Bulletin of Electrical Engineering and Informatics
Vol. 15, No. 6, December 2026, pp. 5149~5161
ISSN: 2302-9285, DOI: 10.11591/eei.v15i6.13354

Explainable CNN benchmark for snake fruit ripeness classification using grad-CAM visualization

Brian Rizqi Paradisiaca Darnoto[1], Sulisetijono[2], Makhziah[3], Ainun Nikmati Laily[4], Arif Yustian Maulana Noor[5]

[1]Department of Informatics, Faculty of Computer Science, University of Jember, Jember, Indonesia
[2]Department of Biology, Faculty of Mathematics and Natural Sciences, Universitas Negeri Malang, Malang, Indonesia
[3]Department of Agrotechnology, Faculty of Agriculture, Universitas Pembangunan Nasional "Veteran" Jawa Timur, Surabaya, Indonesia
[4]Department of Biology Education, Faculty of Tarbiyah and Teacher Training, UIN Sayyid Ali Rahmatullah Tulungagung, Tulungagung, Indonesia
[5]Department of Socioeconomic, Faculty of Agriculture Bio-Industry and Forestry, Universitas Brawijaya, Malang, Indonesia

Article Info

Article history:

Received May 14, 2026
Revised Sep 7, 2026
Accepted Sep 12, 2026

Keywords:

Convolutional neural network
Food security
Gradient-weighted class activation mapping
Ripeness classification
Snake fruit

ABSTRACT

Snake fruit ripeness classification lacks both a standardized image dataset and a systematic multi-architecture convolutional neural network (CNNs) benchmark. This study addresses both gaps by constructing a dataset of 741 images (ripe and unripe, class-balanced in training and validation, imbalanced in test by design) through web scraping and large language model (LLM)-assisted annotation with confidence-based quality filtering, and by benchmarking five transfer learning architectures (VGG16, MobileNetV2, EfficientNetB0, DenseNet121, and ResNet50) with gradient-weighted class activation mapping (Grad-CAM) explainability on metrics relevant to field deployment, reporting results as mean ±95% confidence interval over five random seeds. DenseNet121 achieved the highest mean test accuracy (91.29%±1.67, F1: 0.9135±0.0166), closely followed by MobileNetV2 (90.82%±2.61) with the smallest model size (8.7 MB) and fastest practical inference among the mid-accuracy architectures (0.18 ms); confidence intervals overlap substantially across all five architectures, indicating no single architecture is statistically clearly superior on accuracy alone. Grad-CAM maps confirm that models attend to fruit surface texture rather than background. Dataset, trained models, and the annotation and evaluation pipeline are publicly released to support reproducible research on snake fruit quality assessment.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Brian Rizqi Paradisiaca Darnoto
Department of Informatics, Faculty of Computer Science, University of Jember
Jember, East Java, Indonesia
Email: brianrizqi@unej.ac.id

1. INTRODUCTION

Snake fruit is a commercially significant palm crop endemic to Indonesia and widely cultivated across Southeast Asia, with Java accounting for most national production [1]. Indonesia is a primary global producer, supplying both domestic and international markets, though this crop remains underutilized relative to its nutritional and commercial potential [2]. Most growers manage smallholder plots and rely on visual inspection to judge harvest readiness, a practice that produces inconsistent results across growers and seasons [3].

That inconsistency is exactly what a reliable classifier could fix, but a model returning a ripeness probability with no visual justification is difficult to trust, particularly when a harvest decision cannot be undone. Practical deployment for smallholder farmers demands more than accuracy alone: field systems need

to run on mobile devices, return results quickly, and give interpretable evidence farmers can act on [4]. Classification itself is also harder than it first appears, since ripe and unripe snake fruit share the same scale texture and shape, differing only in a gradual color and firmness shift rather than a visibly distinct defect.

Convolutional neural networks (CNNs) perform consistently well on fruit quality and disease classification across a range of agricultural crops [5]. Transfer learning reduces labeled data requirements to levels attainable in domain-specific settings, making it a practical starting point for crops with limited annotated collections [6]. Interest in agricultural deep learning has expanded steadily across multiple application domains [7]. Systematic benchmarks show that architecture choice substantially influences both accuracy and efficiency in plant image classification tasks [8]. Within this broader literature, published work specific to snake fruit is limited to two studies: Rismiyati and Azhari [9] applied a custom CNN architecture to salak images for binary quality classification and reported 81.45% accuracy and Theng *et al.* [10] classified four snake fruit varieties using 584 images and achieved 84% with ResNet50. Neither evaluates more than one architecture, addresses interpretability, or examines edge deployment feasibility. Beyond CNNs, transformer-based vision models have recently been applied directly to fruit variety classification with strong results, pointing to a complementary architecture family worth benchmarking once a larger snake fruit dataset makes such comparisons meaningful [11]. A large-scale agricultural transfer-learning benchmark demonstrates this benefit concretely across multiple crop-recognition tasks [12]. Deep learning approaches have also been applied directly to fruit freshness classification with strong reported accuracy, supporting CNNs as a viable basis for fruit-condition assessment generally [13].

Automated annotation via large language models (LLMs) with vision capabilities offers a practical alternative to manual labeling: LLM annotators have been shown to match or outperform human crowd workers on text classification tasks [14]. This capability extends to more structured labeling pipelines as well, where LLMs can perform comparably to crowdsourced annotators when given clear task instructions [15]. Vision-language models have recently been evaluated directly for fruit quality assessment in production-oriented settings, reinforcing automated annotation as a practical bridge for crops that still lack labeled collections [16]. Gradient-weighted class activation mapping (Grad-CAM) [17] addresses interpretability by generating class-discriminative heatmaps that localize image regions driving each prediction. In agricultural settings, heatmaps that activate on relevant fruit surface features rather than background artifacts give farmers verifiable grounds for acting on a model recommendation [18]. Several studies identify explainability as a prerequisite for responsible artificial intelligence (AI) deployment in precision agriculture, particularly for decisions with direct economic consequences [19]. Recent edge-deployment work outside snake fruit shows Grad-CAM can pair with a lightweight architecture on-device without sacrificing real-time performance, supporting the deployment case made here [20].

No snake fruit image dataset has been released publicly, preventing independent verification or systematic extension of either prior result [21]. Dataset availability remains a persistent bottleneck in agricultural AI research, since domain-specific datasets require expert annotation, controlled acquisition conditions, and coverage across ripeness stages that are difficult to standardize across growing regions; these constraints have historically limited reproducibility and cross-study comparison, particularly for crops with narrow geographic distributions [22]. Data augmentation has emerged as a practical mitigation strategy in such settings [23]. A recent survey of automated fruit ripeness classification confirms this gap extends well beyond snake fruit: interpretability and dataset standardization remain among the most persistent obstacles across the field, with most reported systems still evaluated on accuracy alone [24]. Missing interpretability analysis in prior snake fruit work is therefore not an isolated omission but one instance of that wider pattern, which this study is positioned to address directly.

