DEVOTION
Journal of Research and Community Service
Volume 5, Number 3, March 2024, pp. 411- 428
e-ISSN: 2797-6068
https://doi.org/10.59188/devotion.v7i8.25749

Performance and Interpretability Analysis of EfficientNetB3 for Guava Leaf Disease Classification Using Score-CAM

Tia Siti Hawa
Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) IKMI Cirebon, Indonesia
Corresponding Author: tiasitihawa4@gmail.com

Keywords:
EfficientNetB3; image classification; guava leaf; transfer learning; score-CAM

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

Copyright © 2026 by Author. Published by
Green Publisher

Abstract

Background: Guava leaf image classification requires both performance evaluation and examination of the visual basis of model decisions.
Objective: This study analyses the performance of EfficientNetB3 and interprets its predictions using Score-Weighted Class Activation Mapping (Score-CAM).
Method: The GuavaLeafVision dataset covers seven classes of disease, pest damage, and healthy conditions. An audit of 1,050 images identified 189 pixel-duplicate copies, leaving 861 unique images, which were split per class into 600 training, 128 validation, and 133 test images in a single random split with one seed (42); near-duplicates were not controlled. ImageNet-pretrained EfficientNetB3 was trained through feature extraction and fine-tuning, and the model was selected by the lowest validation loss.
Results: Fine-tuning reduced the validation loss from 0.229648 to 0.110465. On the test set, the model achieved 97.74% accuracy (130/133; 95% Wilson confidence interval 93.58–99.23%), 97.75% macro-precision, 97.20% macro-recall, and 97.37% macro-F1, with three errors in 133 predictions; the lowest recall was 86.67% for Mealybug Pests. Score-CAM on nine examples showed activation on the leaf together with background or corner responses in some cases.
Conclusion: The results support internal classification and decision inspection but do not establish symptom localization or field generalization; repeated splits, near-duplicate control, and external testing are still required.

INTRODUCTION

Recognizing leaf conditions through digital images is one approach to monitoring plant health. Changes in colour, tissue damage, and surface patterns of leaves can be used as inputs for deep learning-based classification. However, visual similarity between conditions, lighting, and background can affect predictions. The review by Shoaib et al. (2025) shows that disease and pest classification has developed through a range of deep learning approaches, while its application still requires attention to data quality and image-acquisition conditions.

Guava (*Psidium guajava* L.) is an economically important tropical fruit crop. The Food and Agriculture Organization reports guava together with mango and mangosteen as one

commodity group, whose world production reached 48.4 million tonnes in 2017; guava was estimated to account for about 15% of this total, and India, the leading producer, harvested an estimated 19.7 million tonnes of the group, almost exclusively mango and guava (Altendorf, 2019). Bangladesh, where the dataset used in this study was collected, has been described as one of the largest guava-producing countries, with cultivation covering approximately 0.3 million hectares (Nobi et al., 2023).

Leaf diseases and pests threaten this production. Guava plants are susceptible to infectious leaf diseases that lead to significant crop losses, increase production costs, and disrupt the supply of fruit to national and global markets (Nobi et al., 2023), and diseases can greatly reduce both the yield and the quality of guava (Shihab et al., 2025). In a field experiment in Brazil, rust caused by *Puccinia psidii* reduced guava fruit production by around 90% in the absence of chemical control (Martins et al., 2014). Because the seven classes of GuavaLeafVision include both disease symptoms and pest damage, such as caterpillar, cutting weevil, and mealybug infestation, rapid and reliable recognition of these conditions from leaf images is relevant to timely orchard management and reduces dependence on visual inspection by experts.

The review by Pacal et al. (2024) distinguishes classification, detection, and segmentation of plant diseases as tasks with different data requirements and outputs. Demilie, (2024) also shows that the choice of method is related to the availability of data and computing resources. This study focuses on assigning a class to each image, so activation visualizations are not treated as disease segmentation results. For guava leaf classification, Doutoum et al. (2023) evaluated four pretrained CNN architectures on 1,834 images in five categories and reported the highest accuracy of 94.93% using EfficientNetB3. This finding provides an empirical basis for the choice of architecture. EfficientNet itself was developed with compound scaling, which balances the depth, width, and input resolution of the network (Tan & Le, 2019).

Alirezazadeh et al., (2023) improved deep learning-based plant disease classification by adding an attention mechanism to pretrained networks trained with limited data. Transfer learning allows pretrained representations to be exploited before they are adapted to the labels of the research dataset. The EfficientNet family has been studied on several plant targets. Devi et al. (2023) examined EfficientNetV2 for plant disease classification and pest recognition, whereas You et al. (2023) studied SimAM-EfficientNet and its robustness to adversarial attacks. Rani et al. (2025) combined VGG16, EfficientNetB0, and an attention mechanism for chili plant diseases. These studies provide context for the choice of a pretrained model, but they do not show that the variants or their results apply directly to guava leaves. Table 1 summarizes the guava studies most relevant to this work.

Table 1. Summary of related guava studies

Study	Data	Method	Interpretability	Reported result
Doutoum et al. (2023)	1,834 guava leaf images; five categories	Four pretrained CNNs compared	Not reported	EfficientNetB3 accuracy 94.93%

Study	Data	Method	Interpretability	Reported result
Zhou (2026)	Guava fruit and leaf images; five classes	Hybrid transfer-learning models	Grad-CAM	Accuracy 96%
Nobi et al. (2023)	Two guava leaf datasets (Pakistan; Bangladesh); five classes each	GLD-Det (modified MobileNet)	Grad-CAM	Accuracy 0.98 and 0.97
Ahmed et al. (2025)	3,784 guava images; three classes	ResNet-101 transfer learning; 80/10/10 split	Grad-CAM	Accuracy 98.48%
This study	GuavaLeafVision, original version; seven classes; 861 unique images after audit	EfficientNetB3: feature extraction and fine-tuning	Score-CAM	Test accuracy 97.74%; macro-F1 97.37%

Source: compiled from the cited studies and the research data.

Other guava studies have reported similarly high accuracies under different protocols. Nobi et al. (2023) proposed GLD-Det, a lightweight MobileNet-based model with Grad-CAM visualization, and reported accuracies of 0.98 and 0.97 on two guava leaf datasets from Pakistan and Bangladesh. Ahmed et al. (2025) applied ResNet-101 transfer learning to 3,784 guava images in three classes with an 80/10/10 split and reported an accuracy of 98.48%, also supported by Grad-CAM heatmaps. Public guava image datasets have also been released, such as the six-class leaf and fruit dataset of Rajbongshi et al. (2022) and the nine-class dataset of Shihab et al. (2025), although these data articles did not report classification benchmarks. In the literature search conducted for this study, no peer-reviewed study was found that reported a duplicate audit of GuavaLeafVision or evaluated a classifier on its de-duplicated original images.

