

Multi-Source Rice Disease Detection Using Fine-Tuned Swin-Tiny with Grad-CAM Explainability

Engy Youssef^{1*}, Wael Hassan Gomaa^{2*}, Hagar Mohamed^{3*}

Information System Department-Faculty of Computers and Artificial Intelligence, Beni-Suef University, Beni-Suef, Egypt^{1,2,3}
Information System Department-Faculty of Computer Science, Nahda University, Beni-Suef, Egypt¹

Abstract—Rice (*Oryza sativa*) is a staple food for approximately half of the world's population, which makes rapid and accurate detection of rice diseases essential. Most existing rice disease classifiers are trained and evaluated on images collected in a single region, so little evidence is available on their behavior across acquisition sources. This study combines two public field datasets, Paddy Doctor (India) and the Microsoft Rice Disease Classification Challenge dataset (Egypt). The three Egyptian classes are mapped to their equivalent Paddy Doctor categories, which yields a ten-class dataset of 13,077 RGB images. The original images are split into stratified training, validation, and held-out test subsets before any augmentation, augmentation-based class balancing is applied to the training subset only, and all images are processed with Gaussian smoothing and Contrast Limited Adaptive Histogram Equalization (CLAHE). Seven ImageNet-pretrained CNNs (EfficientNetB4, InceptionV3, MobileNetV2, ResNet50, ResNet101, ResNet152, and VGG19) and the transformer-based Swin-Tiny are fine-tuned under an identical two-stage protocol. On the held-out test set, Swin-Tiny achieves the best performance, with an accuracy of $94.17 \pm 0.34\%$ and a macro-F1 score of $94.16 \pm 0.39\%$ over three random seeds, compared with $91.54 \pm 0.64\%$ for the best CNN, EfficientNetB4. When predictions with low softmax scores are referred to an expert, Swin-Tiny classifies 84.8% of the test images with an accuracy of 98.44%. A leave-one-source-out experiment shows that a model trained on one source performs close to chance level on the other, whereas training on the merged dataset reaches 90.94% accuracy on the Egyptian test images. A factorial ablation with EfficientNetB4 shows that CLAHE does not improve accuracy on the merged dataset, and the Grad-CAM explanations are checked with a deletion test.

Keywords—Rice disease classification; deep learning; transfer learning; dataset fusion; Swin Transformer; EfficientNet; Grad-CAM

I. INTRODUCTION

Rice is one of the principal field crops in Egypt, where it has historically occupied approximately 650,000 hectares, or about 20% of the 3.3 million hectares of cultivated land [1]. Recent official statistics show a marked decline in this share. According to the CAPMAS 2023/2024 Annual Bulletin of Crop Areas and Plant Production Statistics [2], the cultivated rice area decreased by 21.5%, from 1.6 million feddans ($\approx 672,000$ ha) in 2022/2023 to 1.3 million feddans ($\approx 546,000$ ha) in 2023/2024. Over the same period, production fell by 19.5%, from 6.2 to 5.0 million tons.

Rice in the Nile Delta is grown under flooded conditions and therefore requires large volumes of irrigation water, while more than 98% of Egypt's water resources are derived from the Nile. The filling and operation of the Grand Ethiopian Renaissance Dam have added pressure on this supply [3], [4]. Higher temperatures and evaporation rates associated with climate change have further increased crop water demand [5], [6]. As a result, the area permitted for rice cultivation is regulated through annual quotas [7], and part of the domestic demand is met by imports [8]. Under these constraints, yield losses caused by diseases on the remaining cultivated area become more costly, and the untargeted application of pesticides carries environmental and health risks [9], [10]. Pathogens and pests cause substantial yield losses in rice worldwide [11], so early and accurate disease diagnosis supports both yield protection and targeted treatment.

Recent advances in computer vision have made image-based disease diagnosis practical in precision agriculture [12], [13], [14]. Most published rice disease classifiers, however, are trained and tested on a single dataset collected in one region and under one acquisition protocol. Reported accuracies therefore provide limited information about how such models behave on images from other sources. In addition, many studies balance or augment the data before splitting it, which can place near-duplicate images in both the training and evaluation subsets and inflate the reported performance.

This study addresses these issues with a three-stage framework, illustrated in Fig. 1. In the first stage, the Paddy Doctor dataset and the Microsoft Rice Disease Classification Challenge dataset are combined into a ten-class dataset. The original images are split into training, validation, and held-out test subsets before augmentation, the training subset is balanced by augmentation, all images are preprocessed with Gaussian smoothing and CLAHE, and eight pretrained architectures are compared under transfer learning. In the second stage, all architectures are fine-tuned by unfreezing the top 30% of their backbones, the best architecture (Swin-Tiny) is selected on the validation subset, and it is retrained with three seeds and evaluated on the held-out test subset. In the third stage, Grad-CAM heatmaps are generated to visualize the image regions that contribute to each prediction.

The contribution of this work lies in the evaluation design and in the measured value of each component of an established pipeline, rather than in a new network architecture. The main

*Corresponding author.

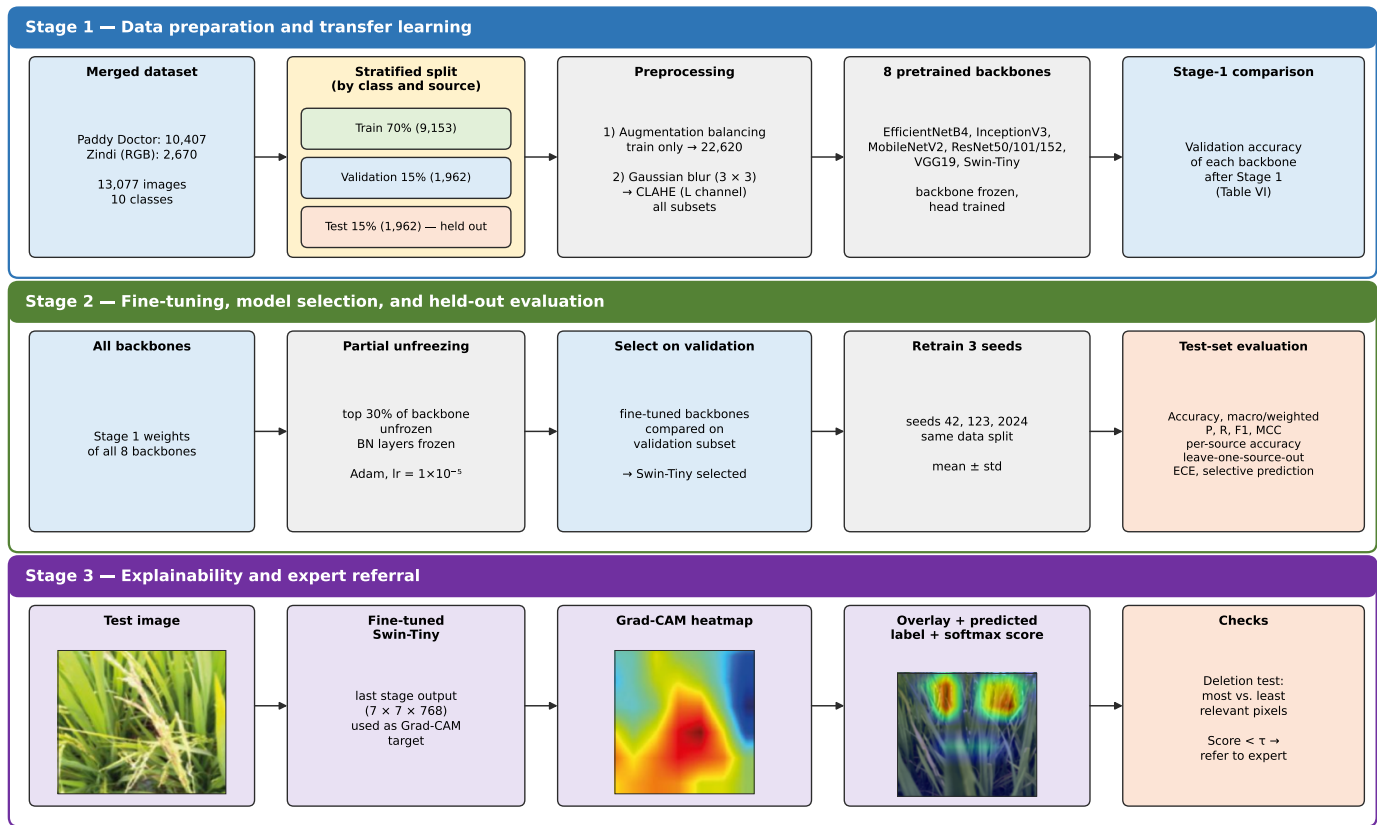


Fig. 1. Summary of the proposed framework.

contributions are as follows:

- Cross-source ten-class dataset with explicit label mapping: The three classes of the Egyptian dataset are mapped to the equivalent Paddy Doctor categories, and per-class, per-source image counts are reported.
- Evaluation protocol: The original images are split before augmentation, a labeled held-out test subset that is not used for training or model selection serves for final reporting, and results are averaged over three random seeds.
- Factorial ablation of the preprocessing pipeline: The separate and combined contributions of Gaussian blurring, CLAHE, and augmentation-based balancing are quantified with the EfficientNetB4 backbone, showing that CLAHE does not improve accuracy on the merged dataset.
- Cross-source analysis: Test performance is reported per source, and a leave-one-source-out experiment on the shared classes measures transfer between the two datasets.
- Comparative benchmark: Eight CNN and transformer architectures are fine-tuned and compared under an identical protocol; the transformer-based Swin-Tiny outperforms all CNNs, including the best CNN, EfficientNetB4.
- Explainability and reliability analysis: Grad-CAM ex-

planations are assessed with a deletion test, and the calibration of the softmax scores is evaluated together with a selective-prediction rule for expert referral.

The rest of this paper is organized as follows. Section II presents background on rice diseases and the deep learning models used. Section III reviews related studies and public rice disease datasets. Section IV formulates the classification problem. Section V describes the two datasets and the merged dataset. Section VI presents the proposed methodology. Section VII reports the experimental setup and results, and Section VIII discusses them, including the limitations of the framework. Section IX concludes the paper.

II. BACKGROUND

Disorders that affect the paddy field are considered one of the biggest challenges in growing rice on a commercial scale, which can cause a substantial fall in yield as well as its quality. Deep learning has achieved strong results in image-based plant disease detection. This section is divided into two parts: the first gives an overview of the major paddy diseases and their symptoms, and the second describes the deep learning architectures evaluated in this study.

A. Paddy Diseases and Their Symptoms

Paddy infections are one of the major problems all around the world affecting rice production. It reduces the crop yield as well as the quality of grains. These illnesses may develop from

unfavorable climatic conditions, such as temperature changes and soil nutrient imbalance, or from biological agents like fungi, bacteria, viruses, nematodes, and viroids [15]. Since many diseases show similar symptoms on leaves, stems and grains, correct disease diagnosis is vital for effective crop management [16]. Paddy diseases can be classified in broad four major classes on the basis of causal agents of the diseases:

1) *Fungal diseases*: These are the most common paddy diseases caused by pathogenic fungi which infect the different parts of rice plant such as leaves, sheaths, stems and grains. The symptoms are lesions, discolouration, rotting and tissue necrosis which greatly reduces the crop productivity. Common fungal diseases are Blast, Brown Spot, Sheath Blight, Rice Sheath Rot, Rice Stem Rot, False Smut and Leaf Smut [15], [17], [18], [19].

2) *Bacterial diseases*: Bacterial infection is usually characterized by water-soaked lesions, yellowing, leaf wilting and tissue blight. Under warm and humid environmental circumstances, these diseases can spread rapidly and can greatly reduce rice yield. Bacterial Blight is one of the most devastating bacterial infections to damage rice crops [19], [20], [21].

3) *Viral diseases*: The viruses are spread largely by insect vectors that negatively affect crop development and grain yield. Typical symptoms of infected plants include leaf yellowing, stunted development and lower production. Rice Tungro disease is one of the most important viral diseases of paddy cultivation from economic point of view [22], [23].

4) *Diseases associated with pests*: Damage by pest insects can give rise to diseases of rice plants. The common signs include leaf feeding damage, injury of stem, reduced tillering and deformed grains, which cause to significant output losses. Rice Hispa and Stem Borers are important insect pests on paddy field [24], [25].

Fig. 2 illustrates the major categories of paddy diseases and their associated symptoms.

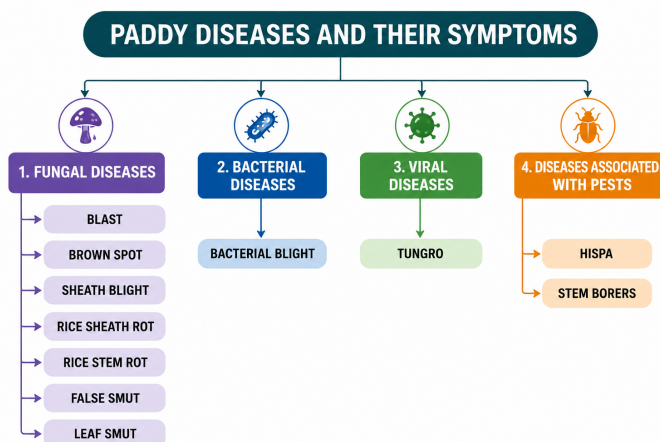


Fig. 2. Major paddy diseases and their symptoms.

B. Deep Learning Models

Deep Learning is a subset of Artificial Intelligence and Machine Learning that uses neural networks with three or more

layers to learn hierarchical representations directly from data. CNNs have long been the dominant approach to image classification tasks such as the detection of plant and rice leaf diseases, and vision transformers have recently become a strong alternative. This subsection reviews the pretrained architectures used in this work, namely seven CNNs (VGG19, ResNet50, ResNet101, ResNet152, InceptionV3, MobileNetV2, and EfficientNetB4) and the transformer-based Swin-Tiny, their performance in recent plant and rice disease classification studies, and their computational requirements on the Kaggle platform.

