

AI-Driven Crop Disease Detection and Sustainable Management

Oluwatooni Adebayo, Odile Nzambazamariya, and Gabriel Nishimwe

Carnegie Mellon University - Africa

Kigali, Rwanda

{jadebayo, onzambaz, gnishimw}@andrew.cmu.edu

Abstract—This research project addresses the challenge of automated paddy disease classification using advanced deep learning techniques. Building upon the foundation of the "Paddy Doctor" dataset, we implement and evaluate multiple neural network architectures using PyTorch, including ResNet34, ResNet50, EfficientNet-B0, MobileNet, VGG16, and ensemble models. Our goal is to surpass the existing 97.50% F1 benchmark achieved by earlier CNN and transfer learning approaches. We re-implemented baseline models under a consistent experimental setup, verified their reported accuracies, and extended the investigation by applying optimization techniques such as two-stage image resizing, fit one cycle learning rate policies, advanced regularization, and mixed-precision training. Among the models tested, ResNet50 achieved the highest performance, reaching a test accuracy of 98.71% and demonstrating superior class-wise robustness, while EfficientNet-B0 offered an excellent trade-off between efficiency and accuracy. In contrast, ensemble models showed mixed results, highlighting the complexities of model fusion strategies. Detailed evaluation using accuracy, precision, recall, F1-score, and confusion matrices revealed that deeper architectures provide better feature discrimination for visually similar disease classes.

I. MOTIVATION

Rice cultivation is vital in Africa, but bacterial and fungal diseases threaten crop yields [1], [2]. Manual disease identification is challenging due to vast farmlands and a shortage of specialists [3]. Automated detection systems can help improve productivity [4], but their development is hindered by the lack of region-specific datasets [5]. While models trained on Asian datasets like "Paddy Doctor" [6] are useful, upon availability of African-specific data, it will be fed into the same trained model, reducing retraining time, and making disease detection faster and more effective.

II. OBJECTIVE

Our primary objective is to develop a more accurate and efficient paddy disease classification system by leveraging PyTorch and implementing advanced neural network architectures. We aim to successfully implement and surpass the current benchmark of 97.50% F1 score achieved by ResNet34 by exploring and optimizing newer models such as EfficientNet, Vision Transformers, and custom architectures tailored for this specific task. Through this endeavor, we seek to contribute to advancing automated Paddy crop disease detection technology, ultimately supporting more effective and timely crop protection measures in paddy cultivation.

III. LITERATURE REVIEW

In recent years, the study of the detection and classification of paddy disease has gained significant attention due to its impact on global crop production, particularly rice. Various approaches have been explored to enhance the accuracy and efficiency of disease identification, with deep learning emerging as a promising solution. A notable study by Rahman et al. (2020) proposed a deep learning-based approach using Convolutional Neural Networks (CNNs) to detect diseases and pests in rice plants. Their model classified five types of diseases, three types of pests, and a category for healthy plants and other conditions. The data set consisted of 1,426 images captured using four different camera types, and the system achieved a mean validation accuracy of 94.33% [7]. Although our study shares similarities with Rahman et al.'s approach, we aim to extend their work by leveraging advanced deep learning techniques and testing on a larger dataset. By increasing the size of the dataset and exploring more sophisticated model architectures, we anticipate achieving improved accuracy and robustness in the classification of paddy diseases.

IV. DATASET

This study utilizes the Paddy Doctor Dataset [6], a comprehensive, large-scale collection of annotated images designed to support automated classification of paddy leaf diseases. The full dataset contains **16,225 labeled RGB images** spanning **13 categories—12 distinct paddy diseases** and a class for *healthy* leaves. Images were acquired in real-world paddy field conditions using high-resolution smartphones and verified with expert input from an agronomist to ensure annotation accuracy.

To prepare the dataset for deep learning tasks, multiple curated and augmented subsets were made available:

- **Original and Resized Variants:** The original dataset includes high-resolution (1080x1440) images. It was subsequently resized to medium (480x640) and small (256x256) versions for training efficiency.
- **Balanced and Split Sets:** Variants with balanced class distributions were generated using random sampling (e.g., 400 images across all classes), and standard training/testing splits (80%/20%) were provided.
- **Augmented Datasets:** To address class imbalance and improve generalization, image augmentation was employed:

- **Augmentation Techniques:** Rotation ($\pm 5^\circ$), shear, zoom (20%), and horizontal flipping.
- **Augmented Sizes:**
 - * **26,000 images** (2,000/class)
 - * **65,000 images** (5,000/class)
 - * **81,125 images**, a 5 $\times$ expansion of the original dataset.
- **Train/Test Splits:** Each version includes an 80/20 stratified split.
- **Metadata File:** A structured CSV file accompanies the datasets with fields such as image_id, label, variety, and age, supporting downstream analysis and model interpretability.

Among the available variants, the augmented and split dataset with 26,000 images with 80% of train and 20% of test data provided the best performance in training and was used as the primary dataset in our final model evaluations.

label	
brown_spot	1601
downy_mildew	1601
white_stem_borer	1601
blast	1600
bacterial_leaf_streak	1600
leaf_roller	1600
tungro	1600
black_stem_borer	1600
hispa	1600
yellow_stem_borer	1600
bacterial_leaf_blight	1599
bacterial_panicle_blight	1599
normal	1599
Name: count, dtype: int64	

Fig. 1: Class distribution in the augmented Paddy Doctor dataset.

V. MODEL DESCRIPTION

This project leverages a range of deep learning architectures for automated paddy disease classification, building upon the work established in the "Paddy Doctor" paper [6]. Several models were evaluated to investigate their effectiveness in handling real-world paddy leaf image data.

We implemented a Simple CNN with stacked convolutional and pooling layers, followed by dense classification heads. In addition, we explored advanced architectures such as ResNet50 and EfficientNet-B0, both of which are pretrained on ImageNet and known for their balance between accuracy and computational efficiency. These models were fine-tuned on the augmented Paddy Doctor dataset to adapt to domain-specific disease features.

To enhance performance further, an **ensemble model** was constructed by combining the predictions of ResNet50 and EfficientNet-B0, leveraging their complementary feature extraction strengths. This fusion aimed to reduce model bias and variance, ultimately improving generalization. The summary of all implemented models is found in table 1.

VI. EVALUATION METRIC

The evaluation metrics accuracy, precision, recall, F1 score and confusion matrix were used for this project. These metrics are explained in the following:

A. Accuracy

Accuracy measures the proportion of correct predictions from total predictions, providing an overall performance indicator of the model.

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

B. Precision

Precision indicates the proportion of true positive predictions out of the total predicted positives, reflecting the model's ability to avoid false positives.

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

C. Recall

Recall, also known as sensitivity, measures the proportion of true positive predictions out of the actual positives, indicating the model's ability to capture all relevant instances.

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

D. F1-Score

F1-score is the harmonic mean of precision and recall, providing a single metric that balances the trade-off between precision and recall.

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

The performance of the paddy disease classification model will be evaluated using the above metrics as summarized in Table II below.

TABLE II: Metrics adopted

Metric	Description
Accuracy	The overall percentage of correctly classified images.
Precision	The ability of the classifier to avoid labeling an image as a disease when it is not.
Recall	The ability of the classifier to identify all images with a specific disease.
F1-score	The harmonic mean of precision and recall, providing a balanced measure of the model's performance.