This paper makes two contributions. First, a novel snake fruit ripeness dataset of 741 images (ripe and unripe, balanced) was constructed through systematic web scraping and LLM-assisted annotation and released publicly, so other researchers can reuse, extend, or benchmark against it directly rather than starting data collection from scratch. Second, five CNN architectures (VGG16, MobileNetV2, EfficientNetB0, DenseNet121, and ResNet50) were benchmarked with Grad-CAM explainability on deployment-relevant metrics including inference time, model size, and classification accuracy, moving snake fruit classification from a single-model accuracy report to a multi-architecture, interpretability-aware benchmark.

Significance here extends beyond snake fruit itself: model size and inference time are reported alongside accuracy because, for a smallholder farmer without reliable connectivity, a model that cannot run on an ordinary phone is not a deployable solution regardless of its accuracy. Reporting these metrics together with Grad-CAM explanations gives other agricultural AI/machine learning (ML) researchers a template for architecture selection in crop quality assessment more generally, not only for ripeness staging, since the same combination of accuracy, deployment cost, and explanation quality matters wherever a trained model's recommendation has to be trusted and acted on by someone who is not a ML practitioner. Reducing post-harvest loss through better ripeness decisions also connects this work to Sustainable Development Goal 2 on zero hunger, and, by cutting food waste

at the farm level, to Sustainable Development Goal 12 on responsible consumption and production. Section 2 describes dataset construction, the CNN benchmark protocol, and Grad-CAM methodology; section 3 presents results, interpretability analysis, and comparison with prior snake fruit studies; section 4 concludes with findings and future directions.

2. METHOD

This section describes the dataset construction pipeline, the CNN benchmark protocol, and the Grad-CAM explainability approach adopted in this study. Figure 1 illustrates this pipeline end to end, from raw image collection through LLM-assisted annotation, quality filtering, class-balanced augmentation, CNN fine-tuning, and Grad-CAM evaluation, giving a visual map of how the raw web-scraped images become the benchmark results reported in section 3.

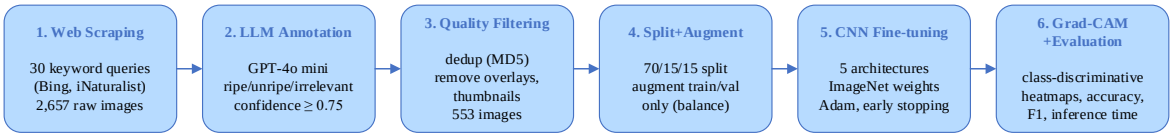


Figure 1. Overview of the dataset construction and CNN benchmarking pipeline, from image collection to Grad-CAM evaluation

2.1. Dataset construction

Prior work on snake fruit quality assessment has not produced publicly available image datasets, preventing independent replication or systematic benchmarking. A novel dataset was therefore constructed through systematic web scraping from Bing Images using 30 keyword queries in Indonesian and English, for example “buah salak matang,” “ripe snake fruit brown,” and “unripe salak palm tree,” supplemented with observations from iNaturalist under a CC-BY-NC license. This yielded 2,657 raw images, reduced to 1,109 unique images after MD5 hash-based deduplication and minimum resolution filtering (100×100 pixels).

Annotation was performed using GPT-4o mini, following evidence that LLM annotators can match or exceed human crowd-worker reliability on text and image labeling tasks [14], with a structured prompt distinguishing three categories: ripe (dark brown scaly skin, ready for harvest), unripe (light brown or greenish skin, immature on palm), and irrelevant. Annotations with confidence scores below 0.75 were discarded, and a secondary quality filtering pass removed images with large text overlays (>20% coverage), YouTube thumbnails, infographics, and photos where a person was the primary subject. Filtering left 553 original images (383 ripe, 170 unripe). Offline augmentation (horizontal and vertical flipping, 90/180/270° rotation, and brightness/contrast jitter) was applied only to unripe training and validation samples, balancing both classes at 272 training and 56 validation images each; the test set stayed entirely unaugmented for both classes, since evaluating on synthetic images would not reflect real-world generalization. This left a final dataset of 741 images: 544 training (272 per class), 112 validation (56 per class), and 85 test images (55 ripe, 30 unripe). Because only original photographs were used for testing, and unripe photographs are scarcer than ripe photographs in the source data, the test set is imbalanced by design: padding it back to 55 unripe images with synthetic samples, as an earlier version of this pipeline did, would have overstated real-world generalization, so a smaller but fully genuine unripe test set was judged the more honest choice, a point discussed further in subsection 3.6. Each image was classified via a system prompt into exactly one of three categories, ripe (dark brown to reddish-brown scaly skin, firm, ready to harvest), unripe (light brown, greenish, or yellowish skin, or immature fruit still on the palm), or irrelevant (not salak fruit, low quality, or heavily watermarked), returned as a JavaScript object notation (JSON) object with a label, confidence score, and brief reason, at a low decoding temperature (0.05) to keep outputs consistent. Prompt text plus the deduplication, quality-filtering, and corrected train/val/test splitting scripts are in the public repository referenced under data availability, to support independent replication.

2.2. Convolutional neural network architecture benchmark

Five CNN architectures were fine-tuned from ImageNet pre-trained weights for binary ripeness classification. All models receive 224×224 red, green, and blue (RGB) input normalized with ImageNet statistics (μ=[0.485, 0.456, 0.406], σ=[0.229, 0.224, 0.225]), and final classification layers were replaced with dropout (0.3 to 0.5) followed by a linear layer with two outputs. VGG16 [25] represents a sequential convolutional design. ResNet50 [26] adds residual connections. EfficientNetB0 [27] uses compound scaling of depth, width, and resolution. MobileNetV2 [28] uses depthwise separable convolution built for mobile

inference. DenseNet121 [29] connects each layer to every subsequent layer through dense connectivity. These five span distinct convolutional design families available as ImageNet transfer-learning backbones, so the benchmark reflects how design family, not only parameter count, relates to accuracy and deployment cost. Dropout was varied within this range per architecture (lower for smaller backbones such as MobileNetV2, higher for VGG16) to counter overfitting in proportion to each architecture's parameter count on a training set of only a few hundred images per class.

Training used the Adam optimizer (learning rate 1×10^{-4}, weight decay 1×10^{-4}) with cosine annealing scheduling (T_max=30) and cross-entropy loss with label smoothing (ε=0.1); early stopping with patience of five epochs prevented overfitting. Training augmentation included random cropping (256→224), horizontal and vertical flipping, color jitter, random perspective distortion, Gaussian blur, and random erasing, each chosen to simulate a specific source of field variation: rotation and flipping for inconsistent camera angle, color jitter and brightness/contrast adjustment for variable lighting, and Gaussian blur for imperfect focus. All experiments ran on an NVIDIA A100-SXM4-40GB GPU under CUDA 12.8 and PyTorch 2.6.0, measuring test accuracy, weighted F1-score, per-class precision and recall, inference time in milliseconds per image, and model size in megabytes. Reported inference time is the mean over 100 forward passes per test image on the same GPU, excluding an initial warm-up pass. To move beyond a single point estimate, each architecture was trained and evaluated across five random seeds, varying classifier-head initialization and augmentation sampling on the same fixed train/validation/test split, with results reported as mean accuracy, F1-score, and inference time, each with 95% confidence intervals rather than a single point estimate. This is a repeated-runs protocol rather than k-fold cross-validation, since the split itself stays fixed across runs, a distinction discussed further in subsection 3.6.

