TIERS Information Technology Journal
Vol.6, No.2, December 2025, pp. 144~157
P-ISSN: 2723-4533 / E-ISSN: 2723-4541
DOI: https://doi.org/10.38043/tiers.v6i2.7159

Efficient Rice Leaf Disease Classification Using Enhanced CAE-CNN Architecture

Destia Suhada[1], I Gede Pasek Suta Wijaya[2], Ida Bagus Ketut Widiartha[3], Minho Jo[4]

destiasuhada96@gmail.com[1], gpsutawijaya@unram.ac.id[2], widi@unram.ac.id[3], minhojo@korea.ac.kr[4]

[1] Master of Information Technology, University of Mataram, NTB, Indonesia
[2,3] Department of Informatics Engineering, University of Mataram, NTB, Indonesia
[4] Department of Computer Convergence Software, Korea University, South Korea

ABSTRACT

This study introduces an enhanced Convolutional Autoencoder–Convolutional Neural Network (CAE–CNN) model designed for efficient and accurate classification of rice leaf diseases. This study aims to develop an architecture that achieves high accuracy while maintaining computational efficiency, serving as an integrative and applicative technical innovation for rice disease detection. The proposed architecture integrates a Squeeze and Excitation Block (SE-Block), Global Max Pooling (GMP), and Separable Convolution to improve feature extraction while reducing the number of parameters and inference time. A total of 7,430 labeled images from five rice disease classes were used for model training and evaluation. The model was optimized using Optuna-based hyperparameter tuning and validated through an ablation and comparative analysis to assess the impact of each component. Experimental results show that the proposed model achieves 99.39% accuracy with only 85,859 parameters, a compact size of 0.28 MB, and inference time at 0.06657 ms/image with 15,213 FPS. These findings demonstrate that the proposed CAE–CNN effectively combines high accuracy and low computational cost, making it highly suitable for real-time and edge-based rice disease classification systems.

Keywords: Rice Leaf Disease; CAE; CNN; SE-Block; Lightweight Deep Learning

Article Info

Received : 20-09-2025
Revised : 11-10-2025
Accepted : 29-12-2025

Correspondence Author:

Destia Suhada
Master of Information Technology,
University of Mataram,
Jl. Majapahit No.62, Gomong, Kota Mataram, NTB, 83115, Indonesia.
Email: destiasuhada96@gmail.com

1. INTRODUCTION

Rice is a vital staple food for nearly half of the global population, including Indonesia, where over 90% of people depend on it as their main food source [1]. Despite producing around 54.4 million tonnes annually [2], Indonesia rice productivity remains low at 5.14 tonnes per hectare, well below the potential yield of 9.2 to 9.5 tonnes [1]. One major factor behind this gap is the prevalence of rice leaf diseases such as Bacterial Blight, Tungro, Blast, and Brown Spot, which significantly reduce both yield and quality. As a result, current research has concentrated on computational techniques to autonomously identify and categorize such diseases, enhancing precision and promoting sustainable agriculture [3].

Earlier studies applied digital image processing techniques using traditional classifiers like SVM, k-NN, and Random Forest. While this model is effective for basic pattern recognition, these models face

significant limitations, like SVM depends heavily on kernel selection [4], k-NN is sensitive to noise [5], and Random Forest can be computationally intensive [6]. These constraints have driven a shift toward deep learning, particularly Convolutional Neural Network (CNN), which can automatically learn spatial features and achieve high classification accuracy [7]. However, most CNNs are computationally heavy, with large parameter counts and memory requirements [8], making them unsuitable for low-resource environments.

To address this, researchers have explored lightweight CNNs as well as hybrid architectures that combine Convolutional Autoencoders (CAE) with CNN, as reported in previous studies by the authors [9] and other research like [10]. CAE is employed because it is capable of producing compact, noise-reduced feature representations [10], which helps lower the computational burden of CNNs without losing crucial information. Studies have shown that integrating CAE with CNN can reduce both the number of parameters and convolutional layers while still achieving high accuracy [9], [10]. Despite the benefits offered by CAE, this component still has certain limitations. In particular, the CAE–CNN model with Global Max Pooling (GMP) in [9], showed misclassification between visually similar diseases such as Brown Spot and Bacterial Blight due to overlapping lesion characteristics under limited lighting or resolution, indicating insufficient discriminative channel-level feature modeling. In addition, Boukhlifa and Chibani (2024) [10] reported that CAE-based lightweight CNNs struggle to generalize across datasets and real-world conditions due to high variability in disease manifestations.

Building upon this, the present study introduces an optimized CAE–CNN architecture that integrates a Squeeze and Excitation Block (SE-Block), GMP, and Separable Convolution within a unified architecture for rice leaf disease classification. The CAE is used to learn robust and compact latent features that mitigate noise and image quality variations, supporting improved robustness within the evaluated dataset. To alleviate the difficulty of distinguishing visually similar disease patterns observed in [9], SE-Blocks are incorporated to model inter-channel dependencies and enhance discriminative feature responses [11]. GMP is retained to reduce parameter growth and overfitting by suppressing redundant spatial information [12], while Separable Convolution further decreases computational complexity without compromising feature expressiveness [13]. Unlike prior studies that examined these components separately [9], [11], [12], their joint integration enables a balanced trade-off between feature representation and computational efficiency, resulting in reduced parameter count, smaller model size, and faster inference. This integrative design provides a practical and efficient solution for real-time rice leaf disease detection in resource-constrained agricultural environments.

