

An empirical study of CNN models and XAI for grape leaf disease detection based on data split ratios

Wasan Alamro¹, Gang Li²

¹Department of Communications and Computer Engineering, Faculty of Engineering, Al-Ahliyya Amman University, Amman, Jordan

²Department of Information Technology, Auckland Institute of Studies, Auckland, New Zealand

Article Info

Article history:

Received Oct 31, 2025

Revised Aug 29, 2026

Accepted Sep 4, 2026

Keywords:

Convolutional neural networks

Data splitting

Explainable artificial intelligence

Gradient-weighted class

activation mapping

Plant disease detection

ABSTRACT

The efficiency of deep learning (DL) models in detecting plant diseases is frequently evaluated on controlled datasets, with little attention paid to data-partitioning techniques that impact generalization in real-world scenarios. This study examines the effects of three data split ratios (80:10:10, 70:15:15, and 50:25:25) on the generalizability of four convolutional neural network (CNN) architectures, DenseNet121, ResNet50, InceptionV3, and VGG16, for grape leaf disease classification across four classes of Black Rot, Esca (Black Measles), Leaf Blight, and Healthy. In addition to the standard performance metrics, we include external validation using an independent Kaggle dataset (9,027 images) to assess true generalizability, complementing the primary PlantVillage dataset (3,251 images). Results indicate that split ratio significantly affects model performance, with DenseNet121 showing outstanding stability across all splits and superior external test accuracy (up to 99.6%), while InceptionV3 demonstrates extreme instability, and ResNet50 shows noticeable variation. Explainable artificial intelligence (XAI) via gradient-weighted class activation mapping (Grad-CAM) reveals that DenseNet121 and ResNet50 focus on pathologically significant leaf regions rather than spurious background correlations. This study offers a framework for assessing DL models in agriculture, highlighting that no single split ratio universally optimizes performance and emphasizing external validation for deploying trustworthy solutions.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Wasan Alamro

Department of Communications and Computer Engineering, Faculty of Engineering

Al-Ahliyya Amman University

Amman, Jordan

Email: w.alamro@ammanu.edu.jo

1. INTRODUCTION

Early detection of crop diseases is critical for food security and agricultural sustainability [1], [2]. Grape diseases such as black rot, leaf blight, and *esca* (black measles) can cause severe crop losses if not promptly managed [3], [4]. Conventional detection methods, such as visual inspection and laboratory diagnosis are inefficient and unsuitable for large-scale monitoring [5]. While image analysis technologies offer more efficient alternatives [6], traditional approaches remain constrained by expert reliance and high costs, and remote sensing faces environmental and equipment challenges [7].

Deep learning (DL), particularly convolutional neural networks (CNNs), has enabled significant advances in diagnostic applications and smart agriculture [8]–[10], offering solutions for precision farming, pest control [11], [12], and disease detection [13], [14]. For grape disease identification, CNNs automate the analysis of leaf images to detect diseases such as black rot, powdery mildew, and downy mildew, providing a

scalable alternative to manual inspection [15], [16]. However, their widespread adoption is constrained by three key challenges including lack of interpretability (the black-box problem), high computational demands, and reliance on large datasets [17]. Explainable artificial intelligence (XAI) addresses the interpretability gap by offering visual insights into model decisions, thereby enhancing trust and usability in agricultural applications [18], [19].

A substantial body of work has applied CNN architectures such as DenseNet121, ResNet50, VGG16, InceptionV3, EfficientNet, Xception, and MobileNet to plant leaf disease detection [20]–[28] and plant classification [29]–[33] generally reporting strong accuracies, for example, 90.12–99.5% across banana and black gram leaf disease using an attention-enhanced DenseNet variant [20], 99.8% using transfer-learned ResNet50 [21], 97.22–98.75% across rice and wheat using an improved VGG16 [22], up to 98.82% across twelve pretrained networks on chilli leaf data [23], and up to 100% using a combination of CNN and vision-transformer models on grape leaf data [15]. Despite this range of strong results, a consistent pattern emerges across these studies, where nearly all rely on a single, fixed data-partitioning ratio, most commonly 80% training/10% validation/10% testing [20]–[23], [25], [27], [29], with a few alternatives such as 70:15:15 [15], 75% training: 25% testing split [24], and 60% training: 40% testing [30], but none testing more than one split ratio within the same study. External validation on an independent dataset is similarly rare, most studies evaluate only on held-out partitions of the same source dataset [15], [20]–[26]. When external testing is attempted, its scope is often minimal, one study, for instance, validated a single model variant on just one external eggplant disease image [19]. XAI techniques such as gradient-weighted class activation mapping (Grad-CAM) and LIME are used in only a subset of this literature [19], [20], [27], and even where present, they are applied in isolation, without examining whether interpretability itself is sensitive to data partitioning or to which architecture is used.

Despite these advances, three critical gaps remain. First, the effect of data-split ratio on model generalizability is unexamined: because every reviewed study fixes a single partitioning scheme, it is unclear whether reported accuracies would hold under a different split, or whether some architectures are more sensitive to this choice than others. Second, external validation on a genuinely independent dataset is rarely performed, and even more rarely at a scale sufficient to draw reliable conclusions about real-world generalization [19]. Third, no reviewed study connects explainability analysis to data-partitioning or cross-architecture comparison, leaving open whether a model's disease-region localization, and not just its accuracy, degrades under distribution shift. Unlike prior studies that rely on fixed split ratios, this work systematically compares three split ratios (80:10:10, 70:15:15, and 50:25:25); validates each configuration on an independent external Kaggle dataset; and integrates Grad-CAM interpretability analysis across both split ratios and architectures to offer insight into real-world generalization.

The main contributions for this study are as follows:

- Unlike prior grape-disease studies that evaluate a single fixed split ratio, this work is the first to systematically quantify how split-ratio choice affects generalizability across four widely used CNN architectures, ResNet50, DenseNet121, VGG16, and InceptionV3, using repeated multi-run training (three split ratios $\times$ three runs $\times$ four models, 36 sessions in total) to ensure statistical reliability rather than relying on single-run results.
- This study also provides one of the few genuinely independent external validations in this domain, using a separate Kaggle grape-disease dataset distinct in source and distribution from the PlantVillage training data, establishing a stronger standard for cross-domain generalization than the single-dataset or single-image external tests common in prior work [19].
- This is the first study in this domain to link Grad-CAM interpretability directly to split-ratio and architecture choice, revealing that classification accuracy and interpretability quality do not always align, for instance, DenseNet121 achieves the highest external accuracy while still showing incomplete Grad-CAM coverage on healthy-leaf regions, and InceptionV3's high internal validation accuracy is not matched by either external generalization or reliable feature localization. Together, these findings show that DenseNet121 achieves the highest and most stable external test accuracy (up to 0.996) across all three split ratios, offering practical, evidence-based guidance for architecture selection under real-world uncertainty about data partitioning.

2. METHOD

This study proposes a multi-stage framework for grape leaf disease classification and interpretability analysis, encompassing data preprocessing, model training under multiple split ratios, external validation, and Grad-CAM-based explainability. The overall workflow is illustrated in Figure 1. The study begins with the dataset unit, followed by data preprocessing, which includes cleaning, resizing, normalizing, and using augmentation techniques, such as rotation, flipping, and scaling. A key factor in developing a robust DL model

for plant disease detection is partitioning the data into training, validation, and test sets. Common split ratios, such as 80:10:10, may lead to over-optimistic performance, especially if the test set does not truly represent real-world variability.

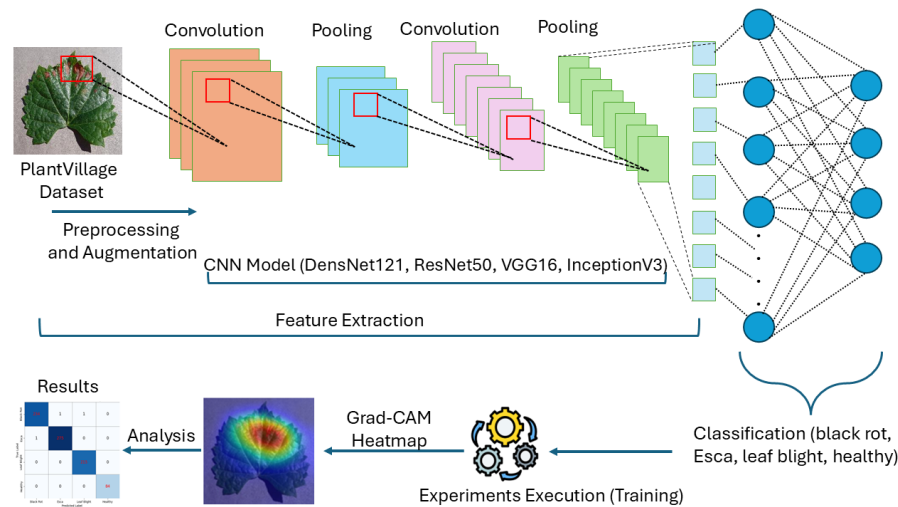


Figure 1. Proposed framework for grape leaf disease classification, illustrating the workflow from data preprocessing to model training, external validation, and Grad-CAM interpretability

2.1. Datasets

A comprehensive experimental framework was designed to systematically investigate the impact of different data split ratios on the generalizability of the model. The proposed framework evaluates the model's performance not only on a standard internal test set but also on a fully independent external dataset. This study utilized two distinct datasets to ensure a thorough evaluation. The primary dataset was the PlantVillage dataset, which is a comprehensive open-source collection featuring 54,305 images of 14 different crop species, including grapes, apples, strawberries, and tomatoes [34]. The other one is the Kaggle grape disease dataset (external test dataset), which is used to evaluate the generalization performance of the models [35].

This study focused on grape leaves, and the primary dataset included four categories: three for grape leaf diseases (Black Rot, Black Measles, and Leaf Blight) and one for Healthy leaves. A total of 3,251 photos are studied, including 944 Black Rot, 1,107 Black Measles, 861 Leaf Blight, and 339 Healthy leaf images, to ensure a balanced dataset for effective model training and evaluation, Figure 2 shows a sample of healthy and infected grape leaves from the used dataset. Specifically, Figure 2(a) shows a healthy grape leaf with uniform green coloration and no visible lesions. Figure 2(b) illustrates Black Rot, characterized by circular brown lesions with darkened margins. Figure 2(c) depicts *Esca* (Black Measles), showing the characteristic interveinal chlorosis (tiger-stripe pattern) with necrotic spots. Figure 2(d) presents Leaf Blight, exhibiting irregular brown patches with yellowing margins. The primary dataset was partitioned using three different split ratios, and then four CNN architectures were tested and trained for grape leaf disease recognition and classification based on the transfer learning approach. Every trained model was further evaluated using an external test dataset. Finally, the classification results of the utilized models (ResNet50, DenseNet121, VGG16, and InceptionV3) were interpreted using the efficient XAI method (Grad-CAM) by generating heatmaps (HMs) for the corresponding models.

The external dataset is sourced from Kaggle and contains 9,027 images with a different distribution: Black Rot (2360), *Esca* (2400), Healthy (2115), and Leaf Blight (2152). This dataset was used exclusively for final testing and was never used during training or validation, simulating a real-world scenario. To confirm the independence of the two datasets, the primary PlantVillage grape subset [34] and the external Kaggle grape disease dataset [35] were obtained from distinct sources with non-overlapping upload histories and differing class distributions, for example a healthy class count of 339 in PlantVillage versus 2115 in the external set.

In previous DL studies, models were typically trained using only the training and validation sets. However, this study adopted a different strategy, in which the data were split randomly into three parts using three split ratios: (80% training, 10% validation, and 10% testing), (70% training, 15% validation, and 15% testing), (50% training, 25% validation, and 25% testing) in splits 1, 2, and 3, respectively. The training set is

used to train the model and optimize its parameters, whereas the validation set helps with hyperparameter adjustment and overall model performance evaluation [15], [36]. It is essential to include a test dataset, because it offers a separate assessment of the model. It guarantees that the outcomes are representative of actual circumstances and removes bias caused by overfitting the training data. This approach increases the reliability of the outcomes and offers more accurate information on how well the model performs on unseen data. The split ratios used in the PlantVillage dataset for training, testing, and validation are listed in Table 1.

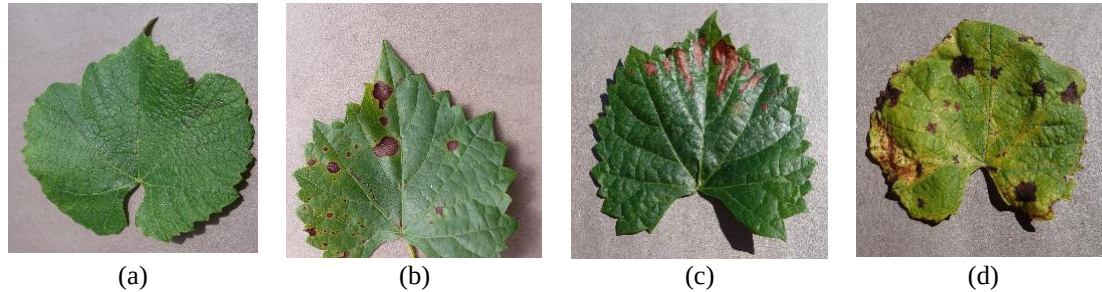


Figure 2. Samples of healthy and infected grape leaves from the PlantVillage dataset: (a) healthy grape leaf, (b) Black Rot, (c) *Esca* (Black Measles), and (d) Leaf Blight

Table 1. Grape crop datasets and their counts in different splits

Class name	Test %			Validation %			Train %			Total (100%)
	10%	15%	25%	10%	15%	25%	80%	70%	50%	
Healthy	34	51	84	34	51	84	271	237	171	339
Black Rot	94	142	236	94	142	236	756	660	472	944
Black Measles (<i>Esca</i>)	111	166	276	111	166	276	885	775	555	1107
Leaf Blight	86	129	215	86	129	215	689	603	431	861

2.2. Data preprocessing

Preprocessing is a crucial step in obtaining raw data for DL models; it maximizes the efficiency and attains the best accuracy. For consistency, all images from both the datasets were preprocessed. First, irrelevant plant leaf data were filtered and manually fixed for any mislabeled data. CNN models often expect images of a specific size. For example, ResNet50, DenseNet121, and VGG16 frequently employ input images of 224×224 pixels. The InceptionV3 model required an input size of 299×299 pixels. The images in the utilized datasets were 256×256 pixels; therefore, they were resized to meet these requirements using TensorFlow's Keras library. This step is critical for consistency, because convolutional operations rely on uniform image sizes. In addition, the pixel values in the images typically range from 0 to 255. Nevertheless, models tend to perform better when the pixel values are scaled between (0 and 1) or (−1 and 1).

To achieve this, pixel values were normalized to the [0, 1] range by dividing each pixel value by 255. The formula for normalization is given as shown in (1):

$$\text{Normalized value} = \frac{\text{Pixel value}}{255} \quad (1)$$

Data augmentation is a fundamental technique in DL, particularly for image-classification tasks. It manipulates training datasets to make them larger and more diverse through a range of transformations such as rotation, flipping, cropping, scaling, and brightness and contrast adjustments [27], [37]. This technique is crucial for training CNN and other DL models. This reduces overfitting by exposing the model to a wider range of image variations. Therefore, the model is encouraged to learn generalizable features rather than memorize specific details of the training data. Real-world variations, such as changing backgrounds, lighting, and orientations, were simulated to achieve this [15].

2.3. Experimental design

The primary dataset was divided into three subsets using three split ratios (80% training, 10% validation, 10% testing), (70% training, 15% validation, 15% testing), and (50% training, 25% validation, 25% testing). To ensure statistical reliability, each model-split combination was trained three times using random seeds (42, 123, 2024), resulting in 36 independent training sessions (4 models×3 splits×3 runs). Stratified random splitting was used to maintain class distribution across all subsets.

2.4. Training configuration

All experiments were conducted on an RTX 3060 GPU with 6 GB memory using CUDA 11.3 and PyTorch 1.12. Unified hyperparameters included the Adam optimizer ($lr = 0.001$ with ReduceLROnPlateau scheduler, $factor = 0.1$, $patience = 3$), batch size of 32, maximum 50 epochs, and early stopping with patience of 6 epochs. Adam was selected for its adaptive learning-rate properties, which reduce the need for manual learning-rate tuning across four architecturally distinct models; early stopping was employed to prevent overfitting. Extensive data augmentation was applied only to training sets including random rotation ($\pm 30^\circ$), horizontal/vertical flipping, and color jitter ($brightness = 0.2$, $contrast = 0.2$). These augmentation parameters were chosen to simulate real-world variations in lighting, orientation, and image quality that the models may encounter during field deployment.

In addition to hyperparameter tuning, each pretrained architecture required specific modifications to its final classification layers to adapt it to the four-class grape disease task. Table 2 summarizes these architecture-specific modifications, including parameter counts, the layers altered, and which layers were frozen during training. As shown in Table 2, VGG16's convolutional layers were frozen during fine-tuning, owing to its substantially larger parameter count ($\sim 138\text{M}$) relative to the other three architectures; freezing these layers reduced training time and overfitting risk while still leveraging its ImageNet-pretrained features. The remaining three models, ResNet50, DenseNet121, and InceptionV3, were fully fine-tuned given their comparatively lower parameter counts and their architectural properties (skip connections and dense connectivity, respectively) that support stable gradient flow even without layer freezing. InceptionV3's auxiliary classifier was retained and adjusted alongside its primary classifier, a design choice known to improve gradient flow and regularization during training.

By iteratively adjusting the calculated gradients and model parameters, the training process spanned up to 50 epochs, balancing the risks of underfitting and overfitting. During evaluation, gradient tracking was disabled to reduce memory consumption and improve computational efficiency.

Table 2. Summary of architecture modifications for transfer learning

Model	Parameters (post-modification)	Final layer modification	Frozen layers	Input size	Output classes
ResNet50	$\sim 23.5\text{M}$	Final fully connected layer replaced, mapping 2048-dim features to 4 output classes	None (fully fine-tuned)	224×224	4
DenseNet121	$\sim 6.96\text{M}$	Classifier layer replaced; final feature output obtained via global average pooling (no separate FC layer)	None (fully fine-tuned)	224×224	4
VGG16	$\sim 134.28\text{M}$	Final classifier layer replaced	Convolutional feature-extraction layers frozen	224×224	4
InceptionV3	$\sim 24.35\text{M}$	Both primary and auxiliary classifier heads replaced	None (fully fine-tuned)	299×299	4

2.5. Performance metrics and model explanation techniques

This study compared the effectiveness of grape disease recognition models using two types of measures: accuracy-related indicators and computational efficiency metrics. The accuracy-related metrics included the precision, recall, and F1-score. In terms of accuracy, the fraction of true predictions is evaluated, whereas precision focuses on the capacity of the model to eliminate false positives. Recall highlights the ability of the model to recognize all the positives. Additionally, the F1-score strikes a compromise between precision and recall by considering both false-positive and false-negative results. As a result, accuracy, recall, and F1-score offer helpful insights into the categorization performance. Precision focuses on reducing false positives. The mathematical expressions for these metrics are defined as (2)-(5):

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \quad (2)$$

$$\text{Precision} = \frac{\text{Number of True positives}}{\text{True positives} + \text{False positives}} \quad (3)$$

$$\text{Recall} = \frac{\text{Number of True positives}}{\text{True positives} + \text{False negatives}} \quad (4)$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

To examine the learning process of the models and identify overfitting or underfitting, training and validation losses were also computed. Additionally, the unique performance of each disease type was examined using a confusion matrix. The sklearn.metrics module, which offers an effective way to compute the accuracy, precision, recall, and F1-score based on predicted and true labels, was used to calculate these metrics. On the other hand, computational efficiency metrics, such as inference time, throughput, and parameter count, quantify the model's performance in real-time. The throughput is the number of images processed per second, whereas the inference time is the speed of the model when analyzing a single image. Furthermore, the size of the model and the number of parameters are important considerations.

They help in determining the computational requirements and viability of deployment on devices with limited resources, such as smartphones or IoT devices. These metrics were measured by counting the number of predicted images and the amount of time required to analyze a dataset. The evaluation criteria were chosen to provide a comprehensive analysis of the ability of the model to effectively identify grape diseases.

Grad-CAM was utilized to enhance the decision-making process of the CNN models. This improves model interpretability and fosters confidence in AI-driven grape disease classification [38], [39]. In addition, it determines the gradients of each model output for a target class and weighs the feature maps of the last convolutional layer. Grad-CAM creates heat maps that overlay the original image. Therefore, areas that discriminate based on class and increasing transparency in the decision-making process should be highlighted [40].

2.6. Statistical analysis

Paired t-tests compared external test accuracies across the three split ratios ($n=3$ per model comparison, $df=2$). A significance level of ($\alpha=0.05$) was used to determine statistical significance. Normality of paired differences was confirmed via Shapiro-Wilk ($p>0.05$ for all pairs), confirming that the assumption of normality was not violated. To account for multiple comparisons, a conservative Bonferroni-corrected threshold of ($\alpha=0.05/6=0.0083$) was applied across the six pairwise comparisons; results below this threshold were considered significant, and those between 0.0083 and 0.05 are reported as suggestive. The 95% confidence intervals (CIs) were calculated using the standard formula ($CI = mean \pm t(0.025, df) \times (\frac{SD}{\sqrt{n}})$) with critical $t=4.303$.

3. RESULTS AND DISCUSSION

In this section, we compare the DL models ResNet50, DenseNet121, VGG16, and InceptionV3 utilized for grape disease diagnosis using the PlantVillage dataset. The models were trained on the primary dataset for different splits in three runs. The models were assessed according to the validation accuracy, training and validation loss, F1-score, and recall. The training was automatically stopped early by defining a patience of 6. Figure 3 shows validation accuracy across three training runs for each model and split configuration. DenseNet121 consistently exceeded 0.98 validation accuracy across all splits with minimal run-to-run variation as displayed in Figure 3(a). ResNet50 performed comparably well but showed occasional instability, particularly at 80:10:10 as shown in Figure 3(b). InceptionV3 exhibited the highest variability, with some runs approaching 1.0 while others started as low as 0.5428 as demonstrated in Figure 3(c). While VGG16 showed the most consistent but lowest peak performance (0.92–0.97) across all splits as shown in Figure 3(d).

Figure 4 summarizes average and standard-deviation performance across validation, test, and external test sets for all models and splits. Figures 4(a) and (b) show the average and standard deviation of validation accuracy, respectively, with DenseNet121 achieving the highest average (0.976–0.981) and lowest variability across splits. Figures 4(c) and (d) present the average and standard deviation of test accuracy, where InceptionV3 exhibits the highest variability among all models. Figures 4(e) and (f) report the average and standard deviation of external test accuracy for all models across split ratios. DenseNet121 achieved the highest and most stable performance across all three evaluation stages (validation: 0.976–0.981; test: 0.985–0.988; external accuracy: 0.988–0.991) with low standard deviations (0.002–0.013), consistent with its dense connectivity property, which promotes feature reuse and gradient flow [41]. ResNet50 showed strong internal performance (validation 0.955–0.972) but a notable external-generalization drop at 70:15:15 (mean 0.963, SD 0.047), suggesting its residual connections benefit intra-domain performance more than cross-domain stability.

On the other hand, InceptionV3 showed the opposite pattern, strong validation accuracy (0.965–0.971) but poor external generalization (0.459–0.679, SD 0.079–0.179), consistent with architectural overfitting, where its parallel filter banks (1×1, 3×3, 5×5 convolutions) were designed for large-scale datasets such as ImageNet (1.2M images) and appear to learn features that transfer poorly to a smaller, domain-specific dataset. VGG16 generalized consistently but at a lower ceiling (validation: 0.954–0.960, SD<0.003; test: 0.949–0.970, SD<0.019; external accuracy: 0.929–0.937, SD<0.020), reflecting its simpler sequential architecture without residual or dense connections. DenseNet121's stability across splits stems from its dense

connectivity, which lets later layers reuse early-layer features directly rather than re-learning them. This effectively regularizes the network [41], so even though it has $\sim 7\text{M}$ parameters, its effective capacity relative to training data stays favorable even at the smallest split, at 50:25:25 (1625 training images), it still achieves 0.988–0.991 external accuracy as shown in Figure 4(e). InceptionV3 shows the opposite pattern for a related reason; its parallel filter banks ($\sim 24\text{M}$ parameters) were designed for ImageNet-scale data, giving it far more capacity than a few thousand grape images warrant. At smaller splits, this mismatch worsens, since the model must fit roughly three times DenseNet121's parameters on the same reduced data, increasing the risk of memorizing source-domain artifacts instead of generalizable disease features.

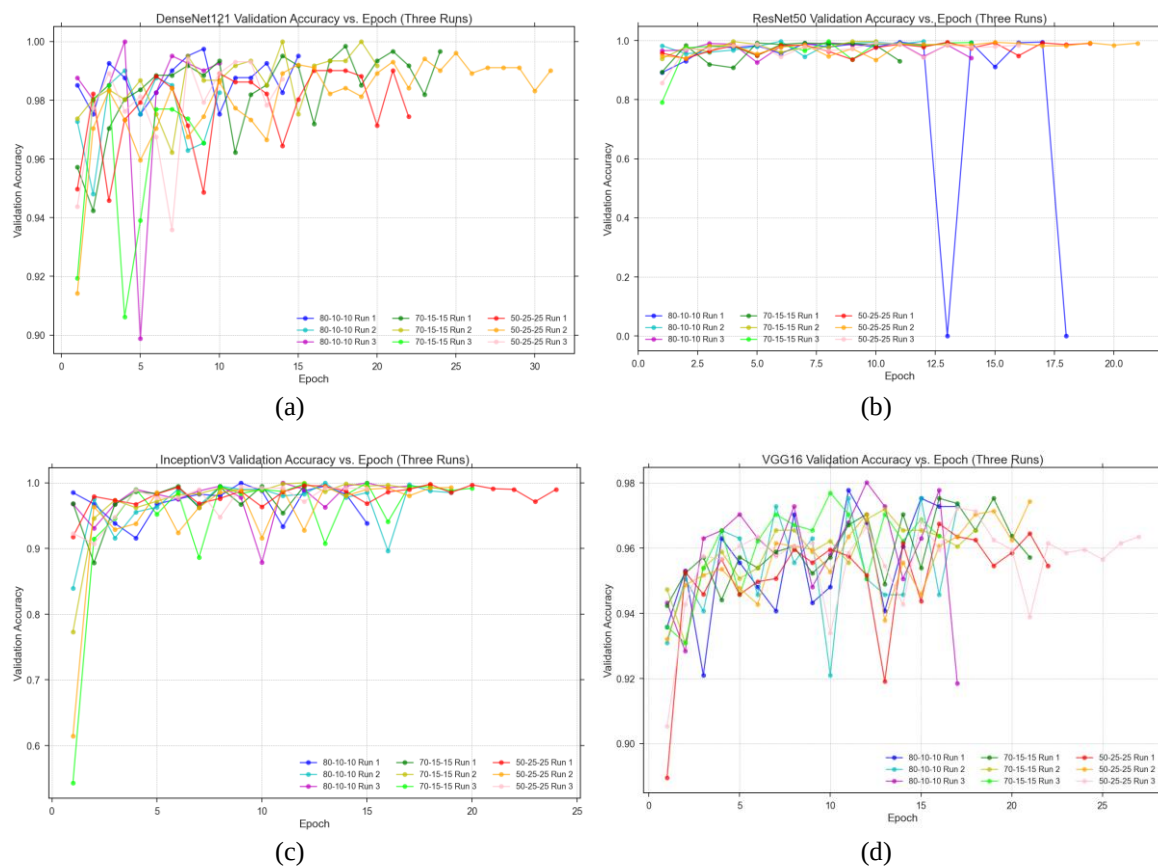


Figure 3. Validation accuracy across three training runs for each CNN model and split configuration; (a) DenseNet121, (b) ResNet50, (c) InceptionV3, and (d) VGG16

DenseNet121's mean external accuracy across the three split ratios was 0.990 (95% CI: [0.985, 0.995]), reflecting its low variance across split ratios, with its best single-split performance reaching 0.996 at 80:10:10 as listed in Table 3. Statistical comparisons confirmed a significant difference between DenseNet121 and VGG16 ($p = 0.003$), which survived Bonferroni correction ($\alpha = 0.0083$); differences among the remaining model pairs ($p = 0.023$ – 0.041) did not survive correction and are reported as suggestive rather than conclusive, where differences between DenseNet121 and InceptionV3 ($p = 0.023$), ResNet50 and VGG16 ($p = 0.041$), ResNet50 and InceptionV3 ($p = 0.030$), and VGG16 and InceptionV3 ($p = 0.030$). While the difference between DenseNet121 and ResNet50 was not significant ($p = 0.325$), indicating comparable generalization capability despite DenseNet121's higher mean accuracy. No split ratio consistently dominated across all four architectures, though 50:25:25 produced comparable or improved generalization for DenseNet121 and ResNet50, underscoring the value of a sufficiently large validation and test allocation for reliable model evaluation.

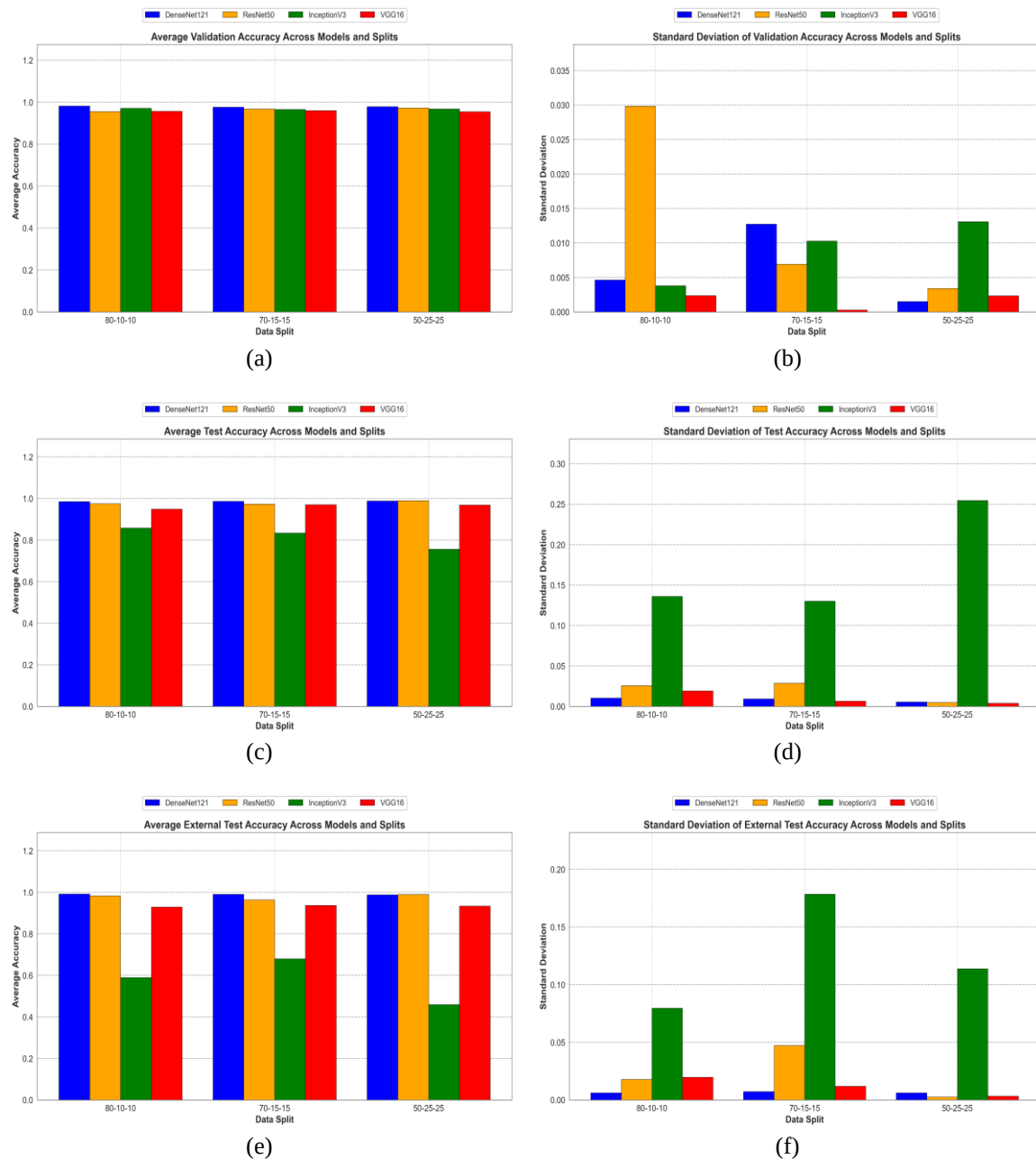


Figure 4. Performance comparison across models and split ratios: (a) average validation accuracy, (b) standard deviation of validation accuracy, (c) average test accuracy, (d) standard deviation of test accuracy, (e) average external test accuracy, and (f) standard deviation of external test accuracy

The confusion matrices at each model's best-performing split, selected based on the highest external test accuracy achieved in three rounds on the Kaggle Grape Disease Dataset, (DenseNet121: 80:10:10, 99.6%; ResNet50: 50:25:25, 99.3%; VGG16: 70:15:15, 95.3%; InceptionV3: 70:15:15, 93.1%) are illustrated in Figure 5. Figure 5(a) shows DenseNet121's confusion matrix at its best-performing split (80:10:10), with the fewest total misclassifications (36/9027, 0.39%) and no errors in the Healthy class. Figure 5(b) presents ResNet50 (50:25:25), with total misclassifications of (67/9027, 0.74%) and modest confusion between Healthy and Black Rot classes. Figure 5(c) shows VGG16 (70:15:15), where errors are concentrated in Healthy-leaf misclassification (237/2115, 11.2%), suggesting difficulty distinguishing pathological symptoms from minor physical damage. Figure 5(d) presents InceptionV3 (70:15:15), showing the most systematic error pattern, frequently confusing Black Rot with *Esca* (413/2360, 17.5%), consistent with over-reliance on color-based features rather than lesion morphology.

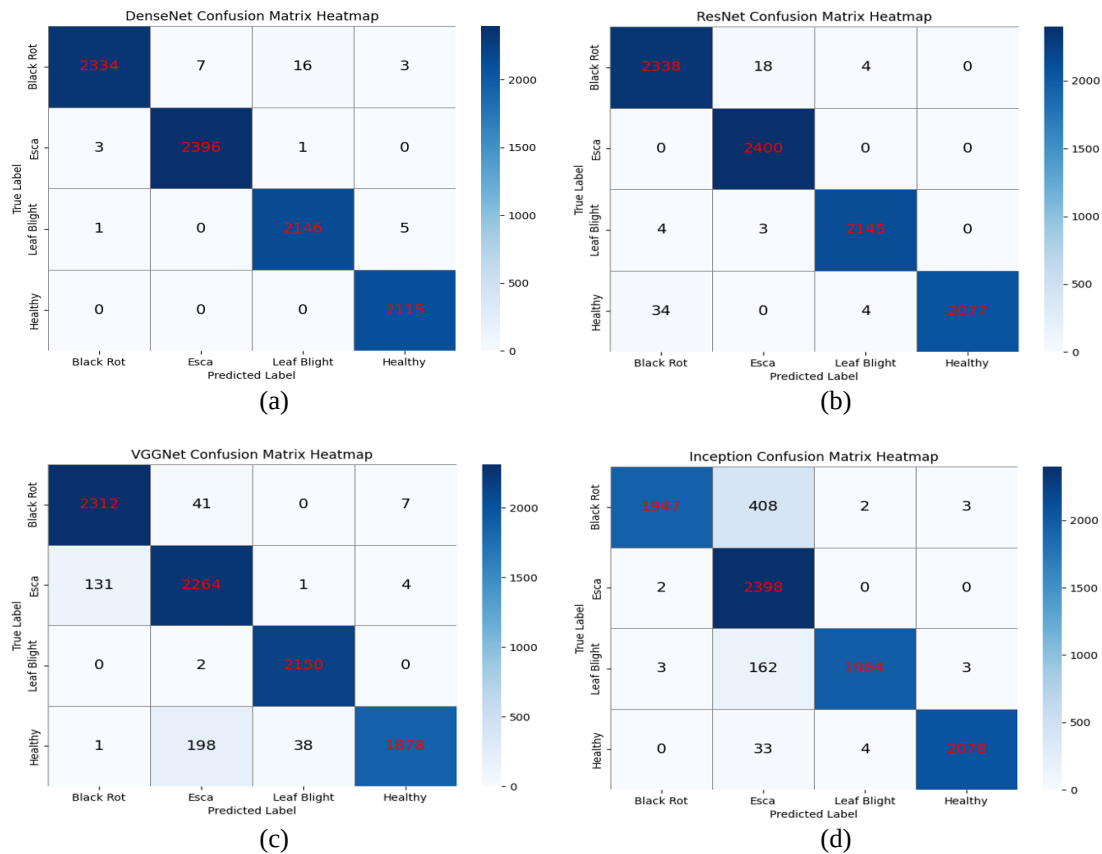


Figure 5. Confusion matrices for each model at their optimal split based on external test performance: (a) DenseNet121 (80:10:10): 99.6% accuracy, near-perfect classification, (b) ResNet50 (50:25:25): 99.3% accuracy, strong performance, (c) VGG16 (70:15:15): 95.3% accuracy, healthy leaf misclassifications, and (d) InceptionV3 (70:15:15): 93.1% accuracy, substantial Black Rot misclassifications

The performance metrics of DenseNet121, ResNet50, VGG16, and InceptionV3 on the external test dataset are summarized in Table 3. DenseNet121 achieved the highest accuracy (99.60%) and lowest loss (0.0197), while ResNet50 offered the fastest inference (0.004222 s) and highest throughput (236.85 images/sec) at a marginal 0.34% accuracy cost, representing a 20% throughput advantage over DenseNet121. This positions ResNet50 as preferable for real-time or high-throughput deployment such as real-time drone surveillance, and DenseNet121 as preferable where accuracy and model footprint (27.8 MB) are the priority such as mobile apps for on-device inference. On the other hand, VGG16 and InceptionV3 offered no compelling efficiency or accuracy advantage over either architecture, with VGG16's large parameter count (134.3M and 524 MB) and InceptionV3's weaker external accuracy (93.13%) and slowest inference (0.011168 s) limiting their practical deployment value.

Table 3. Comparison results for CNN models on external dataset

Model	Loss	Accuracy	Precision	Recall	F1-score	Inference time (sec)	Throughput (images/sec)	Total parameters	Model size (KB)
ResNet50	0.0235	0.9926	0.9926	0.9926	0.9926	0.004222	236.85	23,516,228	92,176
DenseNet121	0.0197	0.9960	0.9960	0.9960	0.9960	0.005065	197.44	6,957,956	27,783
VGG16	0.1267	0.9531	0.9547	0.9531	0.9530	0.006269	159.52	134,276,932	524,532
InceptionV3	0.1990	0.9313	0.9445	0.9313	0.9325	0.011168	89.54	24,354,536	95,484

3.1. Model interpretability

Figure 6 shows Grad-CAM heatmaps across all four disease classes. DenseNet121 and ResNet50 are the top-performing models by consistently localizing disease-relevant regions, aligning well with known disease symptoms [10], [17], [42]. In contrast, VGG16 frequently attended to irrelevant background regions

rather than leaf-specific features, indicating weaker disease-specific feature capture. Similarly, InceptionV3 showed the least reliable localization, particularly misaligning focus for Black Rot and Healthy classes, consistent with its confusion-matrix errors and supporting the interpretation that its poor external generalization reflects genuine feature-learning limitations rather than only dataset-specific noise.

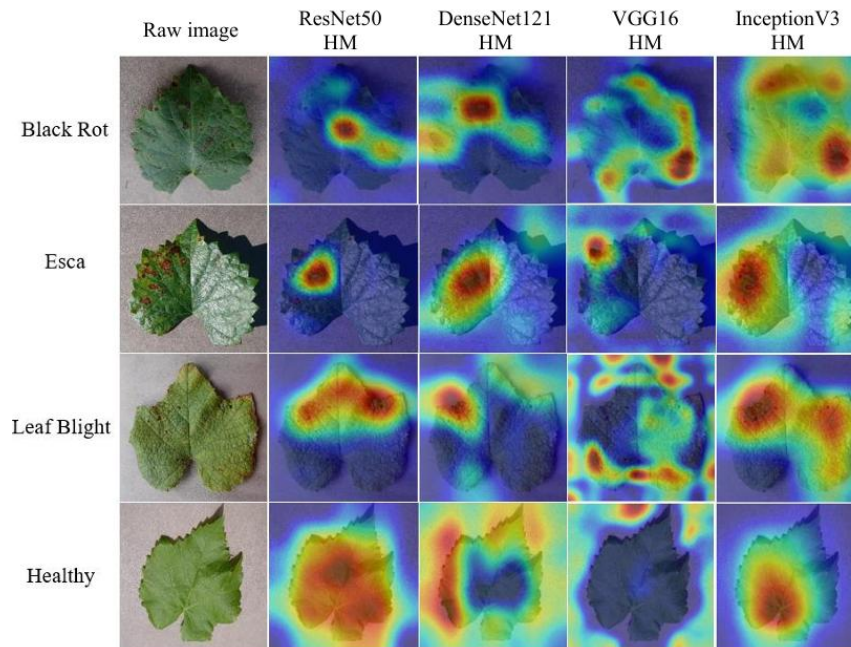


Figure 6. Grad-CAM heatmaps of model focus areas

This study was systematically compared with related work in terms of technique, data split strategy, XAI use, and external validation as shown in Table 4. Unlike most prior studies, which use a single fixed split ratio and rarely validate externally [20], [21], [23], [30], this work evaluates three split ratios per architecture and validates all four models on a fully independent external dataset, while additionally applying Grad-CAM interpretability analysis across both splits and architectures.

Table 4. Comparison of the proposed study with other state-of-arts methods

Reference	Techniques used	Accuracy (%)	Data split ratio	Crop	XAI method	External dataset testing	Dataset size
[20]	DenseNet201 Plus	90.12 99.5	80:10:10	Banana Black gram	Grad-CAM++	No	1289 1007
[21]	ResNet50	99.8	80:10:10	38 plant categories	No	No	70295
[22]	VGG16	97.22 98.75	80:10:10	Rice Wheat	No	Yes	120/80 (Training) 50 (Ext. testing)
[23]	SECNN	99.28	80:10:10	Fruit	No	No	5100
[27]	EfficientNetV2B2	98.96	80:10:10	Tomato	Grad-CAM LIME	No	10000
[30]	Decision tree classifiers Random forests classifier	90 94	60:40	Tomato	No	No	500
This work	DenseNet121	99.60	80:10:10	Grape	Grad-CAM	Yes	3251
	ResNet50	99.26	50:25:25				(training)
	VGG16	95.31	70:15:15				9027 (Ext. testing)
	InceptionV3	93.13	70:15:15				(testing)

3.2. Limitations

This study has several limitations. The PlantVillage dataset's controlled imaging conditions may not fully reflect field variability, and some domain shift is expected between the training and external test distributions. The analysis is limited to four grape disease classes, and reported inference times are hardware-specific (RTX 3060) and may vary across deployment platforms. Additionally, while three runs per split-model combination provide an initial basis for statistical comparison, a larger number of runs would improve the robustness of the reported CIs and significance tests.

4. CONCLUSION

This study evaluated three data split ratios (80:10:10, 70:15:15, 50:25:25) across four CNN architectures for grape leaf disease classification, training on PlantVillage and validating externally on an independent Kaggle dataset. Two implications stand out. First, external validation is essential, several models performed well internally but generalized quite differently on the independent dataset, exposing overfitting that internal metrics alone missed. Second, split-ratio sensitivity is architecture-dependent, not universal; some architectures generalized consistently regardless of partitioning, while others varied substantially, meaning split-robustness should be evaluated alongside accuracy when selecting a model. For deployment, architecture choice should follow operational priority, where the most stable, accurate architecture suits accuracy-critical, resource-constrained applications, while the fastest architecture suits throughput-critical, real-time applications.

Future work should test these patterns under field-realistic conditions including variable lighting, occlusion, and mixed infections, and should extend the approach to other crops to check whether architecture-dependent split sensitivity generalizes. Calibration analysis will also be incorporated to evaluate the reliability of prediction confidence, alongside uncertainty estimation to support decision-making in agricultural contexts where misclassification costs are high. Additionally, future work will benchmark performance against vision transformers (ViT, Swin Transformer) and foundation models fine-tuned for agricultural applications. For mobile deployment specifically, lightweight should be defined against explicit constraints, a target model size and inference latency for a stated device class, rather than used as a general descriptor.

FUNDING INFORMATION

This research received no external funding.

AUTHOR CONTRIBUTIONS STATEMENT

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

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Wasan Alamro	✓	✓		✓	✓	✓	✓	✓	✓	✓		✓	✓	
Gang Li		✓	✓			✓		✓		✓	✓			

C : **C**onceptualization

M : **M**ethodology

So : **S**oftware

Va : **V**alidation

Fo : **F**ormal analysis

I : **I**nvestigation

R : **R**esources

D : **D**ata Curation

O : Writing - **O**riginal Draft

E : Writing - Review & **E**ditng

Vi : **V**isualization

Su : **S**upervision

P : **P**roject administration

Fu : **F**unding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are openly available in [Kaggle] at <https://www.kaggle.com/datasets/mohitsingh1804/plantvillage?resource=download> and <https://www.kaggle.com/datasets/rm1000/grape-disease-dataset-original>. The source code and trained models for this study are available from the corresponding author [initials: WA] upon reasonable request.

REFERENCES

- [1] H. Sandhu, "Bottom-up transformation of agriculture and food systems," *Sustainability*, vol. 13, no. 4, pp. 1–13, Feb. 2021, doi: 10.3390/su13042171.
- [2] L. Liu *et al.*, "A multi-scale feature fusion neural network for multi-class disease classification on the maize leaf images," *Heliyon*, vol. 10, no. 7, 2024, doi: 10.1016/j.heliyon.2024.e28264.
- [3] B. Tugrul, E. Elfatimi, and R. Eryigit, "Convolutional Neural Networks in Detection of Plant Leaf Diseases: A Review," *Agriculture*, vol. 12, no. 8, pp. 1–21, Aug. 2022, doi: 10.3390/agriculture12081192.
- [4] C. Bertsch *et al.*, "Grapevine trunk diseases: Complex and still poorly understood," *Plant Pathology*, vol. 62, no. 2, pp. 243–265, Apr. 2013, doi: 10.1111/j.1365-3059.2012.02674.x.
- [5] J. Xiao, A. K. Aggarwal, U. K. Rage, V. Katiyar, and R. Avtar, "Deep Learning-Based Spatiotemporal Fusion of Unmanned Aerial Vehicle and Satellite Reflectance Images for Crop Monitoring," *IEEE Access*, vol. 11, pp. 85600–85614, 2023, doi: 10.1109/ACCESS.2023.3297513.
- [6] A. Bhujel, N. E. Kim, E. Arulmozhi, J. K. Basak, and H. T. Kim, "A Lightweight Attention-Based Convolutional Neural Networks for Tomato Leaf Disease Classification," *Agriculture*, vol. 12, no. 2, pp. 1–18, Feb. 2022, doi: 10.3390/agriculture12020228.
- [7] I. Pacal *et al.*, "A systematic review of deep learning techniques for plant diseases," *Artificial Intelligence Review*, vol. 57, no. 11, pp. 1–39, Sep. 2024, doi: 10.1007/s10462-024-10944-7.
- [8] A. Arabiat and M. Altayeb, "Driving behavior analytics: an intelligent system based on machine learning and data mining techniques," *Bulletin of Electrical Engineering and Informatics*, vol. 14, no. 3, pp. 2055–2065, Jun. 2025, doi: 10.11591/eei.v14i3.9095.
- [9] D. Amilo *et al.*, "Hybrid machine learning fractional-order framework for mammographic tumor characterization integrating growth dynamics and malignancy prediction," *Discover Artificial Intelligence*, vol. 6, no. 587, pp. 1–27, 2026, doi: 10.1007/s44163-026-01310-3.
- [10] M. E. H. Chowdhury *et al.*, "Automatic and Reliable Leaf Disease Detection Using Deep Learning Techniques," *AgriEngineering*, vol. 3, no. 2, pp. 294–312, May 2021, doi: 10.3390/agriengineering3020020.
- [11] M. Türkoğlu and D. Hanbay, "Plant disease and pest detection using deep learning-based features," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 27, no. 3, pp. 1636–1651, May 2019, doi: 10.3906/elk-1809-181.
- [12] A. M. Mishra *et al.*, "A deep learning-based novel approach for weed growth estimation," *Intelligent Automation and Soft Computing*, vol. 31, no. 2, pp. 1157–1173, 2022, doi: 10.32604/iasc.2022.020174.
- [13] N. K. Trivedi *et al.*, "Early detection and classification of tomato leaf disease using high-performance deep neural network," *Sensors*, vol. 21, no. 23, pp. 1–15, Nov. 2021, doi: 10.3390/s21237987.
- [14] M. Shoaib *et al.*, "An advanced deep learning models-based plant disease detection: A review of recent research," *Frontiers in Plant Science*, vol. 14, pp. 1–22, Mar. 2023, doi: 10.3389/fpls.2023.1158933.
- [15] I. Kunduracioglu and I. Pacal, "Advancements in deep learning for accurate classification of grape leaves and diagnosis of grape diseases," *Journal of Plant Diseases and Protection*, vol. 131, no. 3, pp. 1061–1080, Jun. 2024, doi: 10.1007/s41348-024-00896-z.
- [16] D. Xu, H. Zhao, O. M. Lawal, X. Lu, R. Ren, and S. Zhang, "An Automatic Jujube Fruit Detection and Ripeness Inspection Method in the Natural Environment," *Agronomy*, vol. 13, no. 2, pp. 1–12, Feb. 2023, doi: 10.3390/agronomy13020451.
- [17] S. Rakesh and M. Indiramma, "Explainable AI for Crop disease detection," in *2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, Greater Noida, India: IEEE, Dec. 2022, pp. 1601–1608, doi: 10.1109/ICAC3N56670.2022.10074303.
- [18] E. Dağlarlı, "Explainable Artificial Intelligence (XAI) Approaches and Deep Meta-Learning Models," in *Advances and Applications in Deep Learning*, IntechOpen, 2020, doi: 10.5772/intechopen.92172.
- [19] K. Wei *et al.*, "Explainable Deep Learning Study for Leaf Disease Classification," *Agronomy*, vol. 12, no. 5, pp. 1–15, 2022, doi: 10.3390/agronomy12051035.
- [20] M. K. A. Mazumder, M. M. Kabir, A. Rahman, M. Abdullah-Al-Jubair, and M. F. Mridha, "DenseNet201Plus: Cost-effective transfer-learning architecture for rapid leaf disease identification with attention mechanisms," *Heliyon*, vol. 10, no. 15, p. e35625, Aug. 2024, doi: 10.1016/j.heliyon.2024.e35625.
- [21] I. Z. Mukti and D. Biswas, "Transfer Learning Based Plant Diseases Detection Using ResNet50," in *2019 4th International Conference on Electrical Information and Communication Technology (EICT)*, Khulna, Bangladesh: IEEE, Dec. 2019, pp. 1–6, doi: 10.1109/EICT48899.2019.9068805.
- [22] Z. Jiang, Z. Dong, W. Jiang, and Y. Yang, "Recognition of rice leaf diseases and wheat leaf diseases based on multi-task deep transfer learning," *Computers and Electronics in Agriculture*, vol. 186, p. 106184, Jul. 2021, doi: 10.1016/j.compag.2021.106184.
- [23] B. N. Naik, R. Malmathanraj, and P. Palanisamy, "Detection and classification of chilli leaf disease using a squeeze-and-excitation-based CNN model," *Ecological Informatics*, vol. 69, p. 101663, Jul. 2022, doi: 10.1016/j.ecoinf.2022.101663.
- [24] P. Sahu, A. Chug, A. P. Singh, D. Singh, and R. P. Singh, "Deep Learning Models for Beans Crop Diseases: Classification and Visualization Techniques," *International Journal of Modern Agriculture*, vol. 10, no. 1, pp. 796–812, 2021.
- [25] A. Fuentes, S. Yoon, S. C. Kim, and D. S. Park, "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition," *Sensors*, vol. 17, no. 9, pp. 1–21, Sep. 2017, doi: 10.3390/s17092022.
- [26] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, Feb. 2018, doi: 10.1016/j.compag.2018.01.009.
- [27] A. Debnath *et al.*, "A Smartphone-Based Detection System for Tomato Leaf Disease Using EfficientNetV2B2 and Its Explainability with Artificial Intelligence (AI)," *Sensors*, vol. 23, no. 21, pp. 1–21, Oct. 2023, doi: 10.3390/s23218685.
- [28] A. Kaur, K. Guleria, and N. K. Trivedi, "A deep learning-based model for biotic rice leaf disease detection," *Multimedia Tools and Applications*, vol. 83, no. 36, pp. 83583–83609, Mar. 2024, doi: 10.1007/s11042-024-18730-x.
- [29] A. A. Binguitcha-Fare and P. Sharma, "Crops and weeds classification using convolutional neural networks via optimization of transfer learning parameters," *International Journal of Engineering and Advanced Technology*, vol. 8, no. 5, pp. 2284–2294, 2019.
- [30] J. Basavaiah and A. A. Anthony, "Tomato Leaf Disease Classification using Multiple Feature Extraction Techniques," *Wireless Personal Communications*, vol. 115, no. 1, pp. 633–651, Nov. 2020, doi: 10.1007/s11277-020-07590-x.
- [31] C. R. Dhivyaa, N. Kandasamy, and S. Rajendran, "Integration of dilated convolution with residual dense block network and multi-level feature detection network for cassava plant leaf disease identification," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 11, May 2022, doi: 10.1002/cpe.6879.
- [32] H. N. Ngugi, A. E. Ezugwu, A. A. Akinyelu, and L. Abualigah, "Revolutionizing crop disease detection with computational deep learning: a comprehensive review," *Environmental Monitoring and Assessment*, vol. 196, no. 3, p. 302, Mar. 2024, doi: 10.1007/s10661-024-12454-z.

- [33] A. A. Ahmed and G. H. Reddy, "A Mobile-Based System for Detecting Plant Leaf Diseases Using Deep Learning," *AgriEngineering*, vol. 3, no. 3, pp. 478–493, Jul. 2021, doi: 10.3390/agriengineering3030032.
- [34] M. Singh, "PlantVillage," *Kaggle*. <https://www.kaggle.com/datasets/mohitsingh1804/plantvillage>. (Accessed Jan. 15, 2025).
- [35] R. Mandal, "Grapevine Disease Dataset (Original)," *Kaggle*. <https://www.kaggle.com/datasets/rm1000/grape-disease-dataset-original>. (Accessed Jul. 03, 2025).
- [36] S. Chauhan, M. Singh, and A. K. Aggarwal, "Experimental Analysis of Effect of Tuning Parameters on the Performance of Diversity-Driven Multi-Parent Evolutionary Algorithm," in *2021 IEEE 2nd International Conference on Electrical Power and Energy Systems (ICEPES)*, Bhopal, India: IEEE, Dec. 2021, pp. 1–6, doi: 10.1109/ICEPES52894.2021.9699655.
- [37] S. Amritavarshini, K. S. Ganesh, M. Arvind, K. R. Teja, and D. Vijayan, "Image-Based Plant Leaf Disease Detection using Deep Learning," in *2024 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IconSCEPT)*, Karaikal, India: IEEE, Jul. 2024, pp. 1–6, doi: 10.1109/IconSCEPT61884.2024.10627918.
- [38] N. Nigar, H. M. Faisal, M. Umer, O. Oki, and J. M. Lukose, "Improving Plant Disease Classification with Deep-Learning-Based Prediction Model Using Explainable Artificial Intelligence," *IEEE Access*, vol. 12, pp. 100005–100014, 2024, doi: 10.1109/ACCESS.2024.3428553.
- [39] O. A. D. Ammar *et al.*, "Plant Leaf Disease Detection Using Ensemble Learning and Explainable AI," *IEEE Access*, vol. 12, pp. 156038–156049, 2024, doi: 10.1109/ACCESS.2024.3484574.
- [40] N. Shukla, S. Palwe, Shubham, M. Rajani, and A. Suri, "Plant Disease Detection and Localization using GRADCAM," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 6, pp. 3069–3075, Mar. 2020, doi: 10.35940/ijrte.e6935.038620.
- [41] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, Honolulu, HI, USA: IEEE, Jul. 2017, pp. 2261–2269, doi: 10.1109/CVPR.2017.243.
- [42] K. Mridha *et al.*, "Explainable Deep Learning for Coffee Leaf Disease Classification in Smart Agriculture: A Visual Approach," in *2023 International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*, Ballar, India: IEEE, Apr. 2023, pp. 1–8, doi: 10.1109/ICDCECE57866.2023.10151399.

BIOGRAPHIES OF AUTHORS



Wasan Alamro    is an Assistant Professor at the Department of Communications and Computer Engineering, Al-Ahliyya Amman University, Amman, Jordan, where she has been a faculty member since 2024. She graduated with a first-class honors B.Eng. degree in Electrical Engineering-Telecommunications from Mu'tah University, Jordan, in 2014, and M.Sc. in Wireless Communication Engineering from University of Jordan, Jordan in 2018. She then received the International Scholarship from the New Zealand government in 2020 and completed her Ph.D. in Electrical Engineering from Auckland University of Technology, Auckland, New Zealand, in 2024. Her research interest includes deep learning, machine learning, antenna design, and microwave imaging. She can be contacted at email: w.alamro@ammanu.edu.jo or twasan@ymail.com.



Gang Li    received his B.Eng. degree in Material Science and Engineering from Hunan University, China, in 2003. He later earned an M.S. degree in Information Technology from Auckland Institute of Studies, New Zealand, in 2024. Currently, he is a Senior Software Engineer at Brunton NZ Ltd, where he applies his expertise in advanced computational techniques. His research interests span across multiple domains, including pattern recognition, autonomous driving, flight control, and the application of artificial intelligence in addition-cure silicone rubber material. With a strong interdisciplinary background bridging materials science and information technology, he is dedicated to exploring innovative solutions at the intersection of artificial intelligence and engineering applications. He can be contacted at email: ganglichris@gmail.com.