2.3. Gradient-weighted class activation mapping explainability

Grad-CAM [17] was applied to all five architectures to generate class-discriminative visual explanations. For each architecture, gradients of the predicted class score were computed with respect to final convolutional layer feature maps. The Grad-CAM heatmap was computed as $CAM=ReLU(\Sigma_k \alpha_k \cdot A_k)$, where $\alpha_k=(1/Z)\ \Sigma_{ij}\ (\partial y^c/\partial A^k_{ij})$ represents global average-pooled gradients and A^k denotes the k-th feature map activation; heatmaps were bilinearly upsampled to input resolution and overlaid at 45% opacity using a jet colormap. VGG16 used a modified activation-based CAM to avoid inplace operation constraints in PyTorch; every other architecture took gradients with respect to the last convolutional block before the classifier head, following the standard Grad-CAM convention of targeting the final spatial feature map prior to global pooling.

Algorithm 1. CNN fine-tuning and Grad-CAM generation

```
Input: pretrained backbone fθ, train/val/test loaders, epochs E=30, patience P=5
1: replace final layer with dropout(0.3-0.5) + linear(num_classes)
2: optimizer ← Adam(lr=1e-4, weight_decay=1e-4); scheduler ← CosineAnnealing(T_max=E)
3: best_val_acc ← 0; stale_epochs ← 0
4: for epoch = 1 to E do
5:     train fθ one pass over train_loader (cross-entropy, label smoothing=0.1)
6:     val_acc ← evaluate(fθ, val_loader)
7:     if val_acc > best_val_acc then
8:         best_val_acc ← val_acc; save fθ weights; stale_epochs ← 0
9:     else stale_epochs ← stale_epochs + 1
10:    if stale_epochs ≥ P then break
11: load best saved weights into fθ
12: for each test image x do
13:     y_pred ← fθ(x); record correctness, inference time
14:     A^k ← activations at final convolutional layer for x
15:     α_k ← global-average-pool of ∂y_pred/∂A^k
16:     CAM ← ReLU(Σ_k α_k · A^k); upsample CAM to input resolution
17:     overlay CAM on x at 45% opacity
Output: fine-tuned fθ, test accuracy/F1/inference time, Grad-CAM heatmaps
```

3. RESULTS AND DISCUSSION

This section presents the benchmark results across five architectures, followed by training convergence analysis, per-class classification performance, Grad-CAM explainability analysis, and a comparison with prior snake fruit classification studies.

3.1. Benchmark results

Table 1 summarizes test set performance (n=85: 55 ripe, 30 unripe) for all five architectures, reported as mean ±95% confidence interval over five random seeds following the repeated-runs protocol in subsection 2.2, and Figure 2 presents these results visually. DenseNet121 obtained the highest mean test accuracy at 91.29%±1.67

(F1: 0.9135±0.0166), with MobileNetV2 close behind at 90.82%±2.61. All five confidence intervals overlap substantially: DenseNet121's interval [89.62, 92.96] overlaps MobileNetV2's [88.21, 93.43], VGG16's [86.48, 92.82], and ResNet50's [84.76, 92.66], and even EfficientNetB0's interval [83.23, 89.95] falls just short of non-overlap with DenseNet121. Read honestly, these results mean no architecture is statistically clearly superior to the others in accuracy on this test set; DenseNet121 has the tightest confidence interval and so the most seed-to-seed stable result, itself a useful practical signal even where mean accuracy differences are not significant.

Table 1. CNN architecture benchmark results on snake fruit ripeness test set (n=85: 55 ripe, 30 unripe)

Architecture	Accuracy (%, mean ±95% CI)	F1-score (mean ±95% CI)	Inference (ms, mean ±95% CI)	Size (MB)	Params (M)
DenseNet121	91.29±1.67	0.9135±0.0166	0.62±0.62	27.1	7.0
MobileNetV2	90.82±2.61	0.9081±0.0265	0.18±0.14	8.7	2.2
VGG16	89.65±3.17	0.8961±0.0300	0.07±0.10	512.2	134.3
ResNet50	88.71±3.95	0.8884±0.0386	0.26±0.28	90.0	23.5
EfficientNetB0	86.59±3.36	0.8664±0.0338	0.32±0.33	15.6	4.0

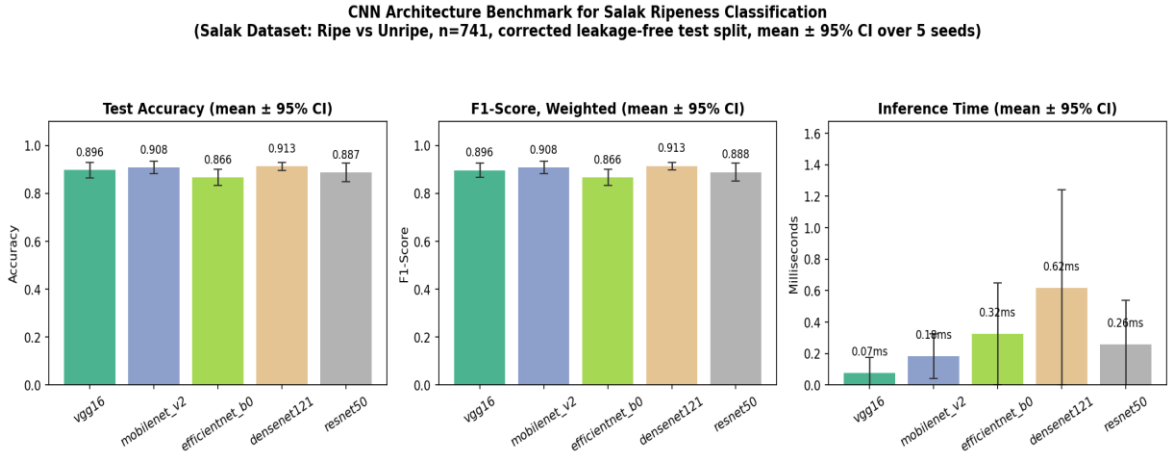


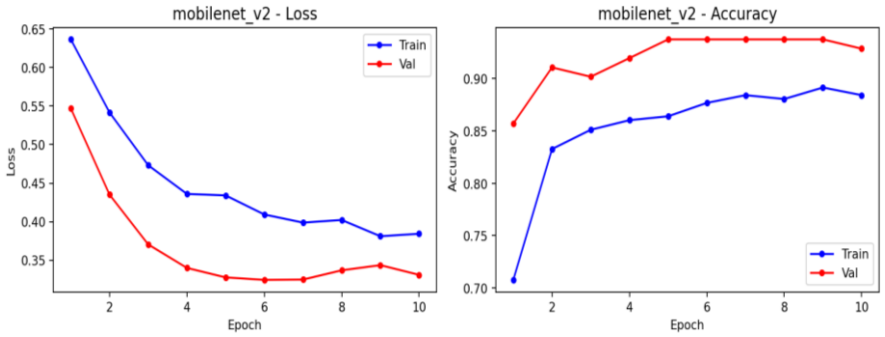
Figure 2. CNN architecture benchmark across all five architectures: (left) test accuracy, (center) weighted F1-score, (right) inference time per image

EfficientNetB0 trended lowest at 86.59%±3.36, though its interval still overlaps every other architecture's; ResNet50 reached 88.71%±3.95 and VGG16 89.65%±3.17. Because accuracy differences are not statistically clear-cut at this sample size, deployment metrics matter more to practitioners than accuracy alone. MobileNetV2 makes the strongest deployment case: it is statistically comparable in accuracy to the highest-scoring architecture while needing only 8.7 MB of storage and 0.18 ms mean inference, supporting real-time classification on commodity smartphones without cloud connectivity, critical where network infrastructure is limited [4].

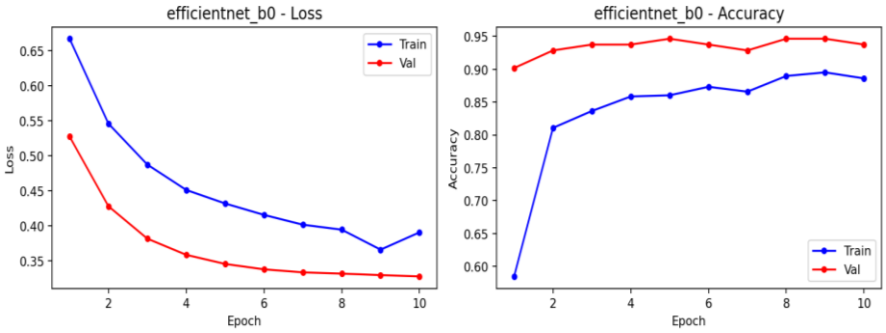
Comparing inference time against accuracy reveals a deployment trade-off distinct from the accuracy ranking. VGG16 has by far the largest footprint (512.2 MB, 134.3 million parameters) yet the fastest mean inference (0.07 ms), consistent with observations on early large-scale classification architectures [30]. DenseNet121, despite the highest mean accuracy, has the slowest and most variable inference time (0.62±0.62 ms), a cost of its dense skip-connectivity. Earlier work showed computational cost and accuracy need not trade off directly [31]. Depthwise separable convolution takes this further: MobileNetV2 achieves strong accuracy-efficiency trade-offs where full convolution would be redundant [32], matching DenseNet121's accuracy with the smallest model size and lowest, most stable inference time. MobileNetV2 remains the most practically viable architecture for resource-constrained field conditions; DenseNet121 is worth considering only where its higher, more variable inference cost is acceptable.

3.2. Training convergence

Figure 3 shows training and validation accuracy curves for MobileNetV2 in Figure 3(a) and EfficientNetB0 in Figure 3(b), both from the seed=42 run representative of the five-seed protocol described in subsection 2.2. Both architectures reach best validation accuracy by epoch 5, MobileNetV2 at 93.75% and EfficientNetB0 at 94.64%, and both stop under early stopping at epoch 10, with no evidence of overfitting despite a relatively small training set.



(a)



(b)

Figure 3. Training and validation loss and accuracy per epoch for: (a) MobileNetV2 and (b) EfficientNetB0, showing the epoch at which each model reached its best validation accuracy

MobileNetV2's validation accuracy plateaus at 93.75% from epoch 5 through epoch 9 before early stopping triggers at epoch 10, consistent with its depthwise separable convolution structure converging quickly on the texture-rich features of snake fruit skin. EfficientNetB0 reaches a higher validation accuracy (94.64%) by epoch 5 but fluctuates between epochs 6 and 10 rather than holding steady, suggesting its compound-scaling design stays sensitive to the small validation set (112 images) even after initial convergence; this fluctuation matches the wider confidence interval EfficientNetB0 shows across seeds in Table 1.

Early stopping with patience of five epochs prevented overfitting across all architectures despite the augmented training set. Fast convergence here reflects how effective ImageNet pre-trained weights are as initialization: feature detectors learned on millions of images give strong low-level representations that transfer directly to fruit surface classification. This confirms transfer learning as a substantively beneficial strategy for agricultural image classification with limited labeled data [6].

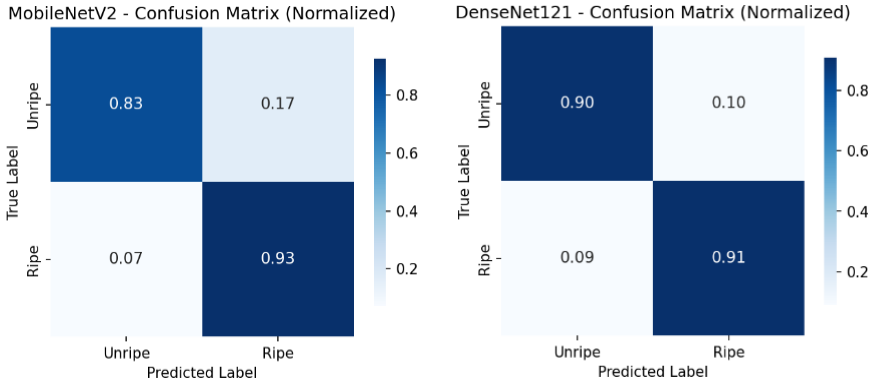
3.3. Per-class classification analysis

Figure 4 presents confusion matrices for MobileNetV2 in Figure 4(a) and DenseNet121 in Figure 4(b) on the test set. MobileNetV2 correctly classifies 51 of 55 ripe samples and 25 of 30 unripe samples in this run, while DenseNet121 correctly classifies 50 of 55 ripe samples and 27 of 30 unripe samples, giving more balanced recall across classes than MobileNetV2 despite an imbalanced test set (55 ripe, 30 unripe). Table 2 reports the corresponding precision, recall, and F1-score for each class for VGG16 and MobileNetV2, the same seed=42 run, which the next paragraph examines in detail.

Recall for unripe snake fruit is now noticeably lower than for ripe fruit in both models: VGG16 recalls 80% of unripe samples against 98% of ripe samples, and MobileNetV2 recalls 83% against 93%. Earlier, uncorrected evaluation masked this gap because part of the unripe test set was synthetically augmented; restricted to genuine unripe photographs, classification on this class turns out to be harder, consistent with training-data scarcity and a narrower visual distribution discussed in subsection 3.6. Precision shows the opposite asymmetry for VGG16, 0.96 for unripe versus 0.90 for ripe: the model rarely mislabels a ripe fruit as unripe but does miss a meaningful share of genuinely unripe fruit.

Offline augmentation applied to unripe training and validation images increases the volume of unripe examples available for learning, but it does not fully substitute for additional real unripe photographs: lower unripe recall, measured on a genuinely unaugmented test set, is the sensitivity flagged in subsection 3.6. Similar

augmentation strategies help models avoid keying on augmentation artifacts as class-discriminative features in other plant recognition tasks [33]. This makes augmentation a practical but imperfect class-balancing strategy for agricultural datasets where minority-class collection is constrained by natural availability of the fruit at a specific growth stage [23], rather than a strategy that eliminates underlying data scarcity.



(a) (b)

Figure 4. Confusion matrices on the test set (n=85): (a) MobileNetV2 and (b) DenseNet121

Table 2. Per-class classification report for VGG16 and MobileNetV2 (test set, n=85; seed=42)

Model/class	Precision	Recall	F1-score
VGG16, Ripe	0.90	0.98	0.94
VGG16, Unripe	0.96	0.80	0.87
MobileNetV2, Ripe	0.91	0.93	0.92
MobileNetV2, Unripe	0.86	0.83	0.85

3.4. Gradient-weighted class activation mapping explainability analysis

Figures 5 and 6 present Grad-CAM visualizations for MobileNetV2 and EfficientNetB0 on eight representative test samples from the seed=42 run used for Table 2 and Figure 4. Sampling in the underlying script draws the first images returned by the test data loader without stratifying by class, so all eight samples shown happen to be unripe fruit; ripe-class Grad-CAM patterns are therefore not visually represented here, a sampling limitation worth stating plainly rather than implying comprehensiveness. For unripe snake fruit, both architectures consistently activate on fruit surface texture and cluster structure, particularly scale boundaries and inter-scale crevices, the same visual cues experienced farmers use when judging harvest readiness by eye. This correspondence between model attention and agronomic expertise gives substantive grounds for trusting model recommendations, beyond a statistical accuracy figure alone [18].

Activation concentrates on lighter-colored skin regions and cluster structure against palm fiber and leaf backgrounds. Unripe fruit photographed on the palm tree shows a characteristic lighter brown or greenish-brown coloration with more loosely packed scales, and both MobileNetV2 and EfficientNetB0 identify these regions as discriminative in most samples shown. MobileNetV2 does misclassify one of the eight samples in Figure 5, predicting ripe for a genuinely unripe cluster, a concrete illustration of the lower unripe recall already quantified in Table 2 and subsection 3.3 rather than a purely qualitative claim of reliability. In several samples, heatmap activation extends to the junction between fruit cluster and palm stem, suggesting contextual palm texture contributes to the unripe classification signal, consistent with how most unripe samples in the dataset were photographed on the tree.

A notable finding is that neither architecture activates strongly on background elements such as leaves, soil, or market display surfaces, even where these occupy large image regions in these unripe samples. This background suppression suggests ImageNet pre-training combined with domain-specific fine-tuning successfully redirects attention from spurious background correlations toward task-relevant foreground features. This property matters for agricultural deployment, where image backgrounds vary substantially across farm locations, lighting conditions, and acquisition devices: models that attend to background rather than fruit surface would be expected to fail under the distribution shift conditions real-world deployment brings [5].

DenseNet121 produces more diffuse activation patterns than MobileNetV2 and EfficientNetB0 despite achieving the highest mean test accuracy in Table 1, a tension between accuracy and interpretability worth surfacing rather than resolving by accuracy alone. This traces to DenseNet's dense connectivity, where

skip connections from all preceding layers dilute spatial specificity in the final feature maps. Agricultural deep learning models need to balance accuracy with interpretability, since focused activation gives farmers a clearer basis for trusting a prediction, which is why MobileNetV2 remains the recommended default in section 4 despite DenseNet121's marginally higher, but statistically overlapping, accuracy. MobileNetV2's depthwise separable convolutions preserve per-channel spatial locality through the final feature map; the dataset bias in subsection 3.6 also affects this comparison. Grad-CAM panels in Figures 5 and 6 come from the same seed=42 run used for Figure 4 and Table 2, though, as noted above, they sample only unripe test images.

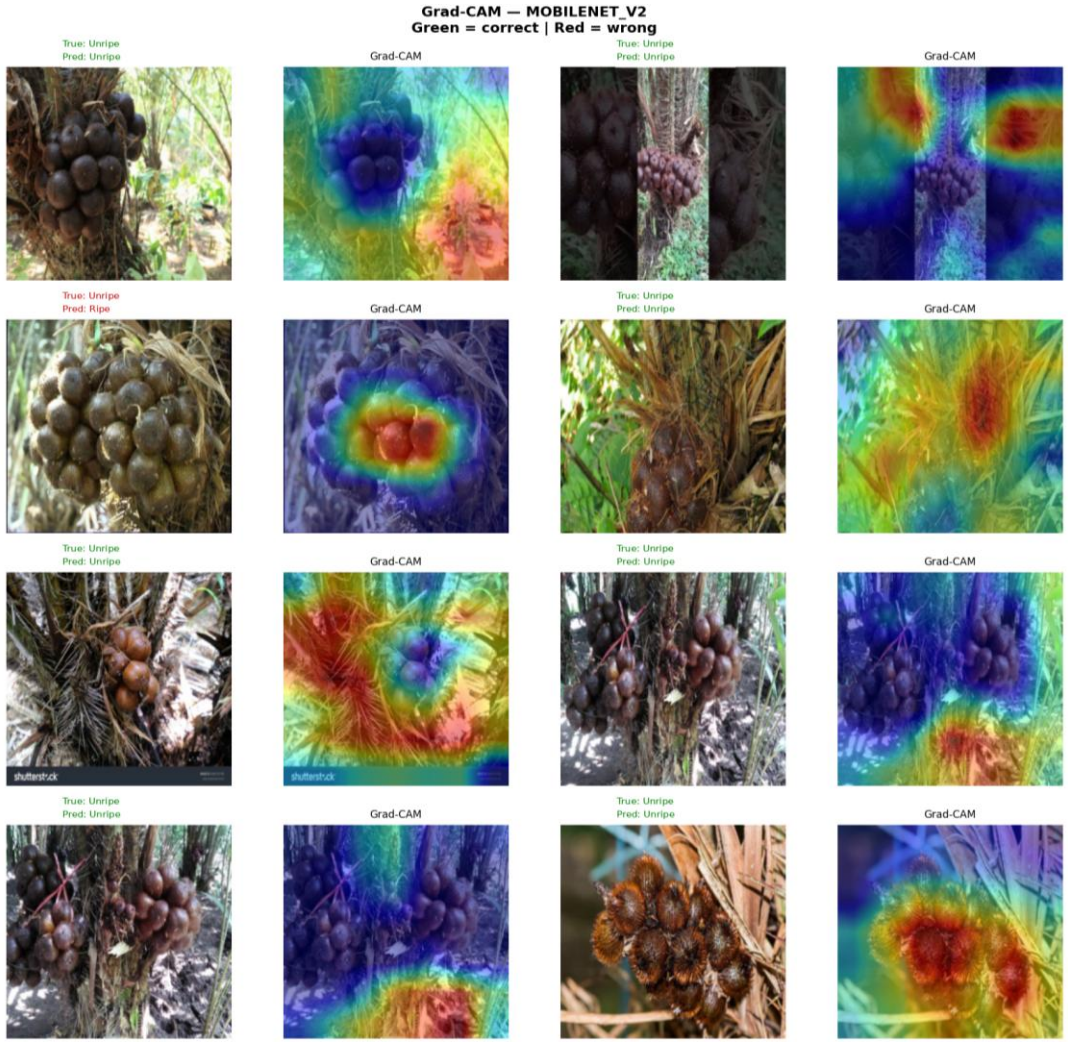


Figure 5. Grad-CAM visualizations for MobileNetV2 on eight representative unripe test samples, each row showing the input image beside its heatmap overlay (green border=correct prediction, red border=incorrect)

3.5. Comparative analysis

Table 3 situates this study within existing snake fruit classification literature. Rismiyati and Azhari [9] achieved 81.45% with a custom CNN architecture on a binary quality classification task, lower than this study's mean accuracy for every architecture benchmarked, including the lowest-performing EfficientNetB0 (86.59%±3.36). This gap is consistent with the substantial improvement in transfer learning practice since 2016: modern ImageNet pre-trained backbones fine-tuned end to end substantially outperform the smaller, non-pretrained convolutional architecture used in the earlier study, even though quality classification (good vs. damaged) in that study involves more visually distinct categories than the ripeness-based task addressed here, where ripe and unripe fruit share the same species-specific texture with only degree-of-maturation differences separating them. Theng *et al.* [10] reported 84% accuracy on a four-class variety classification task using ResNet50 on 584 images, a lower result consistent with the greater difficulty of distinguishing four morphologically similar variants.

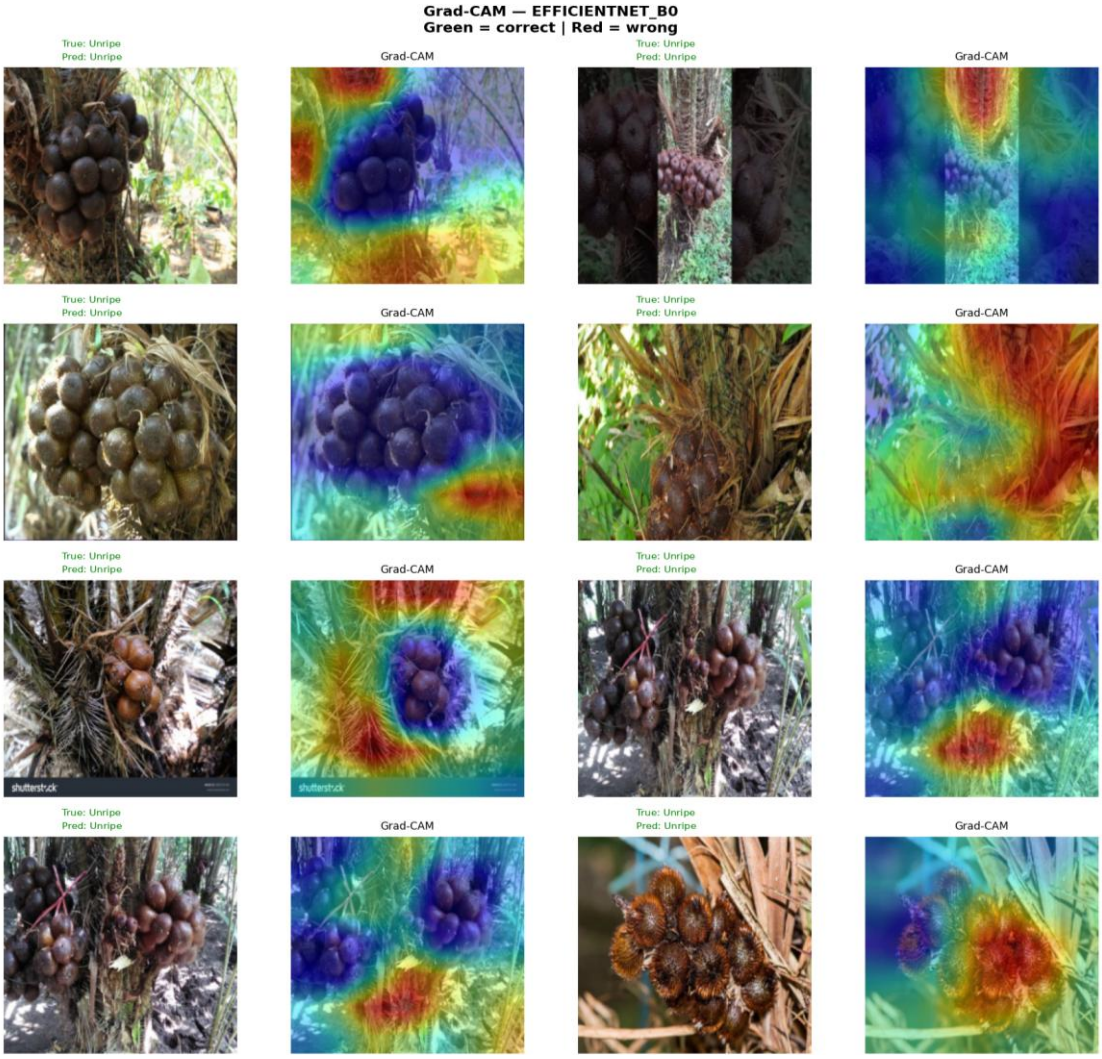


Figure 6. Grad-CAM visualizations for EfficientNetB0 on the same representative test samples as Figure 5, for direct comparison of activation patterns between architectures (green border=correct prediction, red border=incorrect)

Table 3 Comparison with prior deep learning studies on snake fruit classification

Study	Dataset	Architecture	Classes	Accuracy (%)
Rismiyati and Azhari [9]	Not reported	CNN	2 (quality)	81.45
Theng *et al.* [10]	584 images	ResNet50	4 (variety)	84.00
This study	741 images	DenseNet121	2 (ripeness)	91.29
This study	741 images	MobileNetV2	2 (ripeness)	90.82

This study extends prior literature along three dimensions neither previous work addresses. First, a five-architecture comparison reveals performance variation single-architecture studies cannot capture. Second, Grad-CAM analysis provides the first published visual explanation of discriminative features deep learning models use for snake fruit classification, establishing a baseline for interpretability assessment in this domain. Third, deployment-relevant metrics (model size, inference time) are reported alongside accuracy, enabling informed architecture selection for field deployment contexts where computational constraints bind.

Neither prior study releases a public image dataset, evaluates more than one architecture, or reports inference time and model size; both are single-architecture, single-metric benchmarks limited to accuracy. This study addresses all three gaps at once, which is why it is presented here as a benchmark rather than as a single competing model.

Results cluster within roughly five accuracy points of each other, with overlapping confidence intervals throughout subsection 3.1. DenseNet121 has the highest mean accuracy and tightest confidence

interval, while MobileNetV2 is most practical for field deployment given its statistically comparable accuracy, smallest model size, and low, stable inference time. This LLM-assisted annotation pipeline with confidence filtering produced a functional benchmark dataset where none existed publicly for this crop before. A 42% rejection rate during quality filtering points to the noise inherent in web-scraped agricultural image collections, a challenge well documented in agricultural image recognition research [34].

3.6. Limitations

Three limitations constrain how far conclusions from this study can be drawn. First, the unripe snake fruit class is limited to 170 original images across all splits (104 for training before augmentation), against 383 original ripe images; augmented samples make up 62% of unripe training data and 36% of unripe validation data, and the test set stays imbalanced (55 ripe, 30 unripe) because only original photographs were used for testing. Web-scraped images of unripe snake fruit are substantially scarcer than ripe fruit images, since commercial photography predominantly captures harvest-ready fruit. This sampling bias draws the unripe class from a narrower distribution than the ripe class, and markedly lower unripe recall, reported in subsection 3.3 once the test set was corrected to real photographs only (subsection 2.1), confirms this class is genuinely harder rather than merely under-augmented. Because most unripe training data is synthetic, reported accuracy partly measures how well augmentation approximates real unripe field variation rather than purely measuring generalization to novel real photos; this dependency should be tested, not assumed away, as the dataset grows.

Second, all images were acquired via web scraping rather than controlled field photography. Web-sourced images vary substantially in camera type, resolution, lighting conditions, background complexity, and fruit presentation context; this heterogeneity gives some robustness to domain shift, but also introduces noise a quality-filtered field dataset would avoid. A 42% rejection rate during quality filtering shows a substantial fraction of web-scraped images fail to meet minimum quality standards for ML training.

Third, the benchmark uses a fixed train/validation/test split repeated across five random seeds (subsection 2.2) rather than k-fold cross-validation, following established methodology in prior agricultural image classification studies [35]; this captures seed-to-seed variability but not variability from the split itself. Future work should address all three limitations: field image collection from demonstration plots in Lumajang and Pasuruan, dataset extension across growth stages, and k-fold cross-validation to complement the repeated-runs intervals already reported. A semi-supervised or active-learning strategy, using low-confidence predictions to guide targeted collection of real unripe images rather than further synthetic augmentation, is also promising. Benchmarking transformer-based vision models directly against these CNNs, exploring federated learning across farms without pooling raw images, and running real-world field trials would further extend this work.

4. CONCLUSION

This study benchmarks five CNN architectures for snake fruit ripeness classification and evaluates their visual explainability through Grad-CAM, reporting results as mean ±95% confidence interval over five random seeds on a corrected, leakage-free test split. DenseNet121 achieves the highest mean accuracy (91.29%±1.67), closely followed by MobileNetV2 (90.82%±2.61) with the smallest model size (8.7 MB) and a low, stable inference time (0.18 ms per image); because confidence intervals overlap substantially across all five architectures, none of these differences are statistically decisive. MobileNetV2 is therefore the recommended default for deployment rather than the architecture with the nominally highest mean accuracy: DenseNet121 is preferable only where storage and inference latency are not binding constraints, while MobileNetV2 suits cases where deployment cost matters as much as accuracy, the common case for smallholder-facing tools. Grad-CAM analysis confirms that models attend to task-relevant fruit surface features, with MobileNetV2 and EfficientNetB0 showing more localized activation than DenseNet121, a distinction that matters for deployment contexts where prediction transparency counts even when accuracy is comparable.

A novel snake fruit ripeness dataset of 741 images, together with the trained models and Grad-CAM pipeline, was constructed and released, closing the complete absence of any publicly available benchmark for this crop and giving other researchers a reusable pipeline for other palm or tropical fruit crops. Because the recommended MobileNetV2 model is small enough to run on-device, it makes a farmer-facing mobile app a realistic deployment path without cloud connectivity, offering more consistent ripeness classification than unaided visual inspection. Future work should validate model performance on images collected under field conditions and extend classification to multi-class ripeness grading aligned with commercial harvest standards. Originality here is therefore twofold: a released, reusable dataset, and a benchmark that couples deployment cost with a quantifiable account of what each architecture attends to.

ACKNOWLEDGMENTS

This work was supported by a research grant from Universitas Jember under the Riset Jatim Melaju 2026 Program, contract number 3753/DST/UN25.C1/LT/2026. The authors thank the collaborative research team from Universitas Negeri Malang, Universitas Pembangunan Nasional Veteran Jawa Timur, UIN Sayyid Ali Rahmatullah Tulungagung, and Universitas Brawijaya for domain expertise and research coordination support. In accordance with IAES generative AI policy, the authors disclose that GPT-4o mini (OpenAI, version accessed via OpenRouter API, November 2024) was used in the dataset construction phase to perform automated image annotation. Specifically, the model was prompted to classify each image into one of three categories (ripe, unripe, or irrelevant) and provide a confidence score; only annotations with confidence ≥0.75 were retained. All annotations were subsequently reviewed through a quality filtering pipeline by the authors. Generative AI was not used in the analysis or interpretation of results, the writing of this manuscript, or the creation of any figures or tables.

FUNDING INFORMATION

This research was funded by a research grant from DIPA Universitas Jember Tahun Anggaran 2026 through the Riset Jatim Melaju 2026 Program, Skema Kolaborasi Perguruan Tinggi, East Java Provincial Government, Indonesia, under contract number 3753/DST/UN25.C1/LT/2026.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Brian Rizqi Paradisiaca Darnoto	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Sulisetijono		✓				✓		✓	✓	✓	✓	✓		
Makhziah	✓		✓	✓			✓			✓	✓		✓	✓
Ainun Nikmati Laily						✓				✓	✓			
Arif Yustian Maulana Noor					✓			✓		✓				

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST

Authors state no conflict of interest.

DATA AVAILABILITY

Dataset, trained models, and source code are available on Hugging Face Hub at: https://huggingface.co/brianrizqi/salaksmart-models.

REFERENCES

[1] P. H. Čepková, M. Jágr, D. Janovská, V. Dvořáček, A. K. Kozak, and I. Viehmannová, "Comprehensive mass spectrometric analysis of snake fruit: Salak (Salacca zalacca)," *Journal of Food Quality*, pp. 1–12, Mar. 2021, doi: 10.1155/2021/6621811.

[2] P. Mazumdar, H. Pratama, S.-E. Lau, C. H. Teo, and J. A. Harikrishna, "Biology, phytochemical profile and prospects for snake fruit: An antioxidant-rich fruit of South East Asia," *Trends in Food Science & Technology*, vol. 91, pp. 147–158, Sep. 2019, doi: 10.1016/j.tifs.2019.06.017.

[3] M. Rizzo, M. Marcuzzo, A. Zangari, A. Gasparetto, and A. Albarelli, "Fruit ripeness classification: A survey," *Artificial Intelligence in Agriculture*, vol. 7, pp. 44–57, Mar. 2023, doi: 10.1016/j.aiia.2023.02.004.

[4] A. A. AlZubi and K. Galyna, "Artificial intelligence and internet of things for sustainable farming and smart agriculture," *IEEE Access*, vol. 11, pp. 78686–78692, 2023, doi: 10.1109/ACCESS.2023.3298215.

[5] I. Attri, L. K. Awasthi, T. P. Sharma, and P. Rathee, "A review of deep learning techniques used in agriculture," *Ecological Informatics*, vol. 77, Nov. 2023, doi: 10.1016/j.ecoinf.2023.102217.

[6] F. Zhuang *et al.*, "A comprehensive survey on transfer learning," in *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, Jan. 2021,

doi: 10.1109/JPROC.2020.3004555.
[7] M. Albahar, "A survey on deep learning and its impact on agriculture: challenges and opportunities," *Agriculture*, vol. 13, no. 3, Feb. 2023, doi: 10.3390/agriculture13030540.
[8] M. Shoaib *et al.*, "An advanced deep learning models-based plant disease detection: A review of recent research," *Frontiers in Plant Science*, vol. 14, Mar. 2023, doi: 10.3389/fpls.2023.1158933.
[9] R. Rismiyati and S. N. Azhari, "Convolutional neural network implementation for image-based Salak sortation," in *2016 2nd International Conference on Science and Technology-Computer (ICST)*, Yogyakarta, Indonesia, 2016, pp. 77-82, doi: 10.1109/ICSTC.2016.7877351.
[10] L. W. Theng *et al.*, "Salak image classification method based deep learning technique using two transfer learning models," in *Classification Applications with Deep Learning and Machine Learning Technologies*, 2023, pp. 67–105, doi: 10.1007/978-3-031-17576-3_4.
[11] E. Ergün, "High precision banana variety identification using vision transformer based feature extraction and support vector machine," *Scientific Reports*, vol. 15, no. 1, Mar. 2025, doi: 10.1038/s41598-025-95466-0.
[12] Z. Al Sahili and M. Awad, "The power of transfer learning in agricultural applications: AgriNet," *Frontiers in Plant Science*, vol. 13, Dec. 2022, doi: 10.3389/fpls.2022.992700.
[13] M. Mukhiddinov, A. Muminov, and J. Cho, "Improved classification approach for fruits and vegetables freshness based on deep learning," *Sensors*, vol. 22, no. 21, Oct. 2022, doi: 10.3390/s22218192.
[14] F. Gilardi, M. Alizadeh, and M. Kubli, "ChatGPT outperforms crowd workers for text-annotation tasks," in *Proceedings of the National Academy of Sciences*, vol. 120, no. 30, Jul. 2023, doi: 10.1073/pnas.2305016120.
[15] X. He *et al.*, "AnnoLLM: Making large language models to be better crowdsourced annotators," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, 2024, pp. 165–190, doi: 10.18653/v1/2024.naacl-industry.15.
[16] G. Huang, Z. Xu, L. Liu, J. Chen, Z. He, and F. Liu, "Evaluating and deploying large vision-language models for fruit quality assessment in smart agriculture systems," *Computers and Electronics in Agriculture*, vol. 238, Nov. 2025, doi: 10.1016/j.compag.2025.110806.
[17] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 336–359, Feb. 2020, doi: 10.1007/s11263-019-01228-7.
[18] L.-D. Quach, K. N. Quoc, A. N. Quynh, N. Thai-Nghe, and T. G. Nguyen, "Explainable deep learning models with gradient-weighted class activation mapping for smart agriculture," *IEEE Access*, vol. 11, pp. 83752–83762, 2023, doi: 10.1109/ACCESS.2023.3296792.
[19] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018, doi: 10.1109/ACCESS.2018.2870052.
[20] M. J. Karim, M. O. F. Goni, M. Nahiduzzaman, M. Ahsan, J. Haider, and M. Kowalski, "Enhancing agriculture through real-time grape leaf disease classification via an edge device with a lightweight CNN architecture and Grad-CAM," *Scientific Reports*, vol. 14, no. 1, Jul. 2024, doi: 10.1038/s41598-024-66989-9.
[21] W. Shafik, A. Tufail, A. Namoun, L. C. De Silva, and R. A. A. H. M. Apong, "A systematic literature review on plant disease detection: motivations, classification techniques, datasets, challenges, and future trends," *IEEE Access*, vol. 11, pp. 59174–59203, 2023, doi: 10.1109/ACCESS.2023.3284760.
[22] M. H. Saleem, J. Potgieter, and K. M. Arif, "Automation in agriculture by machine and deep learning techniques: A review of recent developments," *Precision Agriculture*, vol. 22, no. 6, pp. 2053–2091, Dec. 2021, doi: 10.1007/s11119-021-09806-x.
[23] Y. Lu, D. Chen, E. Olaniyi, and Y. Huang, "Generative adversarial networks (GANs) for image augmentation in agriculture: A systematic review," *Computers and Electronics in Agriculture*, vol. 200, Sep. 2022, doi: 10.1016/j.compag.2022.107208.
[24] Z. F. R. Ahmed, A. K. Abdalla, N. Kaur, and F. Wu, "Insights into recent developments and obstacles in automated fruit ripeness classification," *Green Technologies and Sustainability*, vol. 4, no. 2, Apr. 2026, doi: 10.1016/j.grets.2025.100302.
[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv Preprint*, Apr. 2015, doi: 10.48550/arXiv.1409.1556.
[26] M. Shafiq and Z. Gu, "Deep residual learning for image recognition: A survey," *Applied Sciences*, vol. 12, no. 18, Sep. 2022, doi: 10.3390/app12188972.
[27] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," *arXiv Preprint*, Sep. 2020, doi: 10.48550/arXiv.1905.11946.
[28] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.
[29] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 2261–2269, doi: 10.1109/CVPR.2017.243.
[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.
[31] C. Szegedy *et al.*, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.
[32] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 1800–1807, doi: 10.1109/CVPR.2017.195.
[33] C. C. Ukwuoma, Q. Zhiguang, M. B. bin Heyat, L. Ali, Z. Almaspoor, and H. N. Monday, "Recent advancements in fruit detection and classification using deep learning techniques," *Mathematical Problems in Engineering*, pp. 1–29, Jan. 2022, doi: 10.1155/2022/9210947.
[34] Y. Chen *et al.*, "Plant image recognition with deep learning: A review," *Computers and Electronics in Agriculture*, vol. 212, Sep. 2023, doi: 10.1016/j.compag.2023.108072.
[35] M. H. Saleem, J. Potgieter, and K. M. Arif, "Plant disease detection and classification by deep learning," *Plants*, vol. 8, no. 11, Oct. 2019, doi: 10.3390/plants8110468.

BIOGRAPHIES OF AUTHORS



Brian Rizqi Paradisiaca Darnoto received his doctoral degree in Informatics from Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia, in 2025. He is currently a Postdoctoral Researcher at Institut Teknologi Sepuluh Nopember and a lecturer at the University of Jember, Indonesia. His research interests include artificial intelligence, natural language processing, and software engineering. He can be contacted at email: brianrizqi@unej.ac.id.



Sulisetijono is currently an associate professor at the Department of Biology, Faculty of Mathematics and Natural Sciences, Universitas Negeri Malang, Malang, Indonesia. His research interests include tropical plant botany, plant morphology, and plant ecophysiology. He serves as host researcher in the Riset Jatim Melaju 2026 collaborative research program. He can be contacted at email: sulisetijono.fmipa@um.ac.id.



Makhziah is currently an associate professor at the Department of Agrotechnology, Faculty of Agriculture, Universitas Pembangunan Nasional "Veteran" Jawa Timur, Indonesia. Her research interests include plant breeding and plant biotechnology. She contributes as a mitra researcher in the Riset Jatim Melaju 2026 collaborative research program. She can be contacted at email: makhziah.agro@upnjatim.ac.id.



Ainun Nikmati Laily is currently a lecturer at the Biology Education, Faculty of Tarbiyah and Teacher Training, UIN Sayyid Ali Rahmatullah Tulungagung, Indonesia. Her research interests include biology education and plant botany. She contributes as a mitra researcher in the Riset Jatim Melaju 2026 collaborative research program. She can be contacted at email: ainunnikmati@uinsatu.ac.id.



Arif Yustian Maulana Noor is currently a lecturer at the Department of Socioeconomic, Faculty of Agriculture Bio-Industry and Forestry, Universitas Brawijaya, Indonesia. His research interests include discrete choice experiments, consumer preference analysis, agribusiness, and willingness-to-pay studies. He contributes as a Mitra Researcher in the Riset Jatim Melaju 2026 collaborative research program. He can be contacted at email: arifyustian@ub.ac.id.