2. RESEARCH METHOD

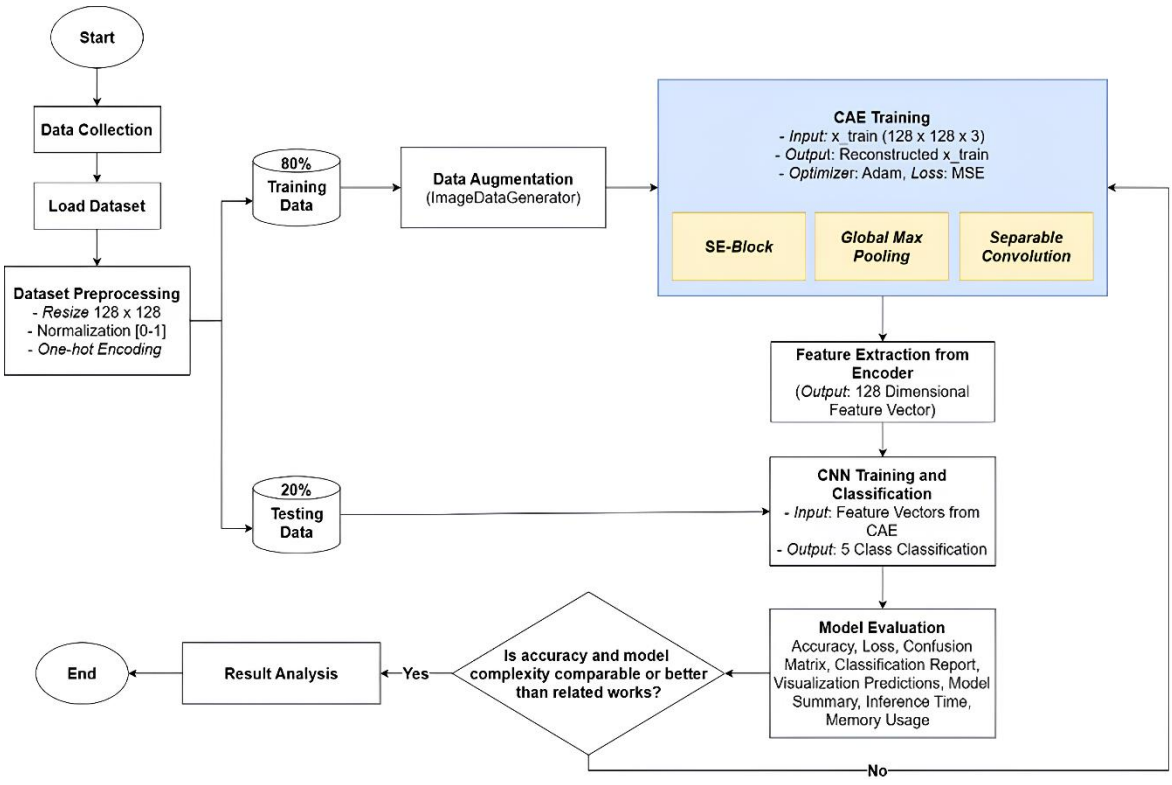


Figure 1. Research Flow

The overall workflow of the proposed research is illustrated in Figure 1, outlining each stage from data collection to model evaluation. The process begins with dataset preprocessing and augmentation, followed by training a CAE integrated with SE-Block, GMP, and Separable Convolution to extract compact feature representations. These features are then used by CNN to classify the five rice leaf disease categories. Model evaluation is conducted using accuracy, precision, recall, and inference efficiency, ensuring that the final design achieves an optimal balance between accuracy and efficiency.

2.1. Research Data

Table 1. Dataset Sources

Source	Dataset Name	Number of Images	Image Size (Pixels)
[14]	Nutrient Deficiency Symtoms in Rice	227	3024 x 3024
[15]	Rice Disease	5.945	1024 x 1024
[16]	Rice Disease Image Dataset	3.960	1024 x 1024
[17]	Rice Leaf Disease	3.081	256 x 256

This study uses publicly available rice leaf disease datasets from Kaggle, as summarized in Table 1. The combined dataset contains five classes, which are Bacterial Blight, Blast, Brown Spot, Tungro, and Healthy. All the images were carefully reviewed for clarity, correct labeling, and overall quality, while those with distortions, overlapping elements, or unclear disease symptoms were removed. Any ambiguous cases were manually checked to ensure accuracy.

2.2. Data Preprocessing

2.2.1. Data Transformation

Since the images came from different sources, all the images were converted to RGB to preserve color information, which is crucial for detecting rice leaf diseases. Color has been shown to play an important role in plant disease classification [18]. All images were then resized to 128x128 pixels to standardize the input size while keeping key visual details. This resolution strikes a balance between computational efficiency and maintaining important features such as color variations and texture [19]. After that, pixel normalization was applied by scaling pixel values to the range [0, 1] for stable and faster convergence during training [20]. Finally, the five disease classes were converted into numerical labels and then transformed into one-hot vectors using the to_categorical() function, allowing the model to compute categorical loss during training [21].

2.2.2. Dataset Splitting

Table 2. Dataset Distribution After Splitting

Class	Total Images	Training (80%)	Testing (20%)
Bacterial Blight	1.584	1.267	317
Blast	1.440	1.152	288
Brown Spot	1.600	1.280	320
Healthy	1.488	1.190	298
Tungro	1.318	1.055	263
Total	7.430	5.944	1.486

The dataset was split into 80% for training and 20% for testing, using stratified sampling to maintain balanced class distribution. This ensures there is enough data to train the model while reserving sufficient samples for a fair evaluation [22].

2.2.3. Data Augmentation

Table 3. Data Augmentation Parameters

Reference	Augmentation Type	Parameter
[23]	Rotation	Up to 20%
[24]	Widht Shift	Up to 20%
	Height Shift	Up to 20%
	Zoom	Up to 20%
	Horizontal Flip	True

Even though the dataset was big enough, there wasn't much visual variety in things like orientation and frame. We used the ImageDataGenerator class from Keras to add more data to the training sample. This helped the model generalize better and avoid overfitting [25].

.

To maintain significant color patterns related to disease, such as the yellowish colors in Tungro and the brown lesions in Brown Spot [3], color-based augmentation, like hue, saturation, or brightness, was not used. This decision was made to avoid changing the original color features that are important for identifying diseases [18]. If the color is randomly modified, it could distort the real appearance of the disease and make one class look like another.

By keeping the original colors unchanged, the model can learn the real color and texture patterns of each disease, which makes its predictions more reliable and easier to interpret in real-world agricultural use. Table 3 shows the chosen parameters for the augmentation. These changes make the data more varied without changing important parts of rice leaf diseases.

