IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 15, No. 2, April 2026, pp. 1646~1663
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i2.pp1646-1663

Unified voting-based ensemble learning for rice leaf disease detection using improved pretrained models

Govindarajan Subburaman, Mary Vennila Selvadurai
Department of Computer Science, Presidency College, University of Madras, Chennai, India

Article Info

Article history:

Received Apr 10, 2025
Revised Jan 13, 2026
Accepted Feb 6, 2026

Keywords:

Data augmentation techniques
Ensemble deep learning
Image preprocessing techniques
Plant disease classification
Rice leaf diseases
Transfer learning models
Voting ensemble methods

ABSTRACT

As a staple food for a large portion of the global population, rice is particularly susceptible to leaf diseases that adversely affect its yield and overall quality. This study utilizes four pretrained convolutional neural network (CNN) models to construct a unified voting-based ensemble approach for rice leaf disease classification. The models include VGG16, DenseNet121, InceptionV3, and Xception. The dataset used in this study was collected from Kaggle and further enriched with images obtained from Google sources. It comprises a total of 4,000 images categorized into six classes: bacterial leaf blight, brown spot, leaf blast, leaf scald, narrow brown spot, and healthy leaves. It was split into training (327 images/class), validation (140 images/class), and testing (200 images/class). Images were normalized to [0,1] and augmented through rotation, flipping, shifting, shear, zoom, brightness, and channel adjustments to improve generalization. Individually, the fine-tuned models achieved accuracies of 91.3% (VGG16), 95.6% (DenseNet121), 92.1% (InceptionV3), and 89.8% (Xception). The ensemble leveraged majority voting (93.6%), weighted voting (96.5%), and soft voting (97%), yielding an absolute gain of 1.4% over the best individual model and 4.8% over the average of all models. To our knowledge, this is the first ensemble combining these four architectures with unified voting for identifying diseases in rice leaves, delivering a scalable and computationally efficient solution suitable in advance diagnosis and timely execution in agricultural settings with limited resources.

Corresponding Author:

Govindarajan Subburaman
Department of Computer Science, Presidency College, University of Madras
Chennai, India
Email: cscresearch2024@gmail.com

1. INTRODUCTION

Rice is one of the world’s most vital crops, sustaining a large portion of the global population and holding significant economic and cultural importance. At present, it constitutes the staple food for over 2.7 billion people, and this number is projected to increase substantially, reaching nearly 3.9 billion by 2025 [1]. Despite rice plant diseases causing an annual 37% loss in production due to insufficient disease identification and management knowledge, effective diagnostic and management applications remain scarce [2]. Among the various diseases that affect rice cultivation, brown spot, blast, and bacterial leaf blight are the most widespread and cause the greatest economic losses. These diseases severely hinder rice plant growth and productivity, often resulting in substantial economic and environmental losses. Early and accurate detection of these diseases within a short time frame is essential, as it can help minimize crop damage and safeguard farmers from substantial financial setbacks [3]. One of India's most popular staple crops is rice, and the country's agricultural industry accounts for about 19.9% of gross domestic product.

Rice crop yield and quality are often reduced by plant diseases, causing economic losses to farmers. Since disease identification based on visual experience is unreliable, an automated and accurate early diagnosis system is essential [4]. Crop health plays a vital role in supporting global food supply and sustainable farming practices. However, various factors can lead to the rapid spread of diseases in crops, causing significant social and economic challenges. These diseases not only hinder plant growth and development but also reduce crop yield and quality, making them a major contributor to productivity loss. Early detection of such illnesses and the timely use of appropriate pesticides are crucial to prevent soil pollution and mitigate its impact [5]. Timely disease diagnosis in crops is crucial for sustainable agricultural development, as it lowers input costs and prevents yield loss. In rice production, early detection of pests and diseases reduces dependency on chemical treatments and contributes to improved productivity. Traditional methods, such as manual observation, are neither feasible nor effective for large-scale farming. However, advancements in information technology have significantly contributed to improving crop productivity while optimizing fertilizer use. Convolutional neural networks (CNNs), a crucial area of artificial intelligence, provide a useful method for classifying plant diseases, and image processing methods have great potential to solve problems in the agricultural industry [6].

While deep learning has significantly improved visual plant disease recognition, the scarcity of some disease instances leads to data imbalance relative to healthy samples. Data collected under natural conditions typically exhibit limited samples of rare disease classes, leading to class imbalance issues in machine learning systems. This imbalance causes supervised learning models to overfit, as decision boundaries become biased toward the dominant classes [7], [8]. By broadening the training dataset's diversity, data augmentation approaches can be used to improve model generalization and overcome this difficulty. Using transformations like rotation (up to 40°), width and height shifts (up to 30%), shear transformation (0.2), zooming (up to 30%), brightness adjustment (between 0.7 and 1.3), channel shifting (up to 30.0), horizontal flipping, and nearest-neighbor filling for empty regions, these techniques create different versions of existing images. These techniques are crucial for balancing datasets, preventing overfitting, and enabling models to learn more robust and unbiased decision boundaries for improved disease classification.

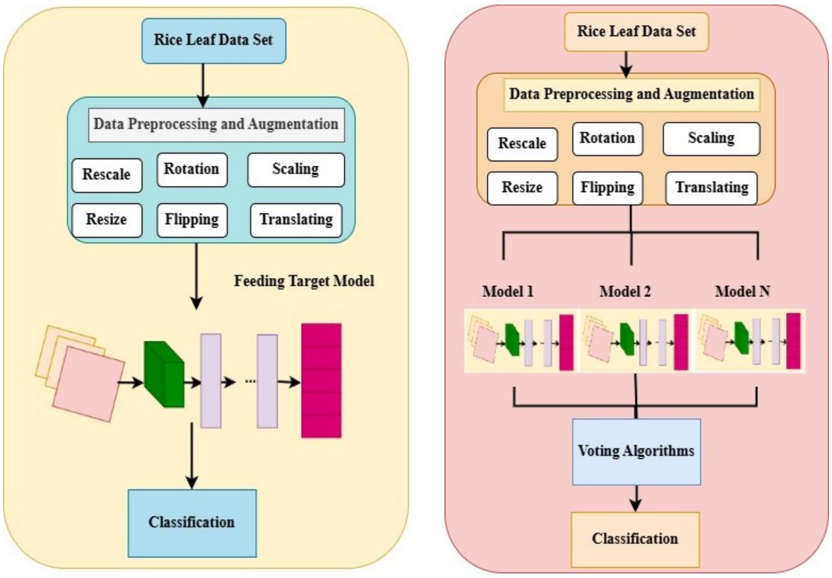
Deep learning methods have been investigated recently for rice leaf disease detection, with very encouraging results. Gogoi *et al.* [9] for instance, reported an accuracy of 93.99% on a dataset including 8,883 images using a three-stage CNN with transfer learning using EfficientNetB5. Leveraging a MobileNet backbone, Wang *et al.* [10] created an attention-based depthwise separable neural network optimized with Bayesian approaches, which achieved 94.65% accuracy on a public rice illness dataset with four disease classifications. During a different study, Krishnamoorthy *et al.* [4] used InceptionResNetV2 and transfer learning to classify paddy disease of the leaves with 95.67% accuracy. These works highlight potential of deep learning models; however, most rely on single pretrained models without ensemble strategies. To address this gap, our study integrates the multiple pretrained CNNs with unified voting to achieve robust rice leaf disease classification.

While current research primarily focuses on single machine learning models for plant disease detection [11], Figure 1 illustrates the comparison between traditional and ensemble classification methods used in this study. Figure 1(a) shows that traditional classification struggles with complex, multi-disease data, leading to lower accuracy. Figure 1(b) illustrates how ensemble classification enhances reliability and generalization by integrating multiple models. However, deep learning methods still face interpretability challenges in understanding disease patterns. For the purpose of enhancing the accurate detection of numerous rice illness of the leaves and highlight the primary causes of their occurrence, this study presents a novel approach that incorporates ensemble learning techniques. Ensemble learning enhances predictive performance by integrating multiple models, and in this study, a deep learning ensemble is constructed using VGG16, DenseNet121, InceptionV3, and Xception. To ensure robust and accurate results, unified voting techniques, majority voting, soft voting, and weighted voting, are utilized to consolidate predictions from individual models, harnessing their combined strengths for improved classification performance.

The key contributions of this research include:

- Compiled a balanced dataset of 4,000 rice leaf images from Kaggle and Google, covering six categories.
- Fine-tuned four pretrained CNN models (VGG16, DenseNet121, InceptionV3, and Xception) for disease classification.
- To enhance model generalization and address data imbalance, data augmentation was applied using transformations such as rotation (up to 40°), width and height shifts (up to 30%), shear (0.2), zoom (up to 30%), brightness adjustment (0.7–1.3), channel shifting (up to 30.0), horizontal flipping, and nearest-neighbor filling.
- Propose the first unified voting ensemble for rice leaf disease detection using these four models.
- Our ensemble (soft voting) achieved 97% accuracy, outperforming individual models.

- The proposed framework demonstrated superior performance, showcasing the effectiveness of ensemble learning and a unified voting scheme for accurate classification of rice diseases of the leaves.



