agriculture

MDPI

Article

Boosting Rice Disease Diagnosis: A Systematic Benchmark of Five Deep Convolutional Neural Network Models in Precision Agriculture

Shu-Hung Lee [1], Qi-Wei Jiang [2], Chia-Hsin Cheng [2,*], Yu-Shun Tsai [2] and Yung-Fa Huang [3,*]

[1] School of Intelligent Manufacturing and Automotive Engineering, Guangdong University of Business and Technology, Zhaoqing 526020, China; umaillee@gmail.com
[2] Department of Electrical Engineering, National Formosa University, Yunlin 632301, Taiwan; wyane1222@gmail.com (Q.-W.J.); nfumsee@gmail.com (Y.-S.T.)
[3] Department of Information and Communication Engineering, Chaoyang University of Technology, Taichung 413310, Taiwan
* Correspondence: chcheng@nfu.edu.tw (C.-H.C.); yfahuang@cyut.edu.tw (Y.-F.H.); Tel.: +886-4-2332-3000 (Y.-F.H.)

Abstract

Rice diseases pose a critical threat to global food security. While deep learning offers a promising path toward automated diagnosis, clear guidelines for model selection in resource-constrained agricultural environments are still lacking. This study presents a systematic benchmark of five deep convolutional neural networks (CNNs)—Visual Geometry Group (VGG)16, VGG19, Residual Network (ResNet)101V2, Xception, and Densely Connected Convolutional Network (DenseNet)121—for rice disease identification using a public leaf image dataset. The models, initialized with ImageNet pre-trained weights, were rigorously evaluated under a unified framework, including 5-fold cross-validation and a challenging out-of-distribution (OOD) generalization test. Our results demonstrate a clear performance hierarchy, with DenseNet121 emerging as the superior model. It achieved the highest OOD accuracy and F1-score (both 85.08%) while exhibiting the greatest parameter efficiency (8.1 million parameters), making it ideally suited for edge deployment. In contrast, architectures with large fully connected layers (VGG) or less efficient feature learning mechanisms (Xception, ResNet101V2) showed lower performance in this specific task. This study confirms the critical impact of architectural design choices, provides a reproducible performance baseline, and identifies DenseNet121 as a robust, efficient, and highly recommendable CNN for practical rice disease diagnosis in precision agriculture.

Keywords: smart agriculture; rice diseases; deep learning; convolutional neural networks (CNNs); densenet; model benchmarking; out-of-distribution generalization

Academic Editors: Maciej Zaborowicz and Francesco Marinello

Received: 9 October 2025
Revised: 19 November 2025
Accepted: 28 November 2025
Published: 30 November 2025

Citation: Lee, S.-H.; Jiang, Q.-W.; Cheng, C.-H.; Tsai, Y.-S.; Huang, Y.-F. Boosting Rice Disease Diagnosis: A Systematic Benchmark of Five Deep Convolutional Neural Network Models in Precision Agriculture. *Agriculture* **2025**, *15*, 2494. https://doi.org/10.3390/agriculture15232494

1. Introduction

Rice stands as a critical staple crop, playing a pivotal role in ensuring global food security. However, its production is persistently threatened by various diseases, leading to substantial economic losses and reduced yields. Plant diseases alone are responsible for staggering global economic losses estimated at approximately $220 billion annually [1]. To mitigate these challenges, it is imperative to equip farmers with the skills and knowledge for early disease identification and prevention, promote scientific cultivation methods, and provide advanced technical support [2,3]. This is particularly crucial in regions like Taiwan, where hot and humid climates create conducive environments for the spread of diseases

such as rice blast, sheath blight, and bacterial leaf blight [4]. Early detection remains a significant hurdle, often resulting in considerable yield losses by the time symptoms become visible and control measures are implemented [5].

The projected growth of the global population to 9.7 billion by 2050 further underscores the urgency for sustainable and efficient agricultural practices [6]. In this context, artificial intelligence (AI), particularly deep learning, offers a promising avenue for automating crop disease diagnosis [7,8]. Among deep learning techniques, Convolutional Neural Networks (CNNs) have proven highly effective for image classification tasks by leveraging spatial hierarchies in visual data [9]. While Vision Transformer (ViT) models have recently emerged as powerful alternatives achieving state-of-the-art performance on large-scale benchmarks [10], their practical deployment in resource-constrained agricultural settings is often hampered by high computational demands and memory consumption [11,12]. Consequently, well-established CNN architectures remain highly competitive for practical agricultural applications due to their computational efficiency, lower memory footprint, and proven effectiveness on medium-sized datasets [13]. This study, therefore, deliberately focuses on establishing a comprehensive benchmark of representative and diverse CNN architectures.

The CNN models selected for this benchmark—VGG16, VGG19, ResNet101V2, Xception, and DenseNet121—represent key evolutionary milestones in deep learning for computer vision. The VGG models [14] are classic architectures that demonstrated the importance of network depth using small convolutional filters. ResNet101V2 [15], an improved version of the pioneering Residual Network, introduced residual learning with skip connections to enable the training of very deep networks. The Xception model [16] extends the idea of Inception by using depthwise separable convolutions to decouple spatial and channel-wise feature learning. Finally, DenseNet121 [17] employs dense connectivity between layers, encouraging feature reuse and achieving high parameter efficiency. By comparing these architectures—which exemplify distinct design philosophies such as plain stacking (VGG), residual learning (ResNet), efficient convolution (Xception), and dense connectivity (DenseNet)—we aim to provide a holistic understanding of their suitability for the specific task of rice disease identification.

This study proposes a deep learning-based image recognition solution and systematically benchmarks the five aforementioned CNN architectures. The primary contributions of this work are threefold:

1. It provides a rigorous and reproducible performance baseline under a unified framework, including 5-fold cross-validation and a challenging Out-of-Distribution (OOD) generalization test.
2. It delivers a clear performance hierarchy and architectural analysis, identifying DenseNet121 as the superior model in terms of both accuracy and parameter efficiency, making it ideally suited for edge deployment.
3. It establishes a crucial foundational benchmark for the community, serving as a reference point for future studies exploring more complex models, including transformers, in this specific domain.

The rest of this paper is organized as follows. Section 2 provides a literature review and describes the dataset and deep learning algorithms used. Section 3 details the system architecture and experimental setup. Section 4 presents a comparative analysis of the five deep learning models and discusses the results. Finally, Section 5 concludes the paper.

2. Literature Review

This section provides a concise overview of the dataset utilized for model training and introduces the fundamental principles of the deep learning algorithms employed in this study.

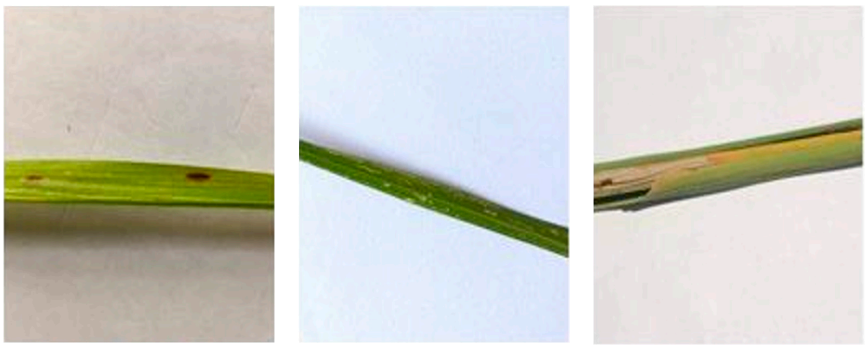
2.1. Rice Leaf Diseases Dataset

The dataset used in this study is sourced from the publicly available Kaggle's Rice disease Dataset (Pereira) [18]. It was selected for its larger scale and more comprehensive sample collection compared to other potential sources, such as the UCI dataset [19], which helps mitigate overfitting and provides a more robust foundation for training deep learning models.

The dataset comprises a total of 2092 images, evenly distributed across four classes with 523 images each. The classes include three prevalent rice diseases—Leaf Blast, Brown Spot, and Hispa—along with a class of Healthy leaves. The original dataset was partitioned by the uploader into training and testing sets. For the purposes of this study, the original training set was further subdivided into a dedicated training subset and a validation subset to facilitate training monitoring and prevent overfitting. This substantial data volume allows for a more reliable evaluation of model performance and generalization ability.

The visual symptoms of the three targeted rice diseases are summarized as follows:

1. Brown spot: It is characterized by elliptical lesions that align with the leaf veins. The center of the lesion is dark brown, surrounded by a red-brown or yellow-brown margin, often appearing in patchy distributions (Figure 1a).
2. Hispa: It manifests as long, narrow, whitish or silvery streaks on leaves caused by insect scraping. Severe infections lead to leaf drying, curling, and browning (Figure 1b).
3. Leaf Blast: This typically produces spindle-shaped lesions on leaves with whitish or gray centers and dark brown to reddish margins. It can also infect the panicle neck, causing rot and yield loss (Figure 1c).



(**a**) Brown spot (**b**) Hispa (**c**) Leaf blast

Figure 1. Representative image samples of the three rice leaf diseases from the dataset: (**a**) Brown Spot, (**b**) Hispa, and (**c**) Leaf Blast.

A summary of the disease characteristics is provided in Table 1.

Table 1. Summary of disease characteristics in rice.

Disease	Shape	Location	Color
Brown spot	Elliptical spots	Leaf surface	Dark brown center, red-brown or yellow-brown surround
Hispa	Linear streaks	Leaf surface	White, silvery
Leaf blast	Spindle-shaped spots	Leaf surface	Gray center, brown margins

2.2. Deep Learning

Deep Learning, a subfield of Artificial Intelligence (AI) and Machine Learning (ML), is founded on Artificial Neural Networks (ANNs) that simulate the interconnections of neurons in the human brain. Its multi-layered architecture—comprising input, hidden, and output layers—enables the hierarchical learning of complex feature representations from data [20]. Convolutional Neural Networks (CNNs), a cornerstone of deep learning, are particularly powerful for processing spatial data and are widely used in applications such as image recognition, speech analysis, and natural language processing.

Activation functions introduce non-linearity into neural networks, enabling them to learn complex patterns. While Sigmoid and Hyperbolic Tangent (Tanh) functions were historically popular, they are prone to the vanishing gradient problem in deep networks. The Rectified Linear Unit (ReLU) function, defined as $f(x) = max(0, x)$, has become the default choice in many CNN architectures due to its computational efficiency and its ability to mitigate the vanishing gradient issue, thereby accelerating convergence [21].

2.2.1. VGG

The VGG (Visual Geometry Group) model, proposed by Simonyan and Zisserman [14], is a classic CNN architecture. Its distinctive characteristic is the systematic use of small 3×3 convolutional kernels and max-pooling layers to increase network depth, thereby enhancing feature learning within a relatively simple and uniform structural design. The model has two main variants, VGG16 and VGG19, which consist of 13 and 16 convolutional layers, respectively, each followed by 3 fully-connected layers (Figure 2). While achieving strong performance in their time, the large number of parameters in the fully-connected layers makes VGG models computationally expensive and memory-intensive compared to more modern architectures [22].

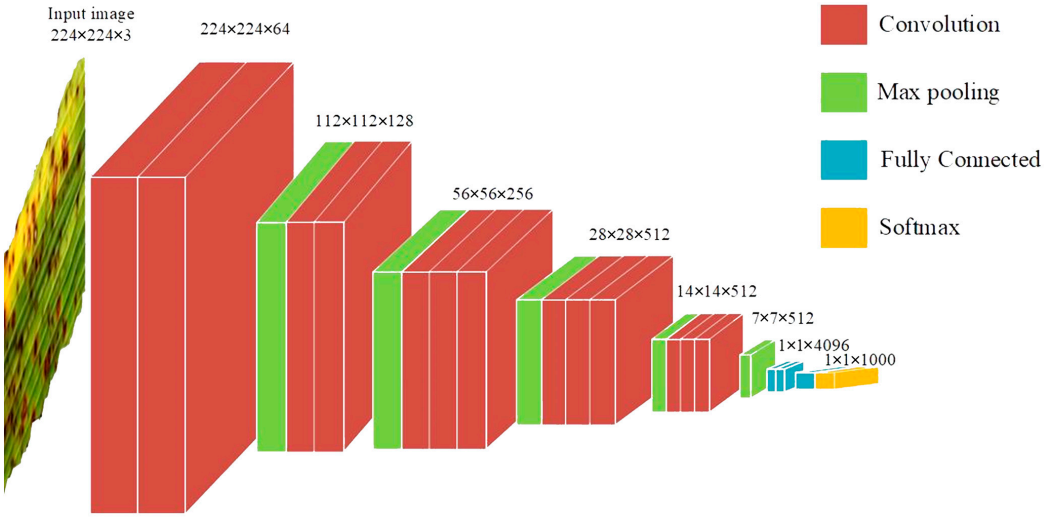


Figure 2. Schematic architecture of the VGG16 model, illustrating its sequential stack of convolutional and fully-connected layers. (Adapted from [14]).

2.2.2. ResNet

The Residual Network (ResNet), proposed by He et al. [15], was designed to address the degradation problem (increasing training error) in very deep networks. Its core innovation is the residual learning block, which incorporates a shortcut connection (identity mapping) between layers (Figure 3). This skip connection allows the network to perform identity mapping by default, making it easier to learn residual functions with reference to the layer inputs. The derivative of the shortcut path is 1, which prevents the issue of vanishing gradients during backpropagation by ensuring a direct flow of gradients [23].

This enables the successful training of networks that are significantly deeper than VGG. In this paper, we leverage the ResNet101V2 architecture, an improved version that uses pre-activation (Batch Normalization and ReLU before the weights) within the residual blocks, enhancing trainability and generalization capability [15].

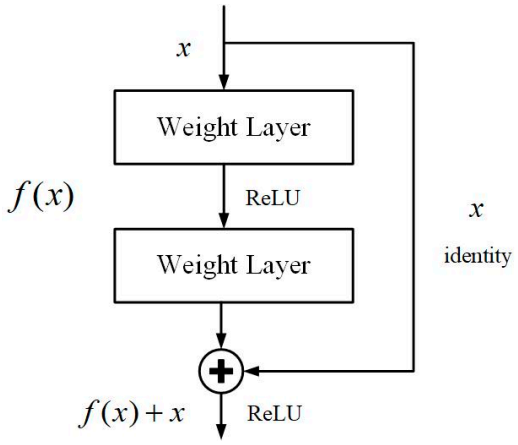


Figure 3. Diagram of a residual learning block with skip connection, the fundamental building block of ResNet architectures (adapted from [15]).

2.2.3. Xception

The Xception architecture, proposed by Chollet [16], builds upon the Inception architecture by replacing standard convolutions with depthwise separable convolutions [24]. This operation factorizes a standard convolution into two separate steps: (1) a depthwise convolution that applies a single filter per input channel to capture spatial features, and (2) a pointwise convolution (a 1×1 convolution) that combines the outputs from the depthwise step across channels. This factorization drastically reduces computational cost and the number of parameters while often achieving equal or better performance [24]. Xception leverages these separable convolutions in a deep network architecture with residual connections, making it both highly efficient and powerful (Figure 4).

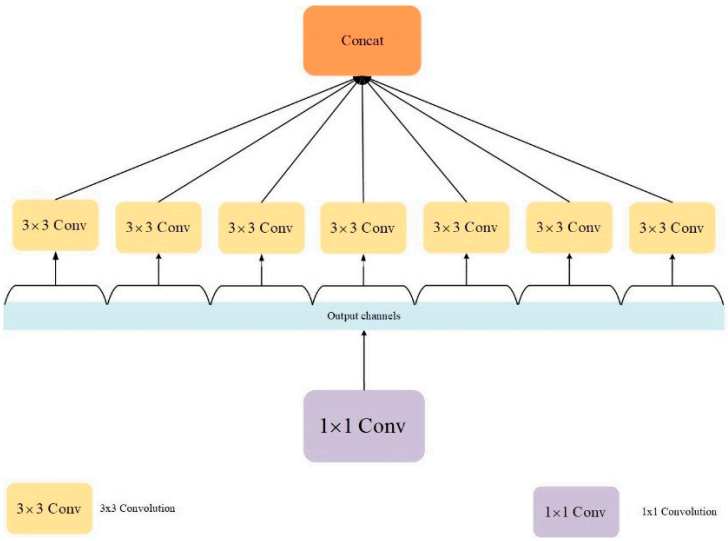


Figure 4. The Xception network architecture, primarily composed of depthwise separable convolution layers with residual connections (adapted from [16]).

2.2.4. DenseNet

The Densely Connected Convolutional Network (DenseNet), proposed by Huang et al. [17], introduces the Dense Block as its core innovation. Within a Dense Block, each

layer is connected to every other layer in a feed-forward fashion (Figure 5). Specifically, the feature maps of all preceding layers are concatenated and used as inputs for each subsequent layer. This dense connectivity encourages feature reuse throughout the network, alleviates the vanishing gradient problem by providing shorter paths for gradient flow, and inherently improves parameter efficiency by reducing the need to relearn redundant feature maps [17]. In this study, we employ the DenseNet121 architecture, which offers an excellent balance between performance and computational cost.

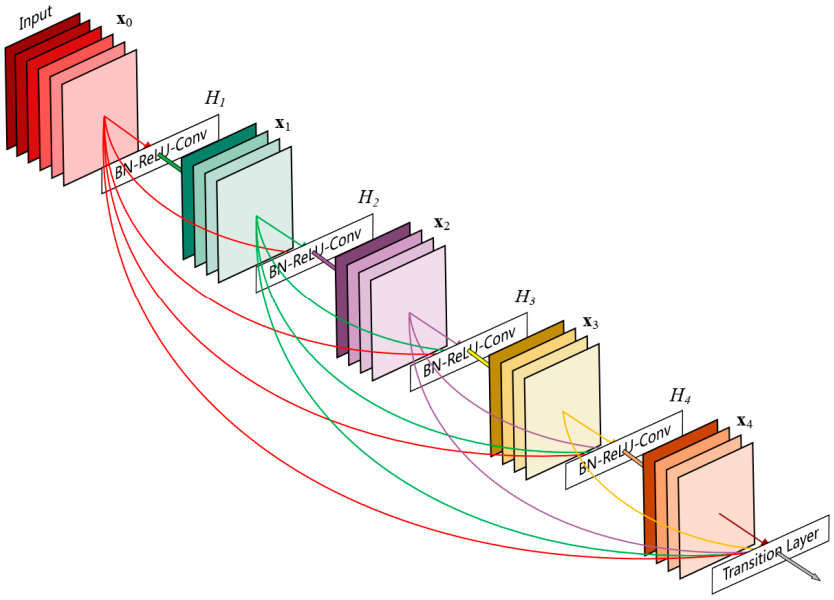


Figure 5. The dense connectivity pattern within a Dense Block of DenseNet, where each layer receives feature maps from all preceding layers (adapted from [17]). The colored arc arrows (Red, Green, Purple, Yellow) illustrate the Dense Connectivity pattern, where the feature maps from all preceding layers are concatenated and fed as input to the current layer (H_l). Each color signifies the connection originating from a specific layer (e.g., Red for x_0, Green for x_1, etc.). The varying colors of the feature map stacks, transitioning from Red x_0 to Brown/Orange x_4, visually represent the flow of data processing and feature extraction progression through the network.

2.2.5. Loss Function and Optimizer

In deep learning, the loss function quantifies the discrepancy between the model's predictions and the actual values. For our multi-class classification task, the Categorical Cross-Entropy (CE) loss is adopted. The training objective is to minimize this loss, which measures the dissimilarity between the true label distribution and the predicted probability distribution [25].

The optimizer is the algorithm that adjusts the model's weights to minimize the loss function. This study utilizes the Adam (Adaptive Moment Estimation) optimizer [26]. Adam combines the advantages of Momentum and RMSprop by computing adaptive learning rates for each parameter. It maintains exponentially decaying averages of past gradients (m_t) and past squared gradients (v_t), which are used for parameter updates. This approach makes Adam well-suited for problems with noisy or sparse gradients, provides stable convergence, and is considered a robust choice for a wide range of deep learning applications [27].

3. System Architecture and Experimental Setup

This section introduces the architecture and operational procedures of the rice disease detection system. A comprehensive description of the image preprocessing pipeline, data

augmentation strategies, model training configuration, and evaluation metrics is provided. Detailed implementation parameters are included to ensure reproducibility.

3.1. System Overview

The overarching architecture of the rice disease detection system, depicted in Figure 6, operates in two distinct phases: Training and Testing/Inference. This structured approach ensures a clear separation between model development and practical application.

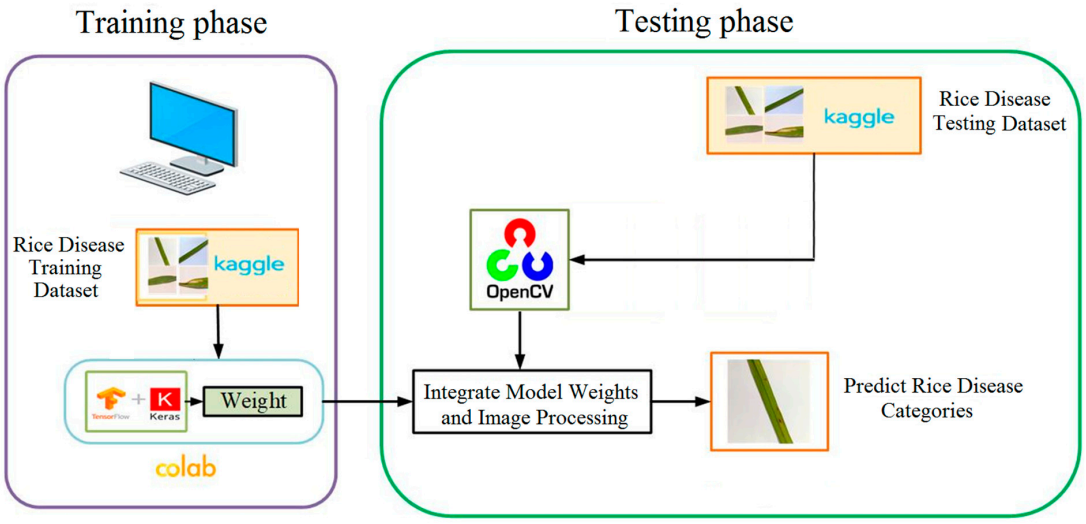


Figure 6. Schematic architecture of the rice disease detection system, illustrating the separate training and testing pipelines.

During the training phase, the models are developed and optimized using the curated Kaggle dataset described in Section 2.1. The image preprocessing, augmentation, and model fine-tuning procedures detailed in subsequent sections are executed. This process is implemented using the Keras library on the TensorFlow platform. The optimal weights obtained from this phase are saved for subsequent use.

The testing phase simulates the system's real-world application. In this phase, the saved model is loaded to classify new, unseen images. For the purpose of this study, the quantitative evaluation presented in the following sections is based on performing inference on the standardized, held-out test set described in Section 2.1. This controlled approach provides a reproducible and unbiased benchmark of the model's core diagnostic performance. The system architecture is modular, establishing a clear pathway for future integration into user-facing applications, such as a mobile app, where a farmer could submit a photo for instant analysis.

3.2. Experimental Scenarios for Robustness Evaluation

To comprehensively evaluate model robustness and generalization, we designed four distinct experimental scenarios (Case A to D). These scenarios systematically investigate the impact of data augmentation volume, out-of-distribution (OOD) generalization, and model resilience to synthetic perturbations. A unified data flowchart illustrating the sample origin and splitting strategy for all scenarios is provided in Figure 7.

A consistent foundation for all experiments was established using the same initial, class-balanced split of the core Kaggle's Rice disease Dataset [18]. The specific composition is as follows:

- Original Training Set: 320 images per class (1280 total).
- Original Validation Set: 80 images per class (320 total).
- Original Test Set: 123 images per class (492 total).

The detailed composition and specific objective of each experimental scenario are summarized in Table 2.

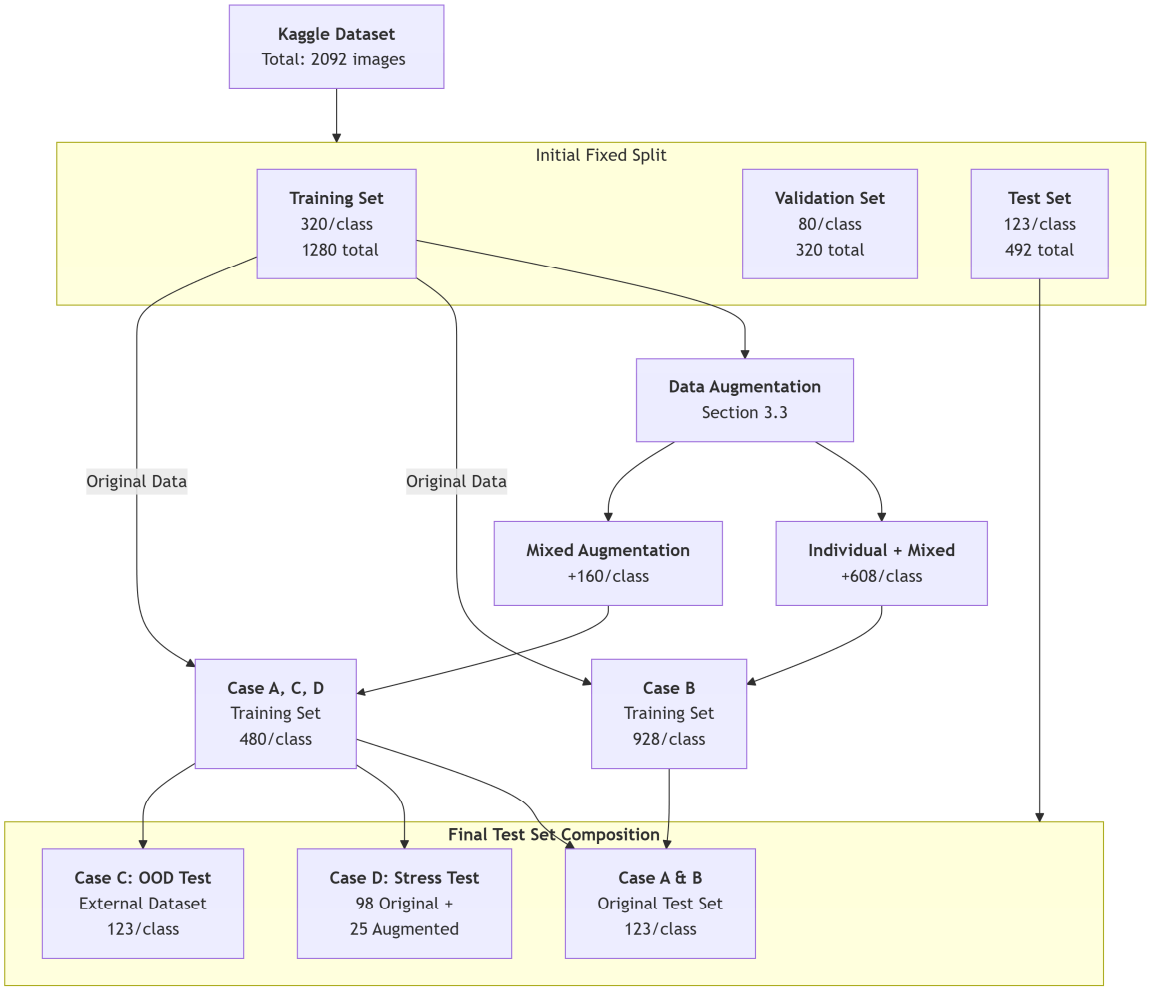


Figure 7. Data flowchart illustrating the origin and splitting strategy of datasets for all experimental scenarios (Cases A–D).

Table 2. Dataset composition and objective for each experimental scenario.

Scenario	Training Set (Per Class)	Augmentation Strategy	Test Set (Per Class)	Primary Objective
Case A (Baseline)	320 (orig.) + 160 (aug) = 480	Mixed Augmentation	123 (original)	Establish a strong baseline with standard augmentation.
Case B (Extended Aug)	320 (orig.) + 608 (aug) = 928	Mixed + Individual Augmentation (Section 3.3)	123 (original)	Probe the effect of significantly increased augmentation diversity and volume.
Case C (OOD)	Identical to Case A	Identical to Case A	123 (external, from [28,29])	Evaluate generalization to a completely unseen data domain (Most rigorous test).
Case D (Stress Test)	Identical to Case A	Identical to Case A	98 (original) + 25 (augmented) = 123	Assess robustness against synthetic variations and potential overfitting to augmentation artifacts.

3.3. Data Preprocessing and Augmentation Pipeline

The data preprocessing and augmentation techniques described in this section were applied to construct the training sets for the experimental scenarios defined in Section 3.2. A rigorous pipeline was implemented to enhance model robustness and combat overfitting.

3.3.1. Preprocessing

All input images from the original and external datasets were consistently processed as follows:

1. Resizing: Images were resized to 224 × 224 pixels, the default input size for the selected CNN architectures.
2. Normalization: Pixel values were normalized to the range [0, 1] by dividing by 255.

3.3.2. Augmentation Strategies

All preprocessing steps were applied consistently, and augmentation was performed exclusively on the designated training sets in real-time during training using the Keras ImageDataGenerator. This prevents data leakage by ensuring the validation and test sets (except for the specified synthetic additions in Case D) remain unmodified and representative of the original data distribution.

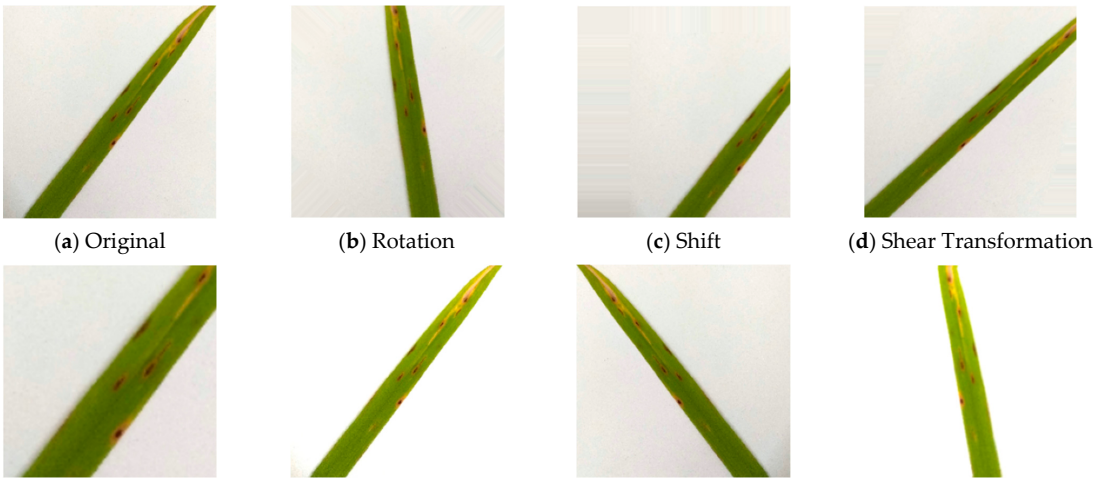
We employed two distinct augmentation strategies:

- Mixed Augmentation (for Cases A, C, and D): This strategy applied a unified transformation that randomly combined all techniques listed in Table 3 in a single pass to each original training image, generating 160 additional samples per class.
- Individual + Mixed Augmentation (for Case B): To investigate the impact of extensive augmentation diversity, this strategy first applied the mixed augmentation. Furthermore, each of the seven augmentation techniques was also applied individually to generate distinct variants, yielding 608 additional samples per class.

Table 3. Parameters for data augmentation techniques applied during training.

Augmentation Technique	Parameters
Rotation	±30 degrees
Width Shift	±20% of total width
Height Shift	±20% of total height
Shear Transformation	Intensity of 0.2 radians
Zoom	Range of ±20%
Brightness Adjustment	[0.8, 1.2] range
Horizontal Flip	Randomly applied with 50% probability

The specific augmentation techniques and their parameters are detailed in Table 3. Figure 8 visually demonstrates the effect of these augmentations on a sample rice leaf image.



(**a**) Original (**b**) Rotation (**c**) Shift (**d**) Shear Transformation

(**e**) Zoom (**f**) Brightness Adjustment (**g**) Horizontal Flip (**h**) Mixed Augmentation

Figure 8. Visual examples of data augmentation techniques applied to a rice leaf image.

3.4. Model Training and Fine-Tuning Strategy

This subsection details the training procedure for the five CNN architectures. All models were initialized with weights pre-trained on the ImageNet dataset [30].

A two-phase fine-tuning strategy was adopted to stabilize training and enhance performance:

- Phase 1: Feature Extraction: The convolutional base of each model was frozen, and only the newly initialized top classification layers were trained for 20 epochs. This allows the model to adapt its new head to the features extracted by the frozen base.
- Phase 2: Fine-Tuning: Subsequently, the top 20% of layers from the convolutional base were unfrozen. The entire model was then trained for an additional 80 epochs with a significantly reduced learning rate (10 times lower than the initial rate). This approach helps prevent catastrophic forgetting and allows for domain-specific adaptation of higher-level features [31].

All models used the Adam optimizer [26] with an initial learning rate of 10^{-4} for the feature extraction phase and 10^{-5} for the fine-tuning phase. A batch size of 32 was used. To ensure robust evaluation, we employed a 5-fold cross-validation strategy for all models. The final performance metrics are the average and standard deviation across all five folds.

The complete set of hyperparameters is summarized in Table 4.

Table 4. Comprehensive model configurations and hyperparameters.

Parameter	VGG16	VGG19	DenseNet121	ResNet101V2	Xception
Input Size	(224,224,3)	(224,224,3)	(224,224,3)	(224,224,3)	(224,224,3)
Top Layers	Flatten, Dense (4096, ReLU) × 2, Dropout (0.5) × 2	Flatten, Dense (4096, ReLU) × 2, Dropout (0.5) × 2	GAP, Dense (512, ReLU), Dropout (0.5)	Flatten, Dense (2048, ReLU)	GAP *, Dense (2048, ReLU)
Output Layer	Dense (4, Softmax)	Dense (4, Softmax)	Dense (4, Softmax)	Dense (4, Softmax)	Dense (4, Softmax)
Optimizer	Adam	Adam	Adam	Adam	Adam
Learning Rate	10^{-4} [a], 10^{-5} [b]	10^{-4} [a], 10^{-5} [b]	10^{-4} [a], 10^{-5} [b]	10^{-4} [a], 10^{-5} [b]	10^{-4} [a], 10^{-5} [b]
Loss Function	categorical_crossentropy	categorical_crossentropy	categorical_crossentropy	categorical_crossentropy	categorical_crossentropy
Epochs	100	100	100	100	100
Batch Size	32	32	32	32	32
EarlyStopping	Patience = 20	Patience = 20	Patience = 20	Patience = 20	Patience = 20
ReduceLROnPlateau	Factor = 0.5, patience = 5	Factor = 0.5, patience = 5	Factor = 0.5, patience = 5	Factor = 0.5, patience = 5	Factor = 0.5, patience = 5

* GAP: Global Average Pooling, 2D; [a] Feature Extraction Phase, [b] Fine-Tuning Phase.

3.5. Model Evaluation Metrics

A multi-faceted evaluation was conducted using a confusion matrix and derived metrics to comprehensively assess model performance. Given that this is a multi-class classification problem involving three disease classes and one healthy class, the evaluation metrics are computed using a "one-vs-rest" (OvR) strategy for each class. In this framework, for any given class (e.g., 'Brown Spot'), that class is considered the "positive" class, while all other classes (e.g., 'Hispa', 'Leaf blast', 'Healthy') are collectively considered the "negative" class. The definitions of the fundamental confusion matrix components for a single class under the OvR strategy are as follows:

- True Positive (*TP*): The number of samples correctly predicted as the positive class. (e.g., A 'Hispa' image is predicted as 'Hispa').
- True Negative (*TN*): The number of samples correctly predicted as not being the positive class. (e.g., A 'Brown Spot' image is predicted as 'Brown Spot', 'Leaf blast', or 'Healthy').
- False Positive (*FP*): The number of samples incorrectly predicted as the positive class. (e.g., A 'Healthy' image is predicted as 'Hispa').

- False Negative (*FN*): The number of samples incorrectly predicted as not being the positive class. (e.g., A 'Hispa' image is predicted as 'Brown Spot').

These per-class *TP*, *TN*, *FP*, and *FN* counts are used to calculate class-specific metrics. The overall model metrics reported in this study (Accuracy, Precision, Recall, *F*1-Score) are macro-averaged. This means the metric (e.g., Precision) is first computed independently for each class, and then the average of these per-class values is taken. Macro-averaging treats all classes equally, which is suitable for this dataset as it helps to ensure that performance on all classes, regardless of their size, contributes equally to the final metric.

Based on the aforementioned definitions, the primary evaluation metrics used in this study are calculated as follows:

- Accuracy: The proportion of total correct predictions (both positive and negative) among the total number of cases examined. It provides an overall measure of correctness.

$$\text{Accuracy } (\%) = \frac{TP + TN}{TP + FP + TN + FN} \times 100\% \tag{1}$$

- Precision: For each class, it measures the proportion of correctly identified positive instances among all instances predicted as positive. A high precision indicates a low false positive rate for that class.

$$\text{Precision } (\%) = \frac{TP}{TP + FP} \times 100\% \tag{2}$$

- Recall: For each class, it measures the proportion of actual positive instances that were correctly identified. A high recall indicates a low false negative rate for that class.

$$\text{Recall } (\%) = \frac{TP}{TP + FN} \times 100\% \tag{3}$$

- *F*1-Score: The harmonic mean of precision and recall, providing a single metric that balances both concerns, especially useful when class distribution is uneven.

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

The Macro *F*1-Score is computed by calculating the *F*1-score for each class independently and then taking the average. This gives equal weight to all classes, making it suitable for assessing performance under potential class imbalance.

Furthermore, Receiver Operating Characteristic (ROC) curves and the corresponding Area Under the Curve (AUC) values were plotted for each class in a one-vs-rest manner. For model interpretability, Grad-CAM (Gradient-weighted Class Activation Mapping) [32] was employed to generate visual explanations of the models' focus areas.

4. Results and Analysis

This section presents a comprehensive evaluation and in-depth analysis of the five deep learning models for rice disease classification. All models were initialized with weights pre-trained on the ImageNet dataset and fine-tuned following the procedure detailed in Section 3. To ensure robust and statistically significant findings, we employed a 5-fold cross-validation strategy. The results reported herein, unless otherwise specified, represent the mean $\pm$ standard deviation across all folds. This approach mitigates the impact of data partitioning variability and provides a reliable estimate of model generalization.

4.1. Comprehensive Performance Benchmark and Architectural Analysis

This subsection establishes the primary performance hierarchy of the models under the most challenging Out-of-Distribution (OOD) scenario (Case C, as defined in Section 3.2) and provides an architectural analysis to explain the observed results. The overall performance metrics are summarized in Table 5. Values are reported as Mean ± Std (%) from 5-fold cross-validation.

Table 5. Comprehensive performance comparison of CNN models under OOD scenario (Case C).

Model	Accuracy	Precision	Recall	F1-Score
VGG16	68.12 ± 2.65	69.20 ± 2.57	71.11 ± 2.29	68.84 ± 2.51
VGG19	64.80 ± 1.94	67.06 ± 1.49	67.75 ± 1.41	66.36 ± 1.70
DenseNet121	85.08 ± 1.07	87.22 ± 1.38	83.75 ± 1.05	85.08 ± 1.15
ResNet101V2	73.98 ± 1.70	75.17 ± 1.62	75.31 ± 1.37	74.89 ± 1.59
Xception	64.30 ± 2.67	65.98 ± 2.75	68.04 ± 2.31	65.66 ± 2.65

As shown in Table 5, the results reveal a clear performance hierarchy. DenseNet121 emerged as the superior model, achieving the highest accuracy (85.08%) and F1-Score (85.08%), along with the lowest standard deviation, indicating robust and stable performance across different data folds. A one-way ANOVA conducted on the accuracy scores confirmed that the observed performance differences were statistically significant ($F(4, 20) = 15.83$, $p < 0.001$). Post hoc Tukey's HSD tests revealed that DenseNet121 significantly outperformed all other models ($p < 0.01$), while ResNet101V2 formed a distinct middle tier, performing significantly better than the VGG models and Xception ($p < 0.05$) [33].

The superior performance of DenseNet121 can be attributed to its innovative dense connectivity pattern [17]. This architecture promotes feature reuse throughout the network, alleviates the vanishing gradient problem, and achieves high parameter efficiency. This design allowed it to effectively leverage pre-trained features from ImageNet even with a simple custom classifier head, resulting in superior generalization without overfitting.

ResNet101V2 delivered solid, mid-tier performance (73.98% accuracy). Its residual learning blocks with pre-activation [15] successfully enabled the training of this very deep network. However, its performance gap behind DenseNet121 suggests that dense connections, which encourage more direct feature propagation and reuse, may be more effective for this specific task.

The VGG models and Xception formed the lower performance tier. The lower performance of VGG16 and VGG19 is likely due to their older, less parameter-efficient architecture [14], with large fully-connected layers being highly susceptible to overfitting. Xception's underperformance was unexpected given its modern design based on depthwise separable convolutions [16]. We hypothesize that its strength may require more data to fully express itself, or that the specific hyperparameters used were not optimal for this architecture on our dataset.

To contextualize these performance findings with deployment practicality, we analyzed the theoretical computational complexity of each architecture, as summarized in Table 6. This analysis reveals that DenseNet121 is the most parameter-efficient architecture by a significant margin, utilizing 8.1 million parameters—94% fewer than VGG16 (138.4 million). This extremely low parameter count, combined with its top-ranking accuracy, makes it a highly promising candidate for future on-device deployment.

In summary, this comprehensive benchmark demonstrates that for the task of rice disease classification, architectures designed for feature reuse and parameter efficiency (DenseNet121) not only converge to higher accuracy but also do so with a drastically lower computational footprint than very deep networks (ResNet101V2) or architectures with

large, over-parameterized classifiers (VGG). This establishes DenseNet121 as the optimal choice when considering both diagnostic precision and practical deployability.

Table 6. Computational complexity comparison of the model architectures.

Model	Parameters (Millions)	Theoretical GFLOPs (Giga)
VGG16	138.4	15.5
VGG19	143.7	19.6
DenseNet121	8.1	2.9
ResNet101V2	44.6	7.8
Xception	22.9	4.3

4.2. Model Diagnosis and Reliability Assessment

Beyond aggregate accuracy, a trustworthy diagnostic system requires that predictions are both discriminative and reliable. This subsection provides a fine-grained diagnosis of the models' capabilities through three critical lenses: class-wise discriminatory power, prediction calibration, and decision interpretability.

4.2.1. Diagnostic Capability with ROC and Precision–Recall Analysis

We first evaluated each model's ability to distinguish between specific disease classes using Receiver Operating Characteristic (ROC) and Precision–Recall (PR) curves. Figure 9 presents the class-wise curves, while Table 7 quantifies the Area Under the Curve (AUC) for both metrics.

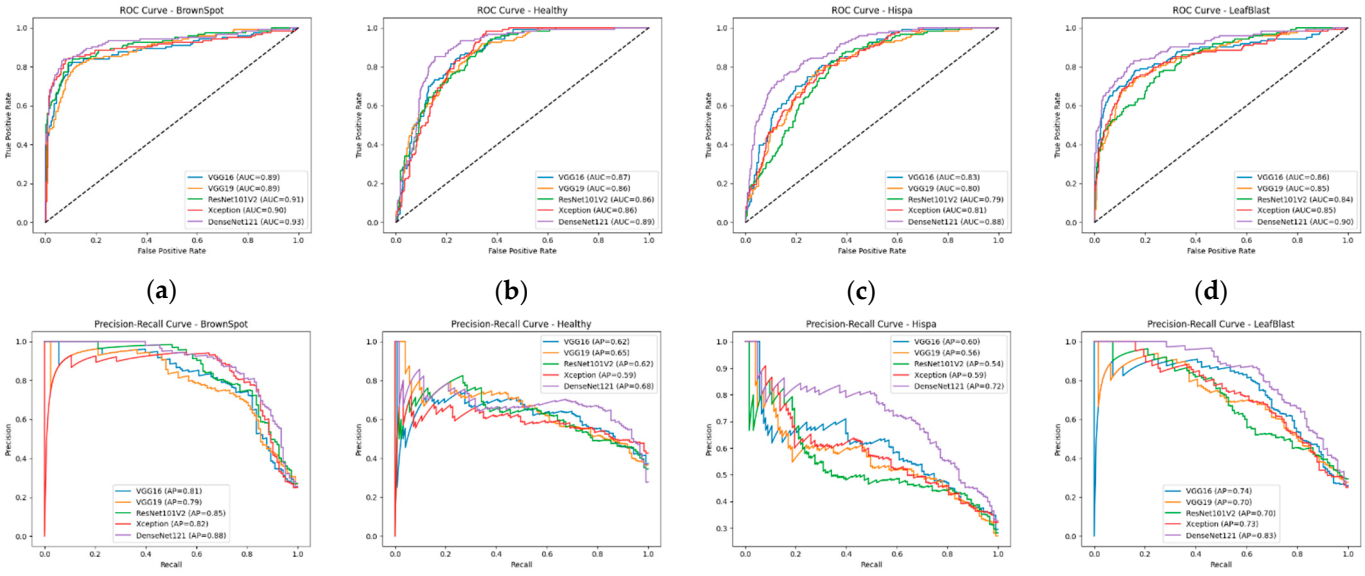


Figure 9. Class-wise evaluation curves. (**a**–**d**) ROC curves for Brown Spot, Healthy, Hispa, and Leaf Blast, respectively. (**e**–**h**) Precision–Recall curves for Brown Spot, Healthy, Hispa, and Leaf Blast, respectively.

As shown in Figure 9 and Table 7, DenseNet121 demonstrated superior and the most robust discriminatory power across all disease classes, achieving the highest AUC-ROC and AUC-PR for every single category. Its AUC-PR for the challenging Hispa class (0.73) was notably higher than all other models by a margin of at least 0.15, underscoring its exceptional capability in learning discriminative features.

Table 7. Class-wise AUC-ROC and AUC-PR for the evaluated models.

Rice Disease	Metric	VGG16	VGG19	DenseNet121	ResNet101V2	Xception
Brown Spot	AUC-ROC	0.89	0.89	0.93	0.90	0.90
	AUC-PR	0.81	0.79	0.88	0.84	0.82
Healthy	AUC-ROC	0.87	0.86	0.90	0.84	0.85
	AUC-PR	0.62	0.65	0.69	0.61	0.60
Hispa	AUC-ROC	0.83	0.80	0.89	0.77	0.81
	AUC-PR	0.60	0.56	0.73	0.48	0.58
Leaf Blast	AUC-ROC	0.86	0.85	0.90	0.85	0.84
	AUC-PR	0.74	0.70	0.84	0.70	0.70

The analysis reveals that diagnostic challenge varies significantly by disease. Brown Spot was the easiest to identify. Hispa was the most challenging, as evidenced by the lowest AUC-PR scores. The significant gap between ROC-AUC and PR-AUC for Healthy and Hispa is a typical signature of a dataset where the "negative" class contains many visually similar instances, making precise classification difficult [34].

4.2.2. Prediction Reliability via Calibration Analysis

For field deployment, a model's predicted confidence must match its actual accuracy. We quantitatively assessed this using Expected Calibration Error (ECE), with results presented in Table 8. The corresponding reliability diagrams for all five models are shown in Figure 10.

(**a**) VGG16 (**b**) VGG19 (**c**) ResNet101V2

(**d**) DenseNet121 (**e**) Xception

Figure 10. Reliability diagrams for the five evaluated CNN models.

Table 8. Class-wise Expected Calibration Error (ECE) for the evaluated models.

Model	Brown Spot	Healthy	Hispa	Leaf Blast	Average ECE
VGG16	0.086	0.081	0.103	0.054	0.081
VGG19	0.081	0.067	0.102	0.068	0.080
ResNet101V2	0.077	0.082	0.132	0.097	0.097
DenseNet121	0.073	0.112	0.158	0.089	0.108
Xception	0.054	0.067	0.078	0.059	0.065

The analysis reveals a critical accuracy-reliability trade-off. Xception is the best-calibrated model overall, with the lowest average ECE (0.065). As visible in Figure 10e, its reliability curve lies closest to the diagonal, indicating its predicted probabilities are the most trustworthy. In contrast, DenseNet121, despite its superior discriminative accuracy, is the most overconfident model, with the highest average ECE (0.108). Its curve in Figure 10d lies substantially below the diagonal for the Hispa (ECE = 0.158) and Healthy (ECE = 0.112) classes. This overconfidence means a high-confidence "Hispa" prediction from DenseNet121 has a much lower true probability of being correct, which could lead to misapplication of treatments [35].

4.2.3. Decision Rationale and Error Analysis Through Interpretability

To validate that the models' decisions are based on pathologically relevant features and to diagnose the root causes of misclassifications, we employed Gradient-weighted Class Activation Mapping (Grad-CAM) [32]. Visualizations for the top-performing DenseNet121 model (Figure 11) on correctly classified samples confirm that its attention is precisely focused on key pathological features: The elliptical lesions of Brown Spot, the linear streaks of Hispa, and the spindle-shaped lesions of Leaf Blast. For healthy leaves, the model shows weak, diffuse activation, confirming it has learned a valid representation of "health" as the lack of conspicuous pathological markers. This provides strong evidence that the model's decision-making process is semantically grounded for correct predictions.

To diagnose the root causes of the predominant misclassifications, we applied Grad-CAM to specific error cases. A summary of the dominant misclassification patterns is provided in Table 9. Our analysis reveals that the primary cause of error is not a failure to localize potentially relevant areas, but rather the inherent visual ambiguity in the image data itself. For instance, a healthy leaf misclassified as Hispa by DenseNet121 (Figure 12a) shows the model's attention activated by natural leaf wrinkles and specular highlights, which visually resemble genuine disease streaks. Conversely, a Hispa-infected leaf misclassified as healthy by ResNet101V2 (Figure 12b) reveals a failure to activate strongly on the faint, early-stage streaks. In another case, Xception misclassifies a Leaf Blast sample as Hispa (Figure 12c) despite correctly localizing the diseased lesion, indicating confusion between the morphological appearance of the two pathologies. In all cases, the models' attention mechanisms are functioning as designed, focusing on plausible regions of interest. The errors are primarily due to the limitations of visual features alone in robustly distinguishing between benign leaf structures, early-stage disease, and morphologically similar pathologies.

In all cases, the models' attention mechanisms are functioning as designed, focusing on plausible regions of interest. The errors are primarily due to the limitations of visual features alone in robustly distinguishing between benign leaf structures, early-stage disease, and morphologically similar pathologies. This analysis underscores that the most substantial immediate barrier to performance is data quality and coverage at the class boundaries.

Table 9. Summary of dominant misclassification patterns and error rates across all models (Case A).

Model	Primary Error Pattern	Error Count	Class-Wise Accuracy (Healthy/Hispa)
DenseNet121	Healthy → Hispa	41	78%/97%
Xception	Hispa → Healthy	37	82%/71%
ResNet101V2	Hispa → Healthy	44	84%/65%
VGG16	Healthy → Hispa	32	71%/80%
VGG19	Healthy → Hispa	27	75%/77%

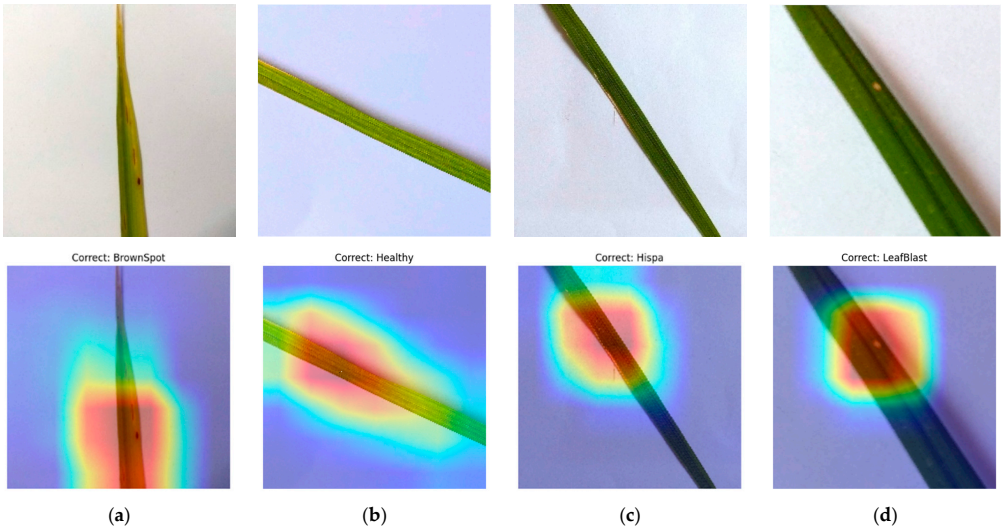


Figure 11. Grad-CAM visualizations for DenseNet121 on correctly classified samples. (**Top**): original images. (**Bottom**): Grad-CAM overlays. (**a**) Brown Spot, (**b**) Healthy, (**c**) Hispa, and (**d**) Leaf Blast. The colors in the heatmap indicate the importance of the corresponding features/regions for the model's prediction. Warmer colors (e.g., red/yellow) denote higher contribution/activation, while cooler colors (e.g., blue/green) denote lower contribution/activation.

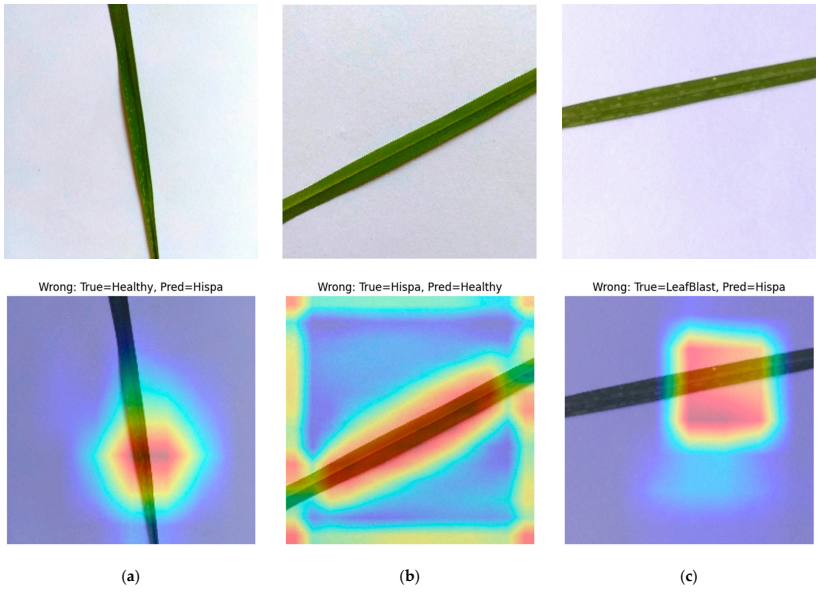


Figure 12. Visual diagnosis of common misclassification root causes. (**Top**): original images. (**Bottom**): Grad-CAM overlays. (**a**) Healthy → Hispa (DenseNet121), (**b**) Hispa → Healthy (ResNet101V2), (**c**) Leaf Blast → Hispa (Xception). The colors in the heatmap indicate the importance of the corresponding features/regions for the model's prediction. Warmer colors (e.g., red/yellow) denote higher contribution/activation, while cooler colors (e.g., blue/green) denote lower contribution/activation.

4.3. Robustness and Generalization Under Challenging Conditions

To evaluate model robustness beyond the primary OOD test, we compared performance across the designed experimental scenarios (Cases A, B, and D), which probe different aspects of model generalization. This multi-scenario assessment provides a holistic view of model behavior under varying data conditions, addressing key concerns regarding practical applicability.

4.3.1. Multi-Scenario Performance and Robustness

The accuracy for each model across all four experimental scenarios is consolidated in Table 10. This comparative view allows for a direct analysis of how each architecture responds to increased data diversity, domain shift, and synthetic perturbations.

Table 10. Model Accuracy (%) Across Different Experimental Scenarios.

Model	Case A (Baseline Aug)	Case B (Extended Aug)	Case C (OOD)	Case D (Stress Test)
VGG16	0.66	0.67	0.68	0.63
VGG19	0.62	0.65	0.65	0.59
ResNet101V2	0.63	0.65	0.74	0.58
DenseNet121	0.71	0.73	0.85	0.64
Xception	0.64	0.68	0.64	0.60

A comparative analysis of the scenarios yields several key insights into model robustness:

- Impact of Augmentation Volume: Comparing Case B (Extended Augmentation) to Case A (Baseline Augmentation), most models showed a slight performance improvement (e.g., DenseNet121: 71% → 73%; Xception: 64% → 68%). This confirms that increasing the diversity and volume of augmented data is beneficial for generalization, albeit with diminishing returns. The consistent but modest gains suggest that while helpful, simply adding more augmented data from the same source distribution has limitations.
- Superior OOD Generalization: The Case C (OOD) results remain the most telling indicator of true generalization ability. Here, DenseNet121 demonstrated exceptional capability (85%), significantly outperforming all other models on completely unseen data from an external source.
- Robustness Under Stress: The Case D (Stress Test) reveals model vulnerability to synthetic perturbations within the test set itself. All models experienced a performance drop compared to their baseline (Case A). DenseNet121, despite a notable drop from 71% to 64%, maintained the highest absolute accuracy, demonstrating relative resilience.

4.3.2. Consistency and Architectural Implications

Across all scenarios, DenseNet121 consistently ranked first, reinforcing the conclusion that it is the most robust and reliable architecture identified in this benchmark. Its performance is sustained across different challenges. The performance of other models, however, showed greater variability. For instance, ResNet101V2's performance was more competitive in the OOD setting but dropped significantly in the stress test. This inconsistent behavior underscores a lack of robustness. The VGG models and Xception formed a stable lower-performance group across all tests.

This systematic evaluation across multiple experimental scenarios validates the primary findings while providing nuanced insights into the models' behavior under different

data conditions. It confirms that DenseNet121 offers the best combination of high accuracy and cross-scenario stability, making it the most suitable candidate for applications where environmental conditions and data sources cannot be perfectly controlled.

4.4. Limitations and Future Work

While this study provides a rigorous and systematic benchmark, it is subject to certain limitations that delineate clear pathways for future research.

The primary limitation lies in the ecological validity of the evaluation. Although we performed an OOD test, all assessments were conducted in a controlled setting using standardized image datasets. The generalization capability of our models must be further tested on images captured in real-field conditions, involving variable lighting, complex backgrounds, and different growth stages. Conducting large-scale, cross-region validation is the ultimate test for assessing model reliability in diverse agricultural environments [36].

Secondly, the trustworthiness of the model predictions presents a critical avenue for improvement. The model outputs are raw, uncalibrated probabilities. For reliable field decision-making, applying post hoc calibration techniques such as Platt scaling or temperature scaling [35] constitutes an essential next step to correct the observed overconfidence.

Thirdly, the scope of our robustness evaluation can be deepened. Future work should include more targeted stress testing, systematically evaluating performance under extreme imaging conditions, such as severe occlusion and subtle disease symptoms.

Finally, the scope of this study deliberately focused on establishing a comprehensive benchmark of CNNs. A direct and fair comparison of this established benchmark against emerging, efficiency-oriented Transformer architectures (e.g., MobileViT, EfficientFormer) is the logical and critical next step [37]. This comparative analysis will help identify the current leading edge of deep learning technology for agricultural applications.

5. Conclusions

This systematic benchmark of five CNN architectures establishes DenseNet121 as the optimal model for rice disease diagnosis, achieving superior accuracy (85.08%) with exceptional parameter efficiency (8.1 M parameters). Its dense connectivity enables effective feature reuse, while its compact design makes it ideally suited for deployment in resource-constrained agricultural environments.

Beyond architectural comparison, our analysis provides critical practical insights: while DenseNet121 excels in diagnostic accuracy, Xception offers better-calibrated probabilities, and the persistent Healthy-Hispa confusion highlights the need for improved data quality at class boundaries. This study delivers a reproducible foundation for model selection in agricultural AI, demonstrating that informed architectural choices can significantly enhance both the performance and practicality of crop disease management solutions.

Author Contributions: Conceptualization, C.-H.C. and Y.-F.H.; Investigation, S.-H.L., Q.-W.J.,C.-H.C. and Y.-S.T.; Methodology, S.-H.L., Q.-W.J., Y.-S.T. and Y.-F.H.; Software, Q.-W.J., C.-H.C. and Y.-S.T.; Supervision, C.-H.C. and Y.-F.H.; Writing—original draft, C.-H.C., S.-H.L., Y.-S.T. and Y.-F.H.; Writing—review and editing, S.-H.L., Q.-W.J., C.-H.C. and Y.-F.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partly supported by National Science and Technology Council (NSTC), Taiwan, grant numbers NSTC 114-2221-E-324-005- and NSTC-114-2221-E-150-015-.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article; further inquiries can be directed to the corresponding authors.

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

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
ANN	Artificial Neural Network
AUC	Area Under the Curve
AUC-PR	Area Under the Precision–Recall Curve
AUC-ROC	Area Under the Receiver Operating Characteristic Curve
CNN	Convolutional Neural Network
DenseNet	Densely Connected Convolutional Network
ECE	Expected Calibration Error
FN	False Negative
FP	False Positive
GAP	Global Average Pooling
Grad-CAM	Gradient-weighted Class Activation Mapping
OOD	Out-of-Distribution
ReLU	Rectified Linear Unit
ResNet	Residual Network
RMSprop	Root Mean Square Propagation
SGD	Stochastic Gradient Descent
Tanh	Hyperbolic Tangent
TN	True Negative
TP	True Positive
UCI	University of California, Irvine
VGG	Visual Geometry Group
ViT	Vision Transformer

References

1. Fukagawa, N.K.; Ziska, L.H. Rice: Importance for Global Nutrition. *J. Nutr. Sci. Vitaminol.* **2019**, *65*, S2–S3. [CrossRef] [PubMed]
2. Kumar, K.S.A.; Karthika, K.S. Abiotic and Biotic Factors Influencing Soil Health and/or Soil Degradation. In *Soil Health*; Springer: Cham, Switzerland, 2020; pp. 145–161.
3. Phadikar, S.; Sil, J.; Das, A.K. Rice Diseases Classification Using Feature Selection and Rule Generation Techniques. *Comput. Electron. Agric.* **2013**, *90*, 76–85. [CrossRef]
4. Zhang, Y.Z. Ecology and Control Measures for Major Rice Diseases in Taiwan. In Proceedings of the Symposium on Rice Health Management, Taipei, Taiwan, 15–17 April 2004.
5. Latif, G.; Abdelhamid, S.E.; Mallouhy, R.E.; Alghazo, J.; Kazimi, Z.A. Deep Learning Utilization in Agriculture: Detection of Rice Plant Diseases Using an Improved CNN Model. *Plants* **2022**, *11*, 2230. [CrossRef] [PubMed]
6. Laborte, A.G.; Gutierrez, M.A.; Balanza, J.G.; Saito, K.; Zwart, S.J.; Boschetti, M.; Murty, M.V.R.; Villano, L.; Aunario, J.K.; Reinke, R.; et al. RiceAtlas, a Spatial Database of Global Rice Calendars and Production. *Sci. Data* **2017**, *4*, 170074. [CrossRef] [PubMed]
7. Sarker, I.H. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Comput. Sci.* **2021**, *2*, 420. [CrossRef] [PubMed]
8. Yamashita, R.; Nishio, M.; Do, R.K.G.; Togashi, K. Convolutional Neural Networks: An Overview and Application in Radiology. *Insights Imaging* **2018**, *9*, 611–629. [CrossRef] [PubMed]
9. Burhan, S.A.; Minhas, S.; Tariq, A.; Hassan, M.N. Comparative Study of Deep Learning Algorithms for Disease and Pest Detection in Rice Crops. In Proceedings of the 2020 International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Bucharest, Romania, 25–27 June 2020; pp. 1–5.
10. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.
11. Khan, A.; Rauf, Z.; Sohail, A.; Aslam, M.S.; Baber, J.; Ullah, H.; Saeed, M.; Alomari, A.A.; Alraddadi, M.O. A Survey of the Vision Transformers and Their CNN-Transformer Based Variants. *Artif. Intell. Rev.* **2023**, *56*, 2917–2970. [CrossRef]
12. Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; et al. A Survey on Vision Transformer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 87–110. [CrossRef] [PubMed]

13. Canziani, A.; Paszke, A.; Culurciello, E. An Analysis of Deep Neural Network Models for Practical Applications. *arXiv* **2016**, arXiv:1605.07678.
14. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
15. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
16. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
17. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269.
18. Pereira, J.R.J. Rice Disease. Kaggle. 2023. Available online: https://www.kaggle.com/datasets/jonathanrjpereira/rice-disease (accessed on 16 November 2025).
19. UCI Machine Learning Repository. Rice Leaf Diseases Dataset. Available online: https://archive.ics.uci.edu/ml/datasets/Rice+Leaf+Diseases (accessed on 24 July 2025).
20. Chen, Y.-C. Applications of Convolution Neural Networks to Predict Clinical Pregnancy from Embryo Microscope Images in In Vitro Fertilization. Master's Thesis, Taipei Medical University, Taipei, Taiwan, 2021.
21. Glorot, X.; Bordes, A.; Bengio, Y. Deep Sparse Rectifier Neural Networks. In Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 11–13 April 2011; pp. 315–323.
22. Huang, T.-Y. Application of Deep Learning Combined with Balanced Experimental Design Method for Pneumonia Classification in Chest X-ray Images. Master's Thesis, National Pingtung University, Pingtung, Taiwan, 2020.
23. Huang, T.-H. Research on Machine Learning for Indoor Positioning Using Multi-Channel Information. Master's Thesis, National Formosa University, Yunlin, Taiwan, 2020.
24. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861. [CrossRef]
25. Lin, C.-T. Matching the Method of Neural Network CNN, LSTM and DNN on the High Confused Mandarin Vowel Recognition. Master's Thesis, National Chung Hsing University, Taichung, Taiwan, 2019.
26. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
27. Ruder, S. An Overview of Gradient Descent Optimization Algorithms. *arXiv* **2016**, arXiv:1609.04747.
28. Jade, T. Rice Disease Image Dataset. Kaggle. 2023. Available online: https://www.kaggle.com/datasets/tiffanyjade/rice-disease-image-dataset (accessed on 16 November 2025).
29. University of Jaffna. Hispa. Roboflow Universe. 2023. Available online: https://universe.roboflow.com/university-of-jaffna-cfjf1/hispa-9f7nz (accessed on 16 November 2025).
30. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
31. Deng, Y.-H. Fine-Tuning Deep Learning Image Classification Parameter Based on Transfer Learning. Master's Thesis, National Taiwan University of Science and Technology, Taipei, Taiwan, 2018.
32. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 618–626.
33. Berrar, D. Cross-Validation. In *Encyclopedia of Bioinformatics and Computational Biology*; Elsevier: Oxford, UK, 2019; pp. 542–545.
34. Valverde-Albacete, F.J.; Peláez-Moreno, C. 100% Classification Accuracy Considered Harmful: The Normalized Information Transfer Factor Explains the Accuracy Paradox. *PLoS ONE* **2014**, *9*, e84217. [CrossRef] [PubMed]
35. Guo, C.; Pleiss, G.; Sun, Y.; Weinberger, K.Q. On Calibration of Modern Neural Networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1321–1330.
36. Arora, A. DenseNet Architecture. Available online: https://amaarora.github.io/posts/2020-08-02-densenets.html (accessed on 15 November 2025).
37. Liu, Z.; Mao, H.; Wu, C.-Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A ConvNet for the 2020s. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 11976–11986.