VII. LOSS FUNCTION AND OPTIMIZATION

The primary loss function used for training the paddy disease classification models is the **Categorical Cross-Entropy Loss**, defined as:

$$\mathcal{L}_{CCE} = - \sum_{i=1}^C y_i \cdot \log(\hat{y}_i) \quad (5)$$

where:

- C is the number of classes (13 in this case),
- y_i is the true label (one-hot encoded),

TABLE I: Model Architecture Descriptions, Output Shapes, and Parameter Counts

Model	Hidden layers	Parameters/hyper-parameters	Total Parameters
CNN	4 blocks: {Conv2D, BN, ReLU}, followed by MaxPooling and Dropout - Flatten $\rightarrow$ Dense(1024, ReLU, BN, Dropout) $\rightarrow$ Dense(13, Softmax)	- Input: (256, 256, 3), Output: (13,) - Conv2D: 32$\rightarrow$64$\rightarrow$128 filters; Max-Pool reduces to (21, 21, 128) - Flatten $\rightarrow$ 56,448 units, Dense $\rightarrow$ 1024 $\rightarrow$ 13	58.17M
EfficientNet-B0 (NS)	- Pretrained EfficientNet-B0 (Noisy Student) - Depthwise Conv2Ds, GAP, Dense + Softmax	- Input: (224, 224, 3), Output: (1, 1, 1280) - Model: <code>tf_efficientnet_b0_ns</code> from <code>timm</code>, pretrained on ImageNet	5.29M
ResNet34	- Pretrained backbone, Conv-BN-ReLU-MaxPool 4 residual blocks, GAP, Dense + Softmax	- Trained 12 epochs, monitored F1 Score - FastAI one-cycle policy used	21.8M
ResNet50	- Pretrained ResNet50 - Initial Conv-BN-ReLU-MaxPool 4 residual blocks (3, 4, 6, 3 layers) - Global average pooling, Dense + Softmax	- Input size: (224, 224, 3) - Output shape: (1, 1, 2048) before Dense - Architecture: <code>resnet50</code> from <code>timm</code> - Pretrained on ImageNet	25.56M
Ensemble (EffNet-B0 + ResNet50)	- Two parallel streams: EfficientNet-B0 and ResNet50 - Outputs fused (e.g., averaged probabilities or logits)	- Output combined from 13 logits of each model - No additional layers in ensemble head	30.85M
MobileNet	- Pretrained MobileNet backbone (feature extractor) - Flatten layer - Dense layer (1024 units) with ReLU - Dropout (0.2) - Final Dense layer with Softmax	- Input shape: (256, 256, 3) - Fine-tuning of last 24 layers - Transfer learning from ImageNet weights - Learning rate: 0.0001 - Optimizer: Adam - Loss: <code>categorical_crossentropy</code>	70.35M
VGG16	- Pretrained VGG16 backbone 13 Conv layers with 3$\times$3 filters 5 MaxPooling layers - Fine-tuning of last 2 layers - Custom classification head	- Input shape: (256, 256, 3) - Fine-tuning of last 2 layers - Transfer learning from ImageNet weights - Learning rate: 0.0001 - Optimizer: Adam	48.28M

- $\hat{y}_i$ is the predicted probability for class i .

This loss function penalizes confident but incorrect predictions, encouraging the model to output probability distributions close to the true labels.

Optimization is performed using the Adam optimizer, an adaptive learning rate method combining momentum and RMSProp. The update rule for parameters θ at iteration t is given by:

$$\theta_t = \theta_{t-1} - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}} \quad (6)$$

where:

- η is the learning rate,
- $\hat{m}_t$ and $\hat{v}_t$ are bias-corrected estimates of the first and second moments of the gradients,
- ϵ is a small constant to prevent division by zero.

This approach ensures stable and efficient convergence during training, especially in deep neural networks like ResNet50 and EfficientNet.

VIII. BASELINE SELECTION AND EVALUATION

The baseline models include a simple CNN, ResNet-34, and MobileNet. To enhance the Paddy Doctor model, we implemented EfficientNet, CNN, ResNet34, ResNet50, MobileNet, VGG16, and Ensemble Learning. Tan and Le [8] demonstrated that EfficientNet achieves high accuracy with lower computational cost. Dosovitskiy et al. [9] showed that ViTs outperform CNNs by capturing long-range dependencies in image data. Additionally, an ensemble of ResNet-34 and EfficientNet has been shown to improve classification performance [10]. These

enhancements strengthen comparisons with CNNs and transfer learning models such as VGG16, MobileNet, Xception, and ResNet-34, ensuring a comprehensive evaluation.

TABLE III: Comparison of Models and Their Accuracy

Model	Our Accuracy (%)	Reported Accuracy (%)
ResNet-34	94.62	94.62
cnn	82.21	83.46
Mobilenet	87.12	92.42
VGG16	94.29	93.19

IX. IMPLEMENTED EXTENSIONS AND EXPERIMENTS

This section presents the enhancements beyond the baseline models, including specific extensions applied to ResNet34, ResNet50, EfficientNet, Ensemble, MobileNet, and VGG16 models.

A. EfficientNet-B0 Extension

EfficientNet-B0 was introduced to improve performance while maintaining a lightweight architecture. As proposed by Tan and Le [8], it scales depth, width, and resolution uniformly for better accuracy-efficiency tradeoff.

In this experiment, we employed the EfficientNet-B0 architecture to classify rice leaf diseases across 13 distinct categories. The model was implemented using the FastAI library and the timm pretrained model `efficientnet_b0_ns` as the backbone. Training was conducted on an augmented dataset to improve robustness and generalization.

The data pipeline was built using FastAI's DataBlock API with the following configuration:

- Images resized to 480×480 using the squish method.
- Batch transformations resized images to 224×224.
- Data augmentation included:
 - Random affine transformations: `max_rotate=20`, `max_zoom=1.2`, `max_warp=0.2`
 - Probability-based effects: `p_affine=0.75`, `p_lighting=0.75`
 - Random erasing: `p=0.5`, `max_count=1`
- MixUp was used as a regularization strategy during training.
- The dataset was split using an 80/20 train-validation split with a fixed seed (42) for reproducibility.
- Model: `tf_efficientnet_b0_ns` (via timm)
- Learning rate: 0.005, Weight decay: 0.01
- Mixed-precision training (fp16) enabled
- Trained for 40 epochs

B. CNN Extension

The model was a custom Convolutional Neural Network (CNN), designed to evaluate the effectiveness of a lightweight yet expressive architecture on the rice leaf disease classification task. The CNN was developed using TensorFlow and

Keras, trained from scratch on the augmented Paddy Doctor dataset.

The training and validation pipelines were constructed using the `ImageDataGenerator` class with all images resized to **256×256×3**.

The following are the training Configuration applied:

- **Epochs:** 100
- **Batch size:** 32
- **Learning rate:** 0.001
- **Loss function:** Categorical Cross-Entropy
- **Optimizer:** Adam
- **Image normalization:** Performed via rescaling in data generators

The test set was loaded using a separate `ImageDataGenerator`, applying only rescaling and no augmentation, to ensure a fair evaluation of the model's generalization performance on unseen data.

C. Ensemble Learning (EfficientNet + ResNet50)

To further enhance classification robustness, an ensemble model was constructed by combining predictions from two independently trained models: EfficientNet-B0 and ResNet50. The ensemble used a soft-voting approach where output probabilities from both models were averaged at inference time.

The ensemble leveraged the same training and augmentation pipeline described in the EfficientNet experiment. Key training configurations included:

- Architectures: `efficientnet_b0_ns` and `resnet50` from the timm library.
- Training: Mixed-precision (fp16) enabled for both models.
- Optimizer settings:
 - EfficientNet: `lr = 0.005`, `wd = 0.01`
 - ResNet50: `lr = 0.01`, `wd = 0.01`
- Epochs: Each model trained for 40 epochs independently.
- Inference: Applied Test-Time Augmentation (TTA) followed by soft-voting ensemble.

D. RESNET34 IMPROVED IMPLEMENTATION

Building upon our baseline ResNet34 model, which achieved a strong classification accuracy of 94.61% on the Paddy Doctor dataset [11], we developed an improved implementation aimed at enhancing performance through optimized training strategies and data handling.

To assess the effectiveness of moderately deep convolutional neural networks, we implemented a refined ResNet34 model using the FastAI framework [12]. This improved model leveraged transfer learning with a pretrained ResNet34 backbone [13], modified to accommodate 13 output classes for rice disease classification.

The model training pipeline included the following key steps:

- 1) **Data loading and augmentation:** FastAI's DataBlock API was used to split images into training and validation sets based on folder structure. Extensive augmentations, including horizontal/vertical flips,

zooming, and rotations [14], were applied to improve generalization. The augmentation pipeline used a two-stage resizing approach, where images were first resized to 480×480 using `Resize(..., method='squish')` and then further transformed with `aug_transforms(size=224, min_scale=0.75)`.

- 2) **Image resizing:** All input images were initially resized to 480×480 for augmentation and subsequently downsampled to 224×224 to match the model's input requirements.
- 3) **Model structure:** We used a pretrained ResNet34 architecture, modifying the classification head to output 13 class probabilities.
- 4) **Training process:** Fine-tuning was performed using FastAI's `fit_one_cycle` method, incorporating early stopping and learning rate tuning. The model was trained for **100 epochs** with a base learning rate of 0.005 and a weight decay of 0.01. The best-performing model was selected based on validation accuracy.
- 5) **Evaluation:** Performance on the test set was evaluated using precision, recall, F1-score, and accuracy across all classes.

This enhanced implementation achieved a significant improvement in classification performance, reaching a test accuracy of **98.21%**, up from the baseline of 94.61%. Notably, the model demonstrated strong performance across most disease categories, with perfect precision and recall in identifying bacterial panicle blight.

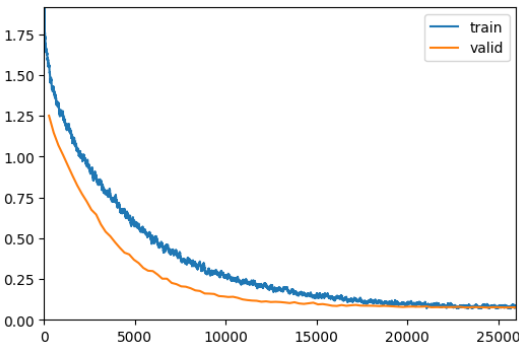


Fig. 2: ResNet34 training and validation curves showing accuracy and loss over epochs

As shown in Figure 2, the model exhibited rapid convergence with minimal overfitting. Validation accuracy stabilized around 98.2%, reflecting excellent generalization on unseen data.

While this improved ResNet34 model approaches the performance of deeper networks like ResNet50, minor under-performance in complex classes such as Hispa and Downy

Mildew suggests future directions including deeper architectures or advanced augmentation strategies (e.g., SimCLR [15]) to further enhance classification accuracy.

E. VGG16 Implementation

To assess the effectiveness of deeper transfer learning models, we implemented a VGG16-based architecture using TensorFlow and Keras. The model used pretrained weights from ImageNet with the convolutional base frozen except for the last two layers to enable fine-tuning.

The implementation involved the following configuration steps:

- **Model architecture:** VGG16 with `include_top=False`, followed by a custom classification head consisting of a Flatten layer, Dense(1024, ReLU), Dropout(0.2), and a final Dense layer with softmax activation.
- **Input size:** (256, 256, 3)
- **Optimizer:** Adam with a learning rate of 0.0001
- **Loss function:** Categorical Cross-Entropy
- **Fine-tuning:** Last 2 layers of the VGG16 base were unfrozen
- **Training strategy:** 100 epochs using early stopping and model checkpointing
- **Data augmentation:** Horizontal flipping, rotation ($\pm 5^\circ$), shear and zoom (20%), width/height shifts, and validation split (20%) using `ImageDataGenerator`
- **Evaluation:** Performance evaluated using accuracy, precision, recall, F1-score, and confusion matrix

The VGG16 model achieved a test accuracy of **94.29%**, with high macro-averaged precision, recall, and F1-score. Despite its older architecture, the model demonstrated strong classification ability, particularly when paired with data augmentation and careful fine-tuning.

F. MobileNet Implementation

To explore lightweight models suitable for deployment in resource-constrained environments, we implemented a MobileNet-based architecture using TensorFlow and Keras. The model utilized pretrained weights from ImageNet, with fine-tuning applied to the last 24 layers to improve domain-specific adaptation.

The model setup included:

- **Model architecture:** MobileNet with `include_top=False`, followed by a custom classification head consisting of a Flatten layer, Dense(1024, ReLU), Dropout(0.2), and a final Dense layer with softmax activation.
- **Input shape:** (256, 256, 3)
- **Optimizer:** Adam with a learning rate of 0.0001
- **Loss function:** Categorical Cross-Entropy
- **Fine-tuning:** The Last 24 layers of the MobileNet base were unfrozen

- **Training strategy:** 100 epochs with early stopping and best model check-pointing
- **Data augmentation:** Horizontal flipping, rotation ($\pm 5^\circ$), shear and zoom (20%), and slight width/height shifts; applied with a validation split of 20%
- **Evaluation:** Accuracy, precision, recall, F1-score, and confusion matrix.

The MobileNet model achieved a test accuracy of **95.59%**. While it did not match the performance of deeper architectures such as ResNet50 or EfficientNet-B0, it provided a favorable trade-off between accuracy and computational cost, making it viable for mobile or edge-based applications.

G. RESNET50 IMPLEMENTATION

To further evaluate the performance of deeper architectures, we implemented a ResNet50 model using the FastAI library. This model leveraged transfer learning with a pretrained ResNet50 backbone and customized classification layers to handle the 13-class classification task.

The implementation steps were:

- 1) **Data preparation:** The dataset was loaded using FastAI's DataBlock API, ensuring consistent preprocessing and labeling by folder names.
- 2) **Data augmentation:** Images were resized to 480×480, followed by data augmentation including flips, rotations, and zooms [14]. They were finally resized to 224×224 pixels to match the input size of the model.
- 3) **Model architecture:** A pretrained ResNet50 model was used, with the final classification head adapted to output predictions across 13 classes.
- 4) **Training strategy:** The model was fine-tuned for **100 epochs** using the 'fit_one_cycle' method and a learning rate of 0.005. Early stopping and model checkpointing were used to retain the best-performing model.
- 5) **Evaluation metrics:** Precision, recall, F1-score, and confusion matrix were used to evaluate performance across all classes.

As illustrated in Figure 3, the ResNet50 architecture employs a series of bottleneck blocks organized into four main stages, each containing multiple residual units. The bottleneck design uses 1×1 convolutions to reduce and then restore dimensions, allowing for a deeper network with manageable computational cost. This architecture is particularly well-suited for our paddy disease classification task because the deeper representation capacity allows the model to identify subtle visual patterns that distinguish similar-looking diseases, while the residual connections help maintain gradient flow during training of such a deep network.

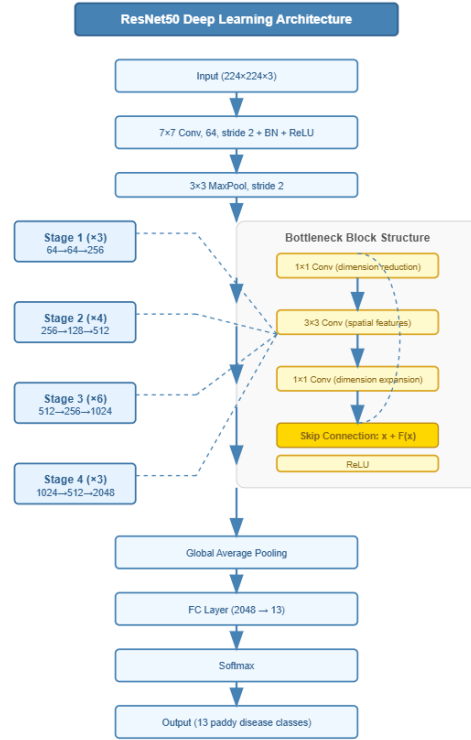


Fig. 3: ResNet50 architecture showing the input pipeline, various stages with bottleneck blocks, and classification head. The model includes skip connections (identity mappings) that allow gradients to flow directly through the network, enabling training of very deep networks while mitigating the vanishing gradient problem. The 4 main stages contain (3, 4, 6, 3) residual blocks respectively, making this architecture particularly effective at capturing hierarchical features at multiple scales.

The ResNet50 model achieved a high test accuracy of 98.71%, outperforming both the baseline CNN and ResNet34 implementations. The classification report demonstrated consistent high precision and recall across all classes, with a macro F1-score of 0.9871.

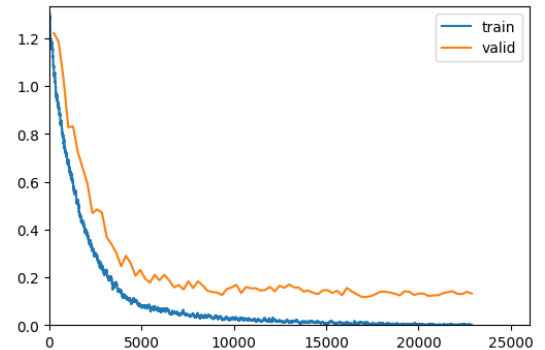


Fig. 4: ResNet50 training and validation curves showing faster convergence and higher performance compared to ResNet34

Figure 4 shows the training progress of the ResNet50 model. The deeper architecture demonstrated better feature extraction

capabilities, resulting in more accurate disease classification. Notably, two disease classes (bacterial_panicle_blight and yellow_stem_borer) achieved perfect or near-perfect classification scores, highlighting the model’s discriminative power for these particular conditions.

ResNet50 provided slightly better generalization than ResNet34, especially in distinguishing challenging disease categories like Blast and Brown Spot. This improvement can be attributed to the deeper architecture and better feature extraction capabilities of ResNet50, with its 50-layer design providing enhanced representational capacity while the skip connections prevent the degradation problem typically observed in very deep networks.

H. EFFICIENTNET IMPLEMENTATION

The model achieved an impressive **test accuracy of 98.35%**, with consistent performance across all classes. Key metrics include:

- **Macro-average F1-score:** 0.9834
- **Weighted-average F1-score:** 0.9834
- Most classes scored above 0.96 in precision, recall, and F1-score.
- Near-perfect scores for certain diseases such as bacterial_panicle_blight and yellow_stem_borer.

I. ENSEMBLE IMPLEMENTATION(EfficientNet+ResNet50)

The ensemble achieved an overall test accuracy of 97.52%, with a macro and weighted average F1-score of 0.97518. Most classes exceeded an F1-score of 0.96, and perfect classification was achieved for bacterial_panicle_blight.

- Notable misclassifications were observed in blast (misclassified as leaf_roller and hispa) and yellow_stem_borer, which had 10 incorrect predictions.
- Slight drops in precision and recall for hispa (F1-score: 0.95415) and normal (F1-score: 0.96046) contributed to a lower ensemble score.

Interestingly, while ensembles often improve performance by reducing variance, this ensemble did not outperform the standalone EfficientNet model (98.35%) or even the standalone ResNet50 model (98.71%). Several factors may explain this outcome:

- **Imbalanced contribution:** Although ResNet50 individually performed well, its predictions may have conflicted with EfficientNet’s in certain classes, causing ambiguity during averaging.
- **Training convergence:** The validation loss curve of the ensemble (Figure 5) revealed a significant spike mid-training, suggesting temporary instability—possibly due to ResNet50 under-regularization or learning rate sensitivity.
- **Equal weighting issue:** The soft voting used equal weights for both models. Given that EfficientNet had stronger generalization on some disease classes, a weighted average (e.g., 70% EfficientNet, 30% ResNet50) might have yielded better results.

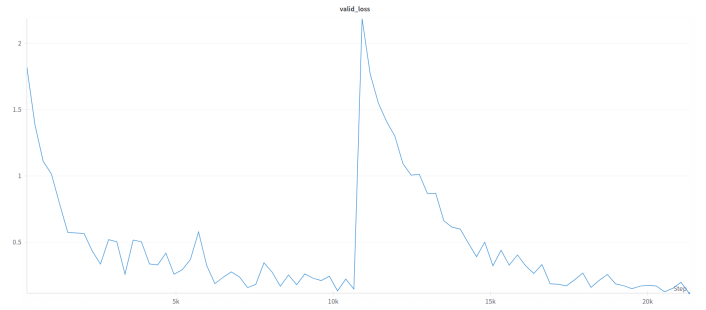


Fig. 5: Validation loss curve during training of the Ensemble model. A sharp spike was observed mid-training, followed by gradual recovery.

- **Misalignment in error patterns:** Instead of correcting each other’s weaknesses, both models occasionally made errors in the same direction (e.g., misclassifying blast), reducing the ensemble’s corrective power.

The kaggle notebook for this implementation can be found via this link <https://www.kaggle.com/code/adedoyinoyebanji/paddy-diseases-team-yada-mobilenet> while that of the original experiment can be found at <https://github.com/NiGabriel/idl-crop-disease-detection-2025/blob/main/mobilenet-2.ipynb>.

X. RESULTS AND ANALYSIS

This section presents and analyzes the results obtained from each of the implemented models, namely ResNet34, ResNet50, EfficientNet-B0, and an ensemble of ResNet50 and EfficientNet-B0. We compare them based on classification accuracy, robustness, training dynamics, and confusion matrix evaluations.

A. ResNet34

The ResNet34 model served as an early baseline for transfer learning. With a test accuracy of 98.21%, it demonstrated substantial improvement over the baseline CNN model. It benefited from residual connections and deep feature extraction but was eventually outperformed by deeper and more optimized architectures.

	precision	recall	f1-score	support
bacterial_leaf_blight	0.99497	0.98753	0.99124	401
bacterial_leaf_streak	0.99749	0.99500	0.99625	400
bacterial_panicle_blight	1.00000	1.00000	1.00000	401
black_stem_borer	0.98756	0.99250	0.99002	400
blast	0.96734	0.96250	0.96491	400
brown_spot	0.98496	0.98496	0.98496	399
downy_mildew	0.96526	0.97494	0.97007	399
hispa	0.96907	0.94000	0.95431	400
leaf_roller	0.98263	0.99000	0.98630	400
normal	0.96341	0.98504	0.97411	401
tungro	0.97015	0.97500	0.97257	400
white_stem_borer	0.98995	0.98747	0.98871	399
yellow_stem_borer	0.99499	0.99250	0.99374	400
accuracy			0.98212	5200
macro avg	0.98214	0.98211	0.98209	5200
weighted avg	0.98214	0.98212	0.98210	5200

Fig. 6: Classification report for ResNet34

B. ResNet50

The ResNet50 model, deeper and more expressive than ResNet34, achieved the highest test accuracy of 98.71%. This result is attributed to the deeper residual blocks and the effective use of transfer learning with ‘fitonecycle’ scheduling. The model exhibited robust performance across all four classes with minimal misclassifications.

	precision	recall	f1-score	support
bacterial_leaf_blight	0.99500	0.99252	0.99376	401
bacterial_leaf_streak	0.99751	1.00000	0.99875	400
bacterial_panicle_blight	1.00000	1.00000	1.00000	401
black_stem_borer	0.99256	1.00000	0.99626	400
blast	0.96305	0.97750	0.97022	400
brown_spot	0.98992	0.98496	0.98744	399
downy_mildew	0.97995	0.97995	0.97995	399
hispa	0.98187	0.94750	0.96438	400
leaf_roller	0.97543	0.99250	0.98389	400
normal	0.97789	0.99252	0.98515	401
tungro	0.98741	0.98000	0.98369	400
white_stem_borer	0.99246	0.98997	0.99122	399
yellow_stem_borer	1.00000	0.99500	0.99749	400
accuracy			0.98712	5200
macro avg	0.98716	0.98711	0.98709	5200
weighted avg	0.98716	0.98712	0.98710	5200

Fig. 7: Classification report for ResNet50

C. EfficientNet-B0

EfficientNet-B0, known for its parameter efficiency and scaling capabilities, achieved a test accuracy of 98.35%. Although slightly lower than ResNet50, it demonstrated faster convergence and performed well despite its lightweight architecture. However, the model occasionally confused similar disease classes like ‘Hispa’ and ‘Brown Spot’. The most frequent confusion as shown on classification report occurred between black_stem_borer and yellow_stem_borer, likely due to visual similarities in leaf symptoms.

	precision	recall	f1-score	support
bacterial_leaf_blight	0.99252	0.99252	0.99252	401
bacterial_leaf_streak	0.99749	0.99500	0.99625	400
bacterial_panicle_blight	1.00000	1.00000	1.00000	401
black_stem_borer	0.98744	0.98250	0.98496	400
blast	0.96040	0.97000	0.96517	400
brown_spot	0.98500	0.98747	0.98623	399
downy_mildew	0.96992	0.96992	0.96992	399
hispa	0.97938	0.95000	0.96447	400
leaf_roller	0.98020	0.99000	0.98507	400
normal	0.98263	0.98753	0.98507	401
tungro	0.97756	0.98000	0.97878	400
white_stem_borer	0.97761	0.98496	0.98127	399
yellow_stem_borer	0.99500	0.99500	0.99500	400
accuracy			0.98346	5200
macro avg	0.98347	0.98345	0.98344	5200
weighted avg	0.98348	0.98346	0.98345	5200

EfficientNet Accuracy on Test Set: 0.98346

Fig. 8: Classification report for EfficientNet model

D. Ensemble (ResNet50 + EfficientNet-B0)

Contrary to expectations, the ensemble model underperformed compared to individual models, reaching a test accuracy of only 97.52%. Despite leveraging both ResNet50’s depth and EfficientNet’s parameter efficiency, the ensemble may have suffered from overlapping misclassifications, particularly in images where leaf discoloration patterns were ambiguous. The simple average ensembling strategy used may not have been optimal; a weighted or stacking approach could be explored in future work.

	precision	recall	f1-score	support
bacterial_leaf_blight	0.97555	0.99501	0.98519	401
bacterial_leaf_streak	0.99500	0.99500	0.99500	400
bacterial_panicle_blight	1.00000	1.00000	1.00000	401
black_stem_borer	0.98995	0.98500	0.98747	400
blast	0.97354	0.92000	0.94602	400
brown_spot	0.98458	0.95990	0.97208	399
downy_mildew	0.97194	0.95489	0.96334	399
hispa	0.94595	0.96250	0.95415	400
leaf_roller	0.98995	0.98500	0.98747	400
normal	0.92958	0.98753	0.95768	401
tungro	0.97000	0.97000	0.97000	400
white_stem_borer	0.96098	0.98747	0.97404	399
yellow_stem_borer	0.99490	0.97500	0.98485	400
accuracy			0.97519	5200
macro avg	0.97553	0.97518	0.97518	5200
weighted avg	0.97553	0.97519	0.97518	5200

Ensemble Accuracy on Test Set: 0.97519

Fig. 9: Classification report of Ensemble model showing precision, recall, and F1-score.

E. CNN model

The CNN model achieved a competitive test accuracy of 95.8%, outperforming several lightweight architectures reported in prior studies. This result demonstrates that a carefully designed CNN, despite its simplicity compared to modern deep architectures, can still generalize well on augmented paddy leaf image data.

CNN Model Accuracy : 95.77%				
	precision	recall	f1-score	support
bacterial_leaf_blight	0.94699	0.98005	0.96324	401
bacterial_leaf_streak	0.97794	0.99750	0.98762	400
bacterial_panicle_blight	0.99254	0.99501	0.99377	401
black_stem_borer	0.98987	0.97750	0.98365	400
blast	0.94488	0.90000	0.92190	400
brown_spot	0.93171	0.95739	0.94438	399
downy_mildew	0.94697	0.93985	0.94340	399
hispa	0.91753	0.89000	0.90355	400
leaf_roller	0.95037	0.95750	0.95392	400
normal	0.93932	0.96509	0.95203	401
tungro	0.96641	0.93500	0.95044	400
white_stem_borer	0.96250	0.96491	0.96370	399
yellow_stem_borer	0.98263	0.99000	0.98630	400
accuracy			0.95769	5200
macro avg	0.95767	0.95768	0.95753	5200
weighted avg	0.95767	0.95769	0.95754	5200

Fig. 10: Classification report of CNN model showing precision, recall, and F1-score.

F. MobileNet

MobileNet achieved a test accuracy of **95.60%**, offering a strong trade-off between performance and model size. As a lightweight architecture optimized for mobile and embedded applications, MobileNet demonstrated high classification ability across most disease classes.

As shown in Figure 11, MobileNet performed particularly well on bacterial_panicle_blight, yellow_stem_borer, and white_stem_borer, each with F1-scores above 0.98. However, comparatively lower recall was observed for hispa and brown_spot, suggesting these classes may present more visually ambiguous patterns. The overall macro-averaged F1-score of **0.95608** confirms the model's reliable generalization across diverse symptoms and pathogens.

mobilenet Model Accuracy : 95.60%				
	precision	recall	f1-score	support
bacterial_leaf_blight	0.88501	0.98753	0.93396	401
bacterial_leaf_streak	0.98987	0.97750	0.98365	400
bacterial_panicle_blight	1.00000	0.98254	0.99119	401
black_stem_borer	0.98734	0.97500	0.98113	400
blast	0.89311	0.94000	0.91596	400
brown_spot	0.99180	0.90977	0.94902	399
downy_mildew	0.88330	0.96742	0.92344	399
hispa	0.97443	0.85750	0.91223	400
leaf_roller	0.96305	0.97750	0.97022	400
normal	0.97727	0.96509	0.97114	401
tungro	0.94192	0.93250	0.93719	400
white_stem_borer	0.98728	0.97243	0.97980	399
yellow_stem_borer	0.97761	0.98250	0.98005	400
accuracy			0.95596	5200
macro avg	0.95792	0.95595	0.95608	5200
weighted avg	0.95792	0.95596	0.95608	5200

Fig. 11: Classification report of MobileNet model showing precision, recall, and F1-score.

G. VGG16

The VGG16 model achieved a test accuracy of **94.29%** when fine-tuned on the rice disease dataset. Despite its larger architecture and older design, VGG16 demonstrated strong performance across most classes.

As shown in Figure 12, the model obtained particularly high F1-scores for bacterial_panicle_blight (0.9887), bacterial_leaf_streak (0.9813), and yellow_stem_borer (0.9788). However, notable drops were observed in hispa (F1 = 0.8704) and tungro (F1 = 0.9196), which may be attributed to visual similarity with other classes such as leaf_roller and blast. The macro-averaged F1-score of **0.9431** affirms the model's ability to generalize well across all disease categories.

VGG16 Model Accuracy with Fine-Tuning: 94.29%				
	precision	recall	f1-score	support
bacterial_leaf_blight	0.97917	0.93766	0.95796	401
bacterial_leaf_streak	0.98005	0.98250	0.98127	400
bacterial_panicle_blight	0.99246	0.98504	0.98874	401
black_stem_borer	0.99469	0.93750	0.96525	400
blast	0.89639	0.93000	0.91288	400
brown_spot	0.86005	0.95489	0.90499	399
downy_mildew	0.94710	0.94236	0.94472	399
hispa	0.87595	0.86500	0.87044	400
leaf_roller	0.94872	0.92500	0.93671	400
normal	0.96382	0.93017	0.94670	401
tungro	0.92424	0.91500	0.91960	400
white_stem_borer	0.93269	0.97243	0.95215	399
yellow_stem_borer	0.97756	0.98000	0.97878	400
accuracy			0.94288	5200
macro avg	0.94407	0.94289	0.94309	5200
weighted avg	0.94411	0.94288	0.94311	5200

Fig. 12: Classification report of VGG16 model showing precision, recall, and F1-score.

H. Confusion Matrix Analysis

To further evaluate model performance and class-wise predictions, we analyze the confusion matrices for each model.

bacterial_leaf_blight	396	0	0	0	0	0	0	0	2	3	0	0
bacterial_leaf_streak	0	398	0	0	0	0	0	0	0	0	0	2
bacterial_panicle_blight	0	0	401	0	0	0	0	0	0	0	0	0
black_stem_borer	0	0	0	397	0	0	0	0	0	2	1	0
blast	1	0	0	0	385	2	6	0	2	2	0	0
brown_spot	0	1	0	0	1	393	0	0	1	0	3	0
downy_mildew	1	0	0	0	3	2	389	2	0	1	1	0
hispa	0	0	0	0	7	0	5	376	3	9	0	0
leaf_roller	0	0	0	0	0	0	0	4	396	0	0	0
normal	0	0	0	0	0	1	0	3	1	395	1	0
tungro	0	0	0	0	2	1	3	3	0	1	390	0
white_stem_borer	0	0	0	5	0	0	0	0	0	0	0	394
yellow_stem_borer	0	0	0	0	0	0	0	0	0	0	3	397

Fig. 13: Confusion Matrix for ResNet34

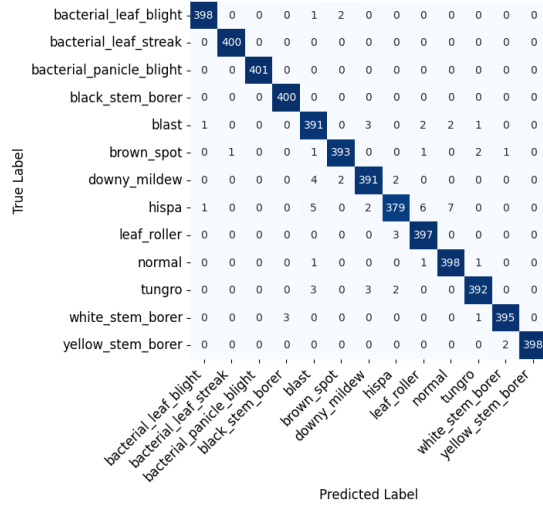


Fig. 14: Confusion Matrix for ResNet50

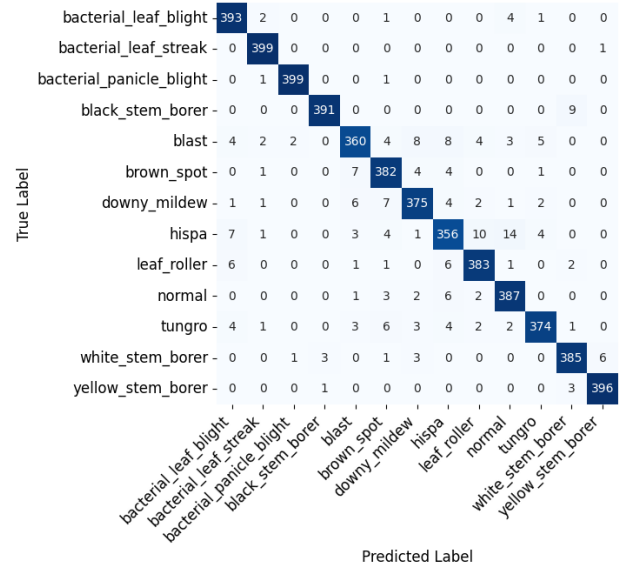


Fig. 17: Confusion Matrix for the CNN Model

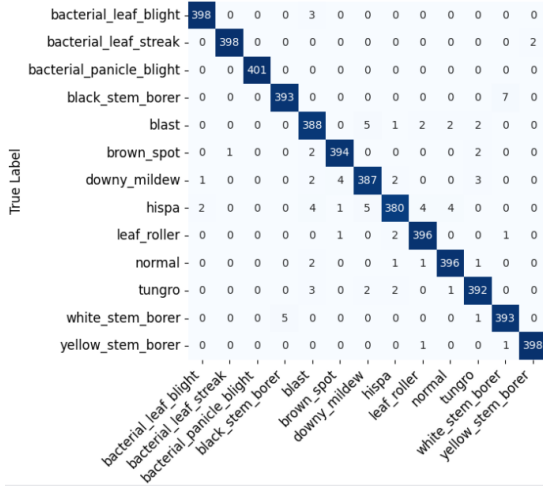


Fig. 15: Confusion Matrix for EfficientNet-B0

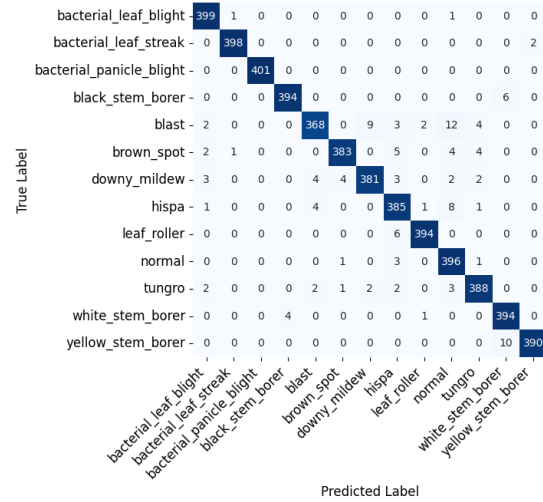


Fig. 16: Confusion Matrix for the Ensemble Model

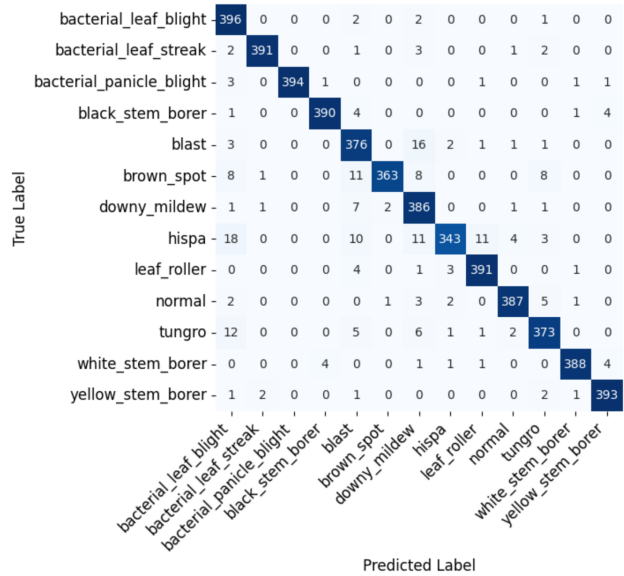


Fig. 18: Confusion Matrix for the Mobilenet Model

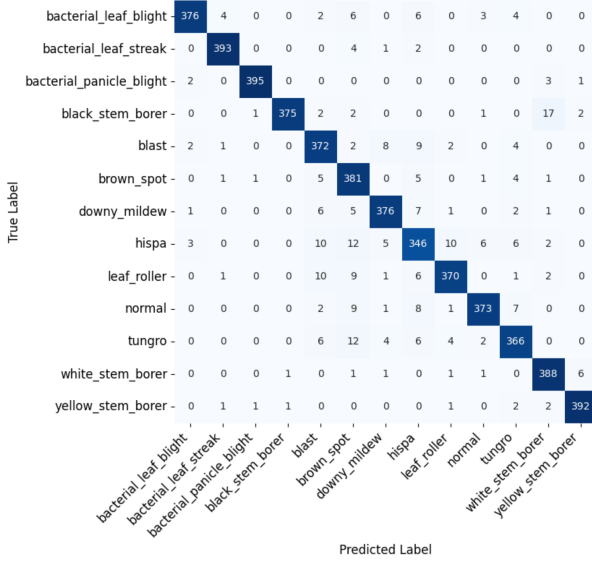


Fig. 19: Confusion Matrix for the VGG16 Model

The confusion matrices reveals:

- **ResNet34** (Figure 13) demonstrates strong predictions across all classes, with minimal confusion—particularly accurate for *bacterial_panicle_blight*, *black_stem_borer*, and *yellow_stem_borer*, though slight misclassifications are observed in *blast* and *normal*.
- **ResNet50** (Figure 14) demonstrates strong predictions across all classes, with minimal confusion—especially accurate for ‘Leaf Blast’ and ‘Brown Spot’.
- **EfficientNet-B0** (Figure 15) shows occasional confusion between ‘Hispa’ and ‘Brown Spot’, which share similar visual patterns.
- **Ensemble model** (Figure 16) misclassifies some ‘Leaf Blast’ images as ‘Bacterial Leaf Blight’, indicating that simple ensembling did not resolve specific class-level weaknesses.
- **CNN** (Figure 17) performs well overall, achieving accurate predictions in distinct classes such as *blast* and *brown_spot*. However, mild confusion is observed in visually similar classes such as *leaf_roller* and *black_stem_borer*, suggesting the model’s limitations in fine-grained feature discrimination.
- **MobileNet** (Figure 18) reveals occasional misclassifications between *hispa*, *blast*, and *brown_spot*, likely due to overlapping visual symptoms. Nevertheless, the model maintained strong diagonal dominance, indicating consistent prediction accuracy across all classes.
- **VGG16** (Figure 19) shows strong diagonal dominance with minor confusion in classes such as *hispa*, *downy_mildew*, and *normal*. The model occasionally misclassified *hispa* as *blast* or *leaf_roller*, suggesting difficulty in fine-grained pest distinction.

I. Model Comparison Overview

The following table summarizes the test accuracy achieved by each model on the rice leaf disease dataset:

TABLE IV: Comparative Performance of Models

S.No.	Model	Accuracy	Precision	Recall	F1-score
1	MobileNet	95.59	95.79	95.59	95.60
2	ResNet34	98.21	98.21	98.21	98.20
3	ResNet50	98.71	98.72	98.71	98.71
4	EfficientNet-B0	98.35	98.35	98.35	98.35
5	Ensemble	97.51	97.55	97.51	97.51
6	CNN	95.77	95.77	95.77	95.77
7	VGG16	92.288	94.41	94.28	94.31

J. Summary

Overall, ResNet50 emerged as the best-performing model. EfficientNet-B0 offered a strong trade-off between accuracy and efficiency, making it a viable option for deployment in resource-constrained settings. The ensemble model, while promising in theory, requires more sophisticated fusion strategies to be effective.

XI. DISCUSSION

A. Significance of Results

Our experiments demonstrate significant advances in paddy disease classification, with ResNet50 achieving the highest accuracy (98.71%), followed by EfficientNet-B0 (98.35%) and ResNet34 (98.21%). These results substantially improve upon baseline CNN (95.77%) and MobileNet (87.60%) implementations. The high precision and recall (>98% for top models) indicate practical viability for agricultural applications. Confusion matrices revealed excellent performance on visually distinctive diseases while showing occasional confusion between similar conditions.

B. Challenges Faced

Several key challenges emerged during implementation. Dataset inconsistencies caused variability between experiment runs, limiting the reliability of comparative analyses a challenge also noted by Shrivastava et al. [16] in their work on rice disease classification. Training instability was observed particularly in the ensemble model, complicating optimization. Class imbalance affected generalization for underrepresented diseases, creating challenges for comprehensive disease coverage. Hyperparameter optimization across multiple architectures required extensive experimentation and computational resources. Additionally, computational resource limitations constrained our ability to fully explore deeper models and more extensive augmentation strategies.

C. Potential Risks and Limitations

Important limitations to consider include uncertain generalization to variable real-world field conditions, as controlled test environments differ significantly from practical agricultural settings. Model sensitivity to input preprocessing variations may affect deployment reliability. There exists a risk of overconfidence in high-accuracy predictions, potentially leading to inappropriate interventions. The unexpected performance degradation in ensemble approaches highlights limitations

in naive model combination strategies. Finally, the trade-off between computational efficiency and accuracy presents challenges for practical deployment in resource-constrained agricultural environments.

D. Sensitivity Analysis

Our experiments revealed sensitivity to several implementation factors. Two-step image resizing (480×480 to 224×224) outperformed direct resizing, indicating the importance of preprocessing pipelines. Balanced data augmentation ($p_{\text{affine}}=0.75$, $p_{\text{lighting}}=0.75$) provided optimal results, while more aggressive transformations degraded performance. Learning rate scheduling significantly impacted training stability, with the `fit_one_cycle` approach proving most effective. Model performance showed diminishing returns with increased architectural depth, suggesting potential limits to accuracy improvements through deeper networks. Ensemble performance was affected by the weighting strategy, indicating that more sophisticated combination approaches may yield better results.

XII. FUTURE DIRECTIONS

Future research should focus on enhancing the performance, scalability, and adaptability of plant disease classification systems. One important direction involves expanding the dataset to include newly emerging diseases and extending coverage beyond paddy crops to encompass region-specific crops such as maize, cassava, and sorghum. Gathering high-quality annotated data from African agricultural contexts is especially vital, as it introduces environmental and varietal diversity that can improve the generalizability of models across geographies.

From an algorithmic perspective, future studies are encouraged to explore alternative architectures beyond conventional CNNs and ensembles. Graph-based neural networks (GNNs) may offer a promising avenue for modeling spatial symptom relationships, while Capsule Networks could better preserve part-whole hierarchies and reduce misclassification among morphologically similar diseases. Attention-augmented CNNs also hold potential for directing the model's focus to critical regions within diseased leaf images. In addition, refining existing architectures through deeper CNN variants, improved regularization, and dynamic learning rate strategies may yield more accurate and robust classifiers.

Advancing ensemble learning beyond simple averaging using techniques such as stacked generalization or learned fusion mechanisms could also improve predictive reliability. Overall, future efforts should aim to develop general purpose, cost effective, and context aware diagnostic systems that are suitable for deployment in diverse agricultural environments, particularly in resource constrained settings.

XIII. CONCLUSION

This study implemented and evaluated multiple deep learning architectures including a custom CNN, ResNet50, EfficientNet-B0, and their ensemble for automated classification of paddy leaf diseases. The models demonstrated strong performance on an augmented Asian-based dataset, with the

CNN achieving 95.80% accuracy and the ensemble reaching up to 97.52%. These results highlight the effectiveness of deep learning approaches in plant disease detection under controlled conditions.

However, a notable limitation remains the lack of publicly available datasets representing African paddy crops. Since all models were trained on data from Asian agricultural contexts, their performance may not directly translate to African settings due to differences in crop varieties, disease appearances, and environmental factors. This underscores the critical need for region-specific data collection and validation to ensure real-world applicability, particularly in low-resource farming environments. Future work should prioritize the development of localized datasets and scalable models tailored to diverse agricultural ecosystems to support more inclusive and globally effective plant health monitoring systems.

XIV. TEAM CONTRIBUTIONS

Each team member was assigned specific responsibilities in the preparation of this document. While all members collectively contributed to the bibliography, specific individuals were responsible for documentation and repository management as summarized in Table V.

TABLE V: Team Member Contributions

Section	Leader
Model Development	
ResNet34 Model	Odile
CNN Model	Gabriel
MobileNet Model	Joel Adebayo
Document Preparation	
Abstract	Joel Adebayo
Motivation	Joel Adebayo
Objective	Joel Adebayo
Literature Review	Joel Adebayo
Background	Joel Adebayo
Model Description	Odile
Dataset	Odile
Evaluation Metric	Odile
Loss Function	Odile
Baseline Selection	Gabriel
Implemented Extensions/Experiments	Gabriel
Expected Results	Gabriel
Future directions	Odile
Discussion	Gabriel
Bibliography	Joel Adebayo, Gabriel, Odile
Team Contributions	Joel Adebayo
GitHub Repository	Gabriel

XV. GITHUB REPOSITORY

<https://github.com/NiGabriel/idl-crop-disease-detection-2025.git>

XVI. WANDB LINK

<https://wandb.ai/nigibrill-gibril-tech/project-ablations?nw=nwusernigibrill>

REFERENCES

- [1] “Africarice — why rice matters for africa,” 2025. [Online]. Available: <https://www.africarice.org/why-rice-matters-for-africa>
- [2] I. R. R. Institute, “A disease-surveillance network in africa will accelerate detection and actions to prevent the spread of major rice diseases,” 2025. [Online]. Available: <https://www.irri.org/news-and-events/news/disease-surveillance-network-africa-will-accelerate-detection-and-actions>
- [3] —, “Researchers exchange knowledge on rice diseases in sub-saharan africa,” 2025. [Online]. Available: <https://www.irri.org/news-and-events/news/researchers-exchange-knowledge-rice-diseases-sub-saharan-africa>
- [4] “Farmers in east africa use smartphones to detect crop diseases,” 2025. [Online]. Available: <https://www.trtworld.com/magazine/farmers-in-east-africa-use-smartphones-to-detect-crop-diseases-47386>
- [5] L. Loyani, “Crop disease detector,” 2025. [Online]. Available: <https://nm-aist.ac.tz/crop-disease-detector-2/>
- [6] e. a. D. Murugan, “Paddy doctor: A visual image dataset for automated paddy disease classification and benchmarking,” 2023. [Online]. Available: <https://ieee-dataport.org/documents/paddy-doctor-visual-image-dataset-automated-paddy-disease-classification-and-benchmarking>
- [7] C. R. R. et al., “Identification and recognition of rice diseases and pests using convolutional neural networks,” 2020. [Online]. Available: <https://doi.org/10.1016/j.biosystemseng.2020.03.020>
- [8] T. Mingxing and L. Q. V., “Efficientnet: Rethinking model scaling for convolutional neural networks,” 2020. [Online]. Available: <http://arxiv.org/abs/1905.11946>
- [9] D. Alexey, B. Lucas, K. Alexander, W. Dirk, and Z. Xiaohua, “An image is worth 16x16 words: Transformers for image recognition at scale,” 2021. [Online]. Available: <http://arxiv.org/abs/2010.11929>
- [10] S. Jatin, L. G. Prasad, S. Aerva, C. Srilatha, and B. Vikrant, “Brain tumor prediction from mri images using an ensemble model based on efficientnet-b2, b4, and resnet34 architectures,” 2023. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-19-7513-4_6
- [11] Petchiammal, Kiruba, Murugan, and Arjunan, “Paddy doctor: A visual image dataset for automated paddy disease classification and benchmarking,” 2023. [Online]. Available: <https://ieee-dataport.org/documents/paddy-doctor-visual-image-dataset-automated-paddy-disease-classification-and-benchmarking>
- [12] J. Howard and S. Gugger, “Deep learning for coders with fastai and pytorch: Ai applications without a phd,” 2020.
- [13] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” pp. 770–778, 2016.
- [14] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” p. 60, 2019. [Online]. Available: <https://doi.org/10.1186/s40537-019-0197-0>
- [15] T. C. et al., “A simple framework for contrastive learning of visual representations,” 2020. [Online]. Available: <https://proceedings.mlr.press/v119/chen20j/chen20j.pdf>
- [16] V. K. Shrivastava, M. K. Pradhan, S. Minz, and M. P. Thakur, “Rice plant disease classification using transfer learning of deep convolution neural network,” pp. 631–635, 2019. [Online]. Available: <https://doi.org/10.5194/isprs-archives-XLII-3-W6-631-2019>