(a) (b)

Figure 1. Comparison of classification approaches for (a) traditional classification and (b) ensemble classification

2. RELATED WORKS

This section presents a summary of earlier studies focused on the detection of rice leaf diseases. To precisely detect various rice leaf diseases, prior research has used a variety of machine learning and deep learning techniques. One approach combined multiple CNN architectures into an ensemble framework to improve classification performance and demonstrated the feasibility of real-time deployment through smartphone applications [12]. Another study introduced a large-scale dataset of rice leaf images and applied ensemble CNN models, showing that lightweight architectures can be optimized for use in mobile and resource-constrained environments [13]. Comparative analyses have also been carried out using standard CNN models. For instance, experiments with InceptionV3, VGG16, AlexNet, MobileNetV2, and ResNet18 on more than 7,000 images achieved high classification accuracy [14]. Similarly, several CNN models, including VGG16, ResNet50, and DenseNet121 were evaluated across multiple datasets containing three to five disease categories, with DenseNet121 found to be the most effective [15]. Hybrid frameworks have also gained attention. An ensemble of CNN and support vector machines (SVM) was developed to classify five major rice diseases, achieving competitive performance [16]. Transfer learning with pre-trained models has been widely explored as well. For example, XceptionNet, ResNet50, DenseNet, VGG19, and SqueezeNet were tested on a multi-class dataset of ten diseases, with XceptionNet showing the best results [17]. Another framework integrated VGG16 with light gradient boosting machine (LightGBM) for four-class rice disease classification and deployed the system via a cloud-based mobile interface [18]. Other research has applied advanced architectures and optimization methods. InceptionResNetV2 was employed for automatic disease classification using transfer learning, yielding high accuracy on a dataset of three disease types [4]. A more complex pipeline integrated preprocessing, segmentation with SegNet, and multi-feature extraction, followed by a deep recurrent neural network (Deep RNN) classifier optimized with a metaheuristic algorithm for improved disease recognition [19]. Table 1 provides a comparative overview of existing approaches, detailing the models implemented, the size of the datasets, the ensemble techniques applied, and the accuracy results reported in each study.

While individual and hybrid methods have shown potential, there remains a need for more robust and adaptable models for rice leaf disease detection. Many existing studies are limited to a few disease types and struggle with generalization across different environmental conditions. The recommended ensemble method enhances the precision and resilience of rice leaf disease detection by integrating the key features of VGG16, DenseNet121, InceptionV3, and Xception. The model is exposed to a variety of training samples through the use of various data augmentation approaches, such as rotation, flipping, and zooming, which enhances the model's generalization and ability to correctly identify varied illness patterns. Through a unified voting approach, including majority, soft, and weighted voting, the model enhances prediction accuracy by aggregating the outputs from multiple models, ensuring the final decision reflects the consensus of diverse characteristics. This method enhances not just the accuracy of the model but also its robustness and ability to generalize, enabling more reliable detection of rice leaf diseases under diverse environmental conditions.

Table 1. Summarizes a comparison of current methods used for detecting rice leaf diseases.

Author	Models/methods	Ensemble	Voting mechanism	Dataset size	Accuracy (%)
Deng *et al.* [12]	DenseNet121, SE-ResNet50, ResNeSt50	Yes	Yes	(33,026 images, 6 diseases)	91
Pai *et al.* [13]	MobileNetV2, GoogLeNet, EfficientNet, ResNet34, DenseNet121, VGG16	Yes	Yes	(18,563 images, 6 diseases)	93.4
Deb *et al.* [14]	InceptionV3, VGG16, AlexNet, MobileNetV2, ResNet18	No	No	(7,096 images, 5 diseases)	96.23
Islam *et al.* [15]	VGG16, ResNet50, DenseNet121	No	No	(Dataset1:386 images, 5 diseases; Dataset2: 120 images, 3 diseases; Dataset3: 278 images, 3 diseases)	91.67, 85.45, 89.85
Haridasan *et al.* [16]	CNN+SVM	Yes	No	(Plant dataset, 5 diseases)	91.45
Mandwariya and Jotwani [17]	XceptionNet, ResNet50, DenseNet121, VGG19, SequeezeNet and CNN	No	No	(4,193 images, 10 diseases)	93.13
Bhowmik *et al.* [18]	VGG16 (transfer learning)+LightGBM (ensemble)	Yes	No	(3,355 images, 4 diseases)	96.49
Krishnamoorthy *et al.* [4]	InceptionResNetV2	No	No	(5,200 images, 3 diseases)	95.67
Bhowmik [18]	SegNet,+Deep RNN	Yes	No	(120 images, 3 diseases	90.5

3. PROPOSED METHOD

The proposed approach employs an ensemble learning classifier for detecting rice leaf diseases. The process involves several stages, including data collection, image preprocessing such as resizing, application of data augmentation techniques, training, and evaluating deep learning models, and finally combining their predictions through a unified voting-based ensemble strategy. To enhance the overall performance of the model, the rice leaf disease dataset underwent preprocessing and data augmentation. The following subsections describe each step of the proposed methodology, which is illustrated in Figure 2.

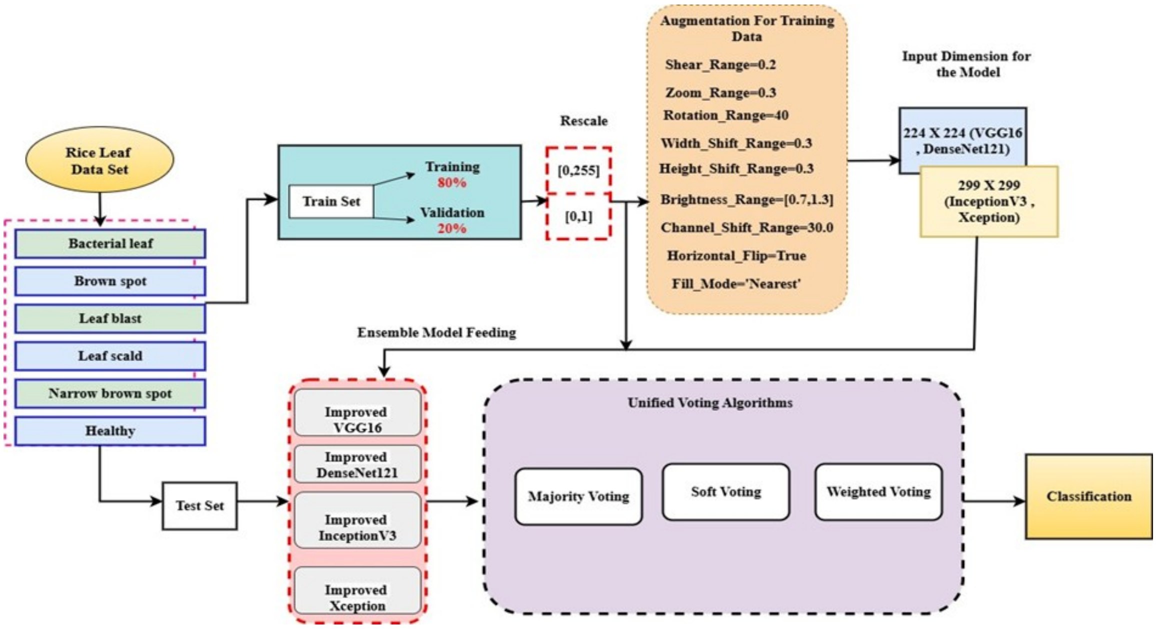


Figure 2. Workflow of proposed ensemble learning with unified voting

3.1. Dataset description

The rice leaf disease dataset was mainly sourced from Kaggle and supplemented with additional images from Google, comprising six classes: five disease types—narrow brown spot, leaf scald, leaf blast, brown spot, and bacterial leaf blight—and one healthy category, as illustrated in Figure 3. A total of 4,000 images were included and divided into training, validation, and testing sets using a 70:30 ratio. Training set contained 327 images per class, validation set 140 images per class, and testing set 200 images per class. Images were rescaled according to the input dimensions required by each pretrained architecture, using 224×224 pixels [20] for VGG16 and DenseNet121, and 299×299 pixels for InceptionV3 and Xception.

Figure 3. Examples of rice leaf disease images

3.2. Preprocessing and augmentation

This stage aims to prepare image data so that it is more consistent and varied to support the model training process.

- Normalization: pixel values were rescaled by a factor of 1/255, ensuring they fall within the [0, 1] range for stable training and faster convergence.
- Augmentation: to increase the variety of training samples and limit overfitting, several image transformation methods were employed. These included controlled rotations of up to 40°, horizontal and vertical translations reaching 30%, geometric distortion using a shear factor of 0.2, zoom operations up to 30%, intensity variation through brightness scaling between 0.7 and 1.3, color channel adjustments up to 30.0, and horizontal mirroring of images.
- Splitting: the available data were organized into separate learning, tuning, and evaluation groups following a predefined ratio, where each class contributed 327 images for model learning, 140 images for parameter validation, and 200 images for final assessment.

3.3. Base models

In this study, four pretrained deep vision architectures were adapted through fine-tuning to classify rice leaf conditions, and the key characteristics of these chosen networks are summarized in Table 2. Transfer learning: transfer learning was applied by starting each network with weights learned from ImageNet and then adapting the models to the rice leaf disease data through additional training. Architectural modifications: the architecture incorporates a feature summarization stage, followed by a nonlinear fully connected layer, a regularization component with a 0.4 drop rate, and a final classification layer that outputs probabilities for six classes.

Table 2. Characteristics of pretrained CNN architectures

Model	Parameters	Notable features	Input size
VGG16	~138 M	Deep but simple sequential architecture	224×224
DenseNet121	~8 M	Dense connectivity, parameter efficiency	224×224
InceptionV3	~24 M	Multi-scale feature extraction	299×299
Xception	~22 M	Depthwise separable convolutions	299×299

3.4. Ensemble learning strategies

To enhance classification accuracy and generalization in rice leaf disease detection, three ensemble methods were applied within a unified voting strategy, where each method contributes to the final decision through combined predictions. This approach improves robustness by leveraging the strengths of multiple models instead of relying on a single classifier. As a result, the overall performance in identifying rice leaf diseases becomes more stable and reliable.