High accuracy alone, however, is not sufficient to explain the quality of an evaluation. Xu et al. (2024) emphasized the importance of the characteristics and construction of datasets in plant disease recognition. Identical or highly similar images distributed across the training and test sets can make an evaluation overly optimistic. An image audit and transparent reporting of the data split are therefore needed before classification results are interpreted. Removing identical duplicates is an initial control, but it does not guarantee that all images come from different leaves.

In addition to the architecture, the representativeness of the data determines the relevance of an evaluation. Ahmad et al., (2023) discussed dataset requirements, generalization, and the development of plant diagnosis tools. Jelali (2024), in a review of tomato diseases and pests, highlighted the differences between field data and images collected under controlled conditions. Results of internal testing therefore need to be distinguished from the ability of a model to cope with variation in field conditions.

Another issue is the interpretability of model decisions. Score-CAM weights activation maps using the scores obtained from forward passes of masked inputs rather than gradients (Wang et al., 2020). Ibrahim & Shafiq, (2022) extended Score-CAM through Augmented

Score-CAM to improve the quality of visual explanations; this study uses an adaptation of Score-CAM, not that Augmented variant. In guava research, Zhou (2026) has already combined fruit and leaf image classification with Grad-CAM, so the use of visual explanations for guava is not novel in itself. The contribution of this study is limited to a documented evaluation of EfficientNetB3 on the seven classes of GuavaLeafVision, a comparison of two training stages based on validation loss, and an examination of Score-CAM on correct and incorrect predictions.

The novelty of this study is therefore stated explicitly in three points. First, it reports a hash-based audit of the original version of GuavaLeafVision, which revealed that a substantial proportion of the images were pixel duplicates concentrated in a few classes, and it evaluates the classifier only after these copies were removed so that no identical image appears in more than one subset. Second, it reports per-class performance together with confidence intervals, so that the uncertainty arising from a small test set is visible rather than hidden behind a single aggregate score. Third, it inspects Score-CAM explanations on both correct and incorrect predictions, including background and corner responses, instead of presenting only heatmaps that appear to support the model.

Ali et al., (2023) positioned the explanation of decisions as an important part of developing trustworthy AI, whereas Bodria et al., (2023) discussed the diversity of explanation methods and their quantitative evaluation. In plant diseases, Prince et al. (2024) combined a CNN-SVM with Grad-CAM visualization. These references support reporting classification performance and visual interpretation as two complementary aspects of evaluation. This study addresses two questions: how well EfficientNetB3 classifies seven classes of guava leaf images, and how Score-CAM activation patterns help to examine the model's predictions. The term disease in the title follows the scope of the dataset classification, which also includes healthy and pest-damage classes. The aim is to produce an overall and per-class evaluation, together with a visual interpretation that distinguishes the model's contributions from biological diagnosis and symptom segmentation.

METHOD

This study used a quantitative experimental design for the evaluation of classification and a descriptive analysis for visual interpretation. The research stages comprised a dataset audit, data splitting, data exploration, preprocessing, two-stage training, model selection, test evaluation, and Score-CAM, as shown in Figure 1. Natarajan et al. (2024) likewise combined fine-tuning with visual analysis to examine plant disease classification decisions. The experiment was run on Google Colab with Python 3.13.16 and TensorFlow 2.21.0, according to the notebook log. A GPU was detected, but the device type was not recorded in the available output. The seed was set to 42.

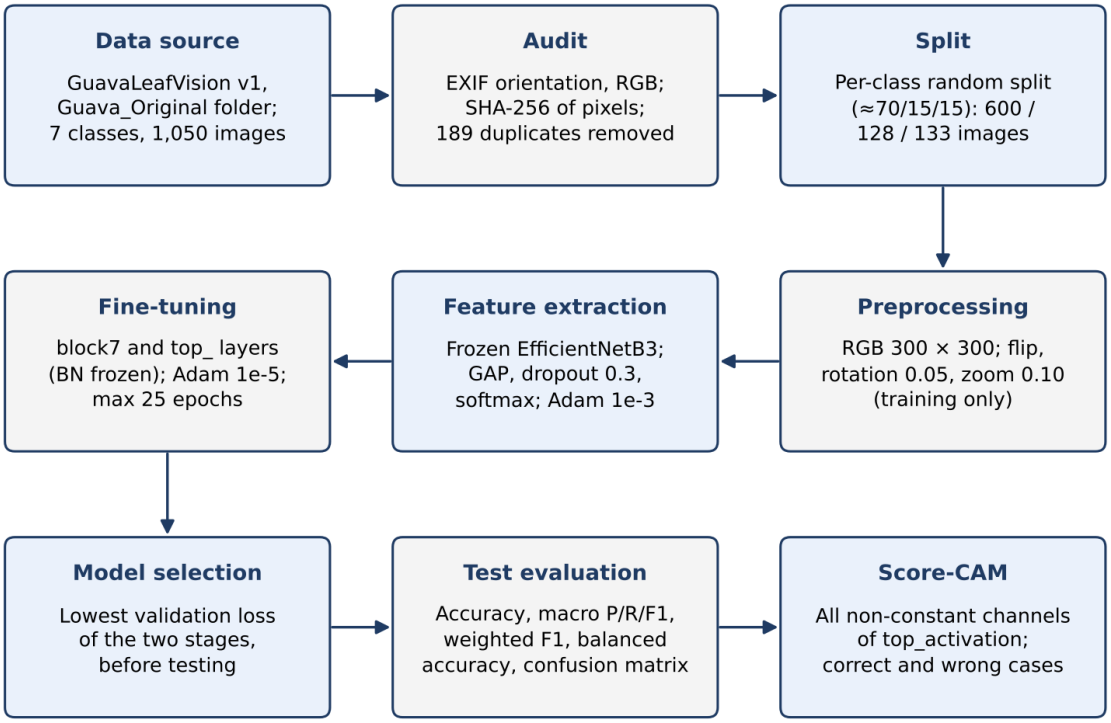


Figure 1. Research workflow

Source: author's design.

The data were obtained from version 1 of GuavaLeafVision, published by Sharfunnabi & Shakib, (2025) through Mendeley Data. According to the provider's description, the images were collected in two guava orchards in Bangladesh and are available in original and augmented versions. This study used the Guava_Original folder, which contains 1,050 images, with 150 images in each of the classes Caterpillars, Cutting Weevil, Die Back, Healthy, Mealybug Pests, Red Rust, and Yellow spot. The term original refers to the dataset version before the provider's augmentation; the images in that version had already been resized to 480 × 480 pixels. The available labels were used without re-verification by plant experts.

