Article

A Comparative Benchmark of Deep Learning Architectures for Guava Disease Classification Using Fruit and Leaf Images

Amani Tahsin Yasin [1], Sabat Abdhulhameed Ali [2] and Elham Tahsin Yasin [3, *]

[1] Nutrition and Dietetics Department, Faculty of Applied Science, Tishk International University, Erbil 44001, Iraq
[2] School of science and engineering, department of computer science and engineering, University of Kurdistan Hewler, Erbil 44001, Iraq
[3] Graduate School of Natural and Applied Sciences, Department of Computer Engineering, Faculty of Technology, Selcuk University, Konya 42100, Türkiye
* Correspondence: ilham.tahsen@gmail.com (E.T.Y.)

Abstract

The diseases of the guava plant are harmful to both yield and fruit quality. Recognizing diseases in the early stages is crucial for orchard management. Comparing deep learning studies published is a challenging task, since they often involve different data sets, preprocessing steps, training procedures, and evaluation methodologies. This study benchmarks the current state of the art in the field of deep learning based on a curated public database (521 images) categorized into five classes related to the healthy and diseased states of the guava plant. Specifically, this study compares ResNet50, DenseNet121, EfficientNet B0, EfficientNetV2 S, ConvNeXt Tiny, MobileNetV3 Large, Swin Transformer Tiny, DeiT Tiny, MaxViT Tiny, and GhostNet algorithms using a curated database of 521 images. Each algorithm was trained using the same data split, pre-processing steps, hyperparameters, and evaluation protocol. Accuracy, balanced accuracy, Macro F1 score, and Matthews Correlation Coefficient (MCC) were calculated. Additionally, confusion matrices, receiver operating characteristics and precision-recall curves were generated, and the efficacy of each algorithm was evaluated using efficiency and statistical tests. ConvNeXt Tiny and ResNet50 produced the highest observed point estimates: accuracy 98.73%, balanced accuracy 98.89%, Macro F1 score 98.78%, MCC 98.43%, and one wrong prediction for the test data set. GhostNet, EfficientNet B0, DenseNet121, and MaxViT Tiny achieved a similar result with a Macro F1 score of 98.74%. MobileNetV3 Large and GhostNet were the fastest algorithms and required 4.75 and 4.76 ms per image for processing, respectively. Overall, ConvNeXt Tiny showed a favorable relationship between predictive performance and computational requirements under the adopted fixed-partition protocol.

Keywords: Guava disease recognition; deep learning; benchmark study; computer vision; precision agriculture; plant disease classification; vision transformers

Citation: Yasin, A. T., Ali, S. A., & Yasin, E. T. (2026). A Comparative Benchmark of Deep Learning Architectures for Disease Classification in Guava Fruit and Leaf Images. *Impact in Agriculture* 2026, **2**, 3. https://doi.org/10.65500/agriculture-2026-003

Received: 23 February 2026 | Revised: 05 April 2026 | Accepted: 09 April 2026 | Published: 17 April 2026

1. Introduction

Guava (*Psidium guajava* L.) is an economically important tropical fruit crop widely cultivated across different regions of the world due to its high nutritional value and increasing market demand [1]. The fruit is particularly rich in vitamin C, dietary fibre, antioxidants,

and several bioactive compounds, which contribute to its importance for both fresh consumption and food-processing applications [2]. Over the past decade, global guava production has increased considerably, with countries such as India, Indonesia, China, Iran, and Taiwan contributing substantially to its overall production [3]. The growing economic importance of guava has encouraged its cultivation on a larger scale; however, maintaining healthy and productive orchards remains challenging because the crop is susceptible to several fungal and bacterial diseases. These diseases can affect both leaves and fruits, resulting in reduced yield, deterioration of fruit quality, and considerable economic losses, particularly when infections are not detected at an early stage [4]. Consequently, timely and reliable disease identification has become an important requirement for effective orchard management and sustainable guava production [5,6].

In conventional orchard management, guava diseases are primarily identified through visual examination of leaves and fruits by farmers or agricultural experts. Although this approach remains widely practiced, its effectiveness largely depends on the experience of the individual and the visibility of disease-specific symptoms. Several guava diseases exhibit similar visual characteristics, particularly during the early stages of infection, which makes reliable differentiation difficult even for experienced personnel [7]. Furthermore, images acquired under actual orchard conditions are affected by variations in illumination, shadows, leaf orientation, occlusion, and complex backgrounds, further increasing the difficulty of visual diagnosis [8]. Manual inspection also becomes increasingly impractical for large plantations because continuous examination of individual plants requires substantial time and human effort. Inaccurate or delayed identification may allow infections to progress and spread to surrounding plants, thereby increasing management costs and reducing overall crop productivity [9]. These limitations emphasize the need for automated approaches capable of providing rapid, objective, and reliable disease recognition under diverse cultivation conditions.

Computer vision and artificial intelligence have emerged as promising alternatives for automating plant disease recognition through the analysis of visual symptoms captured from digital images. Earlier approaches primarily relied on handcrafted colour, texture, and shape features combined with conventional machine learning classifiers. Although such methods achieved satisfactory performance under controlled experimental conditions, their generalization to complex agricultural environments was often limited because manually designed features could not adequately represent the substantial visual variations encountered in field-acquired images [10]. The emergence of deep learning, particularly convolutional neural networks (CNNs), has considerably transformed this process by enabling hierarchical feature representations to be learned directly from image data [11]. CNN-based architectures have subsequently demonstrated strong performance across a wide range of agricultural applications, including plant disease recognition, pest detection, crop classification, and fruit quality assessment [12–17]. More recently, Vision Transformers (ViTs) and hybrid deep learning architectures have received increasing attention because of their ability to capture long-range feature dependencies and complementary image representations for complex classification problems [18,19]. Collectively, these developments have strengthened the application of artificial intelligence in precision agriculture and provided new opportunities for developing scalable and reliable disease-recognition systems.

Several studies have investigated deep learning approaches for automated guava disease recognition using publicly available as well as locally collected image datasets. Transfer learning has been particularly common because pretrained architectures can provide effective feature representations even when the number of labelled agricultural images is relatively limited [20]. Architectures such as VGG, ResNet, DenseNet, MobileNet, EfficientNet, and their variants have been employed to distinguish healthy and diseased guava samples under different experimental settings [21–23]. Recent studies have further explored attention mechanisms, hybrid architectures, and transformer-based networks to improve feature representation and classification performance [9,24–26]. Although the reported results demonstrate the potential of deep learning for guava disease recognition, comparison across these studies remains difficult. Different datasets, preprocessing procedures, augmentation strategies, data-partitioning schemes, training configurations, validation protocols, and evaluation metrics are frequently employed. Consequently, the reported differences in classification performance cannot always be attributed directly to the underlying architecture, as they may also reflect differences in the experimental conditions under which the models were developed and evaluated.

Despite the considerable progress achieved in this area, several limitations remain in the existing literature.

Most studies primarily focus on introducing or evaluating an individual architecture and compare the proposed approach against a relatively small number of baseline models. Moreover, differences in preprocessing, augmentation, training configurations, and validation strategies further limit the possibility of making a fair and consistent comparison among architectures. Model evaluation is also frequently centred on overall classification accuracy, whereas other important aspects, including class-wise predictive behaviour, computational efficiency, inference speed, model complexity, statistical significance, and interpretability, receive comparatively limited attention. A model achieving the highest classification accuracy may not necessarily represent the most appropriate solution for practical deployment when computational requirements, prediction efficiency, and reliability are considered simultaneously. Therefore, a systematic benchmark conducted under identical experimental conditions is required to determine whether observed performance differences are associated with architectural characteristics rather than variations in the experimental protocol.

To address these limitations, this study presents a systematic comparative evaluation of ten representative deep learning architectures for guava disease recognition using a carefully curated public dataset comprising healthy and diseased leaf and fruit images. The evaluated architectures include conventional CNNs, computationally efficient lightweight networks, and transformer-based models, allowing different architectural paradigms to be examined within a common experimental framework. To ensure a fair and consistent comparison, all models were trained and evaluated using the same dataset partition, preprocessing pipeline, augmentation strategy, hyperparameter settings, and experimental protocol. Unlike studies that primarily assess model performance using overall classification accuracy, this study adopts a more comprehensive evaluation framework. Performance is examined using multiple classification metrics, class-wise predictive behavior, confusion matrices, receiver operating characteristic and precision-recall analyses, computational efficiency, and statistical comparisons. Model interpretability is also investigated to better understand the visual information contributing to the predictions of the evaluated architectures. This evaluation makes it possible to examine not only differences in predictive performance but also the computational and practical characteristics of each model. The resulting benchmark provides a reproducible basis for comparing modern deep learning architectures and offers practical guidance for selecting suitable models for automated guava disease recognition and related precision-agriculture applications.

2. Materials and Methods

2.1 Dataset

The experiments employed a publicly accessible guava disease image database which comprised photographs of leaves and fruits of healthy plants as well as a number of disease categories [27]. The images were captured in natural conditions and thus have various symptoms, lighting, viewpoints, and backgrounds. This variety makes the set appropriate for an evaluation which is closer to field images rather than lab conditions. Additionally, the accessibility of the database offers other researchers the opportunity to repeat the experiments or compare additional approaches by utilizing the same set of data.

The collection was inspected before model development, and 6 duplicate images were removed to avoid that duplicates would influence the evaluation. The resulted set consisted of 521 unique images distributed on 5 categories: Disease Free, Phytophthora, Red Rust, Scab, and Styler and Root. Table 1 shows the class distribution and corresponding image-type composition, and figure 1 gives representative examples.

Plant organ is completely associated with class label in this dataset. Disease Free and Red Rust contain only leaf images (213 images in total), whereas Phytophthora, Scab, and Styler and Root contain only fruit images (308 images in total). Consequently, the models may distinguish some categories partly through organ-specific characteristics rather than disease symptoms alone. Because no disease category is represented by both organ types, an organ-balanced five-class subset could not be constructed from the available dataset.

Table 1. Class and image-type composition of the cleaned dataset.

Class	Leaf, n (%)	Fruit, n (%)	Total	Image type
Disease Free	126 (100%)	0 (0%)	126	Leaf
Phytophthora	0 (0%)	114 (100%)	114	Fruit
Red Rust	87 (100%)	0 (0%)	87	Leaf
Scab	0 (0%)	100 (100%)	100	Fruit
Styler and Root	0 (0%)	94 (100%)	94	Fruit
Total	213 (40.9%)	308 (59.1%)	521	—

All images originated from a single publicly available dataset and acquisition location; therefore, no multi-source or combined dataset distribution was available. Although the images retained natural orchard backgrounds, the source dataset did not provide background-category annotations. Consequently, quantitative background-variability metrics could not be calculated retrospectively. This limitation is acknowledged when interpreting the reported in-distribution performance.

Figure 1. Sample images from the guava dataset.

2.2 Data Preparation

The experiments used the publicly available Guava Leaves and Fruits Dataset for Guava Disease Recognition. According to the dataset publication, the images were collected manually in July 2021 from a guava garden at Bangladesh Agricultural University, Mymensingh, Bangladesh, with assistance from domain experts and a plant research centre. Images were captured under orchard conditions using a Nikon D3200 digital single-lens reflex camera, and no sample pretreatment was applied before image acquisition. The original images were subsequently resized to 512 × 512 pixels by the dataset creators. Following duplicate removal, the 521 images were divided into training, validation, and held-out test subsets using one fixed stratified 70:15:15 partition with a random seed of 42. This produced 364 training images, 78 validation images, and 79 test images, as summarized in Table 2. Stratification was used to maintain approximately similar class proportions across the three subsets. The same image assignments were used for all ten architectures, allowing for a direct comparison within one partitioning. However, since the other partitions were not considered, the variability due to the sampling procedure was not accounted for. In other words, these results should be interpreted as the outcome for one specific partition rather than an average across multiple partitions.

The data was not augmented in any way. The images were resized to 224 × 224, converted to tensors, and normalized in RGB channels with ImageNet mean values of 0.485, 0.456, 0.406 and standard deviations of 0.229, 0.224, 0.225. The same preprocessing was applied to the training, validation, and test sets. No transformations were made apart from resizing and normalizing. The same procedure was used for all ten architectures.

Table 2. Dataset partition used for model development.

Subset	Images	Percentage
Training	364	70%
Validation	78	15%
Testing	79	15%
Total	521	100%

2.3 Benchmark Deep Learning Models

The benchmark was conceived to encompass the major families currently utilized for image classification tasks. It thus comprises established convolutional networks, compact models amenable to limited hardware, transformer-based architectures, and a convolution-transformer hybrid. Evaluating this broad class within one protocol enables architectural comparisons beyond the idiosyncrasies of individual experimental setups.

The ten models were ResNet50, DenseNet121, EfficientNet B0, EfficientNetV2 S, ConvNeXt Tiny, MobileNetV3 Large, Swin Transformer Tiny, DeiT Tiny, MaxViT Tiny, and GhostNet. Their designs embody residual learning, dense connectivity, compound scaling, mobile inference, hierarchical attention, data-efficient transformers, and hybrid feature extraction. As can be seen from Table 3, the models also differ significantly in parameter count and computational demand.

ImageNet-pretrained weights were used to initialize all network. The original classification head was replaced with an output layer for the five guava classes, after which all parameters were fine-tuned. Input images, learning-rate schedule, batch size, and stopping rule were kept constant. The controlled design is a critical factor that constrains the degree to which implementation choices could explain differences in performance.

The models span from a bout 3.9 million to 30.4 million trainable parameters. This range is useful in practice: larger networks might learn richer representations but require more memory and computation, whereas compact networks might be preferable for mobile or edge deployment. The benchmark thus takes into account resource usage in addition to

recognition performance. Table 3 summarizes the major network families and their respective parameter counts.

Table 3. Characteristics of the evaluated deep learning models.

Model	Architecture family	Category	Parameters (M)
ResNet50	CNN	Residual network	23.52
DenseNet121	CNN	Dense connectivity	6.96
EfficientNet B0	CNN	Compound scaling	4.01
EfficientNetV2 S	CNN	Improved EfficientNet	20.18
ConvNeXt Tiny	CNN	Modern convolution	27.82
MobileNetV3 Large	Lightweight CNN	Mobile architecture	4.21
Swin Transformer Tiny	Vision Transformer	Hierarchical transformer	27.52
DeiT Tiny	Vision Transformer	Data efficient transformer	5.53
MaxViT Tiny	Hybrid	CNN and Transformer	30.41
GhostNet	Lightweight CNN	Efficient feature generation	3.91

2.4 Experimental Setup

All experiments used the same experimental setup. The training, validation and test subsets, preprocessing steps, optimizer, learning-rate schedule, batch size and model-selection criterion were identical for all architectures. Fixing these variables enabled comparison under a common protocol. However, the observed differences may reflect interactions between each architecture and the shared training configuration and cannot be attributed exclusively to network design.

For transfer learning, each network began with ImageNet weights and received a new fully connected output layer with five neurons. All layers were then fine-tuned with AdamW. The initial learning rate was 3×10^{-4}, weight decay was 1×10^{-4}, and cosine annealing gradually reduced the learning rate during training.

Training was limited to 50 epochs with a batch size of 16. For each architecture, the checkpoint with the highest validation performance was retained and evaluated once on the held-out test set. A random seed of 42 was used for initialization and data shuffling to support reproducibility.

The implementation used PyTorch and the TIMM model library. Experiments ran on Apple Silicon, with the Metal Performance Shaders backend providing hardware acceleration. Table 4 lists the full configuration.

A common optimization configuration was used for all architectures to examine their behavior under one fixed training protocol. This design controls variation in the implemented training settings but does not provide architecture-specific optimization or estimate the maximum attainable performance of each model. As CNNs, transformers, and hybrids may react dissimilarly to variations in learning rate, weight decay, and schedule choice, one should interpret these results as fixed-configuration comparisons, rather than an inherently fair assessment of optimally tuned models.

Table 4. Experimental configuration used for model training.

Parameter	Value
Dataset split	70% Training, 15% Validation, 15% Testing
Input image size	224 × 224 pixels
Pretrained weights	ImageNet
Optimizer	AdamW
Initial learning rate	3×10^{-4}
Weight decay	1×10^{-4}
Learning rate scheduler	Cosine Annealing
Maximum epochs	50
Batch size	16
Random seed	42
Number of classes	5
Framework	PyTorch
Model library	TIMM
Hardware acceleration	Apple Metal Performance Shaders (MPS)

2.5 Evaluation Metrics

No individual metric was sufficient to characterize model performance, thus the evaluation combined overall correctness, class balance, agreement, error patterns, probability discrimination, and computational cost. This was especially important as the five disease categories do not have an equal number of images.

The reported classification metrics were accuracy, balanced accuracy, macro-averaged precision, macro-averaged recall, macro-averaged F1 score, Matthews Correlation Coefficient (MCC), and Cohen's Kappa. Accuracy represents the proportion of correctly classified images across the complete test set. Balanced accuracy is

calculated as the unweighted mean of the recall values obtained for the individual classes. Macro precision, macro recall, and macro F1 are calculated independently for each class and then averaged by assigning equal importance to every class, irrespective of class frequency. Therefore, these macro-averaged measures are not frequency-weighted. MCC and Cohen's Kappa provide complementary measures of prediction quality and agreement while accounting for the multiclass confusion structure and agreement expected by chance, respectively.

Confusion matrices were further analyzed to identify errors made by each model and to reveal whether specific disease pairs were consistently confused. Class-wise F1 scores were calculated to better understand the performance of the model for each category. In addition, receiver operating characteristic and precision-recall curves were also estimated from the predicted probabilities to evaluate the discrimination and confidence of classifications.

The efficiency was evaluated using the number of trainable parameters, the checkpoint size, the mean inference time per image, and the throughput in images per second. These metrics can indicate if a slight improvement of the recognition quality requires a significant increase of the model's size or slowdown in processing speed, which could be especially important for limited hardware capacity.

The pairwise comparisons of model predictions were made using the two-sided exact McNemar test. This test variant assesses discordant predictions for each pair of models directly, without using an asymptotic chi-square approximation. Since the number of comparisons is 45 (each of the 10 models compared with 9 others), the p-values for these pairwise tests were adjusted using the Holm procedure to control the family-wise error rate. The statistical significance of the pair-wise comparison was considered at the level of ($p<0.05$). Given the small size of the test set and the small number of errors obtained, results not significant at the adjusted level were interpreted as insufficient evidence of a difference rather than evidence of predictive equivalence. The Holm correction was used for multiple comparisons. The ranking of models was also done in terms of the main performance measures to provide a concise overview of the comparative performance. The equations (1)-(6) define the basic measures.

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

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

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

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

Balanced Accuracy $$BACC = \frac{1}{C}\sum_{i=1}^{C} Recall_i \quad (5)$$

Matthews Correlation Coefficient $$MCC = \frac{cs - \sum_k p_k t_k}{\sqrt{(s^2 - \sum_k p_k^2)(s^2 - \sum_k t_k^2)}} \quad (6)$$

3. Results and Discussion

3.1 Learning Behavior of the Benchmark Models

Figure 2 below represents the validation accuracy and loss during the process of training. Despite the difference in the design of the presented models, the changes to their accuracy and loss function follow a similar pattern. Accordingly, it can be stated that the majority of the changes happen before the 10th or 20th epoch, after which all the lines indicate the point of saturation. The learning curves indicate that all models could be trained under the selected common configuration. However, convergence does not establish that these hyperparameters were optimal for every architecture. Because architecture-specific tuning was not conducted, the results represent performance under one fixed training configuration.

Most of the CNN and transformer models under consideration exceeded the 95% threshold of validation accuracy quite early. The accuracy of ResNet50, DenseNet121, EfficientNet B0, ConvNeXt Tiny, MobileNetV3 Large, MaxViT Tiny, and GhostNet consistently increased at a relatively stable pace during the early epochs. DeiT Tiny demonstrated a more variable accuracy at the beginning and took more epochs to stabilize. Swin Transformer Tiny and EfficientNetV2 S also exhibited some degree of variance at the start but eventually converged to a stable accuracy. The loss curves also display a comparable pattern. The validation loss decreased substantially at the initial stages of the process and then gradually slowed down. Moreover, after the twentieth epoch, improvements became barely noticeable, suggesting that the algorithm has learned the majority of the significant visual patterns. Most importantly, the validation loss did not demonstrate a significant increase towards the end of the training stage, which implies that, within this paradigm, there was no severe overfitting.

Thus, each architecture reached a stable solution before the 50-epoch limit, although convergence speed varied between models. This confirms that the models could be trained under the shared configuration but does not demonstrate that the selected hyperparameters were optimal for every architecture. The subsequent comparisons should therefore be interpreted as results obtained under one fixed training configuration.

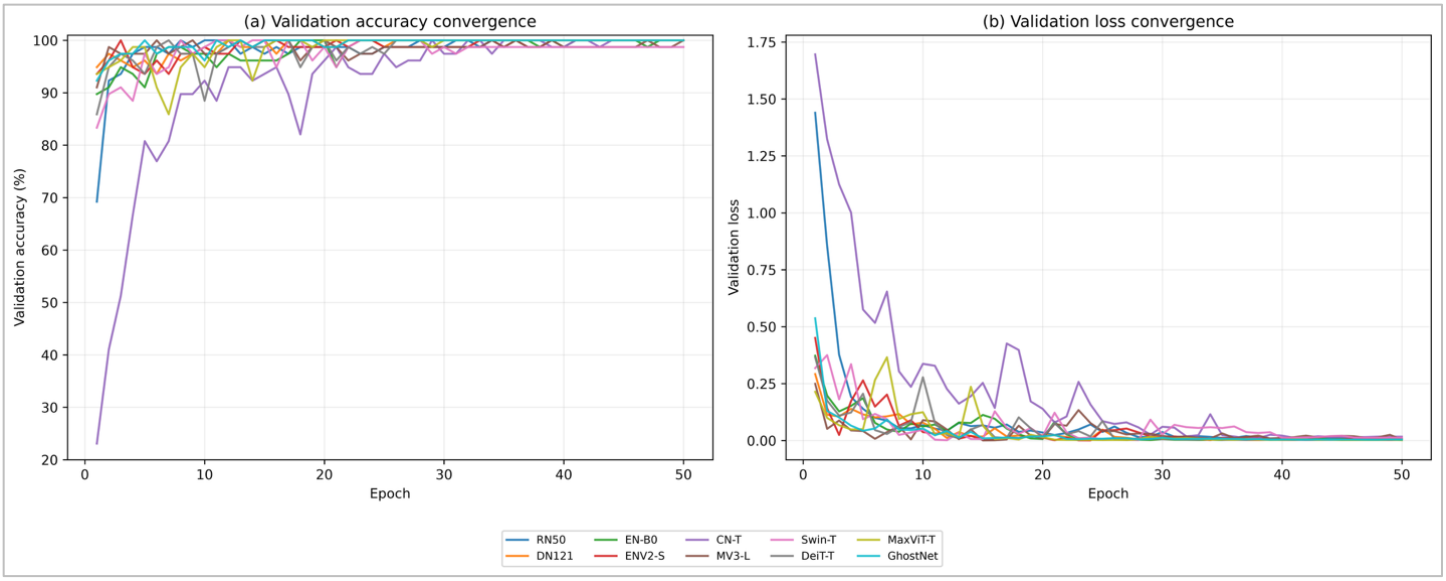

Figure 2. Learning curves of all models

3.2 Overall Benchmark Performance

Table 5 and Figure 3 summarize the performance of the ten architectures on the held-out test set. All models produced high point estimates under the adopted fixed-partition protocol, with observed accuracies ranging from 97.47% to 98.73%. However, the test set contained only 79 images, and the difference between these two accuracy levels corresponded to one prediction. The numerical ordering should therefore be interpreted cautiously.

ConvNeXt-Tiny, ResNet50, GhostNet, EfficientNet-B0, DenseNet121, and MaxViT-Tiny correctly classified 78 of the 79 test images, resulting in an accuracy of 98.73% and a 95% exact binomial confidence interval of 93.15–99.97%. EfficientNetV2-S, Swin-Tiny, MobileNetV3-Large, and DeiT-Tiny correctly classified 77 images, producing an accuracy of 97.47% and a 95% confidence interval of 91.15–99.69%. The substantial overlap between these intervals suggests that the test set does not contain enough information to distinguish between the two model groups in terms of accuracy.

ConvNeXt-Tiny and ResNet50 achieved the highest point estimates for balanced accuracy, Macro F1, and MCC. However, six architectures correctly classified 78 of the 79 test images, while the remaining four correctly classified 77 images. Therefore, the difference between the two observed accuracy levels corresponds to only one prediction. Given the small test set, overlapping confidence intervals, and nonsignificant McNemar comparisons, this numerical ordering should be interpreted as descriptive evidence for the selected partition rather than evidence of meaningful or stable architectural superiority.

Figure 3 compares Macro F1, MCC, and error count. Since Macro F1 and MCC are nearly identical, a large accuracy was not achieved by favoring one class. Six architectures made only one prediction error, while the other four made two errors each. Given the small observed differences between the approaches, accuracy should not be the sole selection criterion. Memory consumption, latency, throughput, and parameter count are more relevant evaluation metrics than accuracy, which are discussed below.

Since the held-out test set only contains 79 images, the accuracy estimates obtained with the architectures should be taken with caution. For this reason, two-sided 95% exact binomial confidence intervals were calculated for each model using the Clopper–Pearson method. They represent the range of possible values for the accuracy metric given the observed number of positive samples. The confidence intervals were used instead of the point-estimate accuracy scores due to the relatively small sample size (n=79), and the values were adjusted with the absolute error count since a 1 prediction difference equals approximately 1.27% in this case.

Accuracy confidence intervals are two-sided 95% exact binomial intervals calculated using the Clopper–Pearson method. Since the test set contained 79 images, one prediction is on average equal to 1.27 percentage points.

Table 5. Overall performance comparison of the benchmark deep learning models.

Model	Accuracy	Accuracy, 95% CI	Balanced Accuracy	Macro F1	MCC	Errors
ConvNeXt-Tiny	98.73	93.15–99.97	98.89	98.78	98.43	1
ResNet50	98.73	93.15–99.97	98.89	98.78	98.43	1
GhostNet	98.73	93.15–99.97	98.89	98.74	98.43	1
EfficientNet-B0	98.73	93.15–99.97	98.89	98.74	98.43	1
DenseNet121	98.73	93.15–99.97	98.89	98.74	98.43	1
MaxViT-Tiny	98.73	93.15–99.97	98.89	98.74	98.43	1
EfficientNetV2-S	97.47	91.15–99.69	97.78	97.49	96.90	2
Swin-Tiny	97.47	91.15–99.69	97.78	97.49	96.90	2
MobileNetV3-Large	97.47	91.15–99.69	97.46	97.46	96.82	2
DeiT-Tiny	97.47	91.15–99.69	97.35	97.43	96.86	2

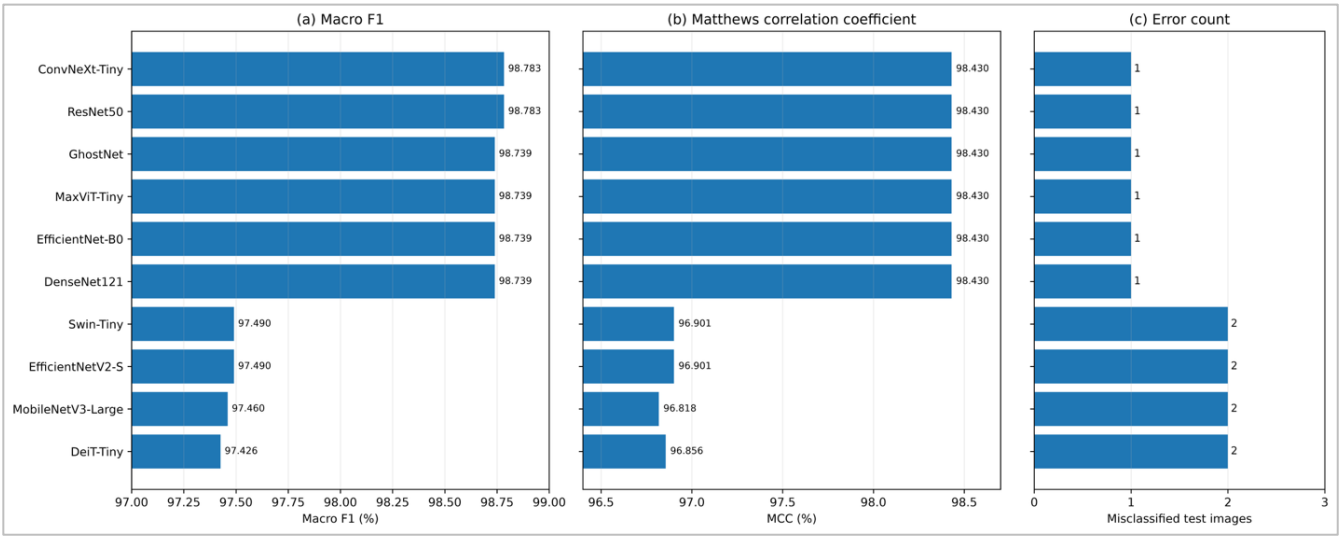

Figure 3. Comparison of (a) Macro F1 score, (b) Matthews Correlation Coefficient (MCC), and (c) misclassification count across the benchmark models.

3.3 Class-wise Performance Analysis

Class-level F1 scores are displayed in Figure 4. Disease Free and Red Rust were recognized most consistently, and most architectures reached 100% F1 for those categories. Scab was also identified reliably, although the scores for ResNet50 and ConvNeXt Tiny turned out to be slightly lower than those of the other models.

Phytophthora and Styler and Root were more challenging. F1 for Phytophthora ranged from 94.1% to 97.1%, while the lowest class specific values occurred for Styler and Root, particularly with MobileNetV3 Large and EfficientNetV2 S, even though the variation remained limited. The overall differences appear to arise from a few difficult images rather than from a consistent inability to recognize an entire class.

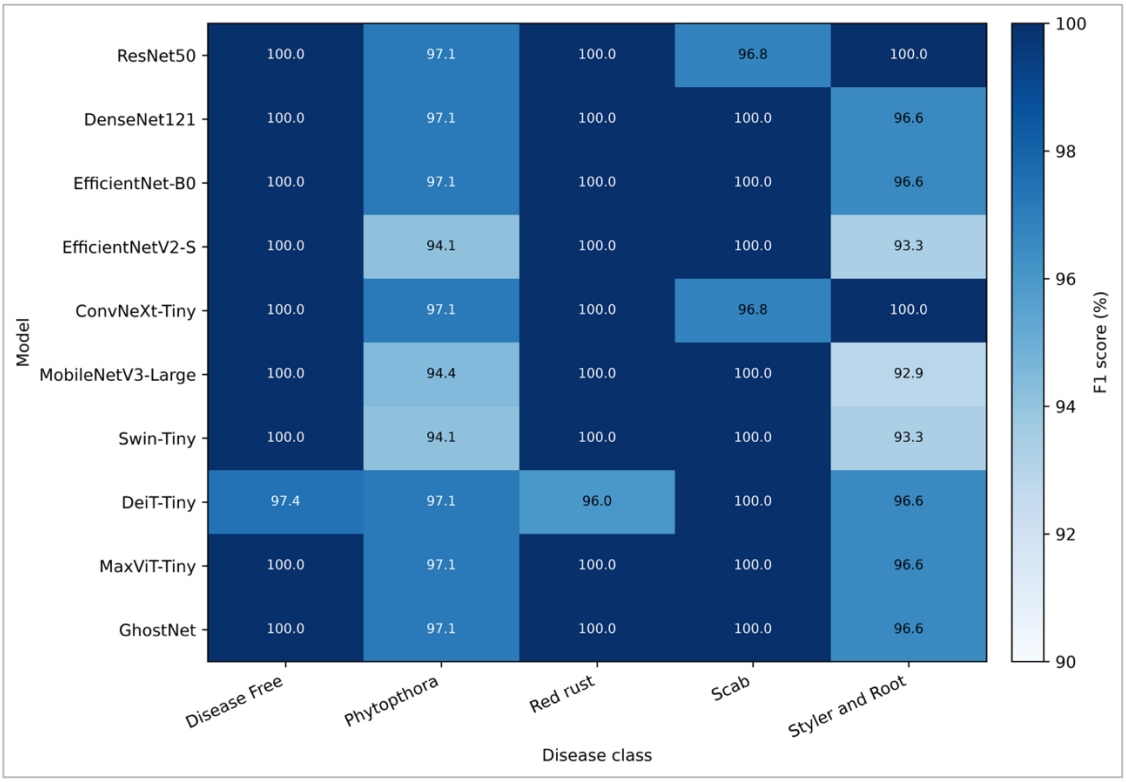

Figure 4. Class wise F1 score comparison of the benchmark deep learning models.

3.4 Error Analysis Using Confusion Matrices

The confusion matrices presented in Figure 5 demonstrate errors at a sample level. DF, Phy, RR, Scab, and SR stand for Disease Free, Phytophthora, Red Rust, Scab, and Styler and Root. The majority of all 10 matrices have clear diagonals, which corresponds to the high summary scores. Multiple classes lack errors for most of the architectures.

The Phy class was the source of errors for ResNet50, DenseNet121, EfficientNet B0, ConvNeXt Tiny, MaxViT Tiny, MobileNetV3 Large, and GhostNet. All these architectures misclassified one Phy image as Scab. EfficientNetV2 S and Swin Tiny committed the same mistake for two images. Since multiple models made the same error, it is possible that the affected pictures have similar characteristics. DeiT Tiny was the only architecture which mixed up an RR image with DF.

Only one or two decisions differ between the models in these matrices. The rest can be seen as challenging or visually similar cases rather than an indication of a general weakness in the architecture. Evaluation using independently collected external datasets is required to determine whether these error patterns persist under different acquisition and environmental conditions. The observed errors were concentrated in two confusion patterns. Most misclassified Phytophthora images were predicted as Scab, suggesting that similarities in fruit-surface lesions, discoloration, or texture may make these categories difficult to distinguish. DeiT-Tiny also classified one Red Rust image as Disease Free, which may reflect weak or localized symptoms, image scale, or insufficient contrast between affected and unaffected leaf regions. These are errors observed within the held-out subset of the GLFD dataset and should not be interpreted as out-of-distribution errors.

No independently collected out-of-distribution dataset was evaluated; consequently, an empirical generalization gap or performance drop cannot be calculated. Nevertheless, several characteristics of the dataset may reduce performance under external conditions. Organ type is completely associated with class label, enabling models to use leaf-versus-fruit characteristics in addition to disease symptoms. Models may also be sensitive to differences in background vegetation, soil, illumination, shadows, viewpoint, image scale, camera characteristics, orchard location, cultivar, season, and symptom severity. The relatively small dataset, single acquisition source, fixed partition, and absence of data augmentation may further increase sensitivity to these changes. External validation is therefore required to determine whether the observed confusion patterns persist and to quantify performance beyond the GLFD dataset.

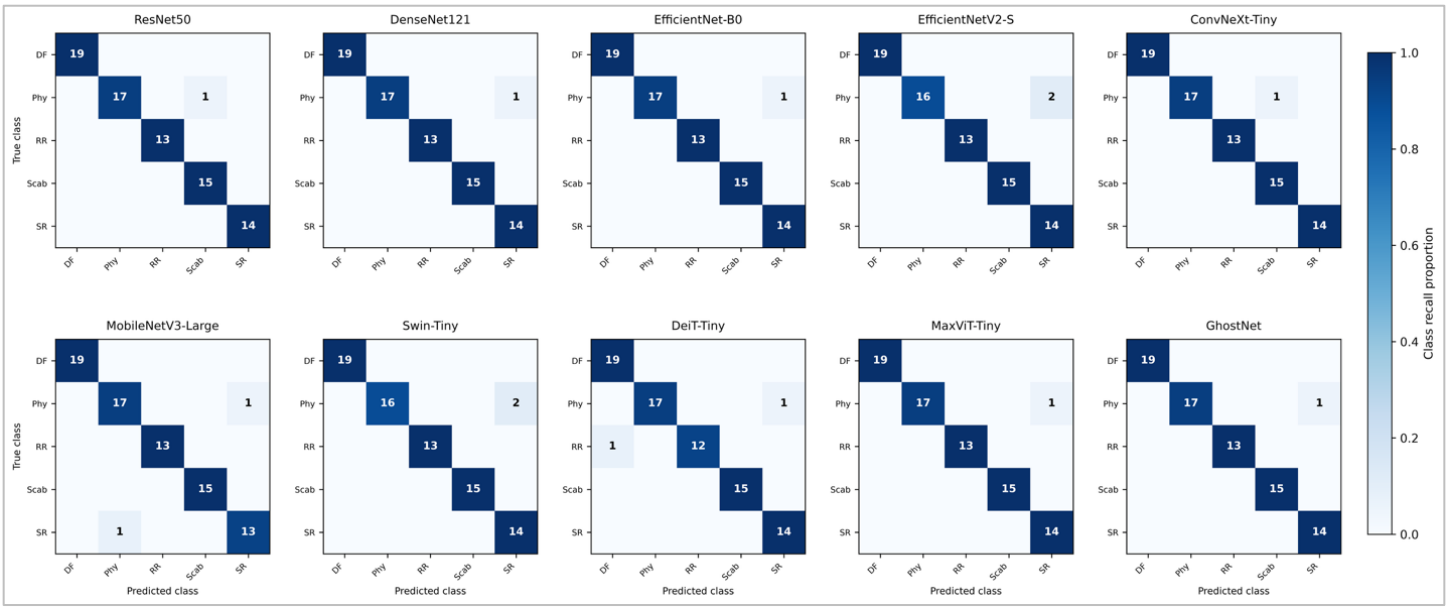

Figure 5. Confusion matrices of the benchmark deep learning models on the held-out test set.

3.5 Computational Efficiency Analysis

Table 6 reports the results of computational measurements, while Figure 6 relates these metrics to the predictive performance. The number of parameters, latency, and storage requirements vary significantly compared to accuracy. In other words, the same classification outcome could be achieved with vastly different computational budgets.

In terms of sheer efficiency, GhostNet proved to be an excellent choice. It boasted 3.91 million parameters, required 15.16 MB of storage, and yielded a Macro F1 score close to the best performer. Another EfficientNet model, B0 variant, had a similarly low number of parameters equal to 4.01 million. MobileNetV3 Large was the fastest convolutional network, with an average of 4.75 ms per image and 210 images processed per second on average. It was a good option for the cases where memory and latency are critical constraints.

Finally, ConvNeXt Tiny, ResNet50, and MaxViT Tiny belonged to the high-cost group of networks. The last one, MaxViT Tiny, had more than 30 million parameters, the largest model size checkpoint in this study, and the worst inference speed in terms of milliseconds per image. Its accuracy advantage over the compact group was barely noticeable, which ruled out this model as an option for this dataset and task due to its high computational burden. Thus, there was no point in opting for larger networks when a smaller one yielded almost identical results.

Table 6. Computational characteristics of the evaluated benchmark models.

Model	Parameters (M)	Inference (ms/image)	Images/s	Checkpoint (MB)
ConvNeXt-Tiny	27.823973	5.11922627	195.34202	106.206183
ResNet50	23.518277	5.28649209	189.161354	90.0185099
GhostNet	3.907913	4.75736973	210.200185	15.1609049
EfficientNet-B0	4.013953	5.03798576	198.492026	15.5902643
DenseNet121	6.958981	5.55267985	180.093221	27.1252213
MaxViT-Tiny	30.406093	7.34439557	136.158243	116.382936
EfficientNetV2-S	20.183893	5.5538692	180.054654	77.8386526
Swin-Tiny	27.523199	5.64690295	177.088221	105.056395
MobileNetV3-Large	4.208437	4.75310127	210.388953	16.2471399
DeiT-Tiny	5.525381	4.99221466	200.311899	21.1332178

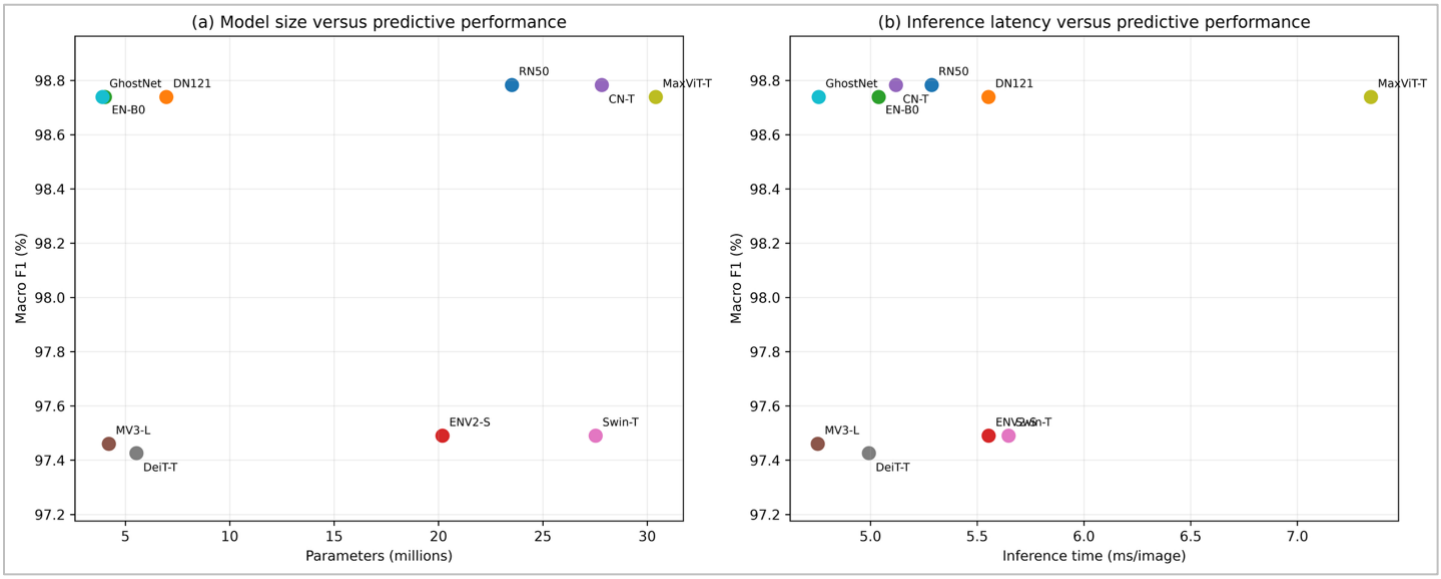

Figure 6. Relationship between computational efficiency and predictive performance of the benchmark models.

The analysis demonstrated that network size was not a reliable indicator of its performance. From the perspective of efficiency, GhostNet and EfficientNet B0 were excellent choices, while ConvNeXt Tiny and ResNet50 still were suitable if one prioritized the highest scores and was not concerned about the large computational footprint.

3.6 Statistical Comparison of Benchmark Models

Figure 7(a) presents a descriptive ranking based on accuracy, balanced accuracy, Macro F1, and MCC. ResNet50, DenseNet121, EfficientNet-B0, ConvNeXt-Tiny, MaxViT-Tiny, and GhostNet received the same average rank because their reported metrics were identical or nearly identical. However, these rankings should be interpreted descriptively. The difference between the two observed accuracy levels corresponded to only one test image, making the ordering highly sensitive to individual predictions.

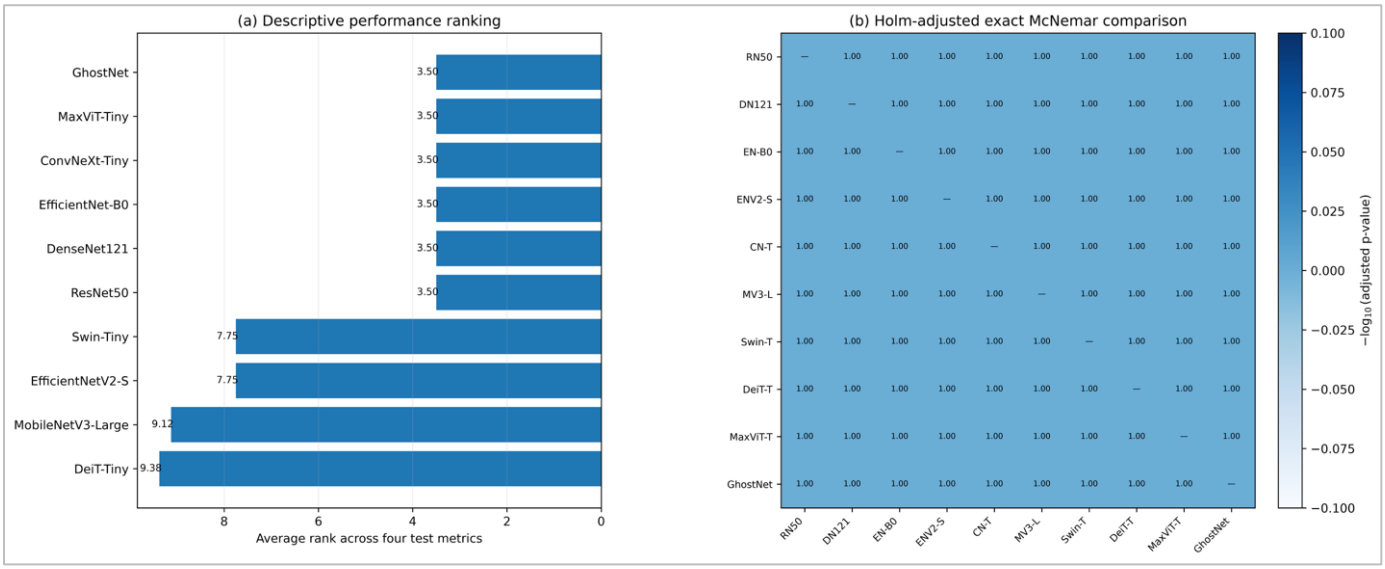

Figure 7. (a) Descriptive ranking of the benchmark models and (b) Holm-adjusted p-values from pairwise exact McNemar tests; all adjusted p-values were 1.000.

Pairwise comparisons were conducted using the two-sided exact McNemar test, followed by Holm correction for the 45 model comparisons. All Holm-adjusted p-values were 1.000, as shown in Figure 7(b), and none reached the predefined significance threshold. Because the exact McNemar procedure was used, the comparisons were

based directly on the discordant prediction counts rather than an asymptotic chi-square statistic.

These non-significant findings do not demonstrate that architecture has equivalent predictive performance. Each model made only one or two errors on the 79-image test set, resulting in very few discordant predictions for pairwise analysis. The tests therefore had limited ability to detect differences. In addition, adjustment across 45 pairwise comparisons produced a conservative inferential setting. The results indicate only that the present held-out test set did not provide sufficient statistical evidence to distinguish the models' prediction patterns.

This interpretation is consistent with the substantially overlapping 95% confidence intervals reported in Table 5. Consequently, the average ranks and McNemar results should be regarded as descriptive evidence specific to the adopted test partition, rather than confirmation of model equivalence or stable architectural superiority

3.7. Study Limitations

A major limitation of this study is the complete confounding between plant organ and class label. Disease Free and Red Rust are represented exclusively by leaf images, whereas Phytophthora, Scab, and Styler and Root are represented exclusively by fruit images. Therefore, the high in-dataset performance cannot be attributed exclusively to disease-specific visual characteristics, because the models may also have learned differences between leaves and fruits. Although the five-class task was evaluated using the available labels, a controlled five-class comparison balanced by organ type was impossible because no class contained both leaf and fruit images. The reported results should therefore be interpreted as classification performance on the original GLFD class structure rather than evidence of pathology-specific recognition.

A further limitation is the size of the held-out test set. With 79 test images, one prediction changes the accuracy by about 1.27%. Consequently, there is barely a difference between 98.73% and 97.47%, which equals one image, and the 95% confidence intervals greatly overlap. Therefore, the ordering of the architectures is uncertain and should not be taken as a given. While the same partition allowed us to directly compare the algorithms on equal footing, it failed to account for variations resulting from different splits. Multiple stratified cross-validation runs, testing on independent partitions, and external validation are required to confidently rank the algorithms.

The pairwise exact McNemar comparisons also had limited statistical power because the models produced only one or two errors on the 79-image test set, leaving very few discordant predictions for analysis. Moreover, applying the Holm correction to 45 model pairs resulted in a conservative comparison. Therefore, the non-significant adjusted p-values cannot be interpreted as evidence that the architectures are predictively equivalent.

The study also relied on one fixed stratified data partition and did not employ data augmentation. Although using identical image assignments and preprocessing operations enabled direct comparison under a controlled protocol, it did not quantify variability caused by alternative partitions. Moreover, the absence of augmentation may have limited the diversity of the training samples and increased sensitivity to characteristics specific to this relatively small dataset. Repeated stratified evaluation across multiple random seeds and carefully designed training augmentation should therefore be considered in future studies.

Potential background bias represents an additional limitation. Due to the original orchard backgrounds being retained, and no background annotations, masking experiments, or controlled background analysis being available, the authors cannot confirm if the models were using disease symptoms only. It is possible the models may have learned to associate vegetation, soil, lighting, shadows, and image composition with certain diseases. This should be regarded as classification performance in-dataset rather than pathology-specific recognition.

Due to using an older and smaller public dataset, the authors are also limited in what benchmark they can use. More recent guava datasets might have more data, more disease categories, and more sample variance. The authors used the dataset to establish reproducibility and comparison to previous research rather than believing it to be better compared to more recent alternatives.

4. Conclusion

This study benchmarked ten deep learning architectures for guava disease recognition within one controlled experimental protocol. The same split, preprocessing, and test procedure were applied to every model. Test accuracy ranged from 97.47% to 98.73%, and Macro F1 ranged from 97.43% to 98.78%. ConvNeXt Tiny and ResNet50 produced the highest observed point estimates: 98.73% accuracy, 98.89% balanced accuracy, 98.78% Macro F1, and 98.43% MCC. Each misclassified one image. GhostNet, EfficientNet B0, DenseNet121, and MaxViT Tiny produced nearly the same predictive performance.

The efficiency results reveal a much larger separation between the models. GhostNet reached 98.74% Macro F1 with 3.91 million parameters and a 15.16 MB checkpoint. MobileNetV3 Large gave the lowest latency, at 4.75 ms per image, and processed about 210 images per second. MaxViT Tiny, by comparison, used 30.41 million parameters and incurred the greatest computational cost without a meaningful improvement over the compact networks. None of the pairwise exact McNemar comparisons reached statistical significance after Holm correction. However, the small test set and very low number of discordant predictions limited the statistical power of these comparisons; therefore, the non-significant results should not be interpreted as evidence of model equivalence.

These findings suggest that lightweight CNNs can be competitive with substantially larger architectures within this particular dataset and fixed experimental protocol. Within the evaluated GLFD dataset and fixed partition, the comparison provides a reproducible preliminary baseline for examining model accuracy and computational efficiency. Its applicability beyond this experimental setting remains to be established through repeated and external evaluation. However, its findings should be interpreted cautiously because they are based on a single public dataset, one fixed data partition, and a held-out test set of only 79 images. In addition, organ type was associated with the class labels, meaning that the models may have learned leaf-versus-fruit characteristics alongside disease symptoms. The overlapping confidence intervals and limited statistical power prevent claims of model equivalence or stable architectural superiority. Future work should employ repeated evaluation and external datasets acquired from different orchards, seasons, devices, and environmental conditions before claims are made regarding field-wide performance.

Author Contributions: A.T.Y. and S.A.A. contributed to the conceptualization, methodology, formal analysis, validation, and review of the final version. E.T.Y. contributed to the conceptualization, methodology, formal analysis, investigation, dataset creation, software, visualization, validation, and writing of the original draft. All authors have read and approved the final manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset analyzed during the current study is openly available in the Mendeley Data repository: https://data.mendeley.com/datasets/x84p2g3k6z/1.

Acknowledgments: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

AI Use Declaration: During the preparation of this work, the authors used a translation tool DeepL and Grammarly solely for language translation and grammar correction. After using these tools, the authors carefully reviewed and edited the text and took full responsibility for the content and accuracy of this publication.

References:

1. Tousif, M.I.; Nazir, M.; Saleem, M.; Tauseef, S.; Shafiq, N.; Hassan, L.; Hussian, H.; Montesano, D.; Naviglio, D.; Zengin, G.; et al. Psidium Guajava L. An Incalculable but Underexplored Food Crop: Its Phytochemistry, Ethnopharmacology, and Industrial Applications. *Molecules 2022, Vol. 27, Page 7016* **2022**, *27*, 7016 https://doi.org/10.3390/MOLECULES27207016.
2. Butt, E.; Altemimi, A.B.; Younas, A.; Butt, M.S.; Jalal, M.; Bhatty, M.; Abdi, G.; Aadil, R.M. Guava (Psidium Guajava): A Brief Overview of Its Therapeutic and Health Potential. *Food Chemistry: X* **2025**, *31*, 103027 https://doi.org/10.1016/J.FOCHX.2025.103027.
3. Guava Production by Country 2026 Available online: https://worldpopulationreview.com/country-rankings/guava-production-by-country (accessed on 05 February 2026).
4. Zakaria, L. Diversity of Colletotrichum Species Associated with Anthracnose Disease in Tropical Fruit Crops—A Review. *Agriculture 2021, Vol. 11, Page 297* **2021**, *11*, 297 https://doi.org/10.3390/AGRICULTURE11040297.
5. Martinelli, F.; Scalenghe, R.; Davino, S.; Panno, S.; Scuderi, G.; Ruisi, P.; Villa, P.; Stroppiana, D.; Boschetti, M.; Goulart, L.R.; et al. Advanced Methods of Plant Disease Detection. A Review. *Agronomy for Sustainable Development 2014 35:1* **2014**, *35*, 1–25 https://doi.org/10.1007/S13593-014-0246-1.
6. Kurniawan, M.B.; Utami, E. Guava Disease Detection and Classification: A Systematic Literature Review. *Telematika* **2025**, *18*, 45–61 https://doi.org/10.35671/TELEMATIKA.V18I1.2901.
7. Singh, R.; Gupta, S.; Aluvala, S.; Riadhwsein, R.; Sharma, G. Proposed Convolutional Neural Network for the Detection of Guava Leaf Diseases. In Proceedings of the 2025 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation, IATMSI 2025; 2025 https://doi.org/10.1109/IATMSI64286.2025.10985607.
8. Rangkuti, A.H.; Samiaji, T.A.; Sebastian, J.; Yora, S.; Maulana, M. Guava Disease Detection Using Machine Learning: A Data-Driven Agriculture Solution. In Proceedings of the Proceedings - 2025 8th International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2025; 2025; pp. 78–83 https://doi.org/10.1109/ISRITI68345.2025.11393115.
9. Raji, P.J.; Meghalingam, R.K. Automated Guava Fruits Disease Classification Using Deep Learning. In Proceedings of the 2025 IEEE International Conference on Electronics, Computing and Communication Technologies, CONECCT 2025; 2025 https://doi.org/10.1109/CONECCT65861.2025.11306537.

10. Srinivas; Satheesh; Rama Santosh Naidu; Neelima *Prediction of Guava Plant Diseases Using Deep Learning*; 2021; Vol. 698; https://doi.org/10.1007/978-981-15-7961-5_135.
11. Thrimavithana, S.T.; Manukalpa, C.S. FoliumX: An Integrated IoT-AI Framework for Real-Time Crop Disease Detection and Environmental Monitoring for Smallholder Farmers. In Proceedings of the Proceedings - IEEE International Research Conference on Smart Computing and Systems Engineering, SCSE 2026; 2026 https://doi.org/10.1109/SCSE70081.2026.11499762.
12. Miah, M.A.; Roshni, S.R.; Anamika, F.T.; Kamal, K.M.S.; Reza, A.W. GuavaNet-XAI: A CNN-Based Framework for Accurate and Interpretable Diagnosis of Guava Diseases. In Proceedings of the 2025 28th International Conference on Computer and Information Technology, ICCIT 2025; 2025; pp. 5353–5358 https://doi.org/10.1109/ICCIT68739.2025.11491449.
13. Pathmanaban; Gnanavel; Anandan, S.S. Guava Fruit (Psidium Guajava) Damage and Disease Detection Using Deep Convolutional Neural Networks and Thermal Imaging. *Imaging Science Journal* **2022**, *70*, 102–116 https://doi.org/10.1080/13682199.2022.2163536.
14. Yasin, E.; Kursun, R.; Koklu, M. Deep Learning-Based Classification of Black Gram Plant Leaf Diseases: A Comparative Study. In Proceedings of the 11th International conference on advanced technologies (ICAT'23); Istanbul-Turkiye, 2023; pp. 59–69 https://doi.org/10.58190/icat.2023.9.
15. Yasin, E.T.; Kursun, R.; Koklu, M. Machine Learning-Based Classification of Mulberry Leaf Diseases. In Proceedings of the Proceedings of International Conference on Intelligent Systems and New Applications; Liverpool, United Kingdom, April 28 2024; Vol. 2, pp. 58–63 https://doi.org/10.58190/ICISNA.2024.91.
16. Hamid, Y.; Wani, S.; Soomro, A.B.; Alwan, A.A.; Gulzar, Y. Smart Seed Classification System Based on MobileNetV2 Architecture. In Proceedings of the 2022 2nd International Conference on Computing and Information Technology (ICCIT); IEEE, 2022; pp. 217–222.
17. Zangana, H.M.; Li, S.; Wani, S. Diffusion Models for Agricultural Imaging: A Systematic Review of Methods, Applications and Future Prospects. *Impact in Agriculture* **2025**, *1*, 1–11 https://doi.org/10.65500/agriculture-2025-003.
18. Mehdipour, S.; Mirroshandel, S.A.; Tabatabaei, S.A. Vision Transformers in Precision Agriculture: A Comprehensive Survey. *Intelligent Systems with Applications* **2026**, *29*, 200617 https://doi.org/10.1016/J.ISWA.2025.200617.
19. Majdalawieh, M.; Martins, C.; Radi, M.; Alaraj, M.; Khan, S. Precision Agriculture in the Age of AI: A Systematic Review of Machine Learning Methods for Crop Disease Detection. *Smart Agricultural Technology* **2025**, *12*, 101491 https://doi.org/10.1016/J.ATECH.2025.101491.
20. Kanani, I.J.; Kaur, M.; Sridhar, R.; Dhenia, R.N.K.; Kumar, J.; Sharma, G.; Banerjee, D. Towards Sustainable Agriculture: Deep Learning for Natural Guava Disease Detection. In Proceedings of the IETACS 2025 - 2025 International Conference on Innovations and Emerging Technologies in AI and Communication Systems: Unifying AI and Communication for a Smarter Tomorrow; 2025; pp. 341–346 https://doi.org/10.1109/IETACS68750.2025.11385380.
21. Tewari, V.; Azeem, N.A.; Sharma, S. Automatic Guava Disease Detection Using Different Deep Learning Approaches. *Multimedia Tools and Applications* **2024**, *83*, 9973–9996 https://doi.org/10.1007/s11042-023-15909-6.
22. Ibrahim, N.J.; Khorsheed, F.H.; Mohammed, Z.H.; Latfa, A.M.; Khudhair, W.B.; Hassan, Z.F.; Alkattan, H. Hybrid CNN-ViT Framework for Guava Leaves and Fruits Disease Detection in Precision Agriculture. In Proceedings of the Proceedings of International Conference on Applied Innovation in IT; 2026; Vol. 14, pp. 49–58.
23. Shrivastava, A.; Habelalmateen, M.I.; Kaur, A.; Praveen; Badhoutiya, A.; Kumar, A. Green Diagnosis: Deep Learning-Based Guava Leaf Disease Classification. In Proceedings of the 2025 IEEE Madhya Pradesh Section Conference, MPCON 2025; 2025; pp. 267–273 https://doi.org/10.1109/MPCON66082.2025.11256545.
24. Kaur, A.; Kukreja, V.; Upadhyay, D.; Aeri, M.; Sharma, R. Multi-Class Guava Disease Classification Using an Efficient and Fine-Tuned DenseNet Model. In Proceedings of the 2024 IEEE 9th International Conference for Convergence in Technology, I2CT 2024; 2024 https://doi.org/10.1109/I2CT61223.2024.10544267.
25. Cheekati, H.; Verma, P.; Vaithiyanathan, D. Guava (Psidium Guajava) Disease Detection Using CNN. In Proceedings of the 2025 2nd International Conference on Pioneering Developments in Computer Science and Digital Technologies, IC2SDT 2025 - Proceedings; 2025; pp. 1–6 https://doi.org/10.1109/IC2SDT68218.2025.11383705.
26. Vijh, S.; Shieh, C.-S.; Jain, V.; Horng, M.-F. Optimized Deep Learning Framework for Plant Disease Detection Using Fine-Tuned Vision Transformers and CNN Model. *SN Computer Science* **2026**, *7* https://doi.org/10.1007/s42979-026-05044-y.
27. Rajbongshi, A.; Sazzad, S.; Shakil, R.; Akter, B.; Sara, U. Guava Leaves and Fruits Dataset for Guava Disease Recognition through Machine Learning and Deep Learning. **2022**, *1* https://doi.org/10.17632/X84P2G3K6Z.1.