3.4.1. Majority voting

Under this approach, each trained model provides its prediction, and the outcome is determined by the label that receives the greatest overall support [21] as in (1).

$$\hat{c} = \max_{j} S_j \quad Where\ S_j = \sum_{m=1}^{M} \delta(m, j) \tag{1}$$

Where $\hat{c}$ is represents selected output category [21], M is denotes total number of participating models, j is indicates a class label, and $\delta(m, j)$ is equals 1 when the m^{th} model predicts class j, and 0 otherwise.

3.4.2. Soft voting

In this approach, probability scores produced by each model are combined by computing their mean values, and the category with the strongest overall confidence is selected. By relying on confidence information rather than only discrete labels, the ensemble benefits from the reliability of all individual predictions [21], [22] as described in (2).

$$\hat{y} = \underset{k}{argmax} \frac{1}{N} \sum_{i=1}^{N} P_i\,(K) \tag{2}$$

Where $P_i(K)$ is probability of class k predicted by model i

3.4.3. Weighted voting

In this method, model weights were determined according to validation performance and overall accuracy. DenseNet121 and InceptionV3, which showed superior results among the selected models, were each given a weight of 0.30, while VGG16 and Xception were assigned lower weights of 0.20 due to their relatively weaker performance. The final class prediction was produced by calculating a weighted combination of the class probability outputs from all models, allowing higher-performing models to have greater impact while still benefiting from the diversity of the ensemble as in (3) [22].

$$\hat{y} = \underset{k}{argmax} \sum_{i=1}^{N} w_i P_i\,(K) \tag{3}$$

Where w_i is weight assigned to model i (e.g., based on accuracy), $P_i(K)$ is probability of class k predicted by model i.

The sequence of steps outlined is used to evaluate the explanations for unified voting in ensemble learning.

- Step 1: input: models {M1, M2, ..., Mn}, test dataset D, true labels Y (optional), voting type, weights W
- Step 2: initialize:
  P = soft predictions (probabilities), C = majority predictions (class labels).
- Step 3: collect predictions:
  For each model M_i:
  Predict probabilities $P_i = M_i.Predict(D)$.
  Compute majority predictions $C_i = argmax(P_i)$.
  Store P_i in P and C_i in C.
- Step 4: aggregate predictions:
  Soft voting: compute $P_{avg} = average(P)$, then $FinalPredictions = argmax(P_{avg})$.
  Weighted voting: normalize W, compute $P_{Weighted} = \sum W[i] . P[i]$, and $FinalPredictions = argmax(P_{weighted})$.
  Majority voting: for each sample, determine the most frequent class $ModeClass$ from C.
- Step 5: final predictions.
- Step 6: end for

3.5. Training configuration

The models were trained with the Adam optimization algorithm using a learning rate of 0.00005, categorical cross-entropy loss, a batch size of 32, and a total of 50 training epochs. To improve generalization, dropout layers and data augmentation were applied during training. The experimental setup utilized multiple hardware accelerators to ensure efficient processing, with CPUs handling data preprocessing, T4 GPUs supporting computationally intensive operations, and TPU v2-8 resources used for highly parallel workloads.

4. ENSEMBLE MODEL–VGG16, DENSENET121, INCEPTIONV3, AND XCEPTION

This research aimed to investigate the effectiveness of combining VGG16, DenseNet121, InceptionV3, and Xception into a single ensemble framework to enhance classification performance and model robustness by exploiting the complementary capabilities of each architecture.

4.1. VGG16

VGG16 is a well-known image recognition architecture developed by an academic research team, characterized by a simple layered structure that relies on small 3×3 convolution filters. It achieved significant success in the 2014 ImageNet large scale visual recognition challenge (2014 ILSVRC), demonstrating the power of deep networks for hierarchical feature extraction [23]. With 16 weight layers, VGG16 influenced the development of deeper networks and its modular design has inspired many subsequent models in computer vision. In this study, the modified VGG16 network processes color images resized to a fixed square resolution suitable for its input layer. The architecture contains thirteen convolution stages arranged in consecutive blocks, with a nonlinear activation applied after each stage. The network begins with an initial stage that applies two feature extraction layers using a small kernel, after which spatial resolution is reduced through pooling. In the next stage, the depth of the feature maps is expanded while maintaining the same kernel dimensions, again followed by a downsampling operation. The third stage further deepens the network by introducing three successive feature extraction layers before reducing spatial size. The final two stages follow a similar design pattern, each employing three feature extraction layers with a higher number of channels, and each stage concludes with a pooling operation to progressively condense the feature maps. Following the feature extraction stages, the model moves into a series of dense layers, where the first two have a high number of units with nonlinear activation, and the final layer reduces the representation to a smaller set of output units. Finally, a SoftMax output layer is applied to perform classification across the target categories, as illustrated in Figure 4.

4.2. DenseNet121

DenseNet121 uses dense connections where each layer connects to all preceding layers, improving information flow, gradient propagation, and training efficiency. With 121 layers, DenseNet121 efficiently captures fine-grained features while keeping the number of parameters manageable, making it well-suited for visual recognition applications [23]. In this research, the modified DenseNet121 model accepts color images resized to a fixed resolution of 224 by 224 pixels. The network begins with a convolutional operation using 64 filters with a relatively large kernel, combined with a downsampling step and a nonlinear activation, followed by a pooling layer that further reduces the spatial dimensions of the feature maps. The model continues with multiple densely connected blocks interleaved with transition layers. The first dense block consists of four sequential layers, each employing convolutions with small and medium-sized kernels, along with batch normalization and nonlinear activation functions. This is followed by a transition layer with a 1×1 convolution of 128 filters and 2×2 average pooling for downsampling. Dense blocks 2, 3, and 4 follow the same structure, each with four layers of 32 filters, separated by transition layers. After the final dense block, global average pooling reduces each feature map to 1×1, followed by a fully connected layer with 1,024 neurons and rectified linear unit (ReLU) activation, and finally a SoftMax layer for classification, as shown in Figure 5.

4.3. InceptionV3

InceptionV3 is an efficient CNN developed by Google, known for its innovative "inception module," which performs parallel convolutions with multiple kernel sizes for multi-scale feature extraction. It optimizes depth and computational cost, achieving high classification accuracy and efficiency, setting benchmarks in computer vision [24]. In this work, the refined InceptionV3 architecture processes color images resized to a higher square resolution suitable for its input layer. The design begins with an initial convolution stage and then applies a sequence of inception units that process information in parallel using filters of different spatial extents, combined with pooling operations to learn features at multiple scales. The first inception module includes convolutions with 64, 128, and 32 filters, followed by max-pooling, while the second module applies convolutions with 128, 192, and 96 filters before pooling. Subsequent inception modules expand in complexity, incorporating wider filter configurations for richer feature extraction. To reduce spatial dimensions, two reduction modules are employed: the first applies down-sampling using 3×3 convolutions and max-pooling after the initial inception modules, while the second further reduces dimensions at deeper levels of the network. After the inception stages, the learned feature representations are passed to a high-capacity dense stage for further abstraction, followed by a regularization step to limit overfitting. The model then produces probability-based outputs to distinguish among the different leaf conditions, as illustrated in Figure 6.

4.4. Xception

Xception, also known as extreme inception, is an advanced vision model developed to achieve strong results in a wide range of visual recognition applications [24]. It builds upon the InceptionV3 design by adopting a refined structure that separates spatial and channel-wise processing. This approach leads to lower computational demands and improved accuracy, as illustrated in Figure 7.