2.3. Proposed Model Architecture

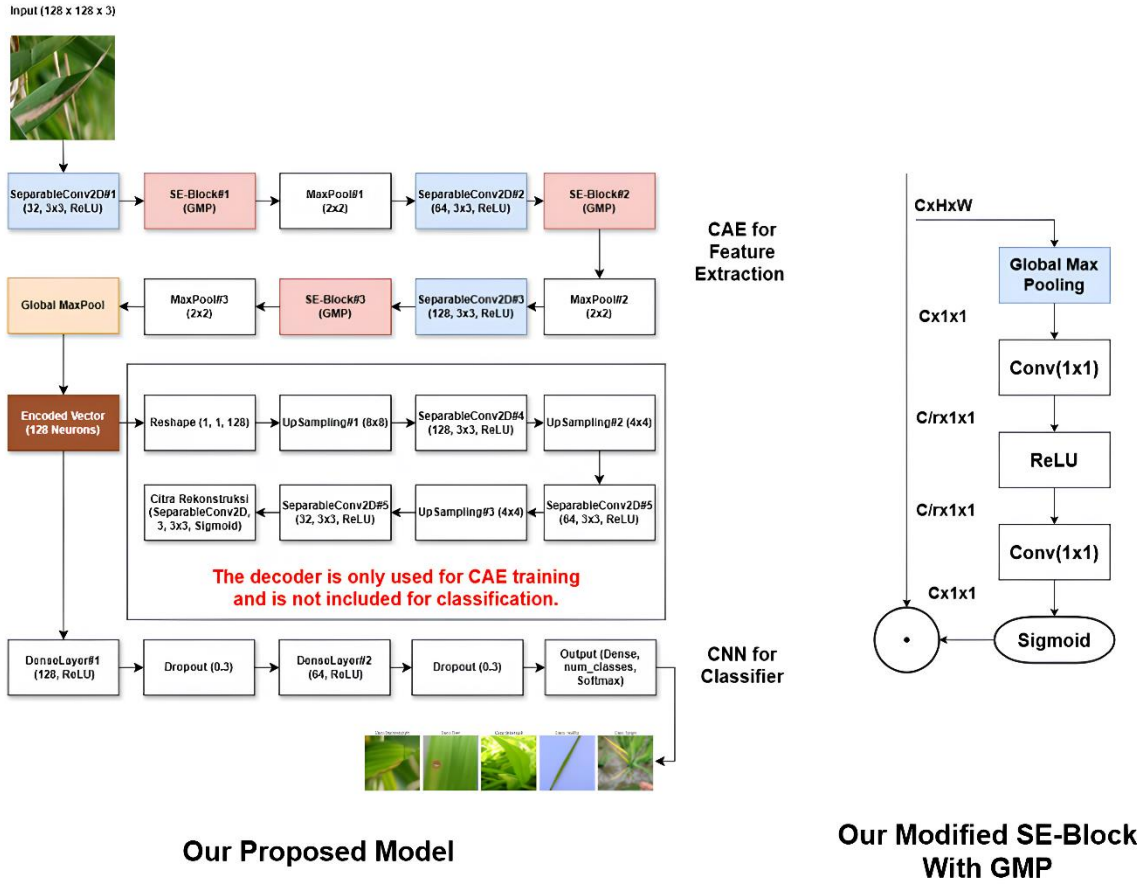


Figure 2. Architecture of the Proposed CAE-CNN and Modified SE Block with GMP

The specific function of each component in the model is described as follows.

2.3.1. CNN for Classification

A Convolutional Neural Network (CNN) is widely used for image classification because it can automatically extract spatial features such as edges, texture, and shapes from an input image [26]. A standard CNN consists of three main components, which are convolutional layers, pooling layers, and fully connected layers that progressively learn hierarchical feature representations [27]. In this study, the CNN is used as the classification component that processes the latent vector generated by the encoder. The classifier consists of two layers with 128 and 64 neurons, followed by a dropout of 0.3 to prevent overfitting [28]. The output layer uses softmax with five neurons corresponding to the five disease classes. The model is trained using the Adam optimizer and categorical cross-entropy loss.

Although previous studies have reported high accuracy in rice disease classification, DenseNet201 has achieved over 99% accuracy [29], CNN often needs a lot of computing power and can overfit when there isn't enough data [8]. By using the high-quality feature representations from the encoder, we hope that this approach can achieve a good balance between speed and accuracy while keeping the number of parameters low.

2.3.2. Convolutional Autoencoder (CAE)

A CAE is an unsupervised framework with encoder and decoder layers, designed for feature extraction while minimizing noise. By using convolutional and pooling operations, CAE maintains spatial integrity and produces compact feature representations [30]. They have proven effective in plant image analysis. For example, Boukhlifa and Chibani (2024) integrated CAE with a CNN to reduce the number of convolutional layers while preserving accuracy in tomato disease classification [10].

In this Study, the CAE is employed as the feature extraction module instead of relying solely on a standard CNN backbone. This design choice is motivated by the limitations of conventional supervised CNN models, which typically require large labeled datasets, are prone to overfitting on imbalanced or limited data, and demand high computational resources, as reported in previous studies [8], [9], [27], [31], [32]. Since the collected rice leaf images originate from multiple public datasets with varying resolutions, lighting conditions, and noise levels, an unsupervised feature extractor is advantageous. CAE learns feature representations through reconstruction, enabling it to capture domain-invariant and noise-resistant patterns without depending on class labels. Prior studies have shown that CAE can produce compact, robust, and denoised latent features that significantly reduce the parameter load of downstream CNN classifiers while retaining discriminative information [10], [30], [33]. Therefore, integrating CAE before the CNN classifier provides a more stable and efficient representation space, supporting low-resource deployment while maintaining high accuracy.

The proposed CAE design uses separable convolution as the main operation for both the encoder and decoder. The encoder has three convolutional layers because this setup has been demonstrated to work well in our previous research [9]. Each layer has 32, 64, or 128 filters with a 3x3 kernel size. After that, an SE-Block and MaxPooling2D are used to get important features while lowering the number of dimensions. Instead of flattening the collected features, GMP is used to summarize them. After that, the features are transformed into a 128-dimensional latent vector. Similar to the encoder, the decoder uses UpSampling2D and Separable Convolution not to extract the feature but to reconstruct the original input image. Its main purpose is to help the encoder learn more effective feature representations [30]. ReLU is used for the hidden layers to keep the pixel values normalized, while sigmoid is used for the output layer. The model is trained with Mean Squared Error (MSE) to make sure that the reconstructed features are very similar to the original inputs [10]. Although minor details may be lost, CAE is an efficient feature extractor [9] and their performance can be further enhanced by integrating SE-Block, GMP, and Separable Convolution.