1) *EfficientNetB4*: EfficientNetB4 is a widely used convolutional neural network architecture for classifying plant diseases. This is because it uses compound scaling to increase the depth, width and input resolution of the network at the same time. It has a high feature extraction ability with a much lower computational cost of around 19.3 million trainable parameters compared to VGG19 and deeper ResNet models. This architecture is quicker to train, consumes less GPU memory and is able to learn the subtle visual features of rice leaf diseases quickly, which supports accurate classification. With these characteristics, EfficientNetB4 is a strong candidate for cloud based applications such as Kaggle and transfer learning where the computing power is limited [26], [27].

2) *MobileNetV2*: MobileNetV2 is a lightweight convolutional neural network based on the depthwise separable convolution and inverted-residual bottleneck blocks suitable for lightweight image classification. This architecture has approximately 3.5 million trainable parameters, which is the smallest architecture among the models evaluated in this study. Therefore, it has much lower computational cost, GPU memory consumption and training time compared with VGG19, ResNet and InceptionV3. Its classification accuracy is generally lower than that of EfficientNetB4 and deeper ResNet variants but MobileNetV2 provides a nice balance between computational efficiency and predictive performance. Its lightweight architecture makes it especially suited for real-time agricultural applications, transfer learning, and cloud-based platforms where fast model training and efficient use of resources are critical [28], [29].

3) *ResNet50*: ResNet50 is one of the most popular convolutional neural network architectures for image classification. It is due to its residual learning framework that enables quick training of deep networks via skip connections. It has approximately 25.6 million trainable parameters, which achieves a trade-off between model complexity and computational efficiency. It requires fewer computational resources than ResNet101 and ResNet152, while still achieving competitive classification performance. The ResNet50 has a good capability in extracting features for complicated lesion pattern recognition for paddy disease classification and has a consistent high prediction accuracy. It also trains faster, consumes less GPU memory and infers faster than other deeper ResNet variants making it a good candidate for transfer learning and cloud environments such as Kaggle [30], [31].

4) *ResNet101*: ResNet101 extends the depth of residual learning architecture to 101 layers and provides more representational power to learn complex image features. It has about 44.5 million trainable parameters which are providing a deeper feature representation than ResNet50 and this can increase the detection of minor lesion features in paddy leaf images.

However, with the increase of network depth, the computational complexity, GPU memory consumption, training time and inference time are also increased. ResNet101 can provide better classification performance than ResNet50, but the gain is generally marginal and depends on dataset properties and training strategies. Hence, ResNet101 is a good choice for applications where higher predictive performance is more important than computational efficiency, while lighter designs are more practical for resource-constrained environments such as Kaggle [32], [31]

5) *ResNet152*: ResNet152 is the deepest architecture in the ResNet family with a total of 152 layers and 60.2 million tunable parameters. The deeper network can extract more discriminative and abstract image features, which can be beneficial for complex image classification tasks. However, such an improvement comes with a cost of significantly increased computational complexity, more GPU memory consumption and a longer training and inference time compared to ResNet50 and ResNet101. ResNet152 has a larger depth so that it can gain a competitive classification performance, but a large depth does not always guarantee a significant accuracy improvement in paddy disease classification, especially when the transfer learning is applied to relatively small datasets. Hence the increased processing cost often does not justify the tiny performance improvement thus making the ResNet152 less appropriate for resource-constrained platforms such as Kaggle [30], [31], [33].

6) *InceptionV3*: InceptionV3 employs a multi-scale feature extraction strategy with factorized convolutional modules, which enables the network to efficiently extract the image features at different spatial scales. It has about 23.9 million trainable parameters, with computational complexity similar to ResNet50, and achieves competitive classification performance. The architecture is effective at extracting discriminative features from the plant disease images with different size and pattern of lesions. However, its classification performance is more sensitive to the image quality, dataset characteristics and preprocessing techniques compared to architectures like EfficientNetB4 and ResNet50. InceptionV3 provides a reasonable trade-off between accuracy and computational cost but may deliver varying results on other agricultural datasets, making it a less reliable option for paddy disease classification than the newer CNN models [34], [35].

7) *VGG19*: VGG19 is one of the first architectures of deep convolutional neural networks and has about 143.7 million trainable parameters that is the largest model among the architectures tested in this study. Its deep sequential architecture has strong feature extraction capability, but large number of parameters leads to high computational complexity, high GPU memory consumption and long training and inference time. VGG19 can achieve competitive classification performance by transfer learning, but its high model complexity increases the risk of over-fitting especially when training with relatively small agricultural datasets. VGG19 is typically less computationally efficient than more recent architectures such as EfficientNetB4, ResNet50 and MobileNetV2, without offering a commensurate improvement in classification performance. Therefore, it is mostly used as a benchmark model rather than a practical option for resource-constrained environments such as Kaggle [36], [37]. Table I shows a comparison of the selected

architectures based on the above discussed characteristics, in terms of model complexity, computational cost, training efficiency, and suitability for paddy disease classification.

8) *Swin-Tiny*: Swin-Tiny is the smallest variant of the Swin Transformer [38], a hierarchical vision transformer that computes self-attention within non-overlapping local windows and shifts the windows between consecutive blocks so that information is exchanged between neighbouring windows. Patch-merging layers reduce the spatial resolution between its four stages, which produces a feature pyramid similar to that of a CNN. With about 28 million parameters and 4.5 GFLOPs, its size is comparable to that of ResNet50, while its attention mechanism can relate image regions that are far apart.

III. LITERATURE REVIEW

This section discusses the application of deep learning for the detection of rice plant diseases. It uses databases such as Elsevier, IEEE Access, Scientific Reports, SpringerLink, ScienceDirect, ResearchGate and PubMed to review studies published between 2020 and 2026. The paper covers various topics such as datasets, plant diseases, accuracy, AI models and explainable AI (XAI). In addition, it provides a comparative summary of the role of AI in crop disease identification.

The review is divided into two parts: the first gives an overview of public rice leaf disease datasets, and the second reviews studies that developed automated rice disease classification methods on the two datasets used in this work.

A. Open-Source Paddy Leaf Disease Datasets

To support the development, training and evaluation of automated systems for classifying rice diseases, several public datasets have been released. This subsection overviews ten well-known open-source rice leaf disease datasets reported in the literature. Table II provides an overview of these datasets in terms of image resolution, number of images, number of disease classes and the corresponding rice diseases represented. Among these datasets, Paddy Disease Classification (Kaggle, India) and Microsoft Rice Disease Challenge (Egypt) are selected for further analysis in this work, as they are relevant and suitable for evaluating the proposed approach.

B. Studies Based on the Same Datasets

This study utilized combined data from both Paddy Disease Classification (Kaggle, India) and Microsoft Rice Disease Challenge (Egypt). The Microsoft Rice Disease Challenge (Egypt) dataset included additional colored images and infrared images sourced from the rice fields of Egypt, while the Paddy Disease Classification (Kaggle, India) dataset included categorized photographs of nine rice maladies and healthy leaves. The two datasets were used to facilitate diversity in data and improve the robustness of the proposed model. The datasets and the merging process are described in Section V. This section's review of the literature is organized according to the datasets used in earlier studies. Specifically, Paddy Disease Classification (Kaggle, India) dataset and Microsoft Rice Disease Challenge (Egypt) dataset researches are discussed separately. No previous review has discussed studies on both datasets together. The investigations are presented separately

TABLE I. ARCHITECTURAL COMPARISON OF THE EVALUATED MODELS

Architecture	Year	Parameters (M)	FLOPs (B)	Model Size	GPU Memory	Training Speed	Key Strength
VGG19	2014	143.7	19.6	Very large	Very high	Slow	Strong baseline; computationally expensive
ResNet50	2015	25.6	4.1	Medium	Moderate	Fast	Balance of accuracy and efficiency
ResNet101	2015	44.5	7.8	Large	High	Moderate	Deeper features for complex patterns
ResNet152	2015	60.2	11.5	Large	High	Slow	High capacity at higher cost
InceptionV3	2015	23.9	5.7	Medium	Moderate	Fast	Multi-scale features for varying lesion sizes
MobileNetV2	2018	3.5	0.3	Very small	Very low	Very fast	Lightweight, efficient deployment
EfficientNetB4	2019	19.3	4.2	Medium	Moderate	Moderate	Compound scaling of depth, width, and resolution
Swin-Tiny	2021	28.3	4.5	Medium	Moderate	Moderate	Hierarchical shifted-window self-attention

TABLE II. COMPARISON OF PUBLICLY AVAILABLE RICE LEAF DISEASE DATASETS

Dataset	Sample Background	No. of Images	No. of Diseases	Name of Paddy Diseases	Limitations	Studies Utilized
Paddy Doctor Dataset (IEEE Dataport, India)	Field	16,225	12	Bacterial leaf blight (BLB), Bacterial leaf streak (BLS), Bacterial panicle blight (BPB), Black stem borer, Blast, Brown spot, Downy mildew, Hispa, Leaf roller, Tungro, White stem borer, Yellow stem borer	Heavily imbalanced: 450 (lowest class) and 2,405 (highest class) images	[39], [40]
Mendeley Data: Rice Leaf Disease (Western tract of Odisha, India)	Field	5,932	4	Bacterial blight, Blast, Brown spot, and Tungro	Limited disease coverage; some classes augmented	[41]
Kaggle, UCI Machine Learning Repository (Gujarat, India)	Plain white or black	120	3	Leaf smut, Brown spot, and Bacterial leaf blight	Limited disease coverage, low sample count, and lack of complex environmental conditions	[42]
Dhan-Shomadhan: Bangladeshi Local Rice	Field and plain white	1,106	5	Brown spot, Leaf scalded, Rice blast, Tungro, and Sheath blight	Limited disease coverage, low sample count, and poor illumination in simple-background images	None
No name was given to this dataset (BRR1, Bangladesh)	A mixture of field and plain	1,426	8	False smut, Brown plant hopper (BPH), Bacterial leaf blight (BLB), Neck blast, Stem borer, Hispa, Sheath blight rot, and Brown spot	Low sample count and combination of different sample environments	[43]
Microsoft Rice Disease Challenge (Egypt)	Field	Total: 7,632 (labeled: 5,340 = 2,670 RGB + 2,670 RG-NIR)	2	Rice blast and Brown spot	Limited disease coverage and class imbalance: 410 (lowest class) and 1,494 (highest class) labeled RGB images	[44], [45]
Kaggle: Rice Leaf Diseases in Taiwan	Field, and 45 images with plain background	929	3	Rice blast, Brown spot, and Herbicide damage	Limited disease coverage, low sample count, combination of original and augmented data, and non-uniform background	None
Kaggle: Philippines Rice Diseases	A mixture of field and plain	1,555	13	Bacterial leaf blight, Bacterial leaf streak, Bakanae, Brown spot, Grassy stunt virus, Narrow brown spot, Ragged stunt virus, Rice blast, False smut, Sheath blight, Sheath rot, Stem rot, and Tungro	Low sample count, poor illumination, and combination of different sample environments	None
Kaggle: Leaf Rice Disease (Southeast Sulawesi, Indonesia)	Plain white	240	3	Bacterial leaf blight, Blast, and Tungro	Limited disease coverage and lack of complex environmental conditions	[46]
Paddy Disease Classification (Kaggle, India)	Field	13,876 (labeled: 10,407)	9	Bacterial leaf blight, Bacterial leaf streak, Bacterial panicle blight, Blast, Brown spot, Dead heart, Downy mildew, Hispa, and Tungro	Class imbalance: 337 (lowest class) and 1,764 (highest class) images	[47], [48], [49]

to allow a clearer examination of the approaches used and the performance achieved on each dataset individually.

A hybrid approach of Xception and OVR SVM has been proposed in [50]. The model combines the feature extraction

capabilities of Xception and the efficiency of OVR SVM, and achieves 94% accuracy in ten rice disease classes including Bacterial Leaf Blight, Brown Spot, Leaf Smut, Rice Blast, Leaf Scald, Sheath Blight, Tungro Virus, Hispa, Brown Planthopper and Healthy Rice. The hybrid model is very accurate and

robust for the real-world scenarios but it adds complexity due to multiple model integration.

The authors in [39] proposed Paddy Doctor, a dataset of 16,225 paddy leaf images in 13 classes and evaluated it using CNN and transfer learning models (VGG16, MobileNet, Xception and ResNet34). The best performance was obtained with ResNet34 with an F1-score of 97.50%. Although the model performs well on benchmarks, the study uses simple augmentation, and doesn't explicitly address class imbalance, which could affect the model's generalization. The study achieves high performance but lacks explainability mechanisms or a detailed post-training analysis to interpret the model predictions. In [47] the authors propose a modified ResNet50 based model to classify ten rice leaf categories, using an augmented version of the Paddy Doctor dataset. The model achieves a high accuracy of 98.03%. The model outperformed some deep learning architectures including VGG, Swin Transformer and MobileNet. But architecture relies on large fully connected layers with flattening. This leads to high computational complexity and may risk overfitting. Also, it lacks explainability mechanisms to interpret predictions.

In [48], EfficientNet-B4 was used for the classification of nine paddy diseases and healthy leaves, using Paddy Doctor dataset, which achieved the test accuracy of 96.91% and outperformed the Vision Transformer (ViT). The model performs well and is computationally efficient, but it has only been tested on a single dataset and does not include explainable AI methods or estimation of disease severity, which limits its practical use.

Authors in [40] introduced a hybrid deep learning framework for paddy leaf disease classification by using MobileNetV3 for feature extraction, GKSO-SA for feature selection and CatBoost classifier, achieving an accuracy of 98.52% on Paddy Doctor dataset. The model performs well with advanced preprocessing and optimization techniques, but lacks explainability tools such as Grad-CAM and practical outputs, such as disease severity estimation or treatment recommendations, which reduces its real-world interpretability and usability for farmers. A public multimodal rice disease dataset comprising RGB and multispectral (RG-NIR) photos taken from Egyptian rice fields was presented in [44]. They proposed a CNN-based pipeline combining RGB images and Normalized Difference Vegetation Index (NDVI) from multispectral images to improve the classification of diseases. In the proposed method, the blast, brown spot and healthy classes obtained F1-score of 90.02%, 83.26% and 81.54%, respectively. The use of RGB+NDVI improved the performance over RGB images but the study was limited to three illness classes and evaluated a single dataset. The improvement was only 1% higher than the RGB-only model.