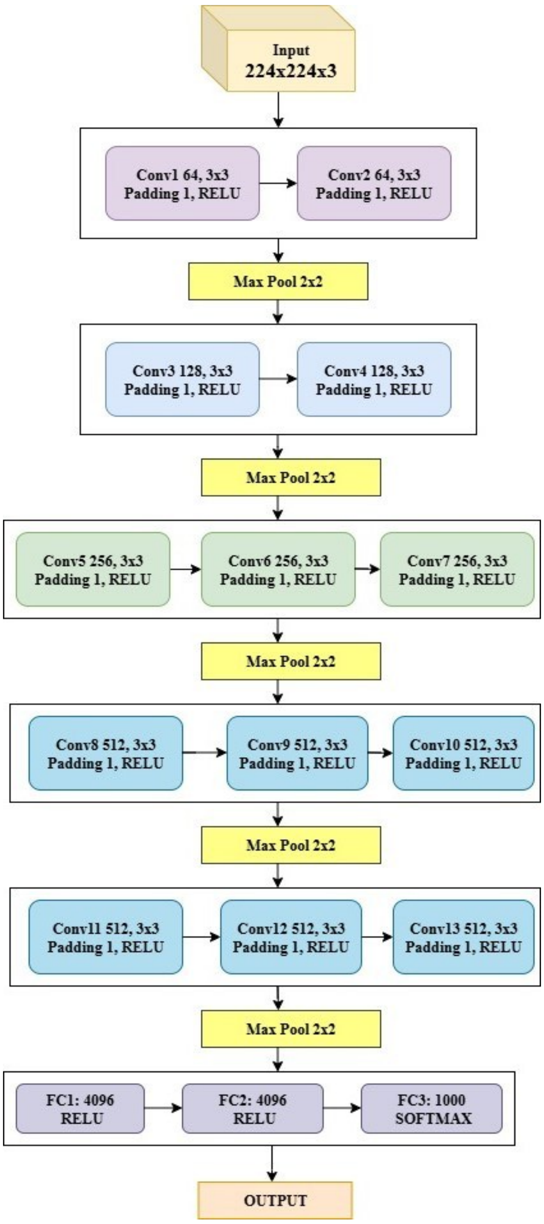


Figure 4. Architecture of the improved VGG16 model

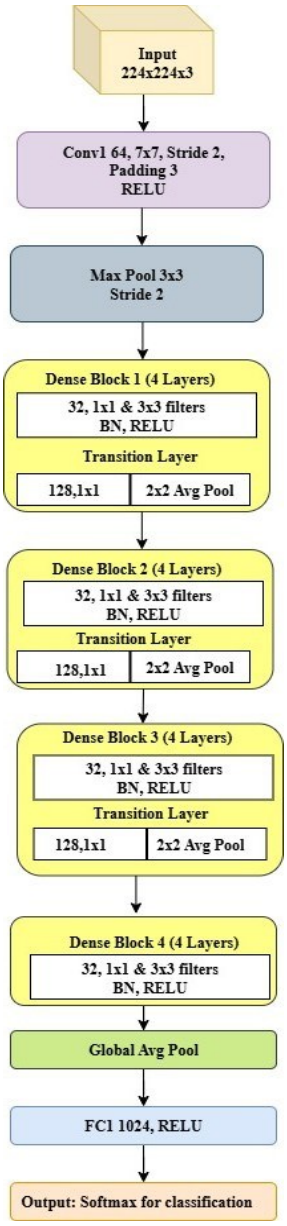


Figure 5. Structure of the enhanced DenseNet121 model

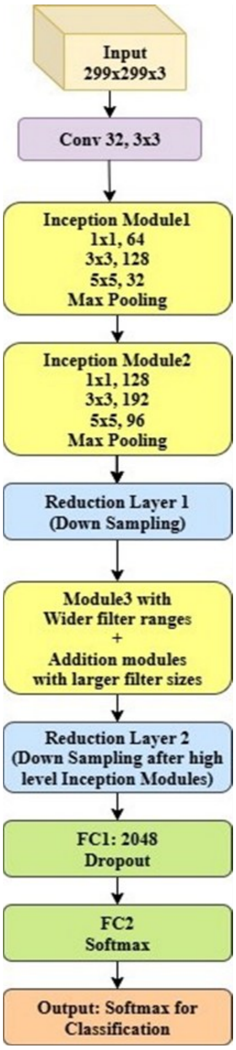


Figure 6. Structure of the enhanced InceptionV3 model

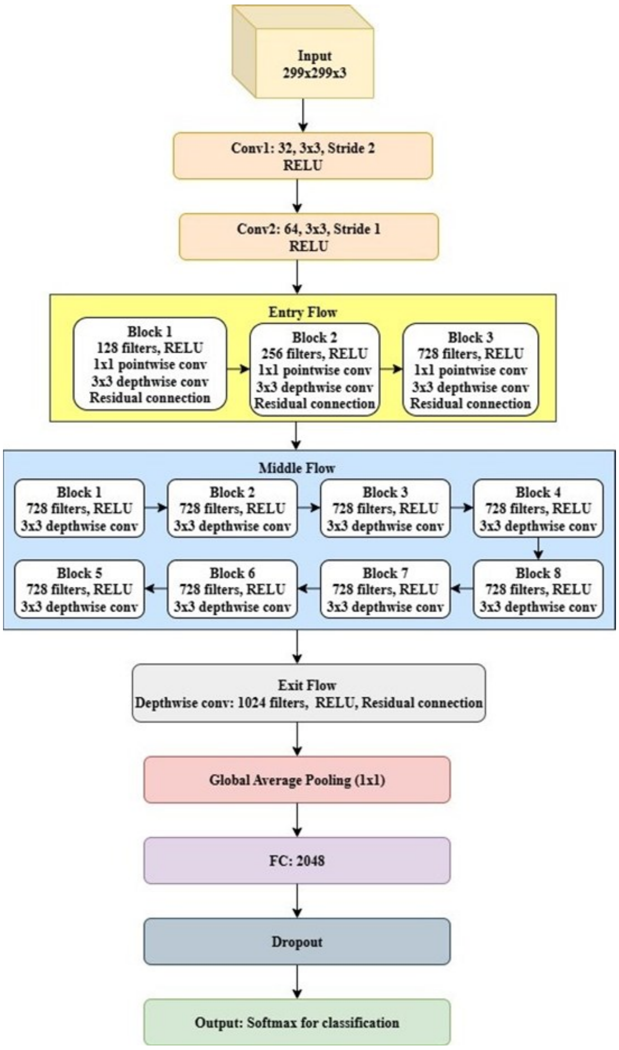


Figure 7. Structure of the enhanced Xception model

The improved Xception model accepts input images of size 299×299×3 (height, width, and RGB channels). The model begins with two initial feature extraction stages, where the first reduces spatial resolution using a smaller number of filters and nonlinear activation, followed by a second stage that increases channel depth while preserving finer spatial detail. The design is organized into three successive processing phases that handle early feature extraction, intermediate representation learning, and final feature refinement. The early feature extraction phase is made up of repeated computational blocks that separate channel-wise and spatial filtering, with shortcut connections included to support stable and efficient learning. This stage consists of three main blocks with 128, 256, and 728 filters, respectively. The immediate representation learning comprises eight identical blocks, each containing three depthwise separable convolutions with 728 filters and ReLU activations, without any down-sampling. In the final feature refinement, depthwise separable convolutions with 1,024 filters are applied, accompanied by residual connections, and the resulting feature maps are then reduced to 1×1 dimension using global average pooling. The final stages employ a dense representation layer for advanced feature abstraction, apply regularization to limit overfitting, and produce probability-based outputs to distinguish among the various rice leaf conditions.

4.5. Ensemble workflow with unified voting

The proposed ensemble workflow is illustrated in Figure 8. In this framework, preprocessed rice leaf images are first fed into four pretrained deep learning models VGG16, DenseNet121, InceptionV3, and Xception to extract features and generate initial class predictions. The predictions from each individual model are subsequently combined within a unified voting framework, which integrates three ensemble strategies: majority voting, soft voting, and weighted voting. In the majority approach, the output corresponds to the class supported by the highest number of models, while the soft approach relies on the average confidence scores from all models to determine the selected class. In weighted voting, models are assigned performance-based weights, and the final prediction is derived from the weighted combination of their outputs, allowing the ensemble to utilize the strengths of each model for more reliable and accurate classification.

In the proposed weighted voting strategy, the weights [0.3, 0.3, 0.2, 0.2] were assigned to the four pretrained models (VGG16, DenseNet121, InceptionV3, and Xception) based on their individual performance accuracies and complementary strengths. DenseNet121 (95.6%) and InceptionV3 (92.1%) demonstrated relatively higher and more stable classification performance, and therefore were assigned slightly higher weights (0.3 each). VGG16 (91.3%) and Xception (89.8%) achieved competitive but slightly lower accuracies, so they were assigned smaller weights (0.2 each).

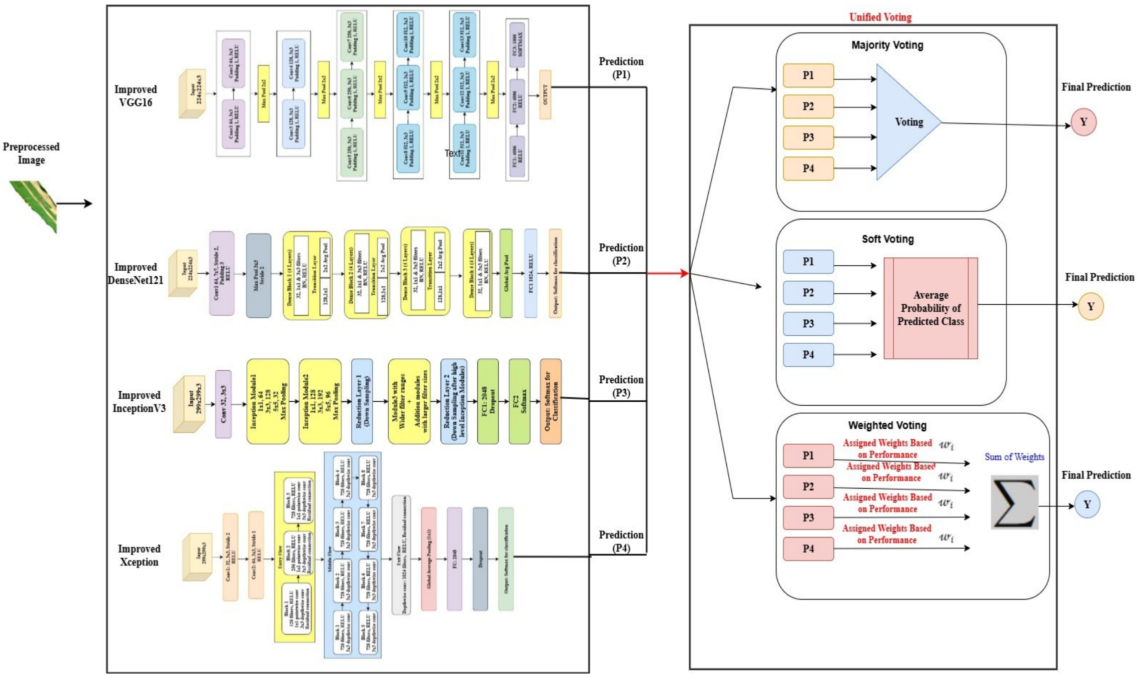
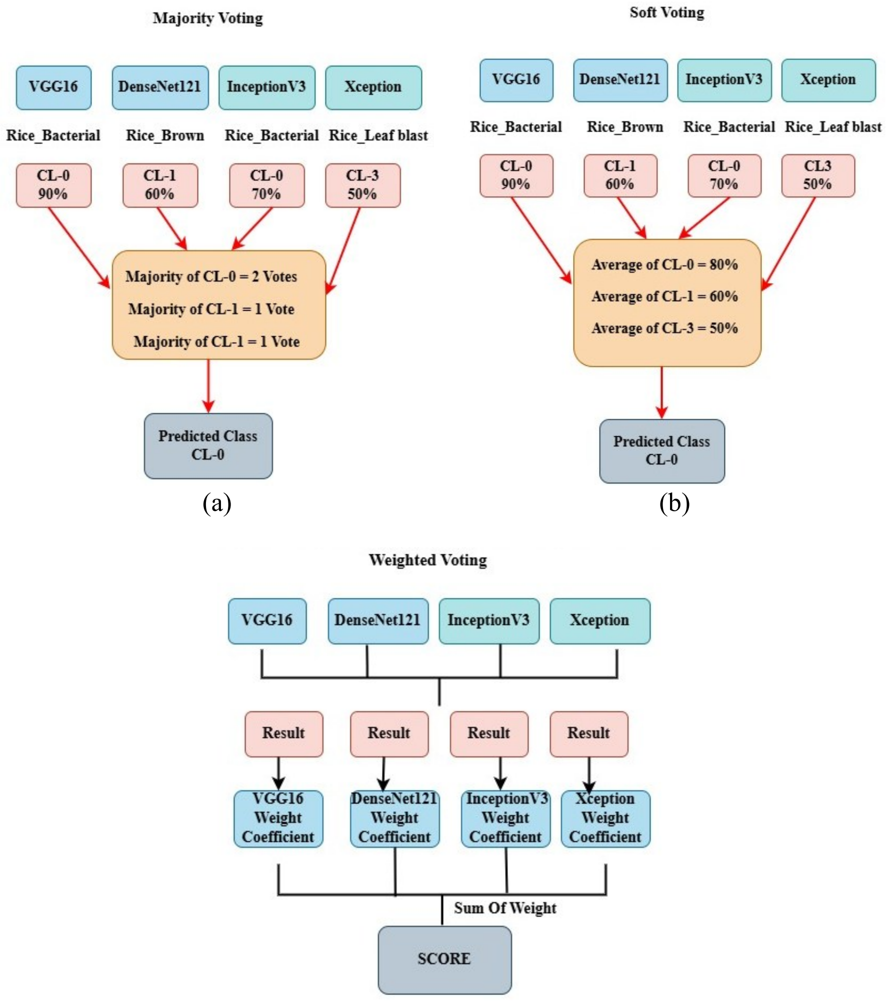


Figure 8. Ensemble workflow of pretrained models with unified voting for rice leaf disease classification

This weighting ensures that models with stronger predictive capabilities have a proportionally greater influence on the final decision, while still retaining the diversity benefits of including all four models. Ensemble learning improves generalization by aggregating predictions from multiple models, thereby reducing the risk that the final output is overly dependent on the biases or weaknesses of any single model. Majority and soft voting already provide robustness by balancing predictions, but weighted voting further enhances performance by emphasizing stronger models. This reduces the likelihood of overfitting, since the ensemble captures a broader representation of features learned across architectures and avoids relying excessively on a single model that may have overfit to the training set. As a result, the unified voting ensemble achieves more stable and accurate predictions across diverse test cases.

4.6. Unified voting

The goal of unified voting in ensemble learning is to combine outputs from several complementary architectures, allowing their individual strengths to collectively enhance overall classification accuracy. This approach improves accuracy by reducing the likelihood of errors made by individual models and mitigates biases inherent in each model, as it aggregates their predictions to counteract the weaknesses of one with the strengths of others. Unified voting enhances robustness to noise and outliers by reducing the influence of anomalies through the combined decisions of all models. Figure 9 depicts the three main approaches for combining model predictions. Figure 9(a) demonstrates the majority-based strategy, in which the outcome corresponds to the label most frequently supported by the participating models. Figure 9(b) illustrates the soft approach, where confidence scores produced by all models are combined, and the category with the strongest overall confidence is selected as the output. Figure 9(c) demonstrates weighted voting, in which higher-performing models are assigned greater weight, allowing them more influence in the final prediction. By combining these diverse models, unified voting ensures that the final prediction is more reliable and accurate than any single model's output, ultimately making it a superior approach for tasks like rice leaf disease classification, where complex data patterns require multiple models for effective analysis.



(a) (b)

(c)

Figure 9. The unified voting of (a) majority voting, (b) soft voting, and (c) weighted voting

5. RESULTS AND ANALYSIS OF EXPERIMENTS

5.1. Development environment setup

The experimental setup and training configuration for all base models are summarized in Table 3. The experiment was conducted using Python 3 in a Google Colab notebook, utilizing the TensorFlow and Keras frameworks to implement and evaluate the proposed rice leaf disease detection models. Libraries including NumPy, SciPy, and Matplotlib were utilized for performing numerical calculations and creating visualizations. Preprocessing and data augmentation were performed using the ImageDataGenerator module, ensuring better generalization of the models.

The networks were enhanced by adding specialized pooling, normalization, regularization, and dense feature-mapping components to boost predictive performance. The learning procedure was improved through an adaptive optimization scheme combined with dynamic learning-rate adjustment and early termination mechanisms to achieve stable and efficient convergence. Model evaluation was performed using various metrics, including confusion matrices, classification reports, and performance scores (accuracy, precision, recall, and F1-score), supported by visualization techniques from Seaborn. To aggregate predictions and further improve accuracy, ensemble methods were implemented using unified voting strategies, while statistical insights, such as mode of predictions, were utilized to enhance decision-making. Hardware accelerators, including CPUs for preprocessing, T4 GPUs for intensive computations, and TPU v2-8 for highly parallel tasks, were employed to achieve efficient execution throughout the experiment.

Table 3. Experimental setup and training configuration

Model	Parameters	Epochs	Batch size	Training time (50 epochs)
VGG16	~138 M	50	32	~35 minutes
DenseNet121	~8 M	50	32	~34 minutes
InceptionV3	~24 M	50	32	~29 minutes
Xception	~22 M	50	32	~29 minutes

5.2. Evaluation of model effectiveness

The performance of the proposed models and methods was assessed using standard evaluation metrics. These including overall correctness, positive prediction reliability, sensitivity, and the harmonic mean of precision and recall. The experiments were conducted using the rice leaf image collection introduced in subsection 3.1.

As illustrated in Figure 10, data augmentation aims to expand the variety of training samples by generating modified versions of the existing images. The dataset includes six categories with 327 samples per category, and the augmentation procedure generates six new versions for every original image. As a result, the total number of samples grows to 1,962 augmented images spanning all categories. Introducing the network to a wider variety of image transformations helps it generalize better and perform more reliably on unseen data. Parameter tuning was performed for each deep learning model to enhance its performance. This step required adjusting key model settings, such as the depth of hidden layers, the learning rate, and regularization factors, with the chosen configurations provided in Table 4.

Figure 11 shows the training and validation accuracy and loss for four pretrained models: VGG16 (Figure 11(a)), DenseNet121 (Figure 11(b)), InceptionV3 (Figure 11(c)), and Xception (Figure 11(d)). DenseNet121 and Xception demonstrate better convergence and stability, while VGG16 and InceptionV3 exhibit moderate fluctuations during training. Additionally, Figure 12 presents the confusion matrices for individual models: VGG16 (Figure 12(a)), DenseNet121 (Figure 12(b)), InceptionV3 (Figure 12(c)), and Xception (Figure 12(d)), reflecting their classification accuracies across all disease categories. Furthermore, Figure 13 depicts the confusion matrices for ensemble voting strategies, including majority voting (Figure 13(a)), soft voting (Figure 13(b)), and weighted voting (Figure 13(c)), illustrating how various ensemble methods affect the overall effectiveness of models. The training results of VGG16, DenseNet121, InceptionV3, and Xception models for rice leaf disease classification showcase remarkable progress in accuracy and loss metrics, reflecting the effectiveness of fine-tuned learning rates and strategic training regimens. As depicted in Figure 11, VGG16 initially achieved 39.12% on the training data and 35.59% on the validation set, gradually rising to 96.51% and 84.41% by the 32nd epoch. Despite its strong training performance, the slight gap between metrics suggests potential overfitting, warranting further regularization.

Similarly, DenseNet121 showed steady improvement, beginning with training and validation accuracies of 36.88% and 38.98% and peaking at 95.50% and 89.15% by epoch 15. Learning rate reductions further refined its performance, achieving 90.85% validation accuracy by epoch 30, though late fluctuations hint at possible overfitting. InceptionV3 demonstrated rapid convergence, reaching validation accuracies above 80% within 15 epochs and stabilizing at 94.49% training accuracy and 89.15% validation accuracy, while the steadily decreasing loss indicated better learning behavior. Finally, Xception initially achieved 35.59% on the training set and 45.42% on the validation set, later improving to 95.93% and 89.15% by the 31st epoch. Adaptive learning rate adjustments effectively mitigated overfitting while optimizing generalization. Across all models, validation loss trends and adaptive fine-tuning highlight their ability to identify complex disease patterns, affirming their potential for practical deployment in real-world agricultural diagnostics. Each model’s effectiveness in identifying rice leaf diseases was assessed in terms of accuracy, precision, recall, and F1-score. VGG16 achieved an accuracy of 91.3%, with precision, recall, and F1-scores of 91.5%, 91.2%, and 91.0%, respectively.

DenseNet121 demonstrated the highest performance with an accuracy of 95.6%, along with precision, recall, and F1-scores of 95.8%, 95.6%, and 95.6%. InceptionV3 achieved an accuracy of 92.1%, with precision, recall, and F1-scores of 92.1%, 92.0%, and 91.7%, respectively. Xception showed slightly weaker results compared to other models, recording an overall correctness of 89.8%, along with reliability, sensitivity, and balanced-score values of 90.3%, 89.6%, and 89.5%, respectively. For soft voting, the predictions from each model (VGG16, DenseNet121, InceptionV3, and Xception) are averaged based on their class probabilities. The decision is produced by selecting the category associated with the strongest average confidence across all participating models. Weighted voting enhances this by applying predefined weights to the predictions from each model. In the code, the weights [25] are specified as [0.3, 0.3, 0.2, 0.2] for VGG16, DenseNet121, InceptionV3, and Xception, respectively. Table 5 summarizes the performance of individual pretrained models and ensemble learning using unified voting methods. DenseNet121 achieved the highest performance among the individual models across all metrics, while soft voting produced the best overall performance among the ensemble methods.

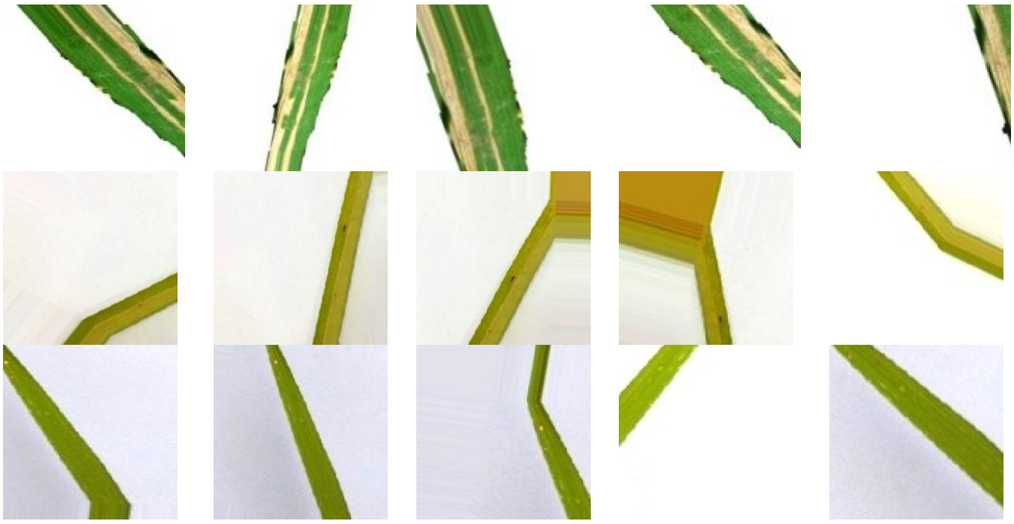
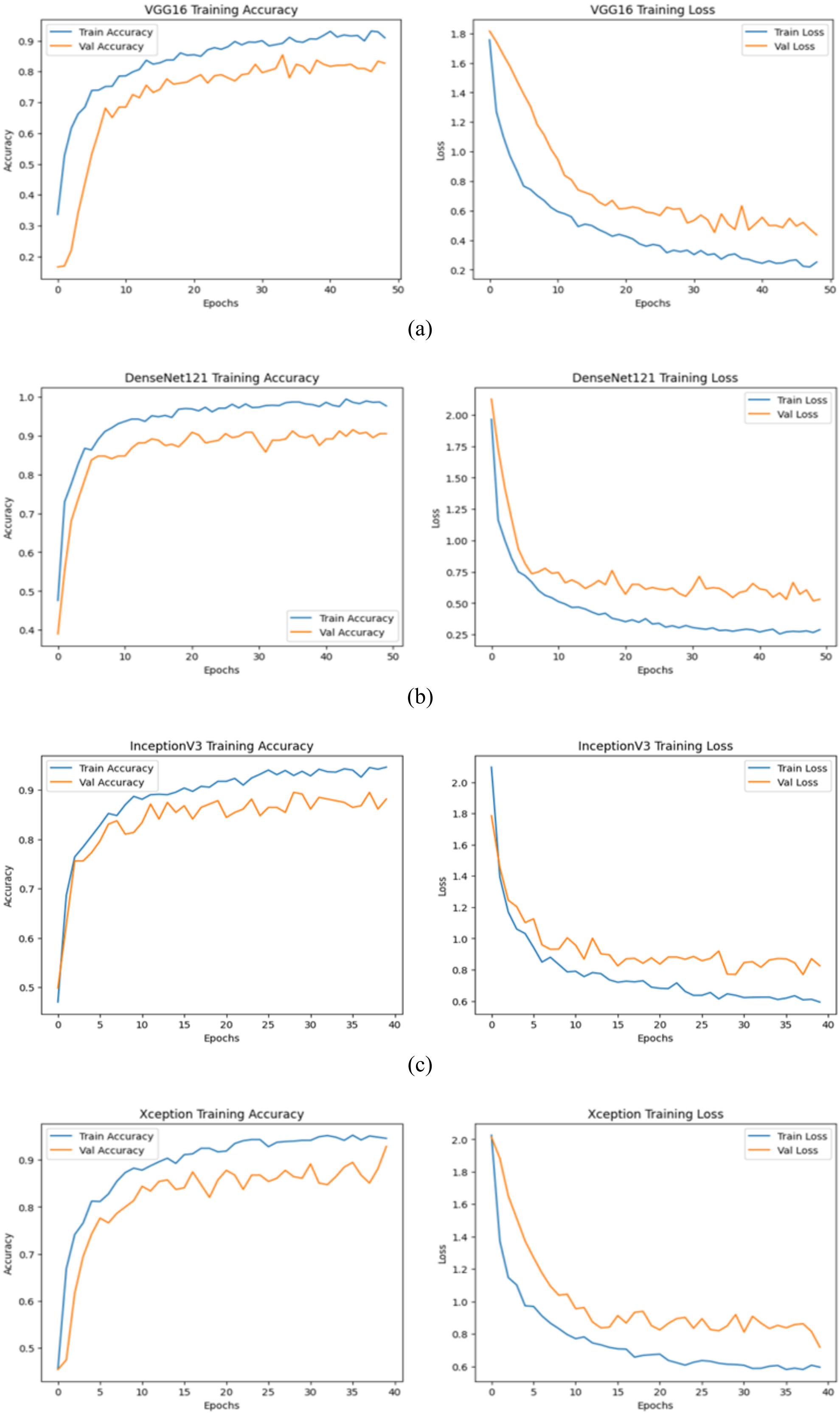


Figure 10. Visual examples of augmented rice leaf disease images

Table 4. Key training settings and their corresponding configurations

Key parameter	Values
Image_size	32
Batch_size	15
Fine_tune_from_layer (VGG16)	100
Fine_tune_from_layer (DenseNet121)	249
Fine_tune_from_layer (InceptionV3)	100
Fine_tune_from_layer (Xception)	0.00005
Learning_rate (optimizer)	40
Early_stopping_patience	6
lr_reduction_patience	50
Epochs	0.4
Dropout_rate	256
Dense layer units	0.001
Dense kernel regularizer	[0.3, 0.3, 0.2, 0.2]
Voting weights (weighted voting)	32

Figure 14 presents a comparison of ensemble and unified voting strategies, demonstrating how each method integrates predictions to enhance classification effectiveness. These weights adjust the influence of each model's output based on their individual performance, giving higher importance to models that perform better. Ensemble (majority) voting combines the class predictions from all models and selects the class that receives the most votes as the final output. Among the individual pretrained models, DenseNet121 achieved the highest accuracy (95.6%). This superior performance can be attributed to its dense connectivity pattern where each layer receives information from earlier layers, allowing features to flow efficiently and be reused throughout the network. Such an architecture supports smoother gradient transmission, encourages repeated use of learned representations, and helps the model identify subtle disease patterns even when the available training samples are limited, making DenseNet121 well suited for analyzing rice leaf conditions. Here, the scipy.stats mode operation determines which class label appears most often among the model outputs, thereby realizing the majority voting rule. By combining these three methods, the ensemble leverages the diversity of predictions from multiple models, improving generalization and reducing the risk of overfitting, thereby increasing classification accuracy and reliability.



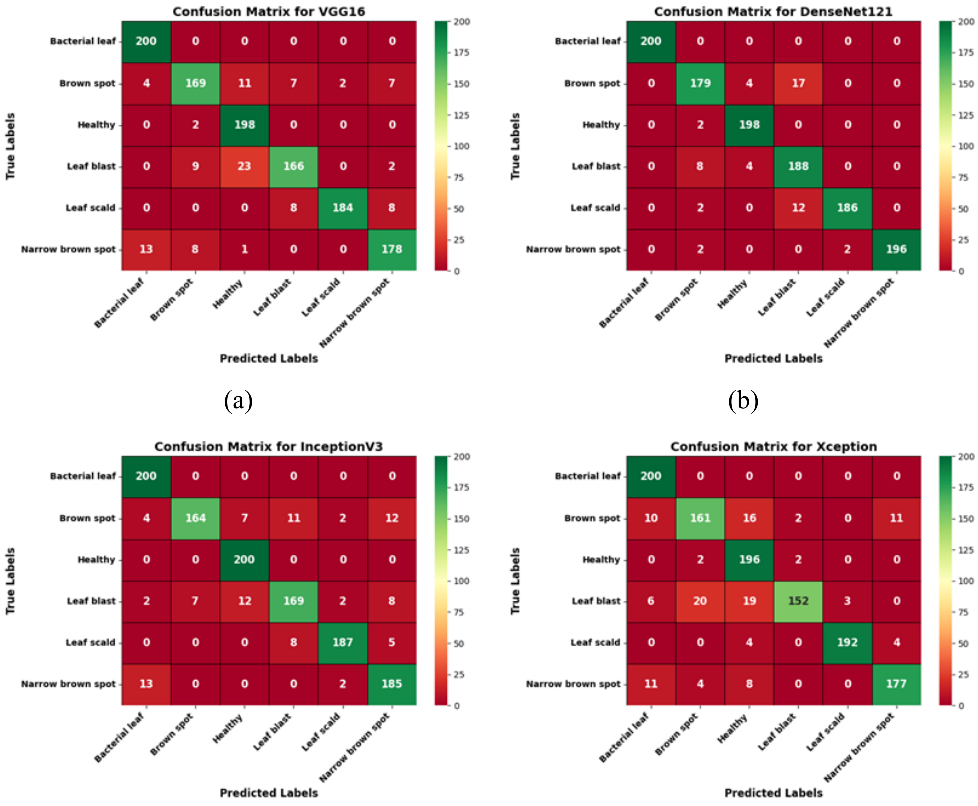
(a)

(b)

(c)

(d)

Figure 11. Comparison of training and validation accuracy and loss for (a) VGG16, (b) DenseNet121, (c) InceptionV3, and (d) Xception models



Confusion Matrix for VGG16

	Bacterial leaf	Brown spot	Healthy	Leaf blast	Leaf scald	Narrow brown spot
Bacterial leaf	200	0	0	0	0	0
Brown spot	4	169	11	7	2	7
Healthy	0	2	198	0	0	0
Leaf blast	0	9	23	166	0	2
Leaf scald	0	0	0	8	184	8
Narrow brown spot	13	8	1	0	0	178

(a)

Confusion Matrix for DenseNet121

	Bacterial leaf	Brown spot	Healthy	Leaf blast	Leaf scald	Narrow brown spot
Bacterial leaf	200	0	0	0	0	0
Brown spot	0	179	4	17	0	0
Healthy	0	2	198	0	0	0
Leaf blast	0	8	4	188	0	0
Leaf scald	0	2	0	12	186	0
Narrow brown spot	0	2	0	0	2	196

(b)

Confusion Matrix for InceptionV3

	Bacterial leaf	Brown spot	Healthy	Leaf blast	Leaf scald	Narrow brown spot
Bacterial leaf	200	0	0	0	0	0
Brown spot	4	164	7	11	2	12
Healthy	0	0	200	0	0	0
Leaf blast	2	7	12	169	2	8
Leaf scald	0	0	0	8	187	5
Narrow brown spot	13	0	0	0	2	185

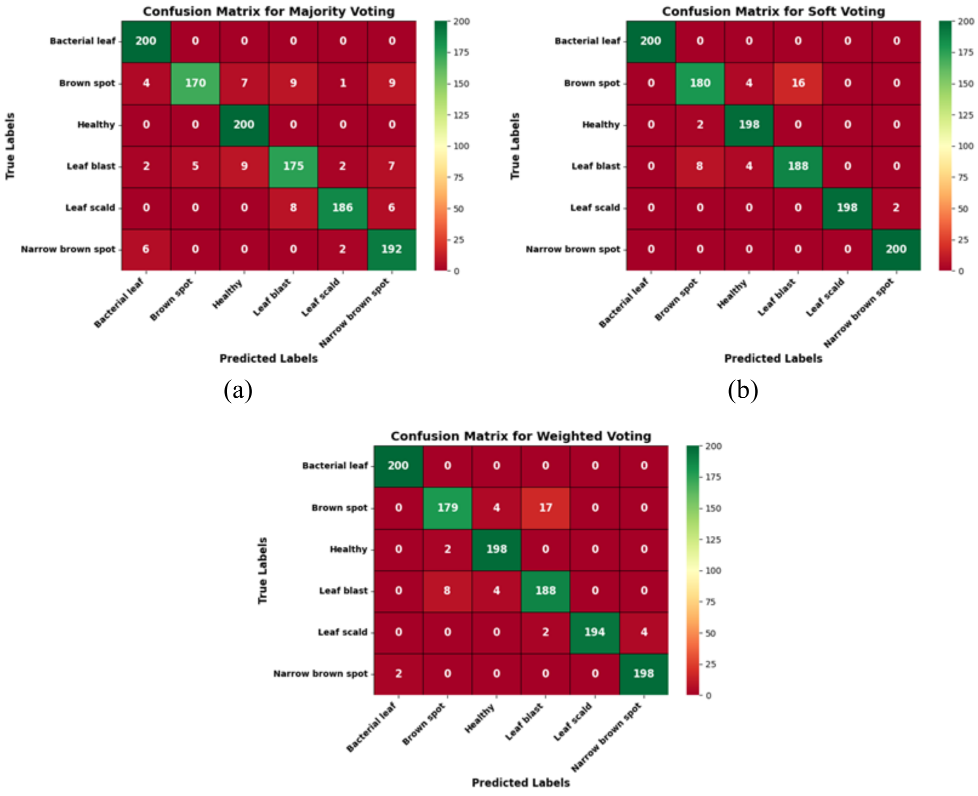
(c)

Confusion Matrix for Xception

	Bacterial leaf	Brown spot	Healthy	Leaf blast	Leaf scald	Narrow brown spot
Bacterial leaf	200	0	0	0	0	0
Brown spot	10	161	16	2	0	11
Healthy	0	2	196	2	0	0
Leaf blast	6	20	19	152	3	0
Leaf scald	0	0	4	0	192	4
Narrow brown spot	11	4	8	0	0	177

(d)

Figure 12. Confusion matrix of (a) VGG16, (b) DenseNet121, (C) InceptionV3, and (d) Xception



Confusion Matrix for Majority Voting

	Bacterial leaf	Brown spot	Healthy	Leaf blast	Leaf scald	Narrow brown spot
Bacterial leaf	200	0	0	0	0	0
Brown spot	4	170	7	9	1	9
Healthy	0	0	200	0	0	0
Leaf blast	2	5	9	175	2	7
Leaf scald	0	0	0	8	186	6
Narrow brown spot	6	0	0	0	2	192

(a)

Confusion Matrix for Soft Voting

	Bacterial leaf	Brown spot	Healthy	Leaf blast	Leaf scald	Narrow brown spot
Bacterial leaf	200	0	0	0	0	0
Brown spot	0	180	4	16	0	0
Healthy	0	2	198	0	0	0
Leaf blast	0	8	4	188	0	0
Leaf scald	0	0	0	0	198	2
Narrow brown spot	0	0	0	0	0	200

(b)

Confusion Matrix for Weighted Voting

	Bacterial leaf	Brown spot	Healthy	Leaf blast	Leaf scald	Narrow brown spot
Bacterial leaf	200	0	0	0	0	0
Brown spot	0	179	4	17	0	0
Healthy	0	2	198	0	0	0
Leaf blast	0	8	4	188	0	0
Leaf scald	0	0	0	2	194	4
Narrow brown spot	2	0	0	0	0	198

(c)

Figure 13. Confusion matrix of (a) majority voting, (b) soft voting, and (c) weighted voting

Table 5. Performance comparison between pretrained models and unified voting methods

Model	Accuracy	Precision	Recall	F1-score
VGG16	0.913	0.915	0.912	0.910
DenseNet121	0.956	0.958	0.956	0.956
InceptionV3	0.921	0.921	0.920	0.917
Xception	0.898	0.903	0.896	0.895
Majority voting	0.936	0.937	0.935	0.937
Soft voting	0.970	0.970	0.970	0.970
Weighted voting	0.965	0.965	0.965	0.965

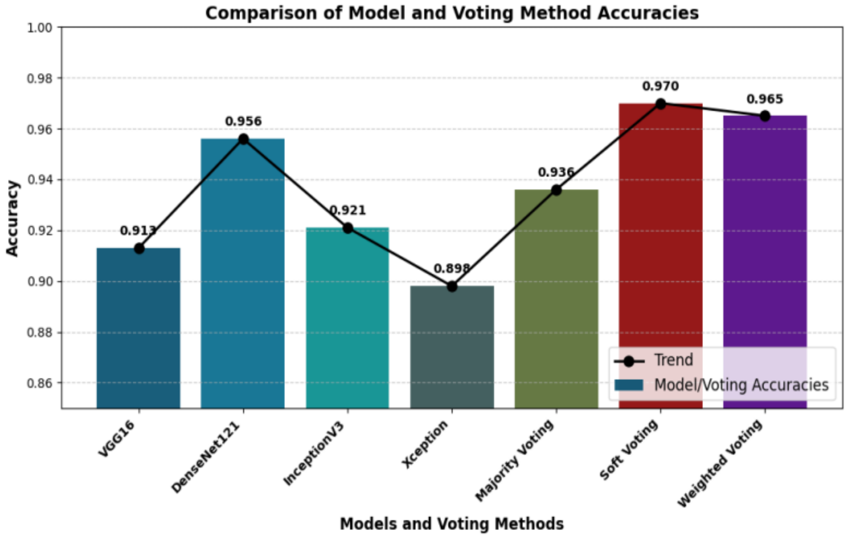


Figure 14. Comparison of model and voting method accuracies

5.3. Limitations

While the ensemble approach achieved high accuracy, some limitations remain. Despite applying extensive augmentation, pretrained models may still overfit due to the relatively small dataset size, with only 327 training images per class. The dataset collected from online platforms such as Kaggle and Google may not fully represent practical field variations, potentially limiting how well the system performs when applied outside controlled settings. Moreover, using an ensemble of multiple large models leads to higher computational overhead, which can restrict scalability in real-time and edge-based deployments. Subsequent studies should focus on gathering broader and more varied data sources, as well as the development of lightweight and efficient ensemble approaches to enhance robustness while maintaining practical applicability.

6. CONCLUSION

The work demonstrates how advanced neural network models can successfully distinguish among six rice leaf conditions, covering multiple disease types along with healthy samples. By employing four well-known pretrained vision architectures such as VGG16, DenseNet121, InceptionV3, and Xception and applying data augmentation techniques, the models achieved strong individual performances. DenseNet121 led the individual model accuracy at 95.56%, followed by InceptionV3 (92.1%), VGG16 (91.30%), and Xception (89.8%). The application of ensemble learning strategies, particularly majority voting, soft voting, and weighted voting, greatly enhanced the classification accuracy. Soft voting achieved the highest accuracy of 97%, while weighted voting also significantly improved the results, achieving an accuracy of 96.5%. These results highlight the importance of unified voting techniques in boosting model performance and ensuring more reliable disease detection compared to individual models. Data augmentation techniques helped improve model generalization, addressing data imbalance and preventing overfitting. This work highlights how integrating several pretrained architectures through ensemble voting can strengthen rice disease identification systems and support the progress of intelligent agricultural practices. Subsequent investigations may consider more sophisticated ensemble approaches, including stacking or boosting, to achieve additional improvements in performance. Incorporating interpretability techniques such as gradient-weighted class activation mapping can improve model transparency by visually indicating the areas of rice leaf images that play a key role in decision-making. Moreover, reducing model size and optimizing performance can allow the system to run on edge hardware, supporting immediate identification of diseases in the field. Overall, this framework aids sustainable farming by enabling timely disease control, promoting healthier crops, and strengthening food supply stability.

FUNDING INFORMATION

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

AUTHOR CONTRIBUTIONS STATEMENT

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

Name of Author	**C**	**M**	**So**	**Va**	**Fo**	**I**	**R**	**D**	**O**	**E**	**Vi**	**Su**	**P**	**Fu**
Govindarajan Subburaman	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Mary Vennila Selvadurai		✓				✓		✓	✓	✓	✓	✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no conflicts of interest regarding the publication of this paper.

DATA AVAILABILITY

The datasets used and analyzed during the current study are available from the corresponding author, [GS], upon reasonable request.

REFERENCES

[1] M. E. Haque, A. Rahman, I. Junaeid, S. U. Hoque, and M. Paul, “Rice leaf disease classification and detection using YOLOv5,” 2022, arXiv:2209.01579.

[2] G. K. V. L. Udayananda, C. Shyalika, and P. P. N. V. Kumara, “Rice plant disease diagnosing using machine learning techniques: a comprehensive review,” *SN Applied Sciences*, vol. 4, no. 11, 2022, doi: 10.1007/s42452-022-05194-7.

[3] M. A. Islam, N. R. Shuvo, M. Shamsojjaman, S. Hasan, S. Hossain, and T. Khatun, “An automated convolutional neural network based approach for paddy leaf disease detection,” *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 1, pp. 280–288, 2021, doi: 10.14569/IJACSA.2021.0120134.

[4] N. Krishnamoorthy, L. V. N. Prasad, C. S. P. Kumar, B. Subedi, H. B. Abraha, and V. E. Sathishkumar, “Rice leaf diseases prediction using deep neural networks with transfer learning,” *Environmental Research*, vol. 198, 2021, doi: 10.1016/j.envres.2021.111275.

[5] P. Tejaswini, P. Singh, M. Ramchandani, Y. K. Rathore, and R. R. Janghel, “Rice leaf disease classification using CNN,” in *IOP Conference Series: Earth and Environmental Science*, 2022, vol. 1032, no. 1, doi: 10.1088/1755-1315/1032/1/012017.

[6] X. Zhang, Y. Qiao, F. Meng, C. Fan, and M. Zhang, “Identification of maize leaf diseases using improved deep convolutional neural networks,” *IEEE Access*, vol. 6, pp. 30370–30377, 2018, doi: 10.1109/ACCESS.2018.2844405.

[7] L. M. R. Longadge, S. S. Dongre, “Class imbalance problem in data mining: review,” *International Journal of Computer Science and Network*, vol. 2, no. 1, 2013.

[8] S. Barua, M. M. Islam, X. Yao, and K. Murase, “2011 reviewers list,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 1, pp. 184–189, 2011, doi: 10.1109/tkde.2012.2.

[9] M. Gogoi, V. Kumar, S. A. Begum, N. Sharma, and S. Kant, “Classification and detection of rice diseases using a 3-stage CNN architecture with transfer learning approach,” *Agriculture*, vol. 13, no. 8, 2023, doi: 10.3390/agriculture13081505.

[10] Y. Wang, H. Wang, and Z. Peng, “Rice diseases detection and classification using attention based neural network and bayesian optimization,” *Expert Systems with Applications*, vol. 178, 2021, doi: 10.1016/j.eswa.2021.114770.

[11] M. S. H. Shovon, S. J. Mozumder, O. K. Pal, M. F. Mridha, N. Asai, and J. Shin, “PlantDet: a robust multi-model ensemble method based on deep learning for plant disease detection,” *IEEE Access*, vol. 11, pp. 34846–34859, 2023, doi: 10.1109/ACCESS.2023.3264835.

[12] R. Deng *et al.*, "Automatic diagnosis of rice diseases using deep learning," *Frontiers in Plant Science*, vol. 12, 2021, doi: 10.3389/fpls.2021.701038.
[13] P. Pai *et al.*, "Deep learning-based automatic diagnosis of rice leaf diseases using ensemble CNN models," *Scientific Reports*, vol. 15, no. 1, 2025, doi: 10.1038/s41598-025-13079-z.
[14] M. Deb, K. G. Dhal, R. Mondal, and J. Gálvez, "Paddy disease classification study: a deep convolutional neural network approach," *Optical Memory and Neural Networks (Information Optics)*, vol. 30, no. 4, pp. 338–357, 2021, doi: 10.3103/S1060992X2104007X.
[15] A. Islam, R. Islam, S. M. R. Haque, S. M. M. Islam, and M. A. I. Khan, "Rice leaf disease recognition using local threshold based segmentation and deep CNN," *International Journal of Intelligent Systems and Applications*, vol. 13, no. 5, pp. 35–45, 2021, doi: 10.5815/ijisa.2021.05.04.
[16] 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, 2023, doi: 10.1007/s10661-022-10656-x.
[17] D. Mandwariya and D. V. Jotwani, "A deep learning approach for automated rice disease detection and classification," *Journal of Advanced Zoology*, vol. 45, no. 3, pp. 316–335, 2024, doi: 10.53555/jaz.v45i3.4313.
[18] A. Bhowmik, M. Sannigrahi, D. Chowdhury, and D. Das, "RiceCloud: a cloud integrated ensemble learning based rice leaf diseases prediction system," in *INDICON 2022 - 2022 IEEE 19th India Council International Conference*, 2022, pp. 1–6, doi: 10.1109/INDICON56171.2022.10039790.
[19] T. Daniya and S. Vigneshwari, "Deep neural network for disease detection in rice plant using the texture and deep features," *Computer Journal*, vol. 65, no. 7, pp. 1812–1825, 2022, doi: 10.1093/comjnl/bxab022.
[20] N. P. Nutulapati, G. A. Macriga, D. Pulugu, and R. T. Murugan, "An ensemble learning approach for early detection and classification of plant diseases," *ICTACT Journal on Image and Video Processing*, vol. 14, no. 4, pp. 3245–3250, 2024, doi: 10.21917/ijivp.2024.0462.
[21] O. A. D. Ammar *et al.*, "Plant leaf disease detection using ensemble learning and explainable AI," *IEEE Access*, vol. 12, pp. 156038–156049, Oct. 2024, doi: 10.1109/ACCESS.2024.3484574.
[22] O. Octavian, A. Badruzzaman, M. Y. Ridho, and B. D. Trisedya, "Enhancing weighted averaging for CNN model ensemble in plant diseases image classification," *Jurnal RESTI*, vol. 8, no. 2, pp. 272–279, 2024, doi: 10.29207/resti.v8i2.5669.
[23] J. Desai and A. Ganatra, "Boosting the accuracy of deep learning models for detecting and classifying multiple diseases in plant leaves using a voting classifier," *14th International Conference on Advances in Computing, Control, and Telecommunication Technologies, ACT 2023*, vol. 2023-June, no. 1, pp. 90–96, 2023.
[24] A. M. P. Aishwarya and P. Reddy, "Ensemble of CNN models for classification of groundnut plant leaf disease detection," *Smart Agricultural Technology*, vol. 6, 2023, doi: 10.1016/j.atech.2023.100362.
[25] X. He *et al.*, "Multi-modal late fusion rice seed variety classification based on an improved voting method," *Agriculture*, vol. 13, no. 3, 2023, doi: 10.3390/agriculture13030597.

BIOGRAPHIES OF AUTHORS

Govindarajan Subburaman completed B.Sc. in Computer Science (2009) from Bharathidasan University, Tiruchirappalli, India and M.Sc. in Computer Science (2012) from Bharathidasan University, Tiruchirappalli, India. He completed his M.Phil. (Computer Science) in the year 2014 from Bharathidasan University, Tiruchirappalli, India and currently pursing Ph.D. in Computer Science from Presidency College, University of Madras, Chennai. His area of interest includes data mining, cloud computing, machine learning, and deep learning. He can be contacted at email: cscresearch2024@gmail.com.

Dr. Mary Vennila Selvadurai completed B.Sc. in Physics (1986) and M.Sc. in Computer Science (1990) from Bharathidasan University, Tiruchirappalli, India. She obtained her M.Phil. (Computer Science) in the year 2002 and Ph.D. in Computer Science (2009) from Mother Teresa Women's University, Kodaikanal. She is currently working as an associate professor and head of the PG and Research Department of Computer Science, Presidency College, University of Madras, Chennai, India. Her areas of interest include networks, grid computing, cloud security, and data mining. She can be contacted at email: maryvennila13@gmail.com.