2.3.3. Integration of SE-Block with Global Max Pooling

The Squeeze and Excitation Block (SE-Block) is a component within a CNN that adjusts the importance of features at the channel level [11]. SE-Block is divided into two steps, which are the squeeze and the excitation. The squeeze step applies Global Average Pooling (GAP), while the excitation step uses fully connected layers with ReLU and sigmoid activations to highlight informative channels [34]. SE-Blocks have shown strong results in plant disease classification, achieving over 99% accuracy when integrated into ResNet-50 [35]. These findings confirm SE-Block effectiveness in improving accuracy without increasing complexity.

In this study, SE-Block is built into each Separable Convolution layer to adjust the weights of the channels. This research substitutes GMP for the normal GAP. This change is based on the fact that rice leaf diseases often show up as small, high-contrast patches. Because of this, GMP is better than GAP at showing these kinds of limited patterns [36]. A research by Md. Sazzadul I. Prottasha. et al. (2021) found that GMP was more accurate than GAP when it came to classifying plant diseases [37]. The SE-Block with GMP works by emphasizing the most important channel properties, processing them over a tiny network, and creating weights that boost the most useful channels while lowering the less useful ones.

2.3.4. Separable Convolution

Separable Convolution is a convolutional method that splits standard convolution into two steps, which are depthwise and pointwise operations. In the depthwise step, a separate filter is applied to each input channel, followed by a 1x1 pointwise convolution to combine the features. In this study, the model uses SeparableConv2D instead of the usual Conv2D. This approach significantly reduces computational cost and the number of parameters while maintaining accuracy [13]. For example, a research by Prottasha and Reza (2022) applied Separable Convolution in their study and achieved about 95% accuracy in rice leaf disease classification, showing that this approach is both efficient and reliable [38].

2.3.5. Global Max Pooling (GMP)

In this study, GMP is used instead of Flatten layer at the end of the encoder section. GMP is a pooling method that can be used to replace the Flatten layer by selecting the maximum activation from each feature map, keeping important information while reducing the number of parameters. Unlike Flatten, which fully connects all the neurons and increases the complexity, GMP efficiently compresses the features, and helps prevent overfitting [12] and has minimal memory usage, making it well-suited for low-resource systems. Our

previous studies also show that GMP can maintain or even improve accuracy in rice leaf disease classification while reducing the number of parameters [9]. By using GMP, it will make sure that only the most useful characteristics are sent to the dense layer [12]. The result is put into a 128-dimensional latent vector.

2.4. Model Training

The training parameters for the proposed CAE-CNN models are summarized in Table 4.

Table 4. Model Training Parameters

Component	CAE (Unsupervised)	CNN (Supervised)
Optimizer	Adam	Adam
Loss Function	MSE	Categorical Crossentropy
Activation Function	ReLU (Hidden Layers), Sigmoid (Decoder)	ReLU (Hidden Layers), Softmax (Output Layer)
Early Stopping	Monitor: val_loss, Patience: 10	Monitor: val_loss, Patience: 10
Epochs	100	100
Batch Size	64	64
Input Data	Rice Leaf Images (128x128x3)	Latent Vectors (128 dimensions)

2.5. Hyperparameter Optimization

Table 5. Hyperparameter Search Space using Optuna

Parameter	Value Options	Description
Target Size	64x64, 128x128	Tests lower and higher input resolutions
Rotation Range	20, 40, 60	Rotation augmentation for data variability
Width Shift Range	0.2, 0.4, 0.6	Horizontal displacement simulation
Height Shift Range	0.2, 0.4, 0.6	Vertical displacement simulation
Zoom Range	0.2, 0.4, 0.6	Zoom in and zoom out variations
Filter 1-3	32, 64, 128, 256	Tests different convolution depths
Reduction	8, 16	Adjusts SE-Block reduction ratios
Dense 1	64, 128 256	Varies first dense layer capacity
Dense 2	32, 64, 128	Adjusts the second dense layer size
Dropout Rate	0.2, 0.3, 0.5	Tests different regularization strengths
Batch Size	32, 64, 128	Balances training stability and efficiency
Epoch	50, 100	Compares shorter and longer training durations

To obtain an optimal configuration, this study employed Optuna, an automated hyperparameter optimization framework using the Tree structured Parzen Estimator (TPE) algorithm with an integrated pruning mechanism to terminate unpromising trials early [39]. The search space included several parameters, as shown in Table 5.

Given the large number of possible combinations, at least 50 trials were conducted, following previous studies that reported this range to be sufficient for convergence [40]. The optimized results were compared with the proposed model to validate whether the proposed model achieved higher accuracy and maintained computational efficiency. In this study, Optuna functioned as a validation and refinement tool rather than a replacement for the main design. A Median Pruner was applied to speed up the search by discarding underperforming trials, while early stopping monitored validation loss to reduce computation time [41]. The primary evaluation metric was validation accuracy, complemented by efficiency measures such as encoder and classifier parameter counts.

2.6. Model Evaluation

2.6.1. Accuracy Metrics

Model performance was evaluated using Accuracy, Precision, Recall, and F1-Score, calculated per class and as macro averages to assess both overall correctness and class. After that, a confusion matrix was used to visualize classification errors [42], while accuracy and loss curves tracked learning performance during training.

2.6.2. Efficiency Metrics

Efficiency was assessed through parameter count, model size, and inference latency to estimate computational cost and deployment feasibility [43]. Visualization of predictions, including true labels, predicted labels, confidence scores, and processing times, supported a clearer interpretation of model behavior. This dual evaluation ensured the model achieved both high accuracy and computational efficiency, making it suitable for practical applications.