In [44] authors proposed a deep learning pipeline for early rice disease identification using a public multimodal dataset of RGB and multispectral images. The proposed method showed that fusion of RGB and multispectral images achieved a better F1-score than the use of RGB images only. The multispectral data were Red, Green and Near-Infrared (RG-NIR) channels. It established the efficiency of multimodal image fusion in enhancing the precision of disease detection. But the research was mainly concentrated on the application of multispectral information and overlooked the hybrid classification framework

or ensemble learning methods to enhance the classification performance.

In [45], the author proposed a deep learning model based on Res2NeXt50 with SwAV self-supervised pre-training, fine-tuning, and self-distillation to classify rice diseases using RGB and RG-NIR images. t-SNE was also used to visualize the prediction distributions and to evaluate the learned feature representations. The proposed method provides important insights into the performance of the model and the effectiveness of self-supervised feature learning, but does not perform K-fold cross-validation or distinguish between RGB-only and RG-NIR-only models. Besides, the study mainly is concerned with presentation and interpretation of models, not with increasing classification ability. Table III illustrates summary of related work.

IV. PROBLEM FORMULATION

Manual inspection of rice fields for disease is time-consuming, error-prone, and difficult to scale in precision agriculture, which motivates automated, image-based identification of rice leaf diseases. This section formulates the classification task and the training procedure so that each quantity corresponds to the implementation described in Sections VI and VII.

Let the labeled dataset be

$$D = \{(x_i, y_i)\}_{i=1}^N, \quad (1)$$

where, x_i is an RGB image and $y_i \in \{0, 1\}^K$ is its one-hot label over $K = 10$ classes. D is partitioned by stratified sampling into disjoint subsets D_{train} , D_{val} , and D_{test} before any augmentation. An augmentation operator A (random rotation, horizontal flipping, and brightness scaling) is applied to D_{train} only, to generate the additional samples defined in Eq. (9) and (10). Every image of the three subsets, including the augmented copies, is then resized to 224×224 pixels and preprocessed by:

$$x'_i = T(x_i) = C(G(x_i)), \quad (2)$$

where, G is a 3×3 Gaussian filter and C applies CLAHE to the luminance channel in the CIELAB color space.

The classifier is composed of a backbone and a classification head:

$$f(x') = g_\phi(h_\theta(x')), \quad (3)$$

where, h_θ is the ImageNet-pretrained backbone and g_ϕ is the classification head. The head outputs logits $z \in \mathbb{R}^K$, which are converted into class scores by the softmax function:

$$\hat{p}_k(x') = \frac{e^{z_k}}{\sum_{j=1}^K e^{z_j}}, \quad (4)$$

and the predicted class is:

TABLE III. COMPARISON OF LITERATURE REVIEW FOR RICE LEAF DISEASE DETECTION MODELS

Ref.	Year	Technique	Performance	Disease Classes	Advantage	Limitation
[50]	2026	Hybrid Model: Xception + OVR SVM	94%	Bacterial Leaf Blight, Brown Spot, Leaf Smut, Rice Blast, Leaf Scald, Sheath Blight, Tungro, Hispa, Brown Planthopper, Healthy Rice	Combines Xception's feature extraction capability with OVR SVM's efficiency and interpretability. Provides robust performance on real-world images.	The hybrid model is more complex due to the integration of multiple models.
[39]	2023	CNN + Transfer Learning (VGG16, MobileNet, Xception, ResNet34); Best: ResNet34	97.50%	12 Paddy Diseases + Normal Leaf (13 classes)	Provides a large annotated dataset of 16,225 images, high performance, and open-source dataset and code.	Uses basic augmentation and does not explicitly address class imbalance, which may limit generalization and introduce bias toward dominant classes. No explainability mechanism or detailed post-training analysis is provided.
[47]	2024	Modified ResNet50 (Enhanced Deep Learning Model)	98.03%	9 Rice Diseases + 1 Healthy	High classification accuracy, enhanced ResNet50, and a larger augmented dataset of approximately 24,000 images.	No class balancing strategy or explainability analysis. Uses a simple train/test split without a validation set.
[48]	2024	EfficientNet-B4 (Transfer Learning, Adam optimizer, Data Augmentation)	96.91%	9 Paddy Diseases + Healthy Leaf	Demonstrates superior performance compared with Vision Transformer (ViT), with high accuracy and fewer parameters. Efficient feature extraction using MBConv makes it suitable for resource-constrained environments.	Trained and evaluated on a single dataset only. Lacks Explainable AI (XAI), disease severity estimation, and cross-dataset validation, with potential overfitting.
[40]	2025	MobileNetV3 + GKSO + SA Feature Selection + CatBoost	98.52%	10 classes (9 diseases + normal)	Uses advanced feature selection and achieves high classification performance on the Paddy Doctor dataset.	No explainability, severity estimation, or treatment output. The overall pipeline is relatively complex.
[49]	2025	Comparative Study: ResNet50 vs. Vision Transformer (ViT)	ResNet50: 92% / ViT: 93%	10 classes (standard paddy diseases + normal)	Provides an academic comparison between CNN and Transformer architectures and shows slightly better test performance for ViT.	Lower test accuracy (92–93%), no explainability using Grad-CAM, no severity or treatment output, and no test-time augmentation (TTA).
[45]	2022	Res2NeXt50 + SwAV + Self-Distillation + t-SNE	N/A	Brown Spot, Blast, Healthy	Uses self-supervised feature learning with effective prediction visualization using t-SNE.	No K-fold validation, no RGB/RG-NIR comparison, and focuses mainly on visualization rather than classification.
[44]	2023	CNN + RGB + NDVI	F1: 90.02%, 83.26%, 81.54%	Blast, Brown Spot, Healthy	RGB + NDVI improves disease detection by incorporating additional vegetation information.	Only three classes are considered, with only a slight improvement over RGB-based input.

$$\hat{y} = \arg \max_k \hat{p}_k(x'). \quad (5)$$

Training minimizes the class-weighted categorical cross-entropy over each mini-batch B :

$$\mathcal{L}(\theta, \phi) = -\frac{1}{|B|} \sum_{i \in B} \sum_{k=1}^K w_k y_{ik} \log \hat{p}_k(x'_i), \quad (6)$$

where, $w_k = 1$ when the training subset is balanced by augmentation, and w_k is given by Eq. (11) otherwise. Regularization is provided by dropout in g_ϕ . Optimization uses the Adam optimizer in two stages. In Stage 1, θ is frozen and only ϕ is updated with learning rate η_1 . In Stage 2, the top fraction r of the backbone layers, excluding batch-normalization layers, is unfrozen, and ϕ together with this subset $\theta_{\text{top}} \subset \theta$ is updated with a smaller learning rate $\eta_2 < \eta_1$:

$$(\theta_{\text{top}}, \phi) \leftarrow \text{Adam}(\nabla_{(\theta_{\text{top}}, \phi)} \mathcal{L}, \eta_2). \quad (7)$$

In both stages, the weights with the lowest loss on D_{val} are retained, and D_{test} is not used for training or model selection. The values of η_1 , η_2 , r , and the stopping criteria are given in Section VII.

To interpret the predictions, Grad-CAM is used to compute a class-discriminative heatmap for class c :

$$L_{\text{Grad-CAM}}^c = \text{ReLU} \left(\sum_k \alpha_k^c A^k \right), \quad (8)$$

where, A^k denotes the k -th feature map of the last convolutional layer (for Swin-Tiny, of the output of its last stage) and α_k^c is the importance weight obtained by global average pooling of the gradient of the class- c logit with respect to A^k .

V. DATASET

This study uses two publicly available rice disease datasets collected in different regions. The first dataset was taken from Kaggle and is called Paddy Disease Classification (Kaggle, India). It includes pictures of rice leaves categorized into ten classes, nine of which have damaged leaves and one of which has healthy leaves. Because it includes a variety of disease symptoms under varied environmental settings, this dataset is appropriate for deep learning model training. The second dataset for rice disease classification based on photos collected in Egypt for rice plant disease classification is retrieved from Zindi, a data science competition platform, under a competition sponsored by Microsoft. Unlike the first dataset, this dataset contains three classes: two diseases (blast and brown spot) and healthy leaves. Merging the two datasets extends the diversity of acquisition conditions for the shared classes; its effect on cross-source generalization is measured explicitly in Section VII.

A. Paddy Disease Classification (Kaggle, India)

Paddy Disease Classification dataset is collected from Kaggle, India. It is the ten-class release of the Paddy Doctor dataset [39] published for a Kaggle competition, and it is referred to as Paddy Doctor in the remainder of this paper. The dataset contains 13,876 photographs of paddy leaves, of which 10,407 labeled training images cover ten classes (nine diseases and healthy leaves); the remaining 3,469 competition test images are unlabeled and were not used. The dataset comprises image data together with other meta data such as type of rice and plant age. The photographs were obtained from the real paddy fields of the villages around the Tirunelveli district, Tamil Nadu, India, under natural agricultural conditions during the growing season of the year 2021. The photographs were obtained with a CAT S62 Pro smartphone camera and then annotated by agricultural professionals for correct classification labels. This dataset is designed to train and evaluate automated systems that classify rice illnesses using computer vision and deep learning techniques. Fig. 3 shows representative samples from the dataset. The nine diseases in the dataset were introduced in Section II [40].

1) Limitations of Paddy Disease Classification (Kaggle, India): However, Paddy Disease Classification dataset also has some limitations that might affect the performance of the model. These limitations are as follows: Paddy Disease Classification dataset might have some class imbalance as shown in Fig. 4, which might affect the learning process [51]. This is because some diseases are less represented compared to others. Moreover, most of the images are taken in the same environments, which may affect a model's applicability to real-world scenarios. Another limitation is that some diseases are similar in nature, and it is difficult to categorize the images [52].

B. Microsoft Rice Disease Challenge (Egypt)

The Microsoft Rice Disease Challenge (Egypt) collection contains 5340 labeled photos from rice fields in Egypt for automating rice disease identification. Each sample in the original dataset contains a pair of photos, comprising an RGB image obtained using a conventional smartphone camera and

a Red-Green-Near-Infrared (RG-NIR) image acquired using a MAPIR Survey3N multispectral camera. In this study, only the RGB subset was used, which contains 2,670 images distributed over three classes: blast (1,494), brown spot (766), and healthy (410). The RGB subset was selected for two reasons: it matches the imaging modality of Paddy Doctor, and excluding the paired RG-NIR images prevents two images of the same plant from appearing in different data subsets. RGB images can also be acquired with ordinary smartphone cameras [44]. The diseases in this dataset were described in Section II. Blast and brown spot show similar visual characteristics, particularly during the early phases of infection, making their distinction problematic for both human specialists and deep learning models. Fig. 5 shows representative RGB samples from the dataset.

1) Limitations of microsoft rice disease challenge (Egypt):

Several factors should be considered for this dataset. First, the dataset only covers a limited number of types of plant diseases, which may limit the model's ability to learn from and grow through multiple features of pathology, thus reducing the model's ability to generalize effectively in an applied agricultural setting. Second, the number of training instances available for deep learning applications is small given the size of the dataset. This limited size may increase the likelihood of obtaining models that overfit and affect how well learned features can transfer to future models. Moreover, the dataset has a significant class imbalance, with some classes having a considerably larger number of training examples than others. This class imbalance may result in the classifier becoming biased toward the widely represented classes, which will negatively impact the ability to classify those with fewer examples available as shown in Fig. 6. Therefore, evaluation metrics may not accurately represent classifier performance overall, as evaluated across all classes. Due to the limitations of this dataset, larger, diverse, and evenly distributed datasets in future studies will be necessary to improve classifier generalization and dependability.

C. Merged Dataset and Evaluation Protocol

The merged dataset contains 13,077 original RGB images across ten classes (Table IV). Three classes (blast, brown spot, and normal) are represented in both sources, while the remaining seven originate only from Paddy Doctor. The unlabeled test images released by both competitions were not used, because no ground-truth labels are available for them.

The original images were partitioned into training (70%, 9,153 images), validation (15%, 1,962 images), and held-out test (15%, 1,962 images) subsets using stratified sampling by both class and source dataset, before any augmentation was applied. Augmentation-based class balancing was then applied to the training subset only, raising each class to the size of the largest training class (2,262 images), so that the balanced training set contains 22,620 images. Gaussian smoothing and CLAHE were then applied to every image of all subsets, including the augmented training images. The validation and test subsets contain only original, non-augmented images. The validation subset was used exclusively for checkpointing, early stopping, model selection, and the selection of the referral threshold, while the test subset was used only for reporting the final results.

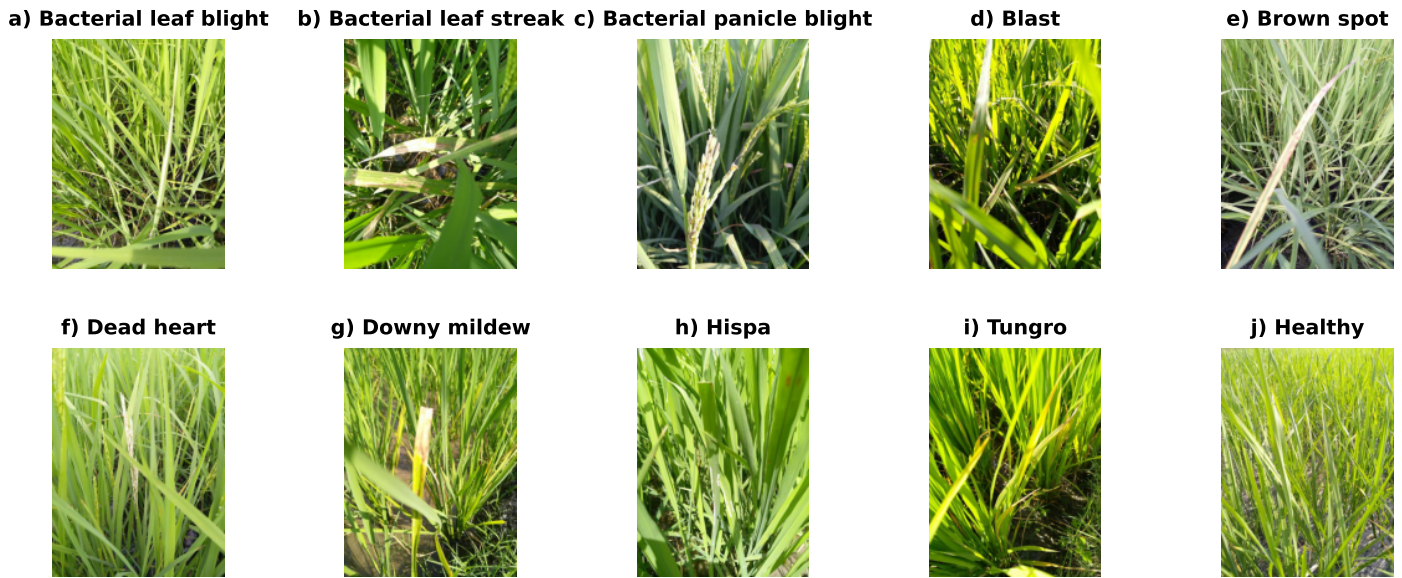


Fig. 3. Representative images of the ten classes in the paddy doctor dataset.

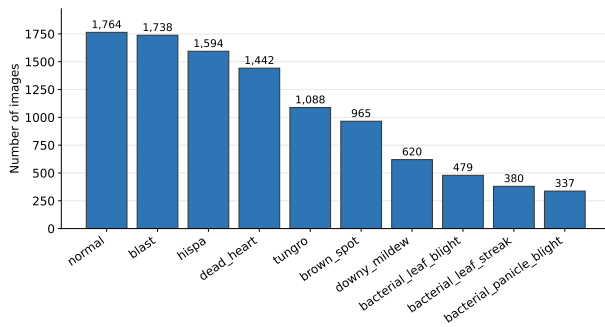


Fig. 4. Class distribution of paddy disease classification (Kaggle, India).



Fig. 5. Sample images from microsoft rice disease challenge (Egypt)

VI. METHODOLOGY

The proposed methodology is a three-stage pipeline which aims to provide an accurate and explainable classification of rice diseases. The first stage comprises the preprocessing pipeline and the adaptation of several pretrained architectures by transfer learning. In the second stage, the models are fine-tuned by unfreezing the top 30% of their backbones, the best architecture is selected on the validation subset, and it is retrained with three random seeds and evaluated on the held-out test subset. In the third stage, Grad-CAM is used to provide visual explanations of the predictions, and predictions with a low softmax score are referred to an expert. Fig. 7 illustrates

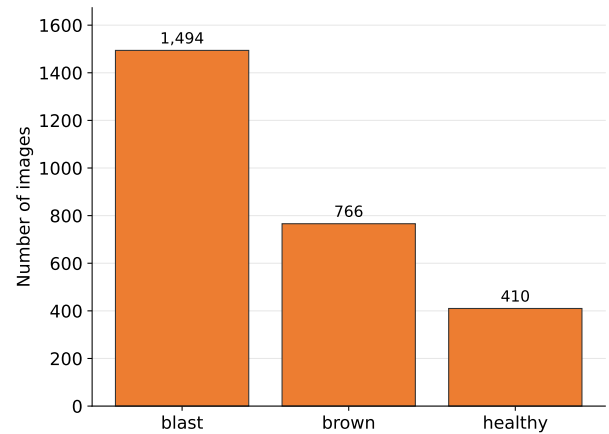


Fig. 6. Class distribution of the Microsoft Rice Disease Challenge (Egypt) dataset (labeled RGB images).

TABLE IV. NUMBER OF ORIGINAL IMAGES PER CLASS AND SOURCE, AND THE TRAIN/VALIDATION/TEST SPLIT

Class	Paddy	Zindi	Total	Train	Val	Test
bacterial_leaf_blight	479	–	479	335	72	72
bacterial_leaf_streak	380	–	380	266	57	57
bacterial_panicle_blight	337	–	337	236	51	50
blast	1,738	1,494	3,232	2,262	485	485
brown_spot	965	766	1,731	1,211	260	260
dead_heart	1,442	–	1,442	1,009	217	216
downy_mildew	620	–	620	434	93	93
hispa	1,594	–	1,594	1,116	239	239
normal	1,764	410	2,174	1,522	325	327
tungro	1,088	–	1,088	762	163	163
Total	10,407	2,670	13,077	9,153	1,962	1,962

the workflow.

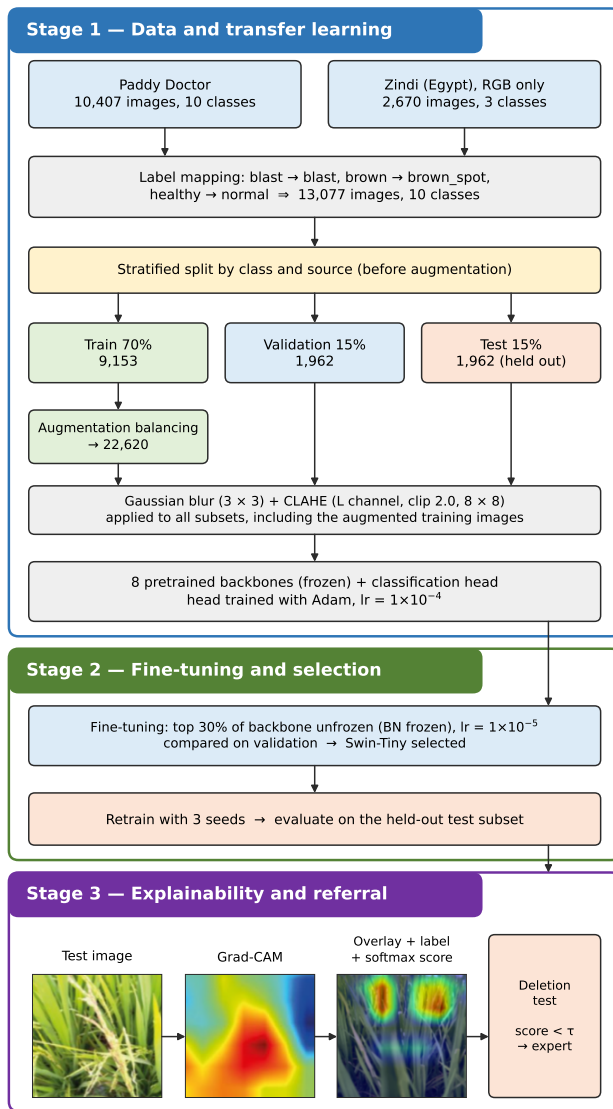


Fig. 7. Detailed workflow of the proposed framework.

A. Transfer Learning

In this study, Transfer Learning (TL) was used to improve the performance of deep learning models in classifying rice diseases, and to reduce the need for large annotated datasets. Instead of training networks from scratch, models pretrained on large-scale image datasets were adapted to the rice disease dataset. This approach allows the models to efficiently learn the disease-specific features such as lesion distribution, color and texture abnormalities, and substantially reduces the training time and computing cost.

Seven pretrained CNN models (EfficientNetB4, InceptionV3, MobileNetV2, ResNet50, ResNet101, ResNet152, and VGG19) and the transformer-based Swin-Tiny were evaluated for rice disease classification. The fine-tuning was conducted by the updating of the weights of the higher layers of every network according to the characteristics of the rice illness dataset, while the useful feature representations obtained during the pre-training were kept. This allows the models to learn how to differentiate between different types of diseases

TABLE V. LABEL MAPPING BETWEEN THE TWO DATASETS

Zindi label	Mapped class	In Paddy	In Zindi
blast	blast	Yes	Yes
brown	brown_spot	Yes	Yes
healthy	normal	Yes	Yes
–	7 remaining classes	Yes	No

and healthy rice leaves. The overall transfer learning process adopted in this work is shown in Fig. 7. Before training, the dataset was prepared through dataset integration, data splitting, augmentation-based balancing of the training subset, and image enhancement of all subsets, as described in the following subsections.

1) *Dataset preprocessing*: The preprocessing operations were performed in the following order: dataset integration, data splitting, augmentation-based balancing of the training subset, and image enhancement of all subsets (including the augmented training images). Splitting precedes augmentation so that no augmented copy of an image can appear in a different subset from its original.

a) *Dataset integration*: The Paddy Disease Classification (Kaggle, India) and Microsoft Rice Disease Challenge (Egypt) datasets were merged into a single ten-class dataset. Only the three classes of the Egyptian dataset required mapping, and each refers to the same disease in both datasets (Table V): rice blast, caused by *Magnaporthe oryzae*; brown spot, caused by *Bipolaris oryzae*; and healthy leaves. Both datasets define these classes by the same visible leaf symptoms, which supports their semantic equivalence. The remaining seven classes come from Paddy Doctor only. Differences in annotation criteria between the two sources cannot be fully excluded; their effect on the shared classes is quantified by the leave-one-source-out experiment in Section VII.

b) *Data splitting*: Data splitting was performed on the original images, before augmentation, as described in Section V-C: 70% for training, 15% for validation, and 15% for the held-out test subset, using stratified sampling by class and source with a fixed random seed. Images were resized to 224 × 224 pixels and loaded in batches of 32 using TensorFlow data pipelines with caching and prefetching for the CNN backbones; the same images and batch size were used for Swin-Tiny in PyTorch.

c) *Dataset balancing*: The merged dataset is imbalanced (Table IV). Two imbalance-handling techniques were used; both operate on the training subset only:

- **Oversampling**: Augmentation-based oversampling was applied to the training subset. The size of the largest training class was determined, and augmented copies of randomly selected images of each smaller class were generated until every class reached that size [53]. The number of augmented images generated for class i is given by Eq. (9):

$$N_{aug,i} = N_{max} - N_i \quad (9)$$

where, $N_{\max}$ is the quantity of samples in the majority class, and N_i is the quantity of samples in class i . After balancing, every class of the training subset contains the same number of images [54]:

$$N'_i = N_{\max}, \forall i \quad (10)$$

- **Class weighting:** In the configurations without over-sampling (the corresponding ablation settings and the leave-one-source-out experiment), class imbalance was handled by weighting the loss in Eq. (6) with class weights computed from the training subset, which gives larger weights to classes with fewer examples. The weight of each class was computed with Eq. (11):

$$w_i = \frac{N}{K \times N_i} \quad (11)$$

where:

- N is the total number of samples.
- K is the number of classes
- N_i is the number of samples in class i [55]

By applying these weights to the training process, the model was able to penalize minority class misclassification more severely, increasing learning's overall balance and fairness.

d) Data augmentation: To increase the variability of the dataset and decrease the likelihood of overfitting, various forms of data augmentation were applied to the training images. These forms of augmentation were applied to better represent real-world variances and increase the robustness of the model. The augmentation operations were random rotation within $\pm 20^\circ$, horizontal flipping, and brightness scaling in the range $[0.9, 1.1]$. Augmentation was applied to the training subset only, to generate the additional samples of the smaller classes. Combining the original and augmented training images yields a balanced training set of 22,620 images (2,262 per class). Fig. 8 shows the class distribution of the training subset before and after balancing.

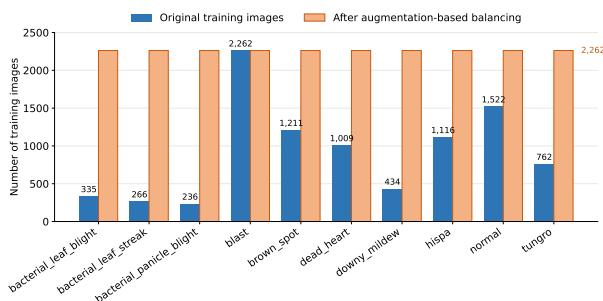


Fig. 8. Class distribution of the training subset before and after augmentation-based balancing. The validation and test subsets are not augmented.

e) Image enhancement: Image enhancement was applied as the last preprocessing step to every image of the training, validation, and test subsets, including the augmented training images. Gaussian smoothing with a 3×3 kernel was used to suppress noise without altering the overall leaf shape. Contrast Limited Adaptive Histogram Equalization (CLAHE) was then applied to the luminance (L) channel of the CIELAB color space, with a clip limit of 2.0 and an 8×8 tile grid (Eq. (12)). Both operations are deterministic per-image transformations and were applied identically to the training, validation, and test subsets. Visually, the combination improves the visibility of lesion patterns and texture detail while preserving natural color characteristics; its effect on classification accuracy is quantified by the ablation study in Section VIII.

$$H_c(i) = \min(H(i), T) \quad (12)$$

where, $H(i)$ denotes the original histogram, T is the clip limit, and $H_c(i)$ is the clipped histogram after contrast limiting. The visualization of the CLAHE effect is presented in Fig. 9.



Fig. 9. Visualization of CLAHE.

B. Fine-Tuning

After the transfer learning stage, the models were fine-tuned to adapt them to the rice disease dataset. The lower layers of the backbone were kept frozen to preserve the generic features learned during pretraining, and only the upper layers were updated; batch-normalization layers remained frozen. The top 30% of the backbone layers were unfrozen for all architectures; this ratio was chosen in preliminary experiments with EfficientNetB4 (Section VIII).

C. Explainable Artificial Intelligence

To improve the transparency and interpretability of the proposed system, Explainable Artificial Intelligence (XAI) was added utilizing Gradient-weighted Class Activation Mapping (Grad-CAM). The fine-tuned model was used to compute class activation heatmaps on the test images, indicating the image regions that contribute most to the predicted class. The heatmaps were overlaid on the original images together with the predicted label and its softmax score. A prototype interface additionally attaches a predefined description of the predicted class (symptoms and common management practices); this description is not specific to the individual image and has not been validated by agronomists.

VII. EXPERIMENTS AND RESULTS

In this section the proposed framework is experimentally evaluated. It begins by explaining the experimental setup, including the computational environment, model architecture, training configuration, and evaluation metrics, followed by the collected data and a full discussion and analysis of the model performance.

A. Experimental Setup

This subsection explains the experimental configuration employed to train and assess the proposed rice disease categorization framework.

1) *Computational environment*: The proposed model was evaluated and implemented using the Kaggle computational environment. The platform provided the following resources:

- Framework: TensorFlow and Keras (CNN backbones); PyTorch with the timm library (Swin-Tiny)
- Programming Language: Python
- GPU Accelerator: NVIDIA Tesla T4
- Dataset Storage Capacity: 214.75 GB
- Free GPU Access: Up to 30 hours per week
- Pre-trained backbones: seven CNNs and Swin-Tiny (ImageNet weights)
- Input Image Size: 224×224 pixels

2) *Model architecture*: The seven CNN backbones were implemented in TensorFlow/Keras and Swin-Tiny in PyTorch using the timm implementation `swin_tiny_patch4_window7_224`. All backbones share the same classification head, preprocessing, data split, and two-stage training schedule. Swin-Tiny achieved the best performance and was selected as the final model, and EfficientNetB4 was the best CNN; both are described below.

a) *EfficientNetB4 backbone*: EfficientNetB4 is a convolutional neural network architecture that is part of the EfficientNet family. It was designed with a compound scaling method that uniformly scales the network depth, width and input resolution with pre-determined scaling coefficients. The design consists of a first convolution layer, and a set of Mobile Inverted Bottleneck Convolution (MBConv) blocks combined with Squeeze-and-Excitation (SE) modules to enhance the channel-wise feature recalibration. These blocks extract hierarchically visual information in a computationally efficient manner. EfficientNetB4 employs residual connections and depthwise separable convolutions inside the MBConv blocks to decrease the number of parameters and computational cost, without compromising performance. The network was originally designed for an input resolution of 380×380 pixels and has more than 19 million parameters. EfficientNetB4 has been widely used as a backbone feature extractor for various computer vision applications, including medical image analysis, plant disease detection, and fine-grained image classification [56] [57], owing to its favorable balance between accuracy and computational cost.

b) *Customized classification head*: The original classification layer of EfficientNetB4 was removed and replaced with a custom classification head. The resulting model is composed of the following parts:

- Input: $224 \times 224 \times 3$ RGB images (reduced from the original 380×380 for computational efficiency).
- Backbone: the ImageNet-pretrained EfficientNetB4 without its top layers (MBConv blocks with Squeeze-and-Excitation); its last convolutional layer (`top_conv`) is used as the Grad-CAM target layer.
- Average pooling with a 2×2 window, followed by flattening of the pooled feature maps.
- A fully connected layer with 220 units and ReLU activation.
- Dropout with a rate of 0.25.
- A fully connected output layer with 10 units and softmax activation, which produces the class scores of Eq. (4).

The same head is used for all CNN backbones.

c) *Swin-Tiny Backbone*: Swin-Tiny splits the input image into 4×4 patches, embeds them linearly, and processes them in four stages with 2, 2, 6, and 2 Swin Transformer blocks, which produce feature maps of size $56 \times 56 \times 96$, $28 \times 28 \times 192$, $14 \times 14 \times 384$, and $7 \times 7 \times 768$. The output of the last stage is used as the Grad-CAM target and is fed to the same classification head as the CNN backbones (Fig. 10). Three implementation details differ from the Keras models: the inputs are normalized with the ImageNet mean and standard deviation, training uses automatic mixed precision, and in Stage 2 the top 30% of the backbone parameter tensors are unfrozen (Swin-Tiny uses layer normalization, so no batch-normalization layers have to be kept frozen).

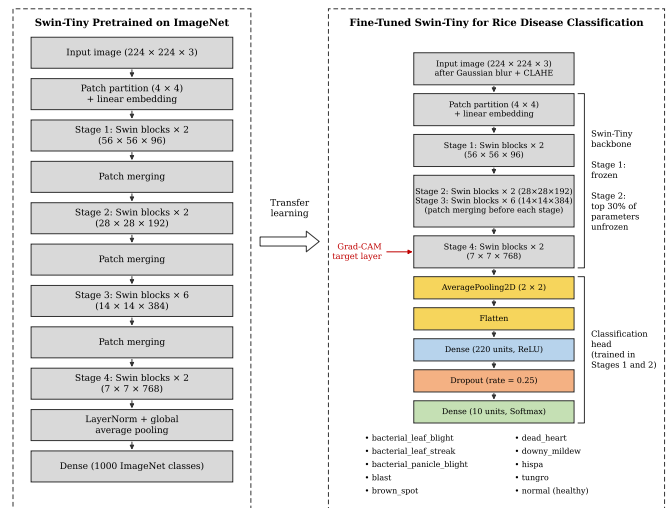


Fig. 10. Architecture of the final model: Pretrained Swin-Tiny (left) and the fine-tuned model with the classification head (right).

3) *Training configuration*: The training configuration follows the formulation in Section IV; the complete pipeline is summarized in Algorithm 1.

- Two-stage optimization with Adam: In Stage 1, the backbone is frozen and the head is trained with a learning rate of 1×10^{-4} for at most 15 epochs. In Stage 2, the top 30% of the backbone layers (excluding batch-normalization layers) are unfrozen and trained together with the head with a learning rate of 1×10^{-5} for at most 30 epochs.
- Loss and regularization: Categorical cross-entropy (Eq. (6)) with dropout of 0.25 in the head. Class weights are used only in the configurations without oversampling.
- Callbacks (both stages): Model checkpointing on the lowest validation loss; early stopping after 6 epochs without improvement in validation loss, restoring the best weights; and reduction of the learning rate by a factor of 0.3 after 3 epochs without improvement, with a minimum of 1×10^{-7} .
- Repetitions: The final configuration is trained with three random seeds (42, 123, and 2024) on the same data split, and results are reported as mean $\pm$ standard deviation.

Algorithm 1 Proposed Training, Fine-Tuning, and Evaluation Framework

Require: Rice leaf disease dataset

Ensure: Best fine-tuned model, evaluation metrics, and Grad-CAM visualizations

Initialize parameters:

IMAGE_SIZE $\leftarrow$ (224, 224), BATCH_SIZE $\leftarrow$ 32,
NUM_CLASSES $\leftarrow$ 10

$\eta_1 \leftarrow 1 \times 10^{-4}$, $\eta_2 \leftarrow 1 \times 10^{-5}$, $E_1 \leftarrow 15$, $E_2 \leftarrow 30$

Load the original images of both datasets and map the Zindi labels.

Split the original images into training (70%), validation (15%), and test (15%) subsets, stratified by class and source. Balance the training subset by augmentation (Eq. (9)–(10)). Apply Gaussian blurring and CLAHE to all subsets.

for each model in {EfficientNetB4, MobileNetV2, ResNet50, ResNet101, ResNet152, InceptionV3, VGG19, Swin-Tiny} **do**

 Load the ImageNet-pretrained backbone and freeze it.

 Attach the classification head.

 Stage 1: train the head with η_1 for up to E_1 epochs (early stopping on validation loss).

 Stage 2: unfreeze the top 30% of the backbone (BN frozen) and train with η_2 for up to E_2 epochs.

 Record validation metrics.

end for

Select the model on the validation subset.

Retrain the selected model and the best CNN with three random seeds.

Evaluate on the held-out test subset: accuracy, macro- and weighted-averaged precision, recall, and F1-score, confusion matrix, per-source accuracy, and calibration.

Generate Grad-CAM heatmaps and overlay them on the test images.

return Final model, test metrics, and Grad-CAM visualizations.

4) *Evaluation criteria:* The framework was evaluated with accuracy, precision, recall, and F1-score, and per-class performance was analyzed with confusion matrices and classification reports. For multi-class evaluation, precision, recall, and F1-score are computed per class and then averaged. Macro-averaging gives each class equal weight, whereas weighted averaging weights each class by its number of test samples; weighted-averaged recall is therefore identical to accuracy. Both averages are reported. The metric definitions are given below. Accuracy [58]: calculates the proportion of accurate predictions relative to the total number of occurrences assessed, calculated in Eq. (13):

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (13)$$

Precision [59]: calculates the ratio of accurately recovered instances to the total retrieved instances within the dataset, calculated in Eq. (14):

$$\text{Precision} = \frac{TP}{TP + FP} \quad (14)$$

Recall [59]: calculates the ratio of accurately recovered instances to the total number of correct instances in the dataset, calculated in Eq. (15):

$$\text{Recall} = \frac{TP}{TP + FN} \quad (15)$$

F1-score [59]: denotes the harmonic mean of recall and precision, calculated in Eq. (16):

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (16)$$

where, TP (True Positive) denotes the number of correctly classified samples belonging to the target class, TN (True Negative) represents the number of correctly classified samples that do not belong to the target class, FP (False Positive) denotes the number of samples incorrectly assigned to the target class, and FN (False Negative) represents the number of target-class samples incorrectly classified as another class.

The calibration of the softmax scores is measured by the expected calibration error (ECE). The test samples are divided into $M = 15$ bins B_m according to their maximum softmax score, and

$$\text{ECE} = \sum_{m=1}^M \frac{|B_m|}{n} |\text{acc}(B_m) - \text{conf}(B_m)|, \quad (17)$$

where, $\text{acc}(B_m)$ and $\text{conf}(B_m)$ are the accuracy and the mean maximum softmax score in bin B_m , and n is the number of test samples.

TABLE VI. STAGE 1 (FROZEN BACKBONE): VALIDATION ACCURACY OF THE EVALUATED ARCHITECTURES AT THE EPOCH WITH THE LOWEST VALIDATION LOSS

Model	Validation accuracy (%)
EfficientNetB4	89.25
InceptionV3	81.55
MobileNetV2	86.95
ResNet50	91.23
ResNet101	89.50
ResNet152	88.99
VGG19	82.87
Swin-Tiny	89.81

TABLE VII. TEST-SET PERFORMANCE OF THE FINE-TUNED ARCHITECTURES (SEED 42; MACRO-AVERAGED PRECISION, RECALL, AND F1-SCORE)

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	MCC
EfficientNetB4	91.74	93.14	91.30	92.14	0.903
InceptionV3	82.47	81.64	84.18	82.67	0.796
MobileNetV2	87.46	88.27	86.60	87.31	0.853
ResNet50	90.11	90.70	89.80	90.21	0.884
ResNet101	89.76	91.04	89.70	90.24	0.880
ResNet152	89.40	90.17	87.53	88.65	0.876
VGG19	86.49	88.18	85.18	86.50	0.842
Swin-Tiny	93.88	93.63	94.22	93.87	0.929

B. Results and Discussion

This section reports the results of the proposed framework under the evaluation protocol of Section V-C. All architectures were trained on the same balanced training subset with the same preprocessing, classification head, and two-stage schedule (seed 42); the frozen-backbone stage is compared on the validation subset, and the fine-tuned models are evaluated on the held-out test subset.

1) *Architecture comparison:* Table VI reports the validation accuracy of each architecture at the end of Stage 1 (frozen backbone), and Table VII reports the test-set performance of each architecture after Stage 2 (top 30% of the backbone unfrozen). At the frozen stage, ResNet50 reached the highest validation accuracy (91.23%), followed by Swin-Tiny (89.81%) and ResNet101 (89.50%). After fine-tuning, Swin-Tiny achieved the highest test accuracy (93.88%), macro-F1 score, and MCC, followed by EfficientNetB4 (91.74%) and the ResNet family (89.40–90.11%). Swin-Tiny also reached a higher validation accuracy than EfficientNetB4 after fine-tuning (94.09% versus 92.46% for seed 2024) and was therefore selected as the final model. The ranking at the frozen stage does not determine the ranking after fine-tuning: ResNet50 moved from first to third place, and EfficientNetB4 from fourth to second.

2) *Performance on the held-out test set:* Table VIII reports the test performance of the final model, Swin-Tiny, and of the best CNN, EfficientNetB4, averaged over three random seeds. Swin-Tiny achieves an accuracy of $94.17 \pm 0.34\%$, a macro-F1 score of $94.16 \pm 0.39\%$, and a Matthews correlation coefficient

TABLE VIII. TEST-SET PERFORMANCE OF THE FINE-TUNED SWIN-TINY AND EFFICIENTNETB4 (MEAN $\pm$ STANDARD DEVIATION OVER THREE SEEDS)

Metric (%)	Swin-Tiny	EfficientNetB4
Accuracy	94.17 ± 0.34	91.54 ± 0.64
Macro precision / recall / F1	94.21 / 94.21 / 94.16	92.89 / 91.48 / 92.11
Weighted precision / recall / F1	94.24 / 94.17 / 94.18	91.62 / 91.54 / 91.54
MCC	0.932 ± 0.004	0.901 ± 0.007
Accuracy on Paddy Doctor images	95.00 ± 0.36	93.34 ± 0.34
Accuracy on Egyptian images	90.94 ± 0.63	84.54 ± 2.00

of 0.932 ± 0.004 , and it outperforms EfficientNetB4 in every seed (93.88%, 94.09%, and 94.55% versus 91.74%, 90.83%, and 92.05%). The standard deviations across seeds are below 0.7 percentage points for both models, which indicates stable training. The gap between training accuracy (above 98%) and validation accuracy (92–94%) reflects the repeated augmented copies of the smallest classes in the balanced training set.

3) *Cross-source analysis:* On the held-out test set, the fine-tuned Swin-Tiny reaches $95.00 \pm 0.36\%$ accuracy on Paddy Doctor images and $90.94 \pm 0.63\%$ on images from the Egyptian dataset, and EfficientNetB4 reaches $93.34 \pm 0.34\%$ and $84.54 \pm 2.00\%$ (Table IX). To measure transfer between sources, using the EfficientNetB4 backbone, a leave-one-source-out experiment was conducted on the three classes present in both datasets (blast, brown spot, and normal). Since no oversampling was applied in this experiment, class weights (Eq. (11)) were used. A model trained on Paddy Doctor images achieves 44.60% accuracy and 34.53% macro-F1 on all Egyptian images (MCC = 0.03), while a model trained on Egyptian images achieves 38.33% accuracy and 36.25% macro-F1 on all Paddy Doctor images (MCC = 0.11). Both results are close to the chance level of a three-class problem, although the same models reach 98.21% and 87.28% accuracy, respectively, on validation images from their own source. This severe domain shift indicates that a model trained on a single source largely learns source-specific cues. In contrast, the models trained on the merged dataset reach 84.54% (EfficientNetB4) and 90.94% (Swin-Tiny) accuracy on the Egyptian test images. Cross-source generalization therefore cannot be assumed from in-source accuracy, and including images from both sources in training is what enables acceptable performance on each of them.

4) *Confidence calibration and selective prediction:* On the held-out test set, the fine-tuned Swin-Tiny has an ECE (Eq. (17)) of 2.68% for seed 42 (3.27% and 3.34% for seeds 123 and 2024), compared with 4.30% for EfficientNetB4 (Fig. 11a). The maximum softmax score separates correct from incorrect predictions with an area under the ROC curve of 0.91; incorrect predictions have a mean maximum score of 0.78, compared with 0.98 for correct predictions, so the model remains moderately over-confident on its errors. The softmax output is therefore reported as a model score rather than as a diagnostic confidence. For operational use, predictions with a maximum softmax score below a threshold τ are withheld and referred to an expert. The threshold was selected on the

TABLE IX. PER-SOURCE TEST ACCURACY AND
LEAVE-ONE-SOURCE-OUT RESULTS

Training Data	Test Data	Acc. (%)	Macro-F1 (%)
Merged, Swin-Tiny	Paddy test images	95.00	94.37 [†]
Merged, Swin-Tiny	Zindi test images	90.94	89.82 [†]
Merged, EfficientNetB4	Paddy test images	93.34	93.43 [†]
Merged, EfficientNetB4	Zindi test images	84.54	84.79 [†]
Paddy only, EfficientNetB4 (3 classes)	All Zindi images	44.60	34.53
Zindi only, EfficientNetB4 (3 classes)	All Paddy images	38.33	36.25

[†] Macro-F1 is computed over the classes present in each source using a single seed.
Merged-model accuracy is averaged over three seeds.

TABLE X. COMPUTATIONAL COST OF THE TWO BEST MODELS
(NVIDIA TESLA T4)

Metric	Swin-Tiny	EfficientNetB4
Framework	PyTorch (FP32)	TensorFlow
Parameters, incl. head (M)	29.0	21.2
Model file size (MB)	110.9 ^a	213.9 ^b
Preprocessing (blur + CLAHE) per image (ms)	2.0	3.1
Inference latency, batch 1 (ms/image)	11.5	21.7
Inference latency, batch 32 (ms/image)	4.9	5.4
End-to-end latency, batch 1 (ms/image)	13.5	24.8
Peak GPU memory during inference (MB)	731	875
Training time per run (min)	45.0 ± 12.3	46.7 ± 8.2

^aweights only; ^bcheckpoint including optimizer state

validation subset as the smallest value for which the accuracy on accepted validation images reaches 98%, giving $\tau = 0.959$. On the test subset, 84.8% of the images are then classified automatically with an accuracy of 98.44%, and the remaining 15.2% are referred (Fig. 11b); the corresponding values for EfficientNetB4 are 82.8% and 97.97%.

5) *Computational efficiency*: Table X reports the computational cost of the two best models, measured on the NVIDIA Tesla T4 GPU used for training. Swin-Tiny has more parameters than EfficientNetB4 but a lower single-image latency in this setting. Because the two models run in different frameworks (PyTorch and TensorFlow), the latency comparison is indicative rather than exact.

6) *Explainable AI results*: Beyond quantitative performance, understanding the basis of the model's predictions is important for real-world use in precision agriculture. The Gradient-weighted Class Activation Mapping (Grad-CAM) technique [60] was also used for visual interpretation of the model's predictions, to show the areas of the image that contributes most to each projected rice illness class.

Grad-CAM produces class-discriminative localization maps highlighting the important regions of the input image for classification by computing the gradients of the predicted class with respect to the feature maps of the last convolutional

layer (for Swin-Tiny, the $7 \times 7 \times 768$ output of its last stage). These gradients are transformed into channel-wise importance weights, to produce a coarse localization heatmap that is fused with the corresponding activation maps. This generated heatmap can be overlaid on the original image of the rice leaf to visually interpret the regions that can have strong influence on the model's decision.

Grad-CAM was applied to correctly classified test images from all ten classes (Fig. 13). In most examples, the highlighted regions lie on or near leaves with visible symptoms, although Grad-CAM provides only a coarse localization at the 7×7 resolution of the last stage. Because the datasets provide no lesion annotations, this visual agreement cannot be measured as an overlap score. A deletion test was therefore used as a quantitative check: for 200 randomly selected test images, pixels were progressively replaced by the image mean, either in order of decreasing Grad-CAM relevance (most relevant first), in order of increasing relevance (least relevant first), or in randomly selected 16×16 blocks, and the relative drop in the predicted-class score was measured (Fig. 12). Removing the 10% and 20% most relevant pixels reduced the predicted-class score of Swin-Tiny by 22.9% and 33.3%, compared with 0.7% and 4.0% for the least relevant pixels and 5.8% and 9.1% for random blocks. Block-wise random removal was used as the baseline because removing scattered random pixels corrupts every 4×4 patch of a patch-based transformer at once; with scattered pixels, the score of Swin-Tiny dropped more than with Grad-CAM-ordered removal (34.6% versus 22.9% at 10%). For EfficientNetB4, removing the 10% most relevant pixels reduced the score by 29.1%, compared with 17.7% for scattered random pixels. These results show that the regions highlighted by Grad-CAM carry much of the evidence used by both models, but they do not establish that every prediction is based on the lesion itself.

VIII. DISCUSSION AND COMPREHENSIVE PERFORMANCE ANALYSIS

This section analyzes the experimental results presented in the previous section. The discussion focuses on the factors that affected the performance of the evaluated models to investigate the contribution of the proposed preprocessing and fine-tuning strategies and to compare the proposed framework with recent studies reported in literature. Finally, some practical limitations of the proposed approach and some potential directions for future improvement are discussed.

A. Overall Performance Analysis

The experimental results show that all the deep learning models tested benefit from transfer learning and fine-tuning, but the classification performance of these models can be significantly different depending on the architectural characteristics of the models. On the merged dataset, Swin-Tiny achieved the best performance among all evaluated architectures, with a test accuracy of $94.17 \pm 0.34\%$, followed by EfficientNetB4 with $91.54 \pm 0.64\%$.

The transformer-based Swin-Tiny achieved the best performance, 2.6 percentage points above the best CNN, EfficientNetB4, and its advantage was largest on the Egyptian images (90.94% versus 84.54%). A plausible explanation is that

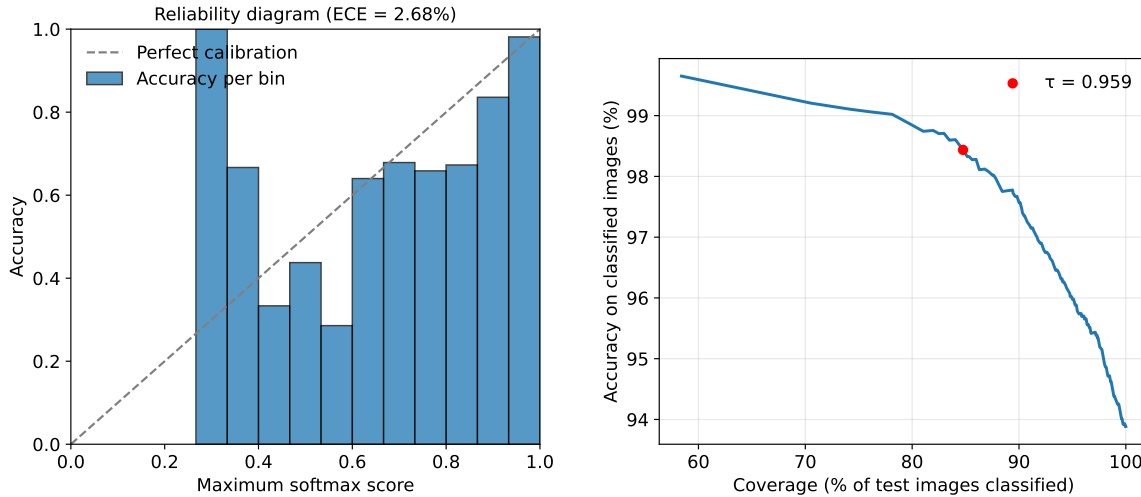


Fig. 11. Calibration and selective prediction of the fine-tuned Swin-Tiny. (a) Reliability diagram of the fine-tuned Swin-Tiny on the held-out test set. (b) Accuracy on the classified test images as a function of coverage; the red point marks the threshold τ selected on the validation subset (seed 42).

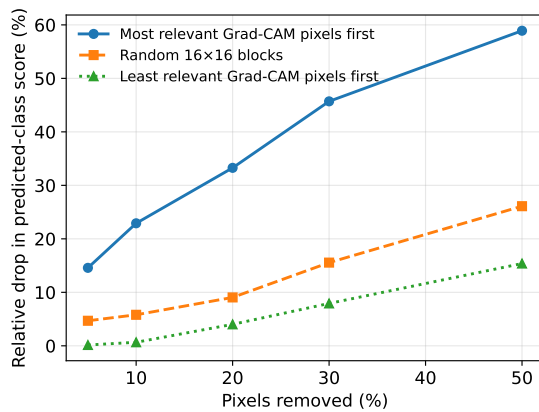


Fig. 12. Deletion test for Swin-Tiny: relative drop in the predicted-class score when the most relevant Grad-CAM pixels, the least relevant pixels, or random 16×16 blocks are removed (200 test images).

shifted-window self-attention aggregates context over larger image regions than the local receptive fields of CNNs, which can help when lesions are small and scattered among leaves in cluttered field images; this interpretation was not tested directly in this study.

VGG19 and the deeper ResNets (ResNet101 and ResNet152) have many more parameters than ResNet50 but do not improve the classification accuracy (Table VII), so their additional computational cost brings no benefit on this dataset. On the other hand, MobileNetV2 achieved reasonable classification performance with much fewer parameters, and was suitable for lightweight deployment, but its compact structure limited its ability to learn highly discriminative disease-specific features.

Among the CNNs, EfficientNetB4 performed best, which is consistent with its compound scaling of depth, width, and resolution.

Overall, Swin-Tiny provides the best accuracy and calibration together with the lowest single-image latency in the tested setting (Table X), whereas EfficientNetB4 remains a competitive CNN alternative with fewer parameters.

B. Impact of Fine-Tuning

Fine-tuning is an integral part of the proposed framework. Transfer learning enables the models to benefit from generic visual features learned from ImageNet, while fine-tuning helps these features to be more specific for rice disease recognition by adapting them to disease-specific visual characteristics.

Updating the high-level feature representations is expected to help the networks discriminate subtle differences between visually similar disease classes. This adaptation is particularly important for rice disease classification as many diseases have similar lesion colors, shapes and texture patterns.

For most architectures, the test accuracy after fine-tuning (Table VII) exceeds the validation accuracy at the frozen stage (Table VI); the largest differences are observed for Swin-Tiny (93.88% versus 89.81%) and VGG19 (86.49% versus 82.87%), whereas ResNet50 does not improve (90.11% versus 91.23%). Because the two values are measured on different subsets, this comparison is indicative only. The ranking also changes substantially between the two stages: ResNet50 ranks first at the frozen stage and third after fine-tuning, Swin-Tiny moves from second to first, and EfficientNetB4 from fourth to second. Model selection should therefore be based on fine-tuned models rather than on frozen-backbone performance.

C. Effect of Layer Unfreezing Ratios

The unfreezing ratio of 30% was selected in preliminary experiments with EfficientNetB4, in which 20%, 30%, and 50% of the backbone layers were unfrozen, and it was retained for all architectures. Other ratios were not evaluated in the final experiments, so the effect of the unfreezing ratio on the other architectures remains to be investigated.

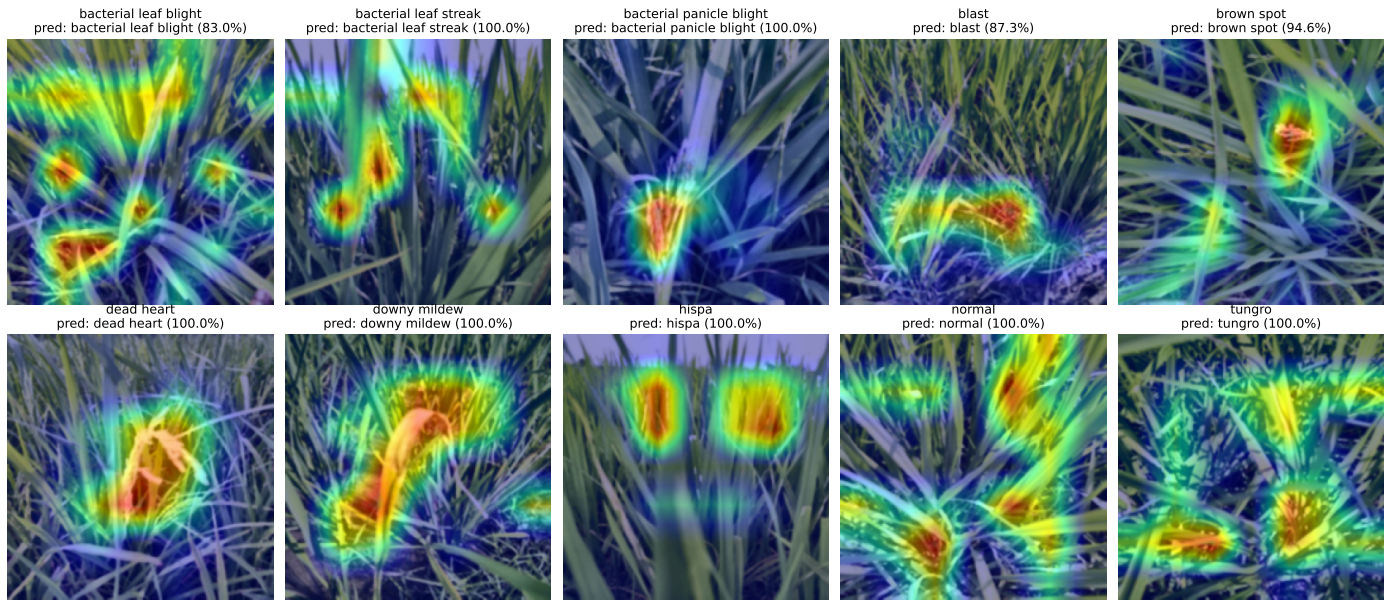


Fig. 13. Grad-CAM overlays of the fine-tuned Swin-Tiny for one correctly classified test image per class, with the predicted class and its softmax score.

TABLE XI. FACTORIAL ABLATION OF THE PREPROCESSING PIPELINE
(EFFICIENTNETB4, HELD-OUT TEST SET)

Blur	CLAHE	Augmentation	Accuracy (%)	Macro-F1 (%)
✓	✓	✓	91.54 ± 0.64	92.11 ± 0.51
✓	–	✓	93.07	93.25
–	✓	✓	92.10	92.71
✓	✓	–	91.59	91.93
–	–	✓	92.76	93.05
–	✓	–	91.44	91.97
✓	–	–	91.44	92.07
–	–	–	92.41	92.81

D. Ablation Study of the Preprocessing Pipeline

A full $2 \times 2 \times 2$ factorial ablation of Gaussian blurring, CLAHE, and augmentation-based oversampling was performed with the EfficientNetB4 backbone, with all other settings unchanged. When oversampling is disabled, class imbalance is handled by class-weighted loss only, so the oversampling itself is ablated. Owing to the computational budget of the Kaggle platform (approximately 30 GPU hours per week), the full configuration was run with three seeds and the remaining seven configurations with one seed (seed 42). Differences smaller than the standard deviation of the full configuration are therefore not interpreted. The results on the held-out test set are summarized in Table XI.

None of the preprocessing components improved test accuracy beyond the variability of the full pipeline across seeds (± 0.64 points). Averaged over the four configurations with and without each component, CLAHE changed test accuracy by -0.75 points (91.67% versus 92.42%), Gaussian blurring by -0.27 points (91.91% versus 92.18%), and augmentation-based oversampling, compared with class-weighted loss, by $+0.65$ points (92.37% versus 91.72%). The best configura-

tion combined blurring and oversampling without CLAHE (93.07%), and oversampling alone reached 92.76%. These single-seed differences are close to the seed variability and should not be over-interpreted, but they consistently show that CLAHE does not improve accuracy on this dataset. A likely reason is that the field images are already well exposed, and contrast enhancement shifts their color and contrast statistics away from those of the ImageNet images used for pretraining.

E. Class-wise Error Analysis

Besides the overall performance metrics, the classification performance of the fine-tuned Swin-Tiny is shown as a confusion matrix in Fig. 14. The test accuracy of 93.88% for this seed indicates that the great majority of test samples were correctly identified, and the confusion matrix is dominated by its diagonal.

The classes with the most distinctive symptoms are recognized most reliably: *dead heart* (F1 = 99.08%), *bacterial panicle blight* (97.03%), and *bacterial leaf streak* (96.49%). The lowest F1 score is obtained for *downy mildew* (87.00%), mainly because of a lower precision (81.31%), followed by *brown spot* (91.76%).

Most errors occur between classes with similar visual symptoms. Brown spot and blast are the most frequently confused pair (15 brown spot images predicted as blast and 9 blast images predicted as brown spot), because both diseases produce dark necrotic lesions of similar shape and color, especially at intermediate stages. Blast and normal images are also confused in both directions (7 and 6 images), which is consistent with small or early lesions that are hard to distinguish from healthy leaves.

Further confusions occur between *hispa* and *normal* (12 images) and between *hispa* and *downy mildew* (5 images), when early insect damage resembles natural leaf texture, and *tungro* images are occasionally predicted as downy mildew (6

images), since both conditions start with diffuse discoloration. Overall, the errors are concentrated among visually similar classes rather than distributed randomly.

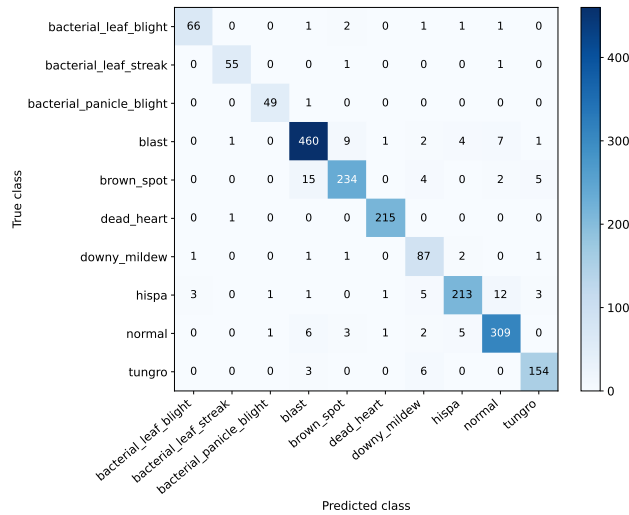


Fig. 14. Confusion matrix of the fine-tuned Swin-Tiny on the held-out test set (seed 42).

F. Comparison with Previous Studies

To position the proposed framework, its performance is listed alongside representative recent studies on rice leaf disease classification in Table XII, which reports the dataset, number of classes, use of explainability techniques, reported accuracy, and evaluation setting of each study. All earlier studies in the table were trained and evaluated on a single dataset, Paddy Doctor, whereas the proposed framework is trained on images from two sources (Paddy Doctor and the Microsoft Rice Disease Challenge dataset) and evaluated on a held-out test set after splitting before augmentation.

Because the reported accuracies were obtained on different datasets, class sets, and data splits, they are not directly comparable, and Table XII is provided to position the proposed framework rather than to rank methods. In particular, several earlier studies do not state whether augmentation was performed before or after splitting, which can strongly affect reported accuracy.

Differently from previous studies which only reported the classification accuracy, the proposed approach also exploits Explainable Artificial Intelligence (Grad-CAM) to visualize the areas of the image that led to the model predictions. In addition, the contribution of each preprocessing component is quantified by a factorial ablation, cross-source transfer is measured explicitly, and the calibration of the softmax scores is evaluated.

The main distinguishing features of the proposed framework are therefore its evaluation protocol and its cross-source analysis, rather than a higher headline accuracy.

G. Comparison of Explainability Methods with Related Work

This subsection compares the explainability methods used in recent rice disease studies (Table XIII) to place the Grad-

CAM analysis of the previous section in context. There is increasing interest in explainability from a deep learning (DL) perspective for plant disease diagnosis, especially as a way to increase transparency and build trust with end-users. However, there are many practical benefits as well. The most popular XAI methods in the literature are LIME, Grad-CAM and SHAP. Grad-CAM produces gradient based spatial heatmaps to highlight regions of interest in an input image for visual inspection by farmers, whereas LIME and SHAP provide explanations based on perturbations or games as shown in Table XIII.

A main difference of the present study is the scope of the classification task: among the studies in Table XIII, it is the only one that applies Grad-CAM to a ten-class problem built from two datasets (MS Rice Egypt and Paddy Doctor). Earlier Grad-CAM studies such as [61] address smaller class spaces, for example four disease classes with 97.86% accuracy.

Furthermore, research that relies only on LIME [62], or that combines several XAI methods [63], [64], is not able to benefit from the visual localization strength of Grad-CAM, which links model attention to specific image regions. This type of spatial explainability is particularly useful in agricultural environments where domain experts can benefit from visual confirmation of the model focus (e.g. lesion areas or discolored tissue) as a valuable validation tool. In summary, the proposed framework combines a ten-class, two-source classification task with Grad-CAM explanations whose relevance is checked by a deletion test.

H. Limitations of the Proposed Framework

The proposed framework has several limitations. First, the computing resources of the Kaggle platform limited GPU memory and training time. Computationally expensive experiments such as extensive hyperparameter tuning, larger model variants, and k-fold cross-validation were therefore not performed, and the ablation configurations other than the full pipeline were run with a single seed.

Second, seven of the ten classes are represented by a single source, so class identity and acquisition source are partially confounded, and a model may exploit source-specific cues such as background or camera characteristics. The leave-one-source-out experiment quantifies cross-source transfer only for the three shared classes; a dataset in which every class is collected in several regions would be required to evaluate cross-source generalization for all ten classes. Differences in annotation criteria between the two datasets also cannot be fully excluded.

Third, the unfreezing ratio of 30% was chosen in preliminary experiments with EfficientNetB4 and was not tuned for the other architectures, and the architectures other than Swin-Tiny and EfficientNetB4 were trained with a single seed. The preprocessing ablation and the leave-one-source-out experiment were carried out with EfficientNetB4 only, and Swin-Tiny was implemented in a different framework (PyTorch) from the CNN backbones (TensorFlow).

Fourth, the model has not been evaluated on images that are strongly blurred, occluded, of low quality, or that show crops or diseases outside the ten classes. A softmax threshold

TABLE XII. COMPARISON BETWEEN THE PROPOSED FRAMEWORK AND PREVIOUS STUDIES

Study	Dataset	Classes	XAI	Accuracy (%)	Evaluation / remarks
Hybrid Xception + OVR-SVM	Paddy Doctor	10	No	94.00	Single dataset
Transfer Learning (ResNet34)	Paddy Doctor	13	No	97.50	Single dataset
Modified ResNet50	Paddy Doctor (augmented)	10	No	98.03	Single dataset
EfficientNet-B4	Paddy Doctor	10	No	96.91	Single dataset
MobileNetV3 + CatBoost	Paddy Doctor	10	No	98.52	Single dataset
CNN vs. ViT	Paddy Doctor	10	No	93.00	Architecture comparison
Proposed (Swin-Tiny)	Merged (Paddy Doctor + MS Rice Egypt)	10	Grad-CAM	94.17 $\pm$ 0.34	Held-out test, 3 seeds; split before augmentation

TABLE XIII. COMPARISON OF EXPLAINABILITY TECHNIQUES USED IN RECENT RICE DISEASE CLASSIFICATION STUDIES

Ref.	Year	Classes / Images	XAI Technique	Accuracy (%)	Public Dataset
[62]	2025	13	LIME	99.10	No
[63]	2025	4	Grad-CAM + LIME	99.49	Yes
[64]	2025	6 diseases (6000 images)	SHAP, LIME, Grad-CAM	82.00	Yes
[61]	2023	4 diseases (5932 images)	Grad-CAM	97.86	Yes
Proposed	-	10 diseases (2 datasets)	Grad-CAM	94.17	Yes

does not reliably detect such out-of-distribution inputs, and dedicated out-of-distribution detection and field validation are required before deployment.

Fifth, the framework uses RGB images only and does not exploit multispectral information, and it does not estimate disease severity. The class descriptions attached to the predictions are predefined and have not been validated by agronomists.

Future work will extend the framework to larger multi-region datasets, severity estimation, out-of-distribution detection, efficient deployment on edge devices, and field testing.

IX. CONCLUSION AND FUTURE WORK

In this study, a deep learning based framework for the automatic classification of rice leaf diseases using transfer learning, fine-tuning, image preprocessing and Explainable Artificial Intelligence was proposed. The primary goal was to develop an accurate classification framework for ten rice disease categories and to evaluate it rigorously on images from two acquisition sources.

The proposed framework uses a preprocessing pipeline that includes Gaussian blurring, CLAHE, and augmentation-based class balancing. The two datasets were merged, split into training, validation, and held-out test subsets before augmentation, and only the training subset was balanced. Eight deep learning architectures were compared using transfer learning and fine-tuning. Swin-Tiny performed best and achieved a test accuracy of $94.17 \pm 0.34\%$ and a macro-F1 score of $94.16 \pm$

0.39% over three seeds, compared with $91.54 \pm 0.64\%$ for the best CNN, EfficientNetB4. With expert referral of low-score predictions, Swin-Tiny classified 84.8% of the test images with an accuracy of 98.44%. The factorial ablation showed that neither Gaussian blurring nor CLAHE improved test accuracy beyond the variability across seeds. The leave-one-source-out experiment showed that a model trained on a single source performs close to chance level on the other source (38–45% accuracy on three classes), whereas training on the merged dataset reaches 90.94% on the Egyptian test images with Swin-Tiny.

The architecture comparison also showed that the ranking of the backbones changes after fine-tuning, so model selection should be based on fine-tuned models. Besides, the class-wise analysis based on confusion matrix showed that most of the samples were correctly classified, whereas the remaining errors were mainly concentrated among the visually similar disease categories. The results show that the framework learns discriminative representations for most of the rice diseases considered.

Along with classification accuracy, Explainable Artificial Intelligence was integrated with Grad-CAM to get visualized insights of the model's predictions. The generated visual explanations aid in identifying the image regions, which have contributed most strongly to the model decisions, thus enhancing the transparency and interpretability of the proposed framework. This is especially relevant in agricultural applications where confidence in a system's predictions can be increased by understanding the basis of an automated diagnosis.

Overall, the findings indicate that combining data from two sources with transfer learning, fine-tuning, and explainability provides an effective approach to automated rice disease classification, whereas contrast enhancement with CLAHE did not contribute to accuracy on this dataset. The framework achieved high test performance and provided additional insight into cross-source transfer, calibration, and classification errors. Several directions can extend the proposed framework.

First, future work could explore estimating disease severity in addition to classifying diseases. Evaluating the level of severity of the infection would give more useful information to make decisions in the agricultural field and would allow the development of systems capable of monitoring the evolution of the disease.

Second, the framework could be expanded using larger and more geographically diverse datasets collected under real-field conditions. A more comprehensive evaluation of model generalization and robustness can be achieved by using images taken under different illumination conditions, backgrounds, camera devices, growth stages and geographical regions.

Another promising direction is the integration of multispectral or hyperspectral imagery with RGB images. Additional spectral information can provide useful discriminative features for distinguishing diseases with visually similar symptoms and can potentially improve early stage disease detection.

Future work could also investigate optimizing the proposed model to run on mobile and edge devices. Model compression, knowledge distillation, quantization and lightweight architectures could be explored to lower the computational demands while preserving high classification performance, allowing real-time disease diagnosis in practical agricultural settings.

Additionally, the explainability component can be extended beyond Grad-CAM by exploring a variety of XAI techniques and testing their consistency and reliability. This might give a better holistic understanding of learnt representations and enhance interpretability of automated predictions of diseases.

Finally, the proposed classification framework can be extended to a more comprehensive intelligent agricultural decision support system by adding disease severity assessment, disease progression monitoring and treatment or management recommendations. Such an extension would turn the system from a disease recognition tool to an integrated tool that can support farmers and agricultural experts throughout the process of disease diagnosis and management.

REFERENCES

- [1] H. Elbasiouny and F. Elbehiry, "Rice production in egypt: The challenges of climate change and water deficiency," in *Climate change impacts on agriculture and food security in Egypt: Land and water resources—Smart farming—Livestock, fishery, and aquaculture*. Cham, Switzerland: Springer, 2020, pp. 295–319.
- [2] Central Agency for Public Mobilization and Statistics (CAPMAS), "Annual bulletin of crop areas and plant production statistics 2023/2024," CAPMAS, Cairo, Egypt, Tech. Rep., 2025. [Online]. Available: <https://www.capmas.gov.eg>
- [3] E. Elsadek, K. Zhang, A. Mousa, G. T. Ezaz, T. L. Tola, H. Shaghaleh, A. A. A. Hamad, and Y. Alhaj Hamoud, "Study on the in-field water balance of direct-seeded rice with various irrigation regimes under arid climatic conditions in egypt using the aquacrop model," *Agronomy*, vol. 13, no. 2, p. 609, 2023.
- [4] S. Ayyad, P. Karimi, L. Ribbe, and M. Becker, "Potential improvements in crop production in egypt and implications for future water and land demand," *International Journal of Plant Production*, vol. 18, no. 3, pp. 313–334, 2024.
- [5] S. Maharjan, W. Li, S. Fazli, A. Tariq, R. Thomas, C. Rakovski, and H. El-Askary, "Enhancing water scarcity resilience in egypt through machine learning-driven phenological crop mapping and water use efficiency analysis," *International Journal of Applied Earth Observation and Geoinformation*, vol. 141, p. 104668, 2025.
- [6] M. Tohamy and H. Mostafa, "Impact of climate change on crop production and water-use efficiency in rashid city, egypt," *Water Science*, vol. 40, no. 1, p. 7, 2026.
- [7] R. Ramadan, "Agrifood policies and challenges of the agrifood system in egypt," *New Medit.: Mediterr J Econ Agric Environ*, vol. 23, pp. 37–48, 2024.
- [8] USDA Foreign Agricultural Service, "Egypt: Grain and feed annual," USDA Foreign Agricultural Service, Global Agricultural Information Network (GAIN), Cairo, Egypt, Tech. Rep. EG2025-0009, 2025. [Online]. Available: <https://www.fas.usda.gov/data/egypt-grain-and-feed-annual-9>
- [9] A. Abdel-Motleb, R. M. Abd El-Hamid, and S. S. Sayed, "Biological diversity associated with pesticides residues in certain egyptian watercourses," *Archives of Environmental Contamination and Toxicology*, vol. 88, no. 4, pp. 419–436, 2025.
- [10] F. Malhat, S. Shokr, S. Heikal, and N. E.-H. Zidan, "Temporal trends, multiple residue incidence, and chronic health risk of pesticides in egyptian onions: A four-year market surveillance," *Journal of Xenobiotics*, vol. 15, no. 6, p. 192, 2025.
- [11] S. Savary, L. Willocquet, S. J. Pethybridge, P. Esker, N. McRoberts, and A. Nelson, "The global burden of pathogens and pests on major food crops," *Nature Ecology & Evolution*, vol. 3, no. 3, pp. 430–439, 2019.
- [12] A. Upadhyay, N. S. Chandel, K. P. Singh, S. K. Chakraborty, B. M. Nandede, M. Kumar, A. Subeesh, K. Upendar, A. Salem, and A. El-beltagi, "Deep learning and computer vision in plant disease detection: a comprehensive review of techniques, models, and trends in precision agriculture," *Artificial Intelligence Review*, vol. 58, no. 3, p. 92, 2025.
- [13] T. Nyawose, R. C. Maswanganyi, and P. Khumalo, "A review on the detection of plant disease using machine learning and deep learning approaches," *Journal of Imaging*, vol. 11, no. 10, p. 326, 2025.
- [14] I. Pacal, I. Kunduracioglu, M. H. Alma, M. Deveci, S. Kadry, J. Nedoma, V. Slany, and R. Martinek, "A systematic review of deep learning techniques for plant diseases," *Artificial Intelligence Review*, vol. 57, no. 11, p. 304, 2024.
- [15] P. Sobiya, K. Jayareka, K. Maheshkumar, S. Naveena, and K. S. Rao, "Paddy disease classification using machine learning technique," *Materials Today: Proceedings*, vol. 64, pp. 883–887, 2022.
- [16] N. Senan, M. Aamir, R. Ibrahim, N. Taujuddin, and W. Muda, "An efficient convolutional neural network for paddy leaf disease and pest classification," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 7, pp. 116–122, 2020.
- [17] J. Chen, D. Zhang, A. Zeb, and Y. A. Nanehkaran, "Identification of rice plant diseases using lightweight attention networks," *Expert Systems with Applications*, vol. 169, p. 114514, 2021.
- [18] M. Aggarwal, V. Khullar, and N. Goyal, "Contemporary and futuristic intelligent technologies for rice leaf disease detection," in *2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*. IEEE, 2022, pp. 1–6.
- [19] A. Haridasan, J. Thomas, and E. D. Raj, "Deep learning system for paddy plant disease detection and classification," *Environmental monitoring and assessment*, vol. 195, no. 1, p. 120, 2023.
- [20] V. K. Shrivastava and M. K. Pradhan, "Rice plant disease classification using color features: a machine learning paradigm," *Journal of Plant Pathology*, vol. 103, no. 1, pp. 17–26, 2021.
- [21] Y. Borhani, J. Khoramdel, and E. Najafi, "A deep learning based approach for automated plant disease classification using vision transformer," *Scientific Reports*, vol. 12, no. 1, p. 11554, 2022.
- [22] S. P. Singh, K. Pritamdas, K. J. Devi, and S. D. Devi, "Custom convolutional neural network for detection and classification of rice plant diseases," *Procedia Computer Science*, vol. 218, pp. 2026–2040, 2023.
- [23] International Rice Research Institute, "Tungro," <https://www.knowledgebank.irri.org/training/fact-sheets/pest-management/diseases/item/tungro>, 2023, IRRI Rice Knowledge Bank. Accessed: July 7, 2026.
- [24] G. Latif, S. E. Abdelhamid, R. E. Mallouhy, J. Alghazo, and Z. A. Kazimi, "Deep learning utilization in agriculture: Detection of rice plant diseases using an improved cnn model," *Plants*, vol. 11, no. 17, p. 2230, 2022.

- [25] H. Andrianto, A. Faizal, F. Armandika *et al.*, "Smartphone application for deep learning-based rice plant disease detection," in *2020 international conference on information technology systems and innovation (ICITSI)*. IEEE, 2020, pp. 387–392.
- [26] A. B. Ayyappan, T. Gobinath, M. Kumar, and A. Sivaramakrishnan, "Rice plant disease detection using convolutional neural networks," *Discover Artificial Intelligence*, vol. 5, no. 1, p. 50, 2025.
- [27] C. G. Simhadri, H. K. Kondaveeti, V. K. Vatsavayi, A. Mitra, and P. Ananthachari, "Deep learning for rice leaf disease detection: A systematic literature review on emerging trends, methodologies and techniques," *Information Processing in Agriculture*, vol. 12, no. 2, pp. 151–168, 2025.
- [28] M. Melinda, R. Maulana, M. Irhamsyah, M. S. Nur, N. Basir, E. Elizar, M. Syafrudin *et al.*, "Lightweight rice leaf disease classification using mobilenetv2: A comprehensive performance evaluation," *Green Intelligent Systems and Applications*, vol. 6, no. 1, pp. 120–132, 2026.
- [29] A. Arya and P. K. Mishra, "Mobilenetv2-incep-m: a hybrid lightweight model for the classification of rice plant diseases," *Multimedia Tools and Applications*, vol. 83, no. 33, pp. 79 117–79 144, 2024.
- [30] S. Mehnaz and M. T. Islam, "Rice leaf disease detection: a comparative study between cnn, transformer and non-neural network architectures," *arXiv preprint arXiv:2501.06740*, 2025.
- [31] A. Setyawan, A. Septiarini, and A. Tejawati, "Comparison of resnet50, resnet101, and resnet152 architectures in image-based rice leaf disease classification," *Jurnal Teknologi Informatika dan Komputer*, vol. 12, no. 1, pp. 383–398, 2026.
- [32] G. W. Wicaksono and A. Andreawan, "Resnet101 model performance enhancement in classifying rice diseases with leaf images," *RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 2, pp. 345–352, 2023.
- [33] R. Sari, H. Asbudi, and F. Susilawati, "Utilizing resnet-50 for deep learning-based rice leaf disease detection," *Brilliance: Res. Artif. Intell.*, vol. 5, no. 2, pp. 1258–1268, 2025.
- [34] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.
- [35] M. T. Ahad, Y. Li, B. Song, and T. Bhuiyan, "Comparison of CNN-based deep learning architectures for rice diseases classification," *Artificial Intelligence in Agriculture*, vol. 9, pp. 22–35, 2023.
- [36] B. Paneru, B. Paneru, and K. B. Shah, "Analysis of convolutional neural network-based image classifications: a multi-featured application for rice leaf disease prediction and recommendations for farmers," *arXiv preprint arXiv:2410.01827*, 2024.
- [37] M. T. Roseno, S. Oktarina, Y. Nearti, H. Syaputra, and N. Jayanti, "Comparing cnn models for rice disease detection: Resnet50, vgg16, and mobilenetv3-small," *Journal of Information Systems and Informatics*, vol. 6, no. 3, pp. 2099–2109, 2024.
- [38] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 10012–10022.
- [39] Petchiammal, B. Kiruba, Murugan, and P. Arjunan, "Paddy doctor: A visual image dataset for automated paddy disease classification and benchmarking," in *Proceedings of the 6th Joint International Conference on Data Science & Management of Data (10th ACM IKDD CODS and 28th COMAD)*, 2023, pp. 203–207.
- [40] C. Subbarayudu and M. Kubendiran, "An automated hybrid deep learning framework for paddy leaf disease identification and classification," *Scientific Reports*, vol. 15, no. 1, p. 26873, 2025.
- [41] P. K. Sethy, N. K. Barpanda, A. K. Rath, and S. K. Behera, "Deep feature based rice leaf disease identification using support vector machine," *Computers and Electronics in Agriculture*, vol. 175, p. 105527, 2020.
- [42] H. B. Prajapati, J. P. Shah, and V. K. Dabhi, "Detection and classification of rice plant diseases," *Intelligent Decision Technologies*, vol. 11, no. 3, pp. 357–373, 2017.
- [43] C. R. Rahman, P. S. Arko, M. E. Ali, M. A. I. Khan, S. H. Apon, F. Nowrin, and A. Wasif, "Identification and recognition of rice diseases and pests using convolutional neural networks," *Biosystems Engineering*, vol. 194, pp. 112–120, 2020.
- [44] Y. A. Alnaggar, A. Sebaq, K. Amer, E. Nacem, and M. Elhelw, "Rice plant disease detection and diagnosis using deep convolutional neural networks and multispectral imaging," in *International Conference on Model and Data Engineering*. Springer, 2022, pp. 16–25.
- [45] B. Muhia, "Visualising multi-sensor predictions from a rice disease classifier," 2022.
- [46] Rukhsar and S. K. Upadhyay, "Rice leaves disease detection and classification using transfer learning technique," in *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*. IEEE, 2022, pp. 2151–2156.
- [47] S. Jain and R. Kumar, "An efficient deep learning model for detection of rice leaf diseases using paddy doctor dataset," in *2024 First International Conference on Software, Systems and Information Technology (SSITCON)*. IEEE, 2024, pp. 1–7.
- [48] J. Padhi, L. Korada, A. Dash, P. K. Sethy, S. K. Behera, and A. Nanthamornphong, "Paddy leaf disease classification using efficientnet b4 with compound scaling and swish activation: A deep learning approach," *IEEE Access*, vol. 12, pp. 126 426–126 437, 2024.
- [49] B. Bhattarai, "Comparative analysis of resnet50 and vision transformer on paddy disease classification," Ph.D. dissertation, 2025.
- [50] S. J. Sidiq, S. Abdullah, and D. Ramesh, "Leveraging deep learning ensembles for rice disease classification: a case study," in *Agricultural Insights from Space*. Amsterdam, Netherlands: Elsevier, 2026, pp. 379–402.
- [51] M. Fachrie, A. Musdholifah, and R. Pulungan, "Effectiveness of data resampling and ensemble learning in multiclass imbalance learning," *Artificial Intelligence Review*, vol. 58, no. 12, p. 368, 2025.
- [52] Z. Mahmood, L. Jamel, D. A. Salem, and I. Ashraf, "Improving learning from the complex multi-class imbalanced and overlapped data by mapping into higher dimension using svm++," *Scientific Reports*, vol. 15, no. 1, p. 31245, 2025.
- [53] W. Zeng, "Image data augmentation techniques based on deep learning: A survey," *Mathematical Biosciences and Engineering*, vol. 21, no. 6, pp. 6190–6224, 2024.
- [54] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of big data*, vol. 6, no. 1, p. 27, 2019.
- [55] J. Hancock, T. M. Khoshgoftaar, and J. M. Johnson, "Informative evaluation metrics for highly imbalanced big data classification," in *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2022, pp. 1419–1426.
- [56] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 97. Long Beach, California, USA: PMLR, 2019, pp. 6105–6114. [Online]. Available: <https://proceedings.mlr.press/v97/tan19a.html>
- [57] S. Nigam, R. Jain, V. K. Singh, S. Marwaha, A. Arora, and S. Jain, "Efficientnet architecture and attention mechanism-based wheat disease identification model," *Procedia Computer Science*, vol. 235, pp. 383–393, 2024.
- [58] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International Journal of Data Mining & Knowledge Management Process*, vol. 5, no. 2, pp. 1–11, 2015.
- [59] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.
- [60] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 336–359, 2020.
- [61] J. Peng, Y. Wang, P. Jiang, R. Zhang, and H. Chen, "Ricedra-net: Precise identification of rice leaf diseases with complex backgrounds using a res-attention mechanism," *Applied Sciences*, vol. 13, no. 8, p. 4928, 2023.
- [62] A. Petchiammal and D. Murugan, "Explainable ai-driven model for paddy disease detection and classification," in *2025 6th International Conference on Data Intelligence and Cognitive Informatics (ICDICI)*. Tirunelveli, India: IEEE, 2025, pp. 1081–1086.

- [63] O. Prabhu, "Interpretable CNN architectures for rice leaf disease detection: A Grad-CAM and LIME-based explainability approach," *International Journal of Computers*, vol. 10, pp. 226–234, 2025.
- [64] H. K. Kondaveeti and C. G. Simhadri, "Evaluation of deep learning models using explainable ai with qualitative and quantitative analysis for rice leaf disease detection," *Scientific Reports*, vol. 15, no. 1, p. 31850, 2025.