Each image was checked for readability, its orientation was corrected according to its EXIF data, and it was converted to RGB. SHA-256 hashes were computed from the image size and the RGB pixel bytes, so that duplicate checking was based on the pixel representation after conversion rather than only on file names or file bytes. For identical hashes within the same class, one copy was retained. Label conflicts were checked by ensuring that a single hash was not associated with more than one class. The prepared copies were saved as PNG files without deleting the sources.

The data were split randomly within each class using train_test_split, with targets of about 70% training, 15% validation, and 15% test. Because the number of members in each class is an integer, the actual proportions differ slightly. The notebook output showed that no groups.csv file was available, so the unit of splitting was the unique image, not a group of leaves or trees. No pHash grouping or StratifiedGroupKFold was used in the reported experiment. The checks ensured that identical hashes did not appear in more than one subset; possible near-duplicates and several photographs of the same leaf remain a limitation.

Because only one random split with seed 42 was evaluated, the reported metrics are point estimates from a single partition, not means and standard deviations across folds or repeated runs. Stratified k-fold cross-validation Kohavi, (1995) and perceptual-hash or embedding-based screening of near-duplicates Barz & Denzler, (2020) were not performed in the reported experiment. To make the uncertainty of the single split explicit, confidence intervals were added to the evaluation, as described below.

RGB images were resized to 300 × 300 pixels when loaded into the model. Pixel values were supplied in the range 0–255 because the Keras implementation of EfficientNet includes internal preprocessing. Augmentation was applied on the fly to the training data only, in the form of horizontal flipping, rotation with a factor of 0.05, and zoom with a factor of 0.10. In Keras, a rotation factor of 0.05 corresponds to a fraction of a full turn, that is, up to about ±18°. The validation and test data were only loaded and resized. Neither CLAHE nor class balancing through oversampling or class weights was applied.

The model consists of an EfficientNetB3 backbone with ImageNet weights and without its built-in classifier, followed by Global Average Pooling, Dropout 0.3, and a seven-unit Dense softmax layer. The model summary recorded 10,794,294 parameters, including 10,759 parameters in the classifier (1,536 × 7 weights plus 7 biases). For logits z_j, the predicted probability of class j is given by Equation (1), where C = 7. Training used the Adam optimizer, a batch size of 16, and sparse categorical cross-entropy, which for an image of true class y is given by Equation (2).

$$p_j = \exp(z_j) / \Sigma_{k=1}^{C} \exp(z_k) \quad (1)$$

$$L = -\log p_y \quad (2)$$

The first stage froze the backbone and trained the classifier; the second stage unfroze the layers whose names begin with block7 and top_, except BatchNormalization layers. The backbone was called with training=False, so the BatchNormalization statistics were not updated. The training configuration is summarized in Table 2.

Table 2. Training configuration

Component	**Feature extraction**	**Fine-tuning**
Initial learning rate	0.001	0.00001
Maximum epochs	15	25
Backbone layers trained	None	block7 and top_, except BatchNormalization
Batch size	16	16
Checkpoint selection	Minimum validation loss	Minimum validation loss
EarlyStopping	Patience 5	Patience 5
ReduceLROnPlateau	Patience 2; factor 0.5	Patience 2; factor 0.5

Source: research data.

Fine-tuning started from the best checkpoint of the first stage. ModelCheckpoint stored the checkpoint with the lowest validation loss; EarlyStopping used a patience of 5, whereas

ReduceLROnPlateau used a patience of 2 with a factor of 0.5 and a minimum learning rate of 10^{-7}. The two best checkpoints were compared on the validation set, and one model was then selected for the final evaluation on the test set. This comparison evaluates two training stages of the same architecture; the study does not compare several CNN architectures.

The maximum of 25 fine-tuning epochs was fixed before training, and the lowest validation loss was reached at the final epoch; the reported model therefore reflects this predefined training budget rather than verified convergence. No baseline architecture, such as ResNet50, MobileNetV3, EfficientNetB0, or a vision transformer, was trained on the same split. The choice of EfficientNetB3 was based on the previous evidence for guava leaves (Doutoum et al., 2023) and on the compound-scaling design of the EfficientNet family (Tan & Le, 2019), not on an internal architecture comparison.

Evaluation used accuracy, per-class precision, recall, and F1-score, macro-F1, weighted F1, balanced accuracy, and the confusion matrix. For each class c, precision, recall, and F1-score are computed from the true positives (TP), false positives (FP), and false negatives (FN) as in Equations (3)–(5). Accuracy is the proportion of correct predictions among the N test images, as in Equation (6). Macro-F1 is the unweighted mean of the F1-scores of the seven classes, as in Equation (7), and in this multi-class setting balanced accuracy equals the mean per-class recall, as in Equation (8). The order of the test data was preserved so that predictions matched the true labels.

$$\text{Precision}_c = TP_c / (TP_c + FP_c) \quad (3)$$

$$\text{Recall}_c = TP_c / (TP_c + FN_c) \quad (4)$$

$$F1_c = 2 \times \text{Precision}_c \times \text{Recall}_c / (\text{Precision}_c + \text{Recall}_c) \quad (5)$$

$$\text{Accuracy} = (\Sigma_c\, TP_c) / N \quad (6)$$

$$\text{Macro-F1} = (1/C)\, \Sigma_c\, F1_c \quad (7)$$

$$\text{Balanced accuracy} = (1/C)\, \Sigma_c\, \text{Recall}_c \quad (8)$$

Interpretation used an implementation of Score-CAM adapted for TensorFlow on the top_activation layer, which has a size of 10 × 10 with 1,536 channels. All non-constant channels were used, without Top-K selection. Each channel map A^k was resized to 300 × 300 and min–max normalized into a mask H^k, as in Equation (9), where Up denotes upsampling and s denotes min–max normalization. The mask was multiplied by the input after the internal normalization of EfficientNet. The probability score of the target class c for the masked image became the channel weight, and the weighted maps were summed, passed through a ReLU, resized, and normalized for visualization, as in Equation (10), where f^c is the softmax probability of class c.

$$H^k = s(\text{Up}(A^k)) \quad (9)$$

$$L^c = \text{ReLU}(\Sigma_k\, f^c(X \circ H^k)\, A^k) \quad (10)$$

Unlike the original formulation, in which the weight of each map is the increase in the target score relative to a baseline image (Wang et al., 2020), this implementation uses the probability score directly, without explicit baseline subtraction or an additional softmax across

channels. These details are implementation choices of this study, not a test of the equivalence of all Score-CAM variants. Masks were processed in batches of eight. For each true class, at most one correct and one incorrect prediction were selected randomly with seed 42; nine examples were available because errors occurred in only two true classes. Heatmaps explain the predicted class, including in incorrect cases. The analysis observed the position of activations relative to the leaf, the visibly damaged parts, and the background. No annotations of symptom areas were available, so IoU or Dice was not computed and the accuracy of localization was not validated.

Because each class had only 14–23 test images, the uncertainty of accuracy and of per-class recall and precision was quantified with the 95% Wilson score interval (Wilson, 1927). For k correct predictions out of n images, with $\hat{p} = k/n$ and $z = 1.96$, the interval is given by Equation (11). Wilson intervals remain within 0–100% and are better suited than the normal approximation to proportions close to 100%, which occur in several classes of this study. These intervals describe sampling uncertainty in the test set only and do not capture variation due to other splits or seeds.

$$CI_{95\%} = [\hat{p} + z^2/(2n) \pm z \sqrt{(\hat{p}(1 - \hat{p})/n + z^2/(4n^2))}] / (1 + z^2/n) \quad (11)$$

RESULTS AND DISCUSSION

Dataset Audit and Distribution

All 1,050 images could be read, and no label conflicts were found for identical hashes. The audit removed 189 pixel-duplicate copies, or 18.00% of the initial total, leaving 861 unique images. The largest reduction occurred in Cutting Weevil (59 images), followed by Mealybug Pests (55) and Caterpillars (46). Because duplication was uneven, the initially balanced distribution changed after cleaning. The actual distribution is presented in Table 3.

Table 3. Results of the data audit and image split

Class	Initial	Duplicates	Unique	Train	Validation	Test
Caterpillars	150	46	104	72	16	16
Cutting Weevil	150	59	91	63	14	14
Die Back	150	0	150	105	22	23
Healthy	150	1	149	104	22	23
Mealybug Pests	150	55	95	66	14	15
Red Rust	150	28	122	85	18	19
Yellow spot	150	0	150	105	22	23
Total	1,050	189	861	600	128	133

Source: research data.

The actual proportions were 69.69% training, 14.87% validation, and 15.45% test; the percentages may differ by 0.01 because of rounding. The training set contained 63–105 images per class, with a ratio of 1.67 between the largest and smallest classes. A size check of the 600 training images showed that all of them were 480 × 480 pixels before resizing. The imbalance in the number of images after the audit is a reason to report per-class metrics and macro-F1, although this study was not designed as an experiment on handling class imbalance.

The concentration of duplicates in three classes also suggests that the provider's original folder already contained repeated copies of some photographs. If these copies had been retained, identical images could have been distributed across the training and test sets, inflating the test results for exactly those classes. The removal step therefore changes not only the class sizes but also the reliability of the per-class estimates (Xu et al., 2024).

The effect of the audit on performance was not measured directly, because no model was trained on the uncleaned data with duplicates allowed across subsets. Studies of general image benchmarks have shown that near-duplicates shared between training and test sets can make reported accuracy overly optimistic (Barz & Denzler, 2020). A comparable training run on the uncleaned folder would therefore be required to quantify how much the audit changes the reported performance of GuavaLeafVision models.

Feature Extraction and Fine-Tuning Results

Table 4. Comparison of checkpoints on the validation set

Stage	Best epoch	Validation loss	Validation accuracy
Feature extraction	13	0.229648	91.41%
Fine-tuning	25	0.110465	96.88%

Source: research data.

Fine-tuning reduced the validation loss by 0.119183 and increased validation accuracy by 5.47 percentage points compared with the feature-extraction checkpoint (Table 4). Because the selection criterion was the lowest validation loss, the fine-tuning checkpoint at epoch 25 was selected as the final model. This result shows the benefit of adapting the final layers of the backbone on the data split used, but it does not yet show that the improvement is consistent across other seeds or splits. Because the best fine-tuning epoch coincided with the maximum of 25 epochs, further training might have reduced the validation loss slightly more.

Expressed with 95% Wilson intervals, validation accuracy was 91.41% (117/128; 85.27–95.13%) for the feature-extraction checkpoint and 96.88% (124/128; 92.24–98.78%) for the fine-tuning checkpoint. The two intervals overlap, so the gain in validation accuracy should be interpreted cautiously, whereas the reduction in validation loss was consistent with the predefined selection criterion.

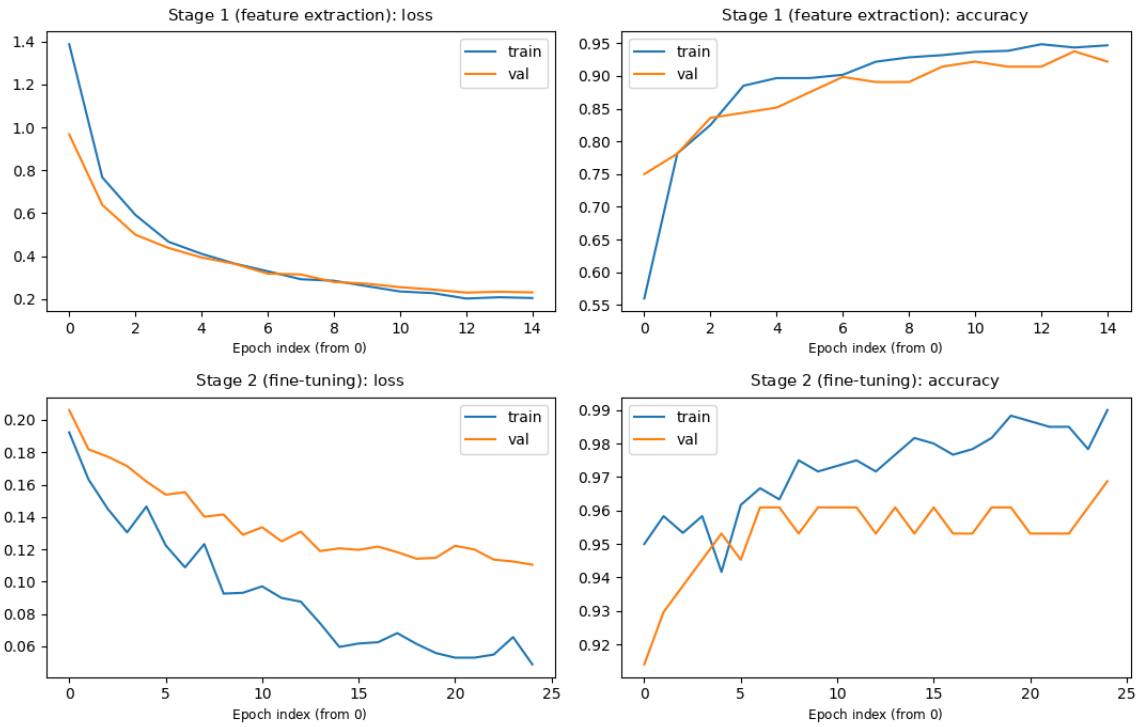


Figure 2. Loss and accuracy curves for feature extraction and fine-tuning
Source: output of the experiment notebook.

The curves in Figure 2 show decreasing training and validation loss in both stages. At the end of fine-tuning, training accuracy was 99.00% and validation accuracy 96.88%, whereas the losses were 0.0488 and 0.1105, respectively. There is a gap between training and validation, but the validation loss did not show a sustained increase up to the end of training. This pattern is not sufficient to conclude that the model is free of overfitting. In addition, the training metrics were computed while augmentation and dropout were active, so they are not fully comparable with the validation evaluation in inference mode.

Classification Performance on the Test Set

The final model produced 130 correct and three incorrect predictions on the 133 test images. Accuracy reached 97.74%, macro-precision 97.75%, macro-recall or balanced accuracy 97.20%, macro-F1 97.37%, and weighted F1 97.73%. The closeness of accuracy and macro-F1 indicates that the overall performance did not come solely from the classes with the most samples. However, each class had only 14–23 test images, so the per-class values remain sensitive to a few errors; a single error changes the recall of a class with 14 test images by more than seven percentage points.

The 95% Wilson confidence interval for test accuracy was 93.58–99.23% (130/133), indicating that the accuracy on comparable data could be several percentage points lower than the point estimate. Because all results were obtained from one random split with one seed (42), this interval reflects sampling uncertainty in the test set only and does not account for variability between splits or training runs.

Table 5. Per-class performance of the final model on the test set

Class	Precision (%)	Recall (%)	F1 (%)	Support
Caterpillars	93.75	93.75	93.75	16
Cutting Weevil	100.00	100.00	100.00	14
Die Back	100.00	100.00	100.00	23
Healthy	100.00	100.00	100.00	23
Mealybug Pests	100.00	86.67	92.86	15
Red Rust	90.48	100.00	95.00	19
Yellow spot	100.00	100.00	100.00	23
Macro average	97.75	97.20	97.37	133
Weighted average	97.89	97.74	97.73	133

Source: research data.

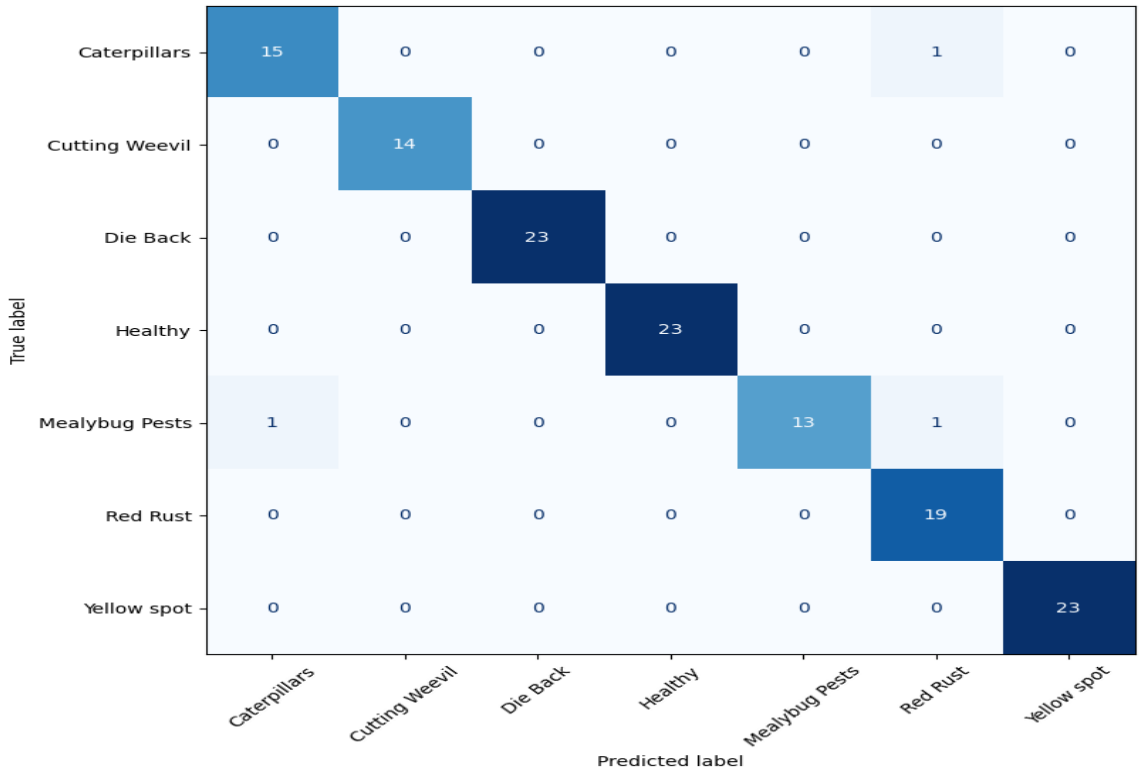


Figure 3. Confusion matrix of the final model on the 133 test images. Rows show the true class and columns the predicted class

Source: output of the experiment notebook.

The errors consisted of one Caterpillars image predicted as Red Rust, one Mealybug Pests image predicted as Caterpillars, and one Mealybug Pests image predicted as Red Rust (Figure 3). Mealybug Pests had the lowest recall, 13/15 or 86.67%, even though its precision was 100%. This means that all Mealybug Pests predictions were correct in this test, but two actual examples of the class were not recognized. Red Rust had a recall of 100% and a precision of 90.48% because all 19 actual examples were correct, but it received two positive predictions from other classes.

The per-class 95% Wilson intervals for recall were wide: 62.12–96.26% for Mealybug Pests (13/15), 71.67–98.89% for Caterpillars (15/16), 78.47–100% for Cutting Weevil (14/14), 83.18–100% for Red Rust (19/19), and 85.69–100% for Die Back, Healthy, and Yellow spot (23/23 each). The precision of Red Rust (19/21) had an interval of 71.09–97.35%. These overlapping ranges indicate that the differences between classes in Table 5 are not statistically conclusive with the present test sizes; results averaged over cross-validation folds would be needed for more stable per-class estimates.

Four classes reached 100% precision, recall, and F1 on the test subset, namely Cutting Weevil, Die Back, Healthy, and Yellow spot. These values describe success on the available samples and do not indicate that these classes can always be recognized in the field. Differences in the amount of training data also cannot be stated as a direct cause of the errors: Cutting Weevil had the smallest training set, yet all of its test samples were classified correctly. Two of the three errors were predictions of Red Rust, which suggests that some damage patterns of Caterpillars and Mealybug Pests share visual characteristics with rust lesions in this dataset.

Score-CAM Interpretation of Correct and Incorrect Predictions

The nine visualizations comprise seven correct and two incorrect predictions. For a Caterpillars image predicted correctly with a probability of 0.925, high activation was located in the centre of the leaf adjacent to perforated tissue (Figure 4). For a Cutting Weevil image predicted correctly with a probability of 0.999, activation was also located around the perforated part of the leaf. For Die Back, the main response overlapped with a leaf object that appeared brown and curled, whereas for Red Rust the response was spread over the central part of the leaf surface. These observations show visual overlap with objects or parts that appear altered, without proving that such tissue is the biological basis of the decisions.

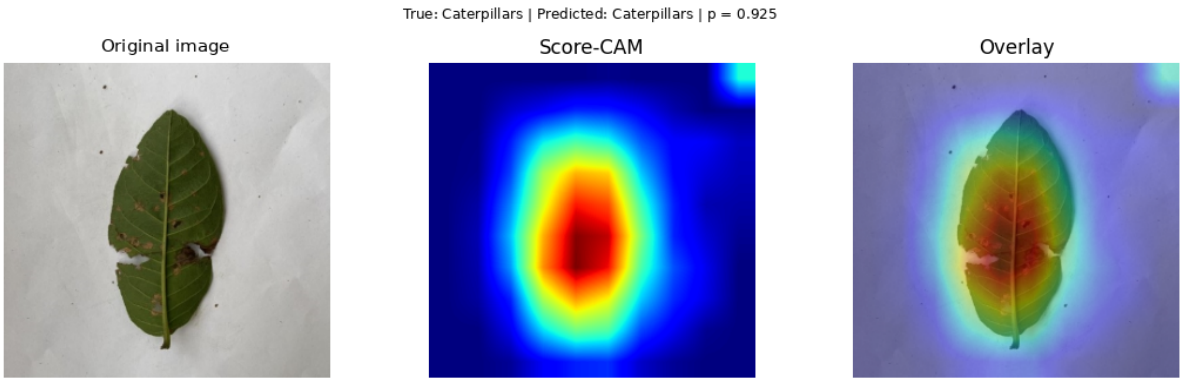


Figure 4. Caterpillars example predicted correctly with a probability of 0.925

Source: output of the experiment notebook.

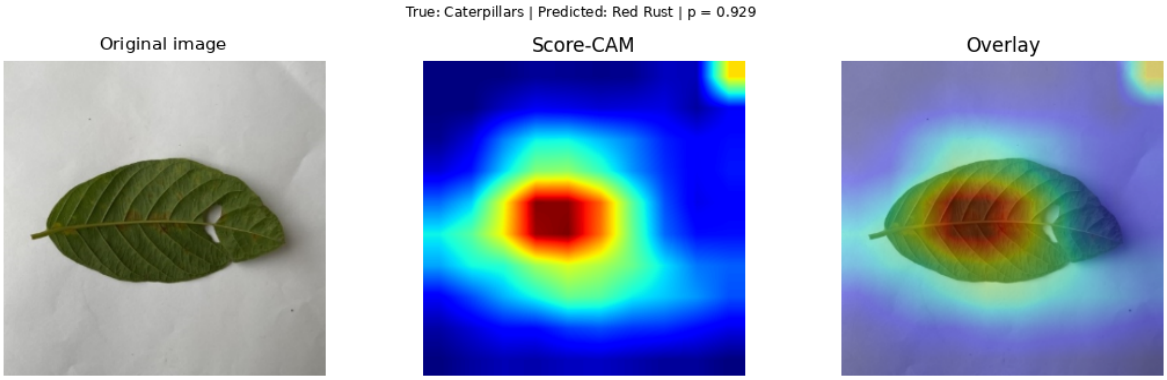


Figure 5. Caterpillars example predicted as Red Rust with a probability of 0.929
Source: output of the experiment notebook.

Figure 5 shows that an incorrect prediction can be accompanied by a high probability. The activation for the Caterpillars example misclassified as Red Rust remained in the centre of the leaf; focus on the leaf object therefore does not guarantee recognition of the correct class. The Mealybug Pests example misclassified as Red Rust had a probability of 0.397, lower than that of the incorrect Caterpillars example. The study has not tested probability calibration, so softmax values are not interpreted as a guarantee of decision reliability.

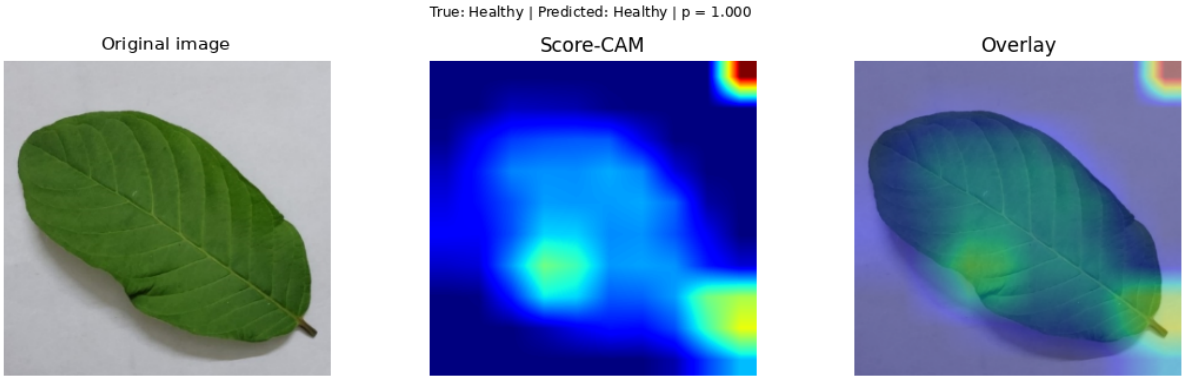


Figure 6. Healthy example with a strong response in the upper right corner outside the leaf
Source: output of the experiment notebook.

In the Healthy and Yellow spot examples, the strongest activation appeared in the upper right corner outside the leaf, whereas the leaf surface received a lower response (Figure 6). A similar corner response was also visible in several other classes. In addition, the correctly predicted Mealybug Pests example showed a broad response over the leaf surface, not only on the visible white pattern. These findings limit the claim that Score-CAM has shown the location of symptoms accurately. The background pattern may be related to features used by the model or to characteristics of the visualization implementation; its cause cannot be determined from the heatmaps alone.

Overall, Score-CAM was useful for examining mismatches between predictions and activation patterns. The heatmaps in this study are relative to each image because they are min–max normalized, so the intensity of red between examples cannot be compared as an absolute magnitude of contribution. The initial feature-map resolution of only 10 × 10 also produces

coarse maps after interpolation. Because no symptom annotations, expert inspection, or controlled perturbation tests were available, the visual results are treated as exploratory explanations, not as validation of localization. Ibrahim & Shafiq, (2022) combined qualitative and quantitative evaluation with sanity checks, which supports the need to test explanations beyond the inspection of heatmaps.

Comparison with Previous Studies and Limitations

Model development also includes the combination of features and architectures. Dai et al., (2024) developed DFN-PSAN with multi-level feature fusion and an attention mechanism, whereas Thakur et al. (2023) combined a CNN and a vision transformer for plant disease classification. Both show alternative model designs, but they were not used as experimental comparators in this study. Assessing superiority over these approaches would require training and testing on the same data split.

Doutoum et al. (2023) reported an accuracy of 94.93% on five categories of guava leaves, whereas Zhou (2026) reported an accuracy of 96% using a hybrid model on five classes of fruit and leaf images together with Grad-CAM. This study obtained 97.74% on seven classes of GuavaLeafVision after removing pixel duplicates. This figure cannot be used to conclude superiority over the two studies because the data, number of classes, test composition, and evaluation protocol differ. The similarity of the research object provides context, whereas a fair comparison of methods requires the same dataset and split.

Similar or higher accuracies have been reported by Nobi et al. (2023), 0.97–0.98 with a MobileNet-based model on two guava leaf datasets, and by Ahmed et al. (2025), 98.48% with ResNet-101 on three guava classes. These studies used different datasets, numbers of classes, and evaluation protocols, so their figures provide context rather than a direct benchmark. The accuracy of 97.74% obtained here lies within the same range, but its 95% interval (93.58–99.23%) shows that differences of a few percentage points between studies of this size should not be interpreted as differences in model quality. Like the present study, both studies used class activation maps to examine predictions, which supports the view that visual explanation is becoming a standard part of guava disease classification, although it does not replace quantitative validation.

The main limitation is that splitting was carried out at the level of unique images without metadata on leaves, trees, or acquisition sessions. Removing 189 identical copies did not eliminate all possible similar images across subsets. The study also used one seed and one split, without external testing, architectural comparators, or measurement of variation between runs. The high performance obtained should therefore be understood as the result of an internal evaluation. This limitation is consistent with the attention to dataset representativeness discussed by Xu et al. (2024).

Habaragamuwa et al. (2024) developed explanations based on disentangled variational autoencoders to show the variation of features associated with plant disease classification. This approach emphasizes that explanations can cover the characteristics of features, not only the position of activations. In this study, Score-CAM was limited to the spatial inspection of decisions; identifying the biological characteristics that cause predictions still requires further examination.

For interpretability, the use of all non-constant channels avoids Top-K selection, but it does not automatically guarantee the quality of the explanations. The implementation uses direct probability weights and masking in the normalized input space; the influence of these choices has not been compared with alternative baselines or other CAM methods. The recurring corner response needs to be examined through background masking, sensitivity tests with respect to model parameters, and expert annotation. These additional tests were not carried out in this experiment and are not claimed as research results.

Quantitative protocols exist for such tests. Model-parameter and data randomization sanity checks can reveal whether a saliency method is sensitive to what the model has learned (Adebayo et al., 2018), and insertion and deletion metrics measure how the class score changes when the most highly weighted pixels are added or removed (Petsiuk et al., 2018). Applying these protocols to the Score-CAM maps of GuavaLeafVision is a priority for future work.

CONCLUSION

This study answered both research questions. After fine-tuning, EfficientNetB3 classified seven classes of guava leaf images on the internal GuavaLeafVision test set with an accuracy of 97.74% (130/133; 95% Wilson confidence interval 93.58–99.23%) and a macro-F1 of 97.37%, while Mealybug Pests had the lowest recall (86.67%). These results come from one random split with one seed after removal of pixel duplicates only, so they represent an internal estimate rather than field performance. Score-CAM showed activation on parts of the leaf but also background and corner responses, so it supports the inspection of decisions rather than the localization of symptoms. The findings imply that duplicate auditing and uncertainty reporting should precede performance claims on this dataset. Future research should control near-duplicates, use cross-validated splits, compare baseline architectures, test external data, and validate explanations quantitatively.

FUNDING

The author states that no financial support has been received for the implementation of this research and/or the publication of this article.

CONFLICT OF INTEREST

The author states that this research was conducted without any commercial or financial relationship that can be interpreted as a potential conflict of interest.

AUTHOR CONTRIBUTION

Tia Siti Hawa: conceptualization, methodology, software, data curation, formal analysis, visualization, and writing – original draft, review, and editing.

GENERATIVE ARTIFICIAL INTELLIGENCE USAGE STATEMENT

The authors state that generative artificial intelligence (AI) was used in the preparation of this manuscript. The authors acknowledge the use of generative AI to help improve language clarity and improve the grammatical structure of manuscripts. The use is carried out in compliance with applicable ethical guidelines regarding the use and disclosure of generative AI in scientific research. Each author has made substantial intellectual contributions to this

work. The entire content of the manuscript has been carefully reviewed to ensure its accuracy, and the author is solely responsible for the integrity and final content of the manuscript

REFERENCES

Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., & Kim, B. (2018). Sanity checks for saliency maps. *Advances in Neural Information Processing Systems*, *31*, 9505–9515.

Ahmad, A., Saraswat, D., & El Gamal, A. (2023). A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools. *Smart Agricultural Technology*, *3*, 100083. https://doi.org/10.1016/j.atech.2022.100083

Ahmed, M., Ahmed, F., Naz, N. S., Mazhar, T., Khan, M. A., Khan, M. A., Ksibi, A., & Abbas, M. (2025). Automated guava disease detection using transfer learning with ResNet-101. *Food Science & Nutrition*, *13*(12), e71348. https://doi.org/10.1002/fsn3.71348

Ali, S., Abuhmed, T., El-Sappagh, S., Muhammad, K., Alonso-Moral, J. M., Confalonieri, R., Guidotti, R., Del Ser, J., Díaz-Rodríguez, N., & Herrera, F. (2023). Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence. *Information Fusion*, *99*, 101805. https://doi.org/10.1016/j.inffus.2023.101805

Alirezazadeh, P., Schirrmann, M., & Stolzenburg, F. (2023). Improving Deep Learning-based Plant Disease Classification with Attention Mechanism. *Gesunde Pflanzen*, *75*(1), 49–59. https://doi.org/10.1007/s10343-022-00796-y

Altendorf, S. (2019). *Major tropical fruits market review 2017*. FAO.

Barz, B., & Denzler, J. (2020). Do we train on test data? Purging CIFAR of near-duplicates. *Journal of Imaging*, *6*(6), 41. https://doi.org/10.3390/jimaging6060041

Bodria, F., Giannotti, F., Guidotti, R., Naretto, F., Pedreschi, D., & Rinzivillo, S. (2023). Benchmarking and survey of explanation methods for black box models. *Data Mining and Knowledge Discovery*, *37*(5), 1719–1778. https://doi.org/10.1007/s10618-023-00933-9

Dai, G., Tian, Z., Fan, J., Sunil, C. K., & Dewi, C. (2024). DFN-PSAN: Multi-level deep information feature fusion extraction network for interpretable plant disease classification. *Computers and Electronics in Agriculture*, *216*, 108481. https://doi.org/10.1016/j.compag.2023.108481

Demilie, W. B. (2024). Plant disease detection and classification techniques: a comparative study of the performances. *Journal of Big Data*, *11*(1), 5. https://doi.org/10.1186/s40537-023-00863-9

Doutoum, A. S., Eryigit, R., & Tugrul, B. (2023). Classification of guava leaf disease using deep learning. *WSEAS Transactions on Information Science and Applications*, *20*, 356–363. https://doi.org/10.37394/23209.2023.20.38

GuavaLeafVision: A Labeled Dataset of Guava Leaf Diseases for Image-Based Detection (2025). https://doi.org/10.17632/3m7gjpjm35.1

Habaragamuwa, H., Oishi, Y., & Tanaka, K. (2024). Achieving explainability for plant disease classification with disentangled variational autoencoders. *Engineering Applications of Artificial Intelligence*, *133*, 107982. https://doi.org/10.1016/j.engappai.2024.107982

Ibrahim, R., & Shafiq, M. O. (2022). Augmented Score-CAM: High resolution visual interpretations for deep neural networks. *Knowledge-Based Systems*, *252*, 109287. https://doi.org/10.1016/j.knosys.2022.109287

Jelali, M. (2024). Deep learning networks-based tomato disease and pest detection: A first review of research studies using real field datasets. *Frontiers in Plant Science*, *15*, 1493322. https://doi.org/10.3389/fpls.2024.1493322

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, *2*, 1137–1145.

Martins, M. V. V., Silveira, S. F. da, & Maffia, L. A. (2014). Guava fruit loss caused by rust. *Summa Phytopathologica*, *40*(2), 107–113. https://doi.org/10.1590/0100-5405/1904

Natarajan, S., Chakrabarti, P., & Margala, M. (2024). Robust diagnosis and meta visualizations of plant diseases through deep neural architecture with explainable AI. *Scientific Reports*, *14*, 13695. https://doi.org/10.1038/s41598-024-64601-8

Nobi, M. M. U., Rifat, M., Mridha, M. F., Alfarhood, S., Safran, M., & Che, D. (2023). GLD-Det: Guava leaf disease detection in real-time using lightweight deep learning approach based on MobileNet. *Agronomy*, *13*(9), 2240. https://doi.org/10.3390/agronomy13092240

Pacal, I., Kunduracioglu, I., Alma, M. H., Deveci, M., Kadry, S., Nedoma, J., Slany, V., & Martinek, R. (2024). A systematic review of deep learning techniques for plant diseases. *Artificial Intelligence Review*, *57*(11), 304. https://doi.org/10.1007/s10462-024-10944-7

Petsiuk, V., Das, A., & Saenko, K. (2018). RISE: Randomized input sampling for explanation of black-box models. *Proceedings of the British Machine Vision Conference 2018*.

Prince, R. H., Mamun, A. A., Peyal, H. I., Miraz, S., Nahiduzzaman, M., Khandakar, A., & Ayari, M. A. (2024). CSXAI: A lightweight 2D CNN-SVM model for detection and classification of various crop diseases with explainable AI visualization. *Frontiers in Plant Science*, *15*, 1412988. https://doi.org/10.3389/fpls.2024.1412988

Rajbongshi, A., Sazzad, S., Shakil, R., Akter, B., & Sara, U. (2022). A comprehensive guava leaves and fruits dataset for guava disease recognition. *Data in Brief*, *42*, 108174. https://doi.org/10.1016/j.dib.2022.108174

Rani, R., Bharany, S., Elkamchouchi, D. H., Ur Rehman, A., Singh, R., & Hussen, S. (2025). VGG-EffAttnNet: Hybrid deep learning model for automated chili plant disease classification using VGG16 and EfficientNetB0 with attention mechanism. *Food Science & Nutrition*, *13*(7), e70653. https://doi.org/10.1002/fsn3.70653

S. Sandhya Devi, R., R. Vijay Kumar, V., & Sivakumar, P. (2023). EfficientNetV2 Model for Plant Disease Classification and Pest Recognition. *Computer Systems Science and Engineering*, *45*(2), 2249–2263. https://doi.org/10.32604/csse.2023.032231

Shihab, M. R., Saim, N. I., Mojumdar, M. U., Raza, D. M., Siddiquee, S. M. T., Noori, S. R. H., & Chakraborty, N. R. (2025). Image dataset for classification of diseases in guava fruits and leaves. *Data in Brief*, *59*, 111378. https://doi.org/10.1016/j.dib.2025.111378

Shoaib, M., Sadeghi-Niaraki, A., Ali, F., Hussain, I., & Khalid, S. (2025). Leveraging deep learning for plant disease and pest detection: A comprehensive review and future directions. *Frontiers in Plant Science*, *16*, 1538163.

https://doi.org/10.3389/fpls.2025.1538163

Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. *Proceedings of the 36th International Conference on Machine Learning*, *97*, 6105–6114.

Thakur, P. S., Chaturvedi, S., Khanna, P., Sheorey, T., & Ojha, A. (2023). Vision transformer meets convolutional neural network for plant disease classification. *Ecological Informatics*, *77*, 102245. https://doi.org/10.1016/j.ecoinf.2023.102245

Wang, H., Wang, Z., Du, M., Yang, F., Zhang, Z., Ding, S., Mardziel, P., & Hu, X. (2020). Score-CAM: Score-weighted visual explanations for convolutional neural networks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 24–25. https://doi.org/10.1109/CVPRW50498.2020.00020

Wilson, E. B. (1927). Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, *22*(158), 209–212. https://doi.org/10.1080/01621459.1927.10502953

Xu, M., Park, J.-E., Lee, J., Yang, J., & Yoon, S. (2024). Plant disease recognition datasets in the age of deep learning: Challenges and opportunities. *Frontiers in Plant Science*, *15*, 1452551. https://doi.org/10.3389/fpls.2024.1452551

You, H., Lu, Y., & Tang, H. (2023). Plant disease classification and adversarial attack using SimAM-EfficientNet and GP-MI-FGSM. *Sustainability*, *15*(2), 1233. https://doi.org/10.3390/su15021233

Zhou, Z. (2026). Explainable deep learning-based comparative study for guava fruit and leaf disease classification: Advancing agricultural diagnostics through AI. *Frontiers in Plant Science*, *17*, 1738585. https://doi.org/10.3389/fpls.2026.1738585