2.7. Ablation Study

An ablation study was conducted to evaluate the individual contribution of each component [44] within the proposed CAE-CNN architecture, including SE-Block, GMP, and Separable Convolution. Variants were created by selectively removing or replacing these modules, and all models were tested under the same

dataset, preprocessing, training protocol, and CPU-based environment. Differences in classification accuracy and computational efficiency, therefore, reflect the impact of each architectural component. The detailed results are presented in Table 7, demonstrating how each module contributes to the overall performance and guiding the final model design.

2.8. Comparative Study

In addition to the ablation study, we conducted a comparative analysis with established CNN models for rice leaf disease classification. Unlike relying on reported results from different studies, all benchmark models were evaluated under the same dataset, preprocessing, training protocol, and CPU-based environment to ensure a fair comparison. Each model was implemented to match its original architecture, and differences in accuracy, model size, parameter efficiency, and inference speed reflect solely architectural variations. Detailed quantitative results are presented in Table 8, highlighting the competitiveness and practical efficiency of the proposed model.

3. RESULTS AND DISCUSSION

3.1. Proposed Model Evaluation

3.1.1. Accuracy and Loss Graph

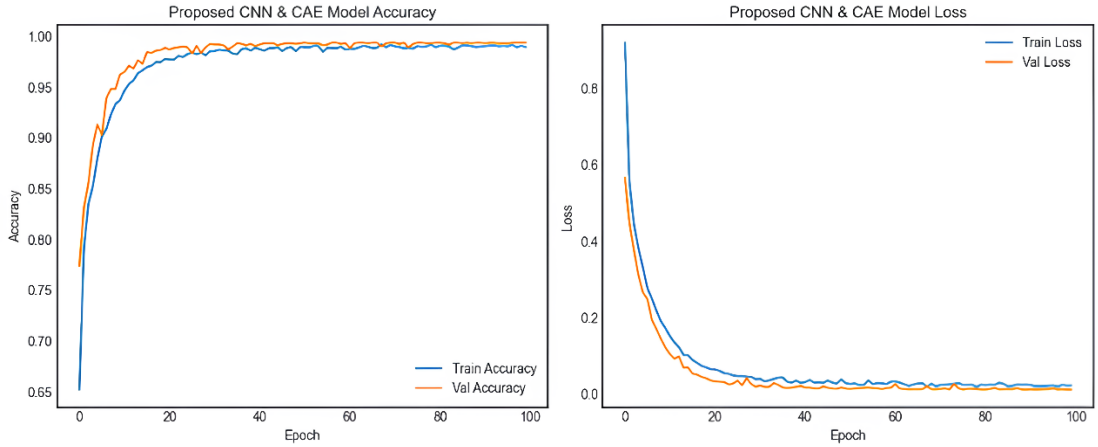


Figure 3. Accuracy and Loss Graph

The training performance of the proposed CAE-CNN model, as illustrated in Figure 3, shows a rapid increase in accuracy during the initial 20 epochs, followed by gradual stabilization up to epoch 100, achieving 98.77% training accuracy and 99.39% validation accuracy. The close alignment of both curves indicates excellent generalization and the absence of overfitting. Similarly, with the accuracy graph, the loss curve exhibits a steep decline in the early epochs and progressively converges toward near zero values, with training loss at 0.0308 and validation loss at 0.0153. These results confirm effective optimization, stable convergence, and a balanced learning process between training and validation datasets.

3.1.2. Classification Report

Table 6. Classification Report

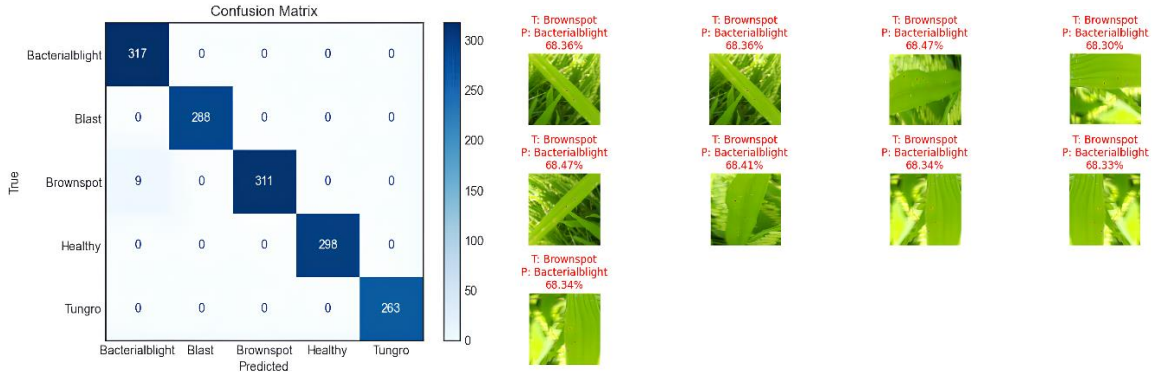
	Precision	Recall	F1-Score	Support
Bacterial Blight	0.97	1.00	0.99	317
Blast	1.00	1.00	1.00	288
Brown Spot	1.00	0.97	0.99	320
Healthy	1.00	1.00	1.00	298
Tungro	1.00	1.00	1.00	263
Accuracy			0.99	1.486
Macro Avg	0.99	0.99	0.99	1.486
Weighted Avg	0.99	0.99	0.99	1.486

The classification report in Table 6 shows that the proposed model performs very well and consistently across all five classes, reaching an overall accuracy of 99%. The classes like Blast, Healthy, and Tungro achieved perfect Precision, Recall, and F1-Score values, meaning the model could clearly distinguish these categories without any errors. A class like Bacterial Blight also produced near-perfect results, with a Precision of 0.97 and a Recall of 1.00, indicating that all diseased samples were correctly detected, although a few leaves
.

might have been slightly misclassified. For the Brown Spot class, the Recall score of 0.97 suggests a small overlap in visual features with other diseases, which occasionally caused minor misclassifications.

The Macro and Weighted Averages, both at 0.99, confirm that the model maintains strong predictive performance and balanced accuracy across all classes, even with different class sizes. Overall, these findings demonstrate that the proposed CAE-CNN model effectively captures the unique visual features of each disease, providing accurate and reliable classification results for rice leaf disease classification.

3.1.3. Confusion Matrix and Misclassified Samples



(a) (b)

Figure 4. (a) Confusion Matrix, (b) Nine Misclassified Samples

The confusion matrix in Figure 4(a) shows that the proposed model achieved very high precision in identifying the five rice leaf disease classes. All classes, like Bacterial Blight, Blast, Healthy, and Tungro, were correctly classified with no errors, indicating strong feature learning and consistent performance under the evaluated experimental conditions. A small number of misclassifications occurred in the Brown Spot class, where nine samples were labeled as Bacterial Blight. This confusion likely happened because of their visual similarity, which is that Brown Spot usually shows dark brown circular lesions, while Bacterial Blight produces yellowish brown streaks with uneven edges. When the Brown Spot lesions are very small or faint, or when lighting conditions cause color distortion, their appearance can resemble the early stage of Bacterial Blight. These nine misclassified samples are shown in Figure 4(b). As shown in Figure 4(b), most samples display overlapping color tones and lesion textures that could confuse even human observers. The model prediction confidence for these samples was moderate, around 68%, indicating that these errors were caused by visual ambiguity rather than model design flaws. Even so, the relatively low number of misclassifications demonstrates the model's reliability and its improved ability to distinguish subtle differences between diseases based on texture and color features.

3.1.4. Prediction Visualization

Figure 5. Prediction Visualization

The prediction visualization in Figure 5 illustrates that the proposed model achieved highly confident and accurate classifications across all five rice leaf disease classes. Each image displays the true label (T) and the predicted label (P), with green text signifying correct predictions. All samples were classified correctly, with confidence levels reaching nearly 100%, confirming the model stability and strong discriminative capability. The consistent performance across diverse conditions, ranging from clear to slightly varied lighting and texture, demonstrates that the model can effectively capture distinct visual patterns such as color intensity, lesion shape, and leaf texture. The absence of misclassification in this visualization supports the quantitative evaluation results, reinforcing that the model is not only accurate in controlled testing but also reliable for real-time disease classification.

3.1.5. Parameter Count, Model Size, and Inference Time

The proposed CAE-CNN model achieved 99.39% accuracy, demonstrating excellent precision in classifying rice leaf conditions. The model is compact and efficient, with a total of 85,859 parameters and a file size of only 0.28 MB, making it lightweight and suitable for devices with limited memory. Inference time is also very fast, at approximately 0.0665 milliseconds per image, corresponding to a throughput of over 15,000 images per second. This combination of high accuracy, small size, and rapid inference highlights the model's effectiveness and practical suitability for real-time field applications in rice disease detection.

3.2. Hyperparameter Optimization Results

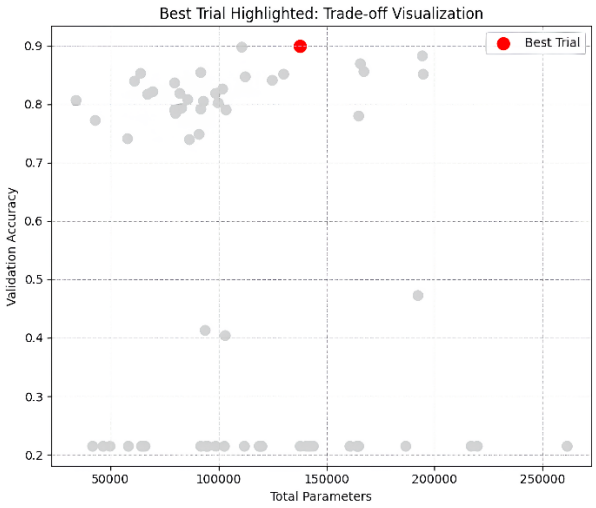


Figure 6. Scatter Plot Trade-off Visualization Optuna

From a total of 134 Optuna trials, the optimization process identified Trial 3 as the best-performing configuration, achieving a validation accuracy of 89.9%. This configuration used a batch size of 128, 100 epochs, dense layer units of 256 and 128, a dropout rate of 0.3, three convolutional layers with 64, 128, and 64 filters, an SE reduction ratio of 16, and an input size of 128×128. Data augmentation included rotation up to 20°, width shift up to 0.2, height shift up to 0.4, and zoom up to 0.4. The total number of parameters for Trial 3 was 137,632.

The scatter plot in Figure 6, illustrates the trade-off between validation accuracy and model complexity. Several configurations reached 85 to 88% accuracy, but many required more parameters or exhibited unstable training behavior. Trial 3 achieved a good balance, maintaining stable learning with reasonable computational cost.

However, when compared to the proposed CAE–CNN model, which achieved 99.39% validation accuracy with only 85,859 parameters, approximately 37% fewer than Trial 3, it is evident that the proposed model already provides an excellent balance between accuracy and efficiency. Optuna served as a challenger, testing 134 architecture combinations using the TPE method. Despite this extensive search, none of the generated models surpassed the proposed model in terms of accuracy or compactness, confirming the strength and efficiency of the CAE–CNN design.

.

3.3. Ablation Study

Table 7. Comparison of Accuracy and Model Complexity Across an Ablation Study

Configuration	Accuracy	Total Parameters	Model Size .h5 (MB)	Inference Time (ms/image)	FPS
Proposed Model	99.39%	85,859	0.28	0.0665	15,213.15
-SE, +GMP, +SC	99.39%	83,651	0.24	0.0707	14,144.61
+SE, -GMP, +SC	99.06%	4,263,779	16.22	0.0817	12,245.54
+SE, +GMP, -SC	99.39%	378,248	0.59	0.0687	14,567.88
-SE, -GMP, +SC	99.33%	4,261,091	16.18	0.0996	10,041.37
-SE, +GMP, -SC	99.39%	375,560	0.55	0.0763	13,098.58
+SE, -GMP, -SC	98.86%	4,556,168	16.52	0.0848	11,793.75
CAE+CNN	99.06%	4,553,480	16.49	0.0755	13,245.08
CNN	99.26%	4,296,261	16.39	1.5057	664.15
Proposed Model with GAP as the Flatten	99.06%	85,859	0.28	0.0694	14,410.22
Proposed Model (SE-Block with GAP)	98.59%	85,859	0.28	0.0770	12,985.30

Based on Table 7, the ablation study was performed to evaluate how SE-Block, GMP, and SC affect model performance and efficiency. The proposed model achieved the best balance, achieving 99.39% accuracy, a compact 0.28 MB model size, and the fastest inference at 0.06657 ms/image with 15,213 FPS. Despite having the second smallest parameter count, it consistently outperformed larger ablation variants.

Besides that, removing the SE-Block had minimal impact on accuracy, but slightly reduced the inference speed, suggesting that SE contributes moderately to feature refinement while adding negligible overhead. On the other hand, removing either GMP or SC resulted in massive increases in parameter count, reaching over 4 million parameters and more than 16 MB, with no accuracy benefit. Latency also increased significantly, from around 0.0817 to 0.0996 ms/image, confirming that both GMP and SC are essential for keeping the model lightweight. In addition, the CAE+CNN baseline achieved strong accuracy but remained computationally heavy, while the CNN-only model performed worst in speed, requiring 1.5057 ms/image and achieving only 664 FPS, far below the lightweight variants.

Further experiments like replacing GMP with GAP reveal that while both pooling types maintain identical parameter counts, GMP consistently offers higher accuracy and faster inference, reinforcing its suitability for capturing localized high contrast features crucial for rice disease identification. Overall, these findings confirm that the proposed architectural combination provides the optimal trade-off between performance, model compactness, and real-time inference capability.

3.4. Comparative Analysis with Related Works

Table 8. The Models Used as Benchmarks for Comparison Rice Leaf Disease Classification with CNN

Source	Algorithm	Accuracy	Parameter (M)	Model Size .h5(MB)	Inference Time (ms/image)	FPS
[29]	DenseNet201	99.39%	18.82	77.46	16.0225	62.41
	EfficientNet-B0	21.53%	4.23	18.18	6.0196	166.12
[32]	MobileNetV2	99.26%	2.44	11.18	2.6081	383.42
	ShuffleNetV2	99.06%	1.42	5.35	1.3273	753.42
[45]	Customized CNN + Separable Convolution	99.26%	0.15	1.77	0.8585	1,164.84
[46]	dCNN	99.26%	0.24	0.91	0.7895	1,266.62
[47]	ImShuffleNet + SE-Block	96.97%	0.36	1.38	18.322	54.58
[48]	EfficientNet B3 + SE-Block	79.07%	10.7	40.83	7.8927	126.70
[49]	LiSA-MobileNetV2+Swish+SE-Block+Separable Convolution	98.86%	0.57	2.17	2.6540	376.79
[9]	Our Previous Research with CAE-CNN & GMP	99.39%	0.375	0.55	0.0763	13,098.58
	Proposed Model	99.39%	0.086	0.28	0.0657	15,213.15

Table 8 above compares the proposed CAE–CNN model with representative CNN-based architectures for rice leaf disease classification under the same dataset, preprocessing pipeline, and CPU-based computational environment. The proposed model achieves a high accuracy of 99.39%, comparable to large-scale architectures such as DenseNet201, while outperforming several lightweight models, including dCNN and ImShuffleNet + SE-Block. These results demonstrate that competitive performance can be achieved without relying on deep or heavy parameter networks when feature extraction and channel-wise attention are carefully optimized.

In terms of efficiency, the proposed model contains only 0.086 M parameters with a compact model size of 0.28 MB, achieving the lowest inference latency with only 0.0657 ms/image and the highest throughput with 15,213 FPS among all evaluated models. Compared with the author's previous CAE–CNN with GMP model (0.375 M parameters, 0.0763 ms/image, 13,098 FPS), the proposed design reduces the parameter count by approximately 77%, decreases inference latency by about 14%, and increases throughput by roughly 16%, while maintaining the same classification accuracy with 99.39%. These improvements highlight the effectiveness of integrating SE-Blocks and Separable Convolution to enhance discriminative feature learning and computational efficiency. Overall, the proposed model achieves a well-balanced trade-off between accuracy and efficiency, making it highly suitable for real time rice leaf disease detection in resource-constrained agricultural environments.

4. CONCLUSION

This study developed a compact and efficient CAE–CNN model for rice leaf disease classification by integrating a CAE with SE-Block, GMP, and Separable Convolution. The experimental results showed that the model achieved 99.39% accuracy with only 85,859 parameters and a model size of 0.28 MB, while keeping the inference time very low, around 0.0657 ms/image with 15,213 FPS. These findings demonstrate a strong balance between accuracy, speed, and efficiency. The ablation study shows that GMP and Separable Convolution play a crucial role in keeping the model lightweight and fast. Removing either component led to a significant increase in parameter count, model size, and inference time, without providing any improvement in accuracy. In contrast, the SE-Block provides only a small improvement in feature representation with minimal computational cost. Although the impact of SE-Block on overall accuracy is minor, it slightly enhances the model's ability to focus on important features without adding significant overhead. Moreover, when compared with larger networks under identical experimental settings with models such as DenseNet201, ImpShuffleNet, and MobileViTV2, the proposed model achieves higher accuracy with far fewer parameters, making it ideal for real-time and resource-limited agricultural systems. In addition, compared with our previous CAE–CNN with GMP, it reduces parameters by 77%, lowers latency by 14%, and increases throughput by 16%.

By addressing the limitations of conventional CNNs, this study demonstrates that careful model design can both advance scientific understanding of efficient CNN architectures and provide a practical solution for real-time disease monitoring in resource-limited agricultural systems. However, the model's ability to generalize is still limited, as cross-validation and testing on diverse external datasets were not performed, which may affect performance on unseen data. Moreover, although misclassification between visually similar diseases such as Brown Spot and Bacterial Blight was reduced, it was not completely eliminated, indicating that fine-grained discrimination under challenging visual conditions remains an open research problem. Future work will focus on validating the model on real devices, evaluating its performance under field conditions, and integrating it with camera-based systems to enable automated and continuous monitoring in support of intelligent and sustainable precision agriculture.

CONFLICT OF INTEREST STATEMENT

The Authors state no conflict of interest.

REFERENCES

[1] Syahri and R. U. Somantri, "Learning from global research to analyze contributing factors in rice yield gap: Bibliometric approach towards Indonesia's self-sufficiency," *Agriculture and Natural Resources*, vol. 57, no. 3, pp. 479–490, May 2023, doi: 10.34044/j.anres.2023.57.3.12.

[2] P. W. Titisari *et al.*, "The role of Kampar watershed in achieving sufficient rice production and sustaining agriculture," *Water Supply*, vol. 24, no. 2, pp. 480–496, Feb. 2024, doi: 10.2166/ws.2024.016.

[3] M. A. Azim, M. K. Islam, M. M. Rahman, and F. Jahan, "An effective feature extraction method for rice leaf disease classification," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 19, no. 2, pp. 463–470, Apr. 2021, doi: 10.12928/TELKOMNIKA.v19i2.16488.

[4] Z. Jun, "The Development and Application of Support Vector Machine," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Jan. 2021. doi: 10.1088/1742-6596/1748/5/052006.

[5] P. Cunningham and S. J. Delany, "K-Nearest Neighbour Classifiers-A Tutorial," *ACM Comput Surv*, vol. 54, no. 6, 2021, doi: 10.1145/3459665.

[6] E. Y. Boateng, J. Otoo, and D. A. Abaye, "Basic Tenets of Classification Algorithms K-Nearest-Neighbor, Support Vector Machine, Random Forest and Neural Network: A Review," *Journal of Data Analysis and Information Processing*, vol. 08, pp. 341–357, 2020, doi: 10.4236/jdaip.2020.84020.

[7] M. Bakr, S. Abdel-Gaber, M. Nasr, and M. Hazman, "DenseNet Based Model for Plant Diseases Diagnosis," *European Journal of Electrical Engineering and Computer Science*, vol. 6, no. 5, pp. 1–9, Sep. 2022, doi: 10.24018/ejece.2022.6.5.458.

[8] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J Big Data*, vol. 8, no. 53, 2021, doi: 10.1186/s40537-021-00444-8.

[9] D. Suhada, I. G. P. S. Wijaya, I. B. K. Widiartha, and M. Jo, "A Lightweight CAE-CNN Model with Global Max Pooling for Efficient Rice Leaf Disease Classification." *Accepted for publication in International Conference on Informatics and Computing (ICIC)*, 2025.

.

[10] G. Boukhlifa and Y. Chibani, "A Lightweight CNN Design Based on Convolutional Autoencoder for Tomato Disease Identification," in *Proceedings - 8th IEEE International Conference on Image and Signal Processing and their Applications, ISPA 2024*, Institute of Electrical and Electronics Engineers Inc., 2024. doi: 10.1109/ISPA59904.2024.10536761.
[11] M. Rouvier and P.-M. Bousquet, "Studying squeeze-and-excitation used in CNN for speaker verification," *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 1110–1115, Sep. 2021, doi: 10.1109/ASRU51503.2021.9687936.
[12] I. B. Venkateswarlu, J. Kakarla, and S. Prakash, "Face mask detection using MobileNet and Global Pooling Block," in *2020 IEEE 4th Conference on Information & Communication Technology (CICT)*, Chennai: IEEE, 2020, pp. 1–5. doi: 10.1109/CICT51604.2020.9312083.
[13] W. Sun, X. Zhang, and X. He, "Lightweight image classifier using dilated and depthwise separable convolutions," *Journal of Cloud Computing: Advances, Systems and Applications*, vol. 9, no. 55, Dec. 2020, doi: 10.1186/s13677-020-00203-9.
[14] guy007, "Kaggle-Nutrient Deficiency Symptoms in Rice ," https://www.kaggle.com/datasets/guy007/nutrientdeficiencysymptomsinrice. [Online]. Available: https://www.kaggle.com/datasets/guy007/ nutrientdeficiencysymptomsinrice
[15] nurnob101, "Kaggle-Rice Disease," https://www.kaggle.com/datasets/nurnob101/rice-disease. [Online]. Available: https://www.kaggle.com/datasets/nurnob101/ rice-disease
[16] minhhuy281, "Kaggle-Rice Disease Image Dataset," https://www.kaggle.com/datasets/minhhuy2810/rice-diseases-image-dataset. [Online]. Available: https://www.kaggle.com/datasets/minhhuy2810/rice-diseases-image-dataset
[17] vbookshelf, "Kaggle-Rice Leaf Disease," https://www.kaggle.com/datasets/vbookshelf/rice-leaf-diseases. [Online]. Available: https://www.kaggle.com/datasets/vbookshelf/ rice-leaf-diseases
[18] Y. hui Zhang, L. Tang, X. jun Liu, L. lei Liu, W. xing Cao, and Y. Zhu, "Modeling dynamics of leaf color based on RGB value in rice," *J Integr Agric*, vol. 13, no. 4, pp. 749–759, 2014, doi: 10.1016/S2095-3119(13)60391-3.
[19] A. W. O. Gama, P. V. J. A. Gunawan, and K. Darmaastawan, "HOG Feature Extraction in Optimizing FK-NN and CNN for Image Identification of Rice Plant Diseases," *Journal of Applied Data Sciences*, vol. 6, no. 3, pp. 1611–1626, Sep. 2025, doi: 10.47738/jads.v6i3.722.
[20] A. Batool and Y. C. Byun, "Lightweight EfficientNetB3 Model Based on Depthwise Separable Convolutions for Enhancing Classification of Leukemia White Blood Cell Images," *IEEE Access*, vol. 11, pp. 37203–37215, 2023, doi: 10.1109/ACCESS.2023.3266511.
[21] L. Jin, J. Yu, X. Yuan, and X. Du, "Fish classification using dna barcode sequences through deep learning method," *Symmetry (Basel)*, vol. 13, no. 9, Sep. 2021, doi: 10.3390/sym13091599.
[22] A. Akhtarshenas and R. Toosi, "An open-set framework for underwater image classification using autoencoders," *SN Appl Sci*, vol. 4, no. 8, Aug. 2022, doi: 10.1007/s42452-022-05105-w.
[23] M. S. I. Sajol, S. T. Alvi, and C. A. A. