-factor experimental design. EfficientNetV2-B0, ResNet-50, and MobileNetV3-Small are evaluated under four scenarios formed by crossing augmentation availability with backbone fine-tuning. Each configuration uses the same data partition and three training seeds, allowing changes in performance to be examined within each architecture. Exact-content checks and perceptual-hash candidate matching are performed before partitioning, and the resulting dataset manifest is locked for subsequent experiments. The methodological contribution lies in integrating these controls with paired ablation analysis and computational evaluation, thereby providing an interpretable assessment of training choices on the selected dataset.

The main objective is to determine how architecture, augmentation, and fine-tuning influence herbal-image classification performance under a common evaluation protocol. Specifically, the study examines whether augmentation benefits a frozen pretrained representation, whether fine-tuning changes this effect, and how predictive performance relates to model size and inference latency. Its contributions comprise an auditable partitioning procedure, a repeated-seed comparison of three pretrained networks, a factorial analysis of augmentation and fine-tuning, and a class-level investigation of residual classification errors. These contributions provide empirical guidance for selecting a training configuration while defining the limits of conclusions drawn from a single dataset. The following section describes the data, preprocessing, experimental scenarios, optimization procedure, and evaluation criteria.

**Copyright © 2026, The Author(s).**
This is an open access article under the CC BY-SA license
https://creativecommons.org/licenses/by-sa/4.0/

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 4, December 2026
https://doi.org/10.58602/jaiti.v4i4.454
Page 717-729

# 2. RESEARCH METHODOLOGY

## 2.1. Research Design

The study employed a controlled image-classification experiment with three architectures, four training scenarios, and three random seeds, yielding 36 runs. Architecture comparisons and within-architecture ablations used an identical training, validation, and test partition [21]. Fig. 1 outlines the workflow, including the separation between optimization, validation-based checkpoint selection, and final test evaluation. The test partition was excluded from gradient updates, learning-rate scheduling, and checkpoint selection.

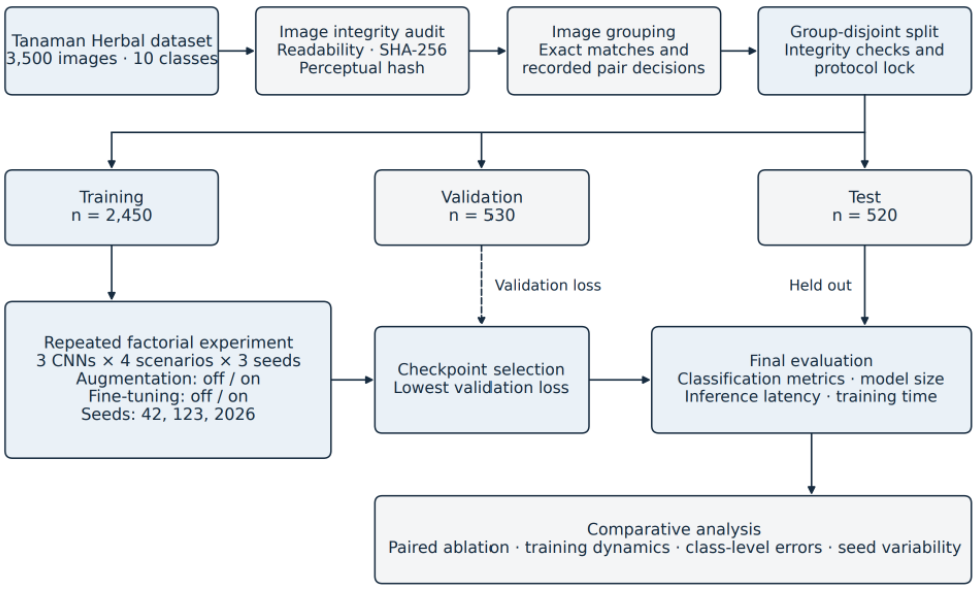



**Figure 1. Research Workflow**

## 2.2. Dataset and Class Definition

The experiments used the publicly available Tanaman Herbal dataset distributed by Ellen Theodora through Kaggle [22]. The analyzed collection contained 3,500 readable RGB images representing ten class labels, with 350 images per class and an image resolution of 1,600 × 1,200 pixels. The original class names were retained to maintain consistency between the source folders, integer labels, prediction files, and confusion matrices. The dataset was treated as a closed-set classification problem: each image was assigned to exactly one of the ten available classes. Table I presents the class distribution and the realized partition sizes.

**Table 1. Dataset Composition and Realized Partition Sizes**

| Class label | Training | Validation | Test | Total |
|---|---|---|---|---|
| Belimbing Wuluh | 245 | 53 | 52 | 350 |
| Jambu Biji | 245 | 53 | 52 | 350 |
| Jeruk Nipis | 245 | 53 | 52 | 350 |
| Kemangi | 245 | 53 | 52 | 350 |
| Lidah Buaya | 245 | 53 | 52 | 350 |
| Nangka | 245 | 53 | 52 | 350 |
| Pandan | 245 | 53 | 52 | 350 |
| Pepaya | 245 | 53 | 52 | 350 |
| Seledri | 245 | 53 | 52 | 350 |
| Sirih | 245 | 53 | 52 | 350 |
| Total | 2,450 | 530 | 520 | 3,500 |

Copyright © 2026, The Author(s).
This is an open access article under the CC BY-SA license
https://creativecommons.org/licenses/by-sa/4.0/

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 4, December 2026
https://doi.org/10.58602/jaiti.v4i4.454
Page 717-729

## 2.3. Image Audit and Group-Based Partitioning

The audit checked image readability, corrected orientation using EXIF information, and converted decoded images to RGB. SHA-256 hashes were calculated for both the original file bytes and the decoded RGB content, with image dimensions included in the latter calculation. This distinguished file-level identity from identical decoded content. Perceptual similarity was screened using a 64-bit perceptual hash obtained from a 32 × 32 grayscale image and the low-frequency 8 × 8 block of its two-dimensional discrete cosine transform. Candidate pairs were identified when their hash Hamming distance was no greater than four. The threshold was used to identify candidates, rather than to establish botanical identity.

Exact matches and candidate pairs recorded as "same" in the pair-decision file were connected using a union–find procedure. Their connected components defined operational image groups. Groups containing conflicting class labels were subject to exclusion under the implemented protocol. Duplicate images were retained within their assigned groups rather than removed. Group-based separation was adopted because dependence between observations can compromise evaluation, as also demonstrated in other biological imaging applications [23]. The available grouping record does not establish independent verification of specimen identity; therefore, the procedure supports claims about the recorded image groups rather than confirmed plant- or session-level independence.

Groups were allocated separately within each class using a randomized greedy procedure with 200 trials and a base split seed of 42. The allocation minimized the sum of squared relative deviations from the target proportions of 70%, 15%, and 15%, while preserving complete groups. The realized counts are reported in Table I. Automated checks verified the absence of group, file-hash, and decoded-pixel-hash overlap across partitions. The split manifest, configuration, source code, and pair-decision file were associated with SHA-256 checksums so that subsequent training and evaluation could reject incompatible inputs. Because specimen and acquisition-session identifiers were unavailable, unrecognized dependencies could remain despite these checks.

## 2.4. Preprocessing and Data Augmentation

Images were resized to a 224 × 224 input using aspect-ratio-preserving letterboxing. The longest dimension was resized to 224 pixels with bicubic interpolation, and the remaining area was filled by reflection padding. This procedure preserved image proportions without requiring leaf segmentation. Its motivation is consistent with research showing that geometric distortion during input standardization can affect leaf recognition [7], although the present implementation did not reproduce that study's segmentation-based standardization method. Pixel values were converted to tensors and normalized using the mean and standard deviation associated with each pretrained model configuration.

For augmentation-enabled scenarios, training images underwent random horizontal flipping with probability 0.5, rotation within ±15°, and colour jitter with brightness, contrast, and saturation factors of 0.15, 0.15, and 0.10, respectively. Rotation used bilinear interpolation and a fill colour derived from the pretrained normalization mean. These operations were applied online after letterboxing. Validation and test images received only deterministic resizing, padding, tensor conversion, and normalization. Augmentation research encompasses both manually specified and learned transformation policies [24], [25], [26]; the present experiment deliberately used one fixed policy to isolate its effect from that of fine-tuning. No synthetic image-generation procedure or augmentation-policy search was included.

## 2.5. Pretrained Models and Experimental Scenarios

Three ImageNet-1K-pretrained models were instantiated through the timm library: tf_efficientnetv2_b0.in1k, resnet50.a1_in1k, and mobilenetv3_small_100.lamb_in1k. Each original classification layer was replaced with a randomly initialized ten-output layer. The selected models provided different parameter counts and computational characteristics under a common input resolution. Their comparison concerns these specific pretrained configurations, because differences between pretraining recipes cannot be separated from architectural differences in this experiment.

Copyright © 2026, The Author(s).
This is an open access article under the CC BY-SA license
https://creativecommons.org/licenses/by-sa/4.0/

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 4, December 2026
https://doi.org/10.58602/jaiti.v4i4.454
Page 717-729

Table 2 defines the two-factor design. All scenarios began with head-only training. During the second phase, S1 and S2 continued to update only the classifier, whereas S3 and S4 enabled gradients for all model parameters. Batch Normalization running statistics remained fixed in every scenario; its affine parameters were trainable when full fine-tuning was enabled. Each model–scenario combination was repeated with seeds 42, 123, and 2026 while retaining the same split manifest.

**Table 2. Factorial Training Scenarios**

| Scenario | Training augmentation | First phase | Second phase |
|---|---|---|---|
| S1 | Disabled | Classifier only | Classifier only |
| S2 | Enabled | Classifier only | Full model remains frozen except classifier |
| S3 | Disabled | Classifier only | All parameters trainable; BN statistics fixed |
| S4 | Enabled | Classifier only | All parameters trainable; BN statistics fixed |

## 2.6. Optimization and Checkpoint Selection

Training minimized multiclass cross-entropy using AdamW with a batch size of 16 and weight decay of $10^{-4}$. The first phase comprised five epochs at a learning rate of $10^{-3}$. The second phase used an initial learning rate of $10^{-4}$ for a maximum of 30 additional epochs. A fresh optimizer and scheduler were initialized at the phase transition. During the second phase, ReduceLROnPlateau monitored validation loss and multiplied the learning rate by 0.5 after three non-improving epochs according to its configured patience. Early stopping used a patience of seven epochs and a minimum validation-loss improvement of $10^{-4}$. Table 3 summarizes these settings. Optimization surveys identify the update algorithm and its associated settings as important components of neural-network training [27].

**Table 3. Optimization and Reproducibility Settings**

| Component | Setting |
|---|---|
| Input size | 224 × 224 pixels |
| Loss | Multiclass cross-entropy |
| Optimizer | AdamW |
| Batch size | 16 |
| Weight decay | $10^{-4}$ |
| Head-training phase | 5 epochs; learning rate $10^{-3}$ |
| Second phase | Maximum 30 epochs; initial learning rate $10^{-4}$ |
| Learning-rate scheduler | ReduceLROnPlateau; factor 0.5; patience 3 |
| Early stopping | Second phase; patience 7; minimum improvement $10^{-4}$ |
| Checkpoint criterion | Lowest validation loss across both phases |
| Training seeds | 42, 123, and 2026 |
| Precision | Float32; TF32 disabled |
| Data-loader workers | 0 |

The selected checkpoint was the one with the lowest validation loss observed across both phases. Checkpoint updates required a strict decrease in loss; the early-stopping minimum improvement was a separate condition. Thus, the final epoch was not necessarily the selected epoch. Python, NumPy, and PyTorch random generators were seeded, cuDNN benchmarking was disabled, and deterministic execution settings were requested. These controls reduce avoidable variability but do not guarantee identical results across different hardware or software environments. Package versions, device information, model identifiers, and normalization parameters were recorded with each run.

## 2.7. Classification Metrics and Paired Ablation Analysis

Each selected checkpoint was evaluated on the same 520 test images. Accuracy, macro-precision, macro-recall, and macro-F1 were calculated, with zero assigned to undefined per-class precision or recall terms. Macro-F1 was calculated by averaging class-specific F1 scores rather than by taking the harmonic mean of macro-precision and macro-recall [28]. This explicit definition follows the need for clear metric specification emphasized by Opitz [25]. For K = 10 classes, the principal measures were

Copyright © 2026, The Author(s).
This is an open access article under the CC BY-SA license
https://creativecommons.org/licenses/by-sa/4.0/

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 4, December 2026
https://doi.org/10.58602/jaiti.v4i4.454
Page 717-729

$$\text{Accuracy} = \frac{\sum_{c=1}^{K} TP_c}{N}, \qquad F_{1,\text{macro}} = \frac{1}{K}\sum_{c=1}^{K} \frac{2TP_c}{2TP_c + FP_c + FN_c} \tag{1}$$

Performance was summarized as the mean and sample standard deviation across the three seeds. These summaries quantify training variability conditional on the fixed partition, rather than uncertainty over independently sampled datasets. For metric M and seed r, augmentation effects were estimated as M(S2,r) − M(S1,r) for a frozen backbone and M(S4,r) − M(S3,r) for a fine-tuned backbone. Fine-tuning effects were estimated as M(S3,r) − M(S1,r) without augmentation and M(S4,r) − M(S2,r) with augmentation. The interaction contrast was

$$I_r = [M(S4,r) - M(S3,r)] - [M(S2,r) - M(S1,r)] \tag{2}$$

The contrasts in (2) and the corresponding main comparisons were calculated within architecture and paired by seed. Differences in percentage-valued metrics were reported in percentage points. Pairing by seed provides a consistent comparison index but does not imply identical random operations across scenarios. The analysis was descriptive, and performance differences were not designated statistically significant without a separate inferential procedure.

### 2.8. Training Dynamics, Error Analysis, and Computational Evaluation

Training and validation loss and macro-F1 were recorded at every epoch to examine convergence, phase-transition behaviour, and separation between training and validation performance. For augmentation-enabled scenarios, training metrics describe transformed training inputs, whereas validation metrics describe deterministic inputs; this distinction was considered when interpreting their differences. Class-level precision, recall, F1, and confusion matrices were examined to identify recurring errors. Confusion-matrix rows represented ground-truth labels and columns represented predictions. When matrices were aggregated across seeds, their entries represented repeated predictions of the same test images rather than additional independent observations.

Computational evaluation included total parameter count, recorded training time, and batch-size-one inference latency. Latency was measured using a tensor already resident on the execution device, with 50 warm-up iterations followed by 500 timed forward passes. Inference mode was enabled, and CUDA execution was synchronized around timing boundaries when applicable. The median latency was calculated for each run. These measurements characterize model-forward latency and exclude image decoding, preprocessing, host-to-device transfer, and application overhead.

## 3. RESULT AND DISCUSSIONS

### 3.1. Dataset Audit and Partition Integrity

The audited dataset comprised 3,500 readable images from ten herbal plant classes, with 350 images per class. All images were retained, and no cross-class groups were reported in the final audit. Group-constrained partitioning produced 2,795 groups, including 2,566 singleton groups and 229 multi-image groups. The latter contained 934 images, representing 26.69% of the dataset. The largest group contained 103 images. Table 4 presents the partition sizes and recorded integrity checks. Each class contributed 245 training images, 53 validation images, and 52 test images, maintaining equal class representation within each partition.

**Table 4. Dataset Partitioning and Recorded Integrity Checks**

| Partition or check | Images | Groups | Images per class |
|---|---|---|---|
| Training | 2,450 | 1,841 | 245 |
| Validation | 530 | 492 | 53 |
| Test | 520 | 462 | 52 |
| Total | 3,500 | 2,795 | 350 |
| Group overlap across partitions | — | 0 | — |
| Exact-pixel overlap across partitions | 0 | — | — |

An exploratory inspection of the embedded review thumbnails covered all 103 images in the largest operational component, g001403, assigned to Lidah Buaya. As illustrated by the selected examples in

Copyright © 2026, The Author(s).
This is an open access article under the CC BY-SA license
https://creativecommons.org/licenses/by-sa/4.0/

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 4, December 2026
https://doi.org/10.58602/jaiti.v4i4.454
Page 717-729

Fig. 2, the component includes elongated leaf images with visible variation in colour, curvature, surface markings, and margins. The manifest records 103 distinct decoded-pixel hashes, indicating that this component is not a set of exact pixel duplicates. Its construction includes 911 candidate-pair links recorded as "same". All 103 images were assigned to the training partition.




**Figure 2. Selected thumbnails from the 103-image Lidah Buaya component g001403**

The recorded audit detected no group or exact-pixel overlap across partitions. All 36 experimental runs also reported the same partition-manifest checksum, supporting comparisons on a common test set. These checks establish consistency under the implemented grouping procedure but do not prove specimen-level independence. Perceptual similarity can connect images transitively, and membership in a large component does not establish that all constituent images depict the same specimen. Without plant-identity and acquisition-session metadata, the results should therefore be interpreted as performance under the recorded group-constrained split.

## 3.2. Classification Performance across Models and Training Scenarios

The test performance across three architectures and four training scenarios is summarised in Table 5. S1 and S2 used classifier-head training without and with augmentation, respectively, whereas S3 and S4 incorporated fine-tuning without and with augmentation. Each configuration was repeated using seeds 42, 123, and 2026. Results are expressed as the mean and sample standard deviation across seeds. Macro-recall equals accuracy because every test class contains 52 images and is therefore omitted from the table to avoid numerical duplication.

**Table 5. Test Classification Performance Across Three Seeds (%)**

| Model | Scenario | Accuracy | Macro-precision | Macro-F1 |
|---|---|---|---|---|
| EfficientNetV2-B0 | S1 | 97.24 ± 0.29 | 97.31 ± 0.26 | 97.24 ± 0.29 |
| EfficientNetV2-B0 | S2 | 95.58 ± 0.33 | 95.77 ± 0.27 | 95.59 ± 0.33 |
| EfficientNetV2-B0 | S3 | 98.97 ± 0.11 | 98.99 ± 0.10 | 98.97 ± 0.11 |
| EfficientNetV2-B0 | S4 | 99.23 ± 0.19 | 99.24 ± 0.18 | 99.23 ± 0.19 |
| ResNet-50 | S1 | 95.77 ± 0.00 | 95.84 ± 0.00 | 95.76 ± 0.00 |
| ResNet-50 | S2 | 95.19 ± 0.19 | 95.36 ± 0.21 | 95.20 ± 0.20 |
| ResNet-50 | S3 | 99.04 ± 0.38 | 99.05 ± 0.38 | 99.04 ± 0.39 |
| ResNet-50 | S4 | **99.29 ± 0.29** | **99.31 ± 0.29** | **99.30 ± 0.29** |
| MobileNetV3-Small | S1 | 94.81 ± 0.58 | 94.90 ± 0.54 | 94.82 ± 0.57 |
| MobileNetV3-Small | S2 | 93.14 ± 0.97 | 93.65 ± 0.99 | 93.20 ± 0.98 |
| MobileNetV3-Small | S3 | 98.14 ± 0.22 | 98.18 ± 0.20 | 98.14 ± 0.22 |
| MobileNetV3-Small | S4 | 98.46 ± 0.38 | 98.50 ± 0.36 | 98.47 ± 0.38 |

**Copyright © 2026, The Author(s).**
This is an open access article under the CC BY-SA license
https://creativecommons.org/licenses/by-sa/4.0/

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 4, December 2026
https://doi.org/10.58602/jaiti.v4i4.454
Page 717-729

ResNet-50 under S4 achieved the highest mean accuracy and macro-F1. However, its macro-F1 advantage over EfficientNetV2-B0 under S4 was only 0.065 percentage points, calculated from unrounded values. Across the three seeds, these configurations produced 11 and 12 errors, respectively, on repeated evaluations of the same 520 images. This small numerical difference does not establish statistical superiority. EfficientNetV2-B0 provided the strongest frozen-backbone baseline, whereas fine-tuning substantially narrowed its advantage over ResNet-50. MobileNetV3-Small achieved 98.47% macro-F1 under S4, demonstrating competitive classification performance with a substantially smaller parameter count.

### 3.3. Effects of Data Augmentation and Fine-Tuning

Paired macro-F1 differences calculated within each seed are reported in Table 6. Augmentation during head training reduced performance for every model and seed, with mean decreases of 1.64, 0.56, and 1.62 percentage points for EfficientNetV2-B0, ResNet-50, and MobileNetV3-Small, respectively. After fine-tuning, augmentation produced smaller positive mean effects of 0.26, 0.26, and 0.33 percentage points. The improvement occurred for all EfficientNetV2-B0 seeds but only two of three seeds for each remaining architecture.

Fine-tuning improved macro-F1 in every paired comparison, both with and without augmentation. The largest gain occurred for MobileNetV3-Small when moving from S2 to S4, at 5.27 percentage points. This suggests that adaptation of the pretrained representations was more influential than augmentation alone under the evaluated protocol. However, these contrasts compare complete training regimes and do not isolate backbone adaptation from all associated optimisation changes.

The positive interactions effects in Table 6 indicate that augmentation became more beneficial after backbone adaptation. A plausible explanation is that feature adaptation helped accommodate the transformations applied during training, although the aggregate results do not verify this mechanism. The interaction also partly reflects the negative augmentation effect under head training and should not be interpreted as the direct benefit of adding augmentation to S3. The direct augmentation gains after fine-tuning remained below 0.34 percentage points for all models, and no claim of statistical significance is made.

**Table 6. Paired Macro-F1 Effects of Augmentation and Fine-Tuning**

| Comparison | EfficientNetV2-B0 | ResNet-50 | MobileNetV3-Small |
|---|---|---|---|
| Augmentation during head training: S2 − S1 | −1.64 ± 0.47 | −0.56 ± 0.20 | −1.62 ± 0.43 |
| Augmentation during fine-tuning: S4 − S3 | +0.26 ± 0.11 | +0.26 ± 0.62 | +0.33 ± 0.43 |
| Fine-tuning without augmentation: S3 − S1 | +1.74 ± 0.38 | +3.28 ± 0.39 | +3.32 ± 0.76 |
| Fine-tuning with augmentation: S4 − S2 | +3.64 ± 0.38 | +4.09 ± 0.44 | +5.27 ± 0.80 |
| Interaction: (S4 − S3) − (S2 − S1) | +1.90 ± 0.37 | +0.82 ± 0.80 | +1.95 ± 0.08 |

### 3.4. Training Dynamics and Class-Level Error Analysis

The complete EfficientNetV2-B0 histories for all four scenarios and three seeds are shown in Fig. 3. S1 and S2 completed 35 epochs in every run. Their minimum-validation-loss checkpoints occurred at epochs 35/35/35 and 34/33/34, respectively, in seed order 42/123/2026. S3 stopped at epochs 16/18/20 and selected checkpoints at epochs 9/11/15. S4 stopped at epochs 16/35/21 and selected checkpoints at epochs 9/31/14. These differences show that augmentation did not consistently accelerate convergence. In particular, the S4 run with seed 123 exhibited a transient validation-loss increase before recovering. Checkpoint selection must therefore follow the specified validation-loss criterion rather than the final epoch or the highest test score.

Copyright © 2026, The Author(s).
This is an open access article under the CC BY-SA license
https://creativecommons.org/licenses/by-sa/4.0/

**Journal of Artificial Intelligence and Technology Information**
**Website : https://ejournal.techcart-press.com/index.php/jaiti**
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 4, December 2026
**https://doi.org/10.58602/jaiti.v4i4.454**
Page 717-729

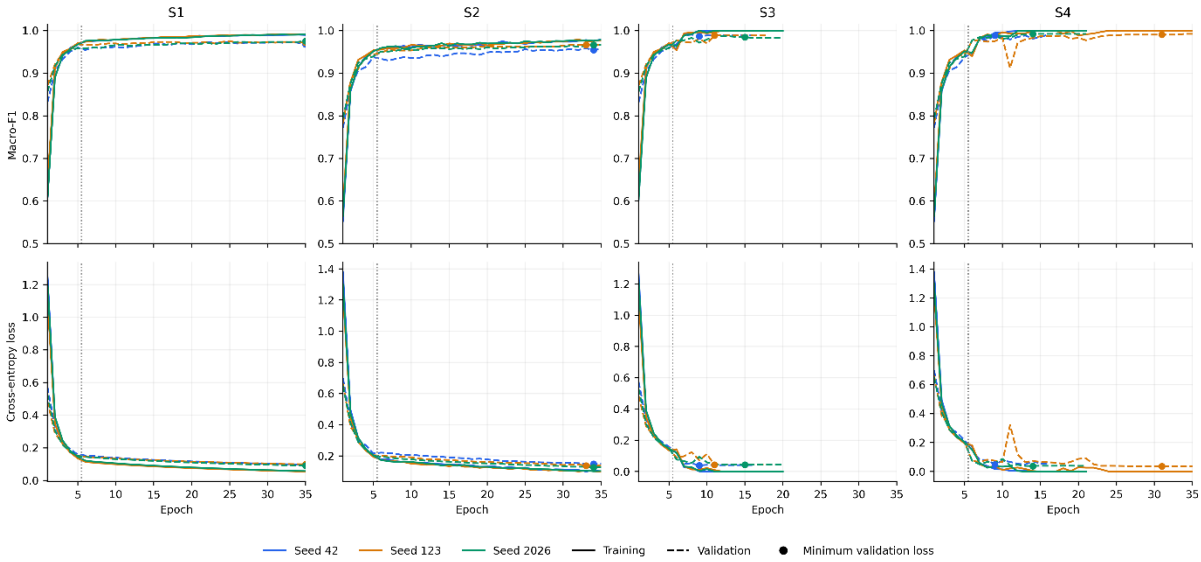



**Figure 3. Training and validation macro-F1 (top) and cross-entropy loss (bottom) for EfficientNetV2-B0 across S1–S4 and three seeds.**

The S4 confusion matrices summed across the three seeds are compared in Fig. 4. ResNet-50, EfficientNetV2-B0, and MobileNetV3-Small produced 11, 12, and 24 errors, respectively. Each matrix represents 1,560 predictions on the same 520 test images evaluated three times, rather than 1,560 independent images. The largest error count for ResNet-50 occurred in Lidah Buaya, with four predictions assigned to Pandan. EfficientNetV2-B0 made four Kemangi errors, including three assignments to Jeruk Nipis. MobileNetV3-Small made six Kemangi errors, split equally between Belimbing Wuluh and Jeruk Nipis. Kemangi-to-Jeruk Nipis confusion occurred three times for each architecture, indicating a shared error direction, although aggregate counts cannot determine whether identical images were involved.

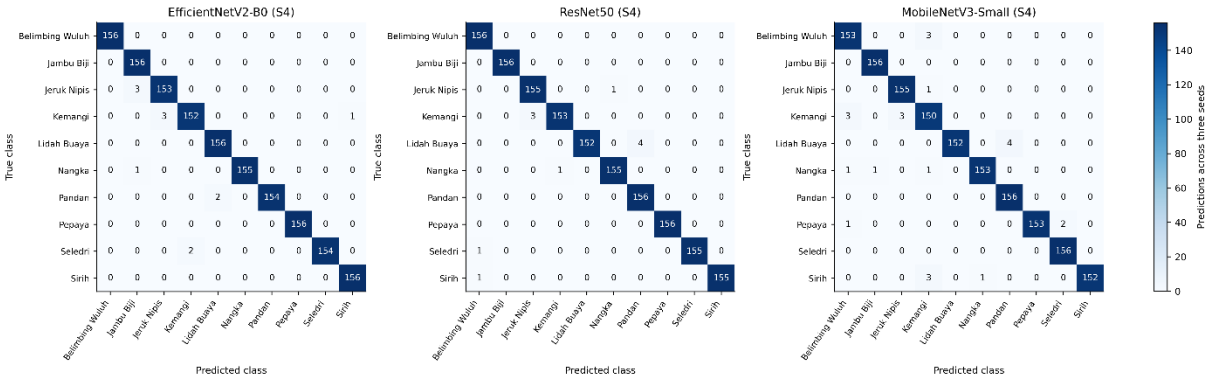



**Figure 4. S4 test confusion matrices summed across seeds 42, 123, and 2026**

The patterns in Fig. 4 show that similar overall accuracies can conceal different class-level weaknesses. Across various model architectures, Kemangi proved to be the most challenging class to classify, frequently being misidentified as Jeruk Nipis. In stark contrast, Jambu Biji achieved flawless recall across all S4 models, demonstrating that its distinct visual features were consistently captured by the learned representations. This pattern indicates that residual errors were localized within a handful of specific class pairs rather than being spread evenly across the entire dataset. EfficientNetV2-B0 correctly classified all Lidah Buaya observations across the three S4 evaluations but misclassified two Pandan observations as Lidah Buaya; ResNet-50 displayed errors in the reverse direction. Jambu Biji attained 100% recall for all three architectures, although false-positive assignments to this class remained possible. These results support examining precision, recall, and confusion direction together.

**Copyright © 2026, The Author(s).**
This is an open access article under the CC BY-SA license
**https://creativecommons.org/licenses/by-sa/4.0/**

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 4, December 2026
https://doi.org/10.58602/jaiti.v4i4.454
Page 717-729

Attribution to leaf shape, colour, background, or illumination requires direct inspection of the error images and cannot be established from confusion counts alone.

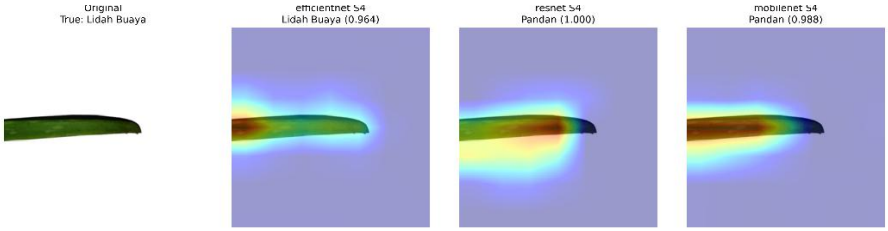



**Fig. 5.** Qualitative comparison of S4 predictions and Grad-CAM overlays for a Lidah Buaya image.

A complementary error pattern is shown in Fig. 5. EfficientNetV2-B0 correctly classified the Lidah Buaya image with a displayed score of 0.964, whereas ResNet-50 and MobileNetV3-Small predicted Pandan with scores of 1.000 and 0.988, respectively. All three heatmaps overlap portions of the elongated leaf, demonstrating that an apparently object-aligned response is not sufficient to ensure correct classification. The incorrect predictions also show that a high model score should not be interpreted as guaranteed correctness; the displayed value of 1.000 is rounded and does not demonstrate calibrated certainty.

### 3.5. Model Size and Computational Cost

Model size and computational measurements under S4 are compared in Table 7. MobileNetV3-Small contained 1.53 million parameters, approximately 93.51% fewer than ResNet-50 and 73.97% fewer than EfficientNetV2-B0. Its mean macro-F1 was 0.83 percentage points below ResNet-50, suggesting a useful accuracy–size trade-off when storage is constrained. Nevertheless, parameter count alone did not determine measured latency: the difference between MobileNetV3-Small and ResNet-50 was modest, whereas EfficientNetV2-B0 exhibited higher latency despite having fewer parameters than ResNet-50.

**Table 7. Model Size and Recorded Computational Cost Under S4**

| Model | Parameters (million) | Inference latency (ms) | Training time (min) |
|---|---|---|---|
| EfficientNetV2-B0 | 5.87 | 23.89 ± 7.23 | 53.54 ± 22.03 |
| ResNet-50 | 23.53 | 13.42 ± 0.49 | 73.99 ± 15.56 |
| MobileNetV3-Small | 1.53 | 12.73 ± 0.41 | 98.38 ± 76.87 |

The timing results are specific to the recorded execution environment. Single-image forward-pass latency excludes image loading and preprocessing and should not be interpreted as end-to-end application latency. Training duration also varied substantially, particularly for MobileNetV3-Small, and depended on stopping epoch and execution conditions. These measurements therefore do not establish that the smallest model is intrinsically slower to train or that its deployment advantage will persist across devices. Controlled timing and memory measurements on the intended target hardware are required before deployment claims are made.

### 3.6. Implications and Limitations

The main experimental finding is the dependence of augmentation effectiveness on the training regime. Backbone fine-tuning consistently improved performance, whereas augmentation reduced performance under frozen-backbone training and produced only modest gains after fine-tuning. ResNet-50 achieved the highest mean classification scores, but its advantage over EfficientNetV2-B0 was too small to support a general superiority claim. MobileNetV3-Small offered a substantially smaller model at a limited reduction in macro-F1.

These findings offer valuable practical insights for developing herbal-plant identification systems, particularly regarding resource allocation and model deployment. Within the evaluated framework, directing computational power toward fine-tuning the model's backbone yielded far more substantial performance gains than simply expanding a standard data augmentation pipeline, suggesting that practitioners handling mid-sized medicinal plant datasets should prioritize representation adaptation.

**Copyright © 2026, The Author(s).**
This is an open access article under the CC BY-SA license
https://creativecommons.org/licenses/by-sa/4.0/

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 4, December 2026
https://doi.org/10.58602/jaiti.v4i4.454
Page 717-729

Furthermore, the marginal accuracy gap between EfficientNetV2-B0 and ResNet-50 demonstrates that choosing an architecture should extend beyond raw performance metrics; operational factors like inference latency, hardware availability, memory footprint, and maintenance costs hold equal weight in mobile or edge-based applications. In these resource-constrained scenarios, lightweight models like MobileNetV3-Small stand out as highly viable candidates, delivering competitive macro-F1 scores despite their minimal parameter count. Consequently, even though this study did not test physical hardware deployments, the evidence strongly implies that compact architectures remain essential options whenever strict hardware limits override minor advantages in accuracy.

# 4. CONCLUSION

This study evaluated three pretrained convolutional neural networks under four training scenarios and three random seeds to examine the effects of augmentation and fine-tuning on herbal-image classification. Fine-tuning consistently improved macro-F1 across all architectures, whereas augmentation reduced performance when the backbone remained frozen and provided modest gains after fine-tuning. ResNet-50 with augmentation and fine-tuning achieved the highest mean accuracy of 99.29% and macro-F1 of 99.30%. However, its small numerical advantage over EfficientNetV2-B0 did not establish statistical superiority. MobileNetV3-Small achieved a macro-F1 of 98.47% with approximately 1.53 million parameters, offering a useful balance between predictive performance and model size. These findings support evaluating augmentation jointly with the adaptation strategy and provide practical guidance for developing image-based tools to assist herbal-plant identification.

The findings are limited to ten classes, one fixed partition, and three training seeds. Although the audit detected no overlap between recorded groups or identical image content across partitions, the absence of specimen and acquisition-session identifiers prevents confirmation of specimen-level independence. Furthermore, the reported variability reflects training randomness rather than uncertainty across populations or acquisition conditions. Future research should validate the models on independently collected images, verify related-image groups through expert inspection, and compare group-constrained and matched image-level partitions to quantify the influence of partitioning. Repeated group-based evaluation and dependence-aware confidence intervals would strengthen performance estimates. For practical applications, MobileNetV3-Small merits further assessment on resource-constrained devices, including end-to-end latency, memory consumption, and recognition reliability under variable illumination and backgrounds. These improvements would clarify the extent to which the observed performance supports dependable use beyond the evaluated dataset.

# CRediT AUTHORSHIP CONTRIBUTION STATEMENT

**Titik Misriati**: Writing – review & editing, Writing – original draft, Validation, Methodology, Conceptualization. **Riska Aryanti**: Writing – original draft, Formal analysis, Data curation, Conceptualization. **Siti Marlina**: Writing – original draft, Software, Investigation. **Sarifah Agustiani**: Writing – original draft, Software, Validation. **Henny Leidiyana**: Writing – original draft, Software, Visualization.

# 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.

# FUNDING STATEMENTS

The author(s) received no financial support for the research, authorship, and/or publication of this article.

**Copyright © 2026, The Author(s).**
This is an open access article under the CC BY-SA license
https://creativecommons.org/licenses/by-sa/4.0/

**Journal of Artificial Intelligence and Technology Information**
**Website : https://ejournal.techcart-press.com/index.php/jaiti**
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 4, December 2026
**https://doi.org/10.58602/jaiti.v4i4.454**
Page 717-729

# DECLARATION OF GENERATIVE AI AND AI-ASSISTED TECHNOLOGIES IN THE WRITING PROCESS

The authors used ChatGPT-6 to improve the writing clarity of this paper. They reviewed and edited the AI-assisted content and take full responsibility for the final publication.

# REFERENCES

[1] A. K. Mulugeta, D. P. Sharma, and A. H. Mesfin, "Deep learning for medicinal plant species classification and recognition: a systematic review," *Front. Plant Sci.*, vol. 14, no. 1286088, Jan. 2024, doi: 10.3389/fpls.2023.1286088.

[2] T. P. Tran, F. Ud Din, L. Brankovic, C. Sanin, and S. M. Hester, "Advancements in Medicinal Plant Identification Using Deep Learning Techniques: A Comprehensive Review," *Vietnam Journal of Computer Science*, vol. 13, no. 02, pp. 149–182, May 2026, doi: 10.1142/S2196888825300017.

[3] M. S. Ikrar Musyaffa, N. Yudistira, M. A. Rahman, A. H. Basori, A. B. Firdausiah Mansur, and J. Batoro, "IndoHerb: Indonesia medicinal plants recognition using transfer learning and deep learning," *Heliyon*, vol. 10, no. 23, p. e40606, Dec. 2024, doi: 10.1016/j.heliyon.2024.e40606.

[4] P. B R and N. S. Rani, "DIMPSAR: Dataset for Indian medicinal plant species analysis and recognition," *Data Brief*, vol. 49, no. 109388, p. 109388, Aug. 2023, doi: 10.1016/j.dib.2023.109388.

[5] S. Islam *et al.*, "BDMediLeaves: A leaf images dataset for Bangladeshi medicinal plants identification," *Data Brief*, vol. 50, no. 109488, p. 109488, Oct. 2023, doi: 10.1016/j.dib.2023.109488.

[6] N. Lankasena, R. N. Nugara, D. Wisumperuma, B. Seneviratne, D. Chandranimal, and K. Perera, "Misidentifications in ayurvedic medicinal plants: Convolutional neural network (CNN) to overcome identification confusions," *Comput. Biol. Med.*, vol. 183, no. 109349, p. 109349, Dec. 2024, doi: 10.1016/j.compbiomed.2024.109349.

[7] G. Li, R. Zhang, D. Qi, and H. Ni, "Plant-Leaf Recognition Based on Sample Standardization and Transfer Learning," *Applied Sciences*, vol. 14, no. 18, p. 8122, Sep. 2024, doi: 10.3390/app14188122.

[8] B. R. Pushpa, N. S. Rani, M. Chandrajith, N. Manohar, and S. S. K. Nair, "On the importance of integrating convolution features for Indian medicinal plant species classification using hierarchical machine learning approach," *Ecol. Inform.*, vol. 81, no. 102611, p. 102611, Jul. 2024, doi: 10.1016/j.ecoinf.2024.102611.

[9] S. Kapoor and A. Narayanan, "Leakage and the reproducibility crisis in machine-learning-based science," *Patterns*, vol. 4, no. 9, p. 100804, Sep. 2023, doi: 10.1016/j.patter.2023.100804.

[10] J. Bernett *et al.*, "Guiding questions to avoid data leakage in biological machine learning applications," *Nat. Methods*, vol. 21, no. 8, pp. 1444–1453, 2024, doi: 10.1038/s41592-024-02362-y.

[11] B. R. Pushpa and N. S. Rani, "Ayur-PlantNet: An unbiased light weight deep convolutional neural network for Indian Ayurvedic plant species classification," *J. Appl. Res. Med. Aromat. Plants*, vol. 34, no. 100459, p. 100459, Apr. 2023, doi: 10.1016/j.jarmap.2023.100459.

[12] S. Sharma and M. Vardhan, "AELGNet: Attention-based Enhanced Local and Global Features Network for medicinal leaf and plant classification," *Comput. Biol. Med.*, vol. 184, no. 109447, p. 109447, Jan. 2025, doi: 10.1016/j.compbiomed.2024.109447.

[13] B. R. Pushpa, S. Jyothsna, and S. Lasya, "HybNet: A hybrid deep models for medicinal plant species identification," *MethodsX*, vol. 14, no. 103126, p. 103126, Jun. 2025, doi: 10.1016/j.mex.2024.103126.

[14] C. P. Lee, K. M. Lim, Y. X. Song, and A. Alqahtani, "Plant-CNN-ViT: Plant Classification with Ensemble of Convolutional Neural Networks and Vision Transformer," *Plants*, vol. 12, no. 14, p. 2642, Jul. 2023, doi: 10.3390/plants12142642.

[15] F. Sheth, I. Chatter, M. Jasra, G. Kumar, and R. Sharma, "Herbify: an ensemble deep learning framework integrating convolutional neural networks and vision transformers for precise herb identification," *Plant Methods*, vol. 21, no. 1, p. 104, Jul. 2025, doi: 10.1186/s13007-025-01421-5.

[16] A. Lapkovskis, N. Nefedova, and A. Beikmohammadi, "Automatic fused multimodal deep learning for plant identification," *Front. Plant Sci.*, vol. 16, no. 1616020, Aug. 2025, doi: 10.3389/fpls.2025.1616020.

[17] H. M. Hama, T. Sh. Abdulsamad, and S. M. Omer, "Houseplant leaf classification system based on deep learning algorithms," *Journal of Electrical Systems and Information Technology*, vol. 11, no. 1, p. 18, Apr. 2024, doi: 10.1186/s43067-024-00141-5.

[18] A. Hosna, E. Merry, J. Gyalmo, Z. Alom, Z. Aung, and M. A. Azim, "Transfer learning: a friendly introduction," *J. Big Data*, vol. 9, no. 1, p. 102, Oct. 2022, doi: 10.1186/s40537-022-00652-w.

[19] A. Mumuni and F. Mumuni, "Data augmentation: A comprehensive survey of modern approaches," *Array*, vol. 16, no. 100258, p. 100258, Dec. 2022, doi: 10.1016/j.array.2022.100258.

**Copyright © 2026, The Author(s).**
This is an open access article under the CC BY-SA license
**https://creativecommons.org/licenses/by-sa/4.0/**

**Journal of Artificial Intelligence and Technology Information**
**Website : https://ejournal.techcart-press.com/index.php/jaiti**
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 4, December 2026
**https://doi.org/10.58602/jaiti.v4i4.454**
Page 717-729

[20] Z. Yang, R. O. Sinnott, J. Bailey, and Q. Ke, "A survey of automated data augmentation algorithms for deep learning-based image classification tasks," *Knowl. Inf. Syst.*, vol. 65, no. 7, pp. 2805–2861, Jul. 2023, doi: 10.1007/s10115-023-01853-2.

[21] S. Kapoor *et al.*, "REFORMS: Consensus-based Recommendations for Machine-learning-based Science," *Sci. Adv.*, vol. 10, no. 18, May 2024, doi: 10.1126/sciadv.adk3452.

[22] E. Theodora, "Tanaman Herbal," Kaggle. [Online]. Available: https://www.kaggle.com/datasets/ellentheodora/tanaman-herbal

[23] M. Rosenblatt, L. Tejavibulya, R. Jiang, S. Noble, and D. Scheinost, "Data leakage inflates prediction performance in connectome-based machine learning models," *Nat. Commun.*, vol. 15, no. 1, p. 1829, Feb. 2024, doi: 10.1038/s41467-024-46150-w.

[24] T.-H. Cheung and D.-Y. Yeung, "A Survey of Automated Data Augmentation for Image Classification: Learning to Compose, Mix, and Generate," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 10, pp. 13185–13205, Oct. 2024, doi: 10.1109/TNNLS.2023.3282258.

[25] T. Kumar, R. Brennan, A. Mileo, and M. Bendechache, "Image Data Augmentation Approaches: A Comprehensive Survey and Future Directions," *IEEE Access*, vol. 12, pp. 187536–187571, 2024, doi: 10.1109/ACCESS.2024.3470122.

[26] K. Alomar, H. I. Aysel, and X. Cai, "Data Augmentation in Classification and Segmentation: A Survey and New Strategies," *J. Imaging*, vol. 9, no. 2, p. 46, Feb. 2023, doi: 10.3390/jimaging9020046.

[27] R. Abdulkadirov, P. Lyakhov, and N. Nagornov, "Survey of Optimization Algorithms in Modern Neural Networks," *Mathematics*, vol. 11, no. 11, p. 2466, May 2023, doi: 10.3390/math11112466.

[28] J. Opitz, "A Closer Look at Classification Evaluation Metrics and a Critical Reflection of Common Evaluation Practice," *Trans. Assoc. Comput. Linguist.*, vol. 12, pp. 820–836, Jun. 2024, doi: 10.1162/tacl_a_00675.

**Copyright © 2026, The Author(s).**
This is an open access article under the CC BY-SA license
**https://creativecommons.org/licenses/by-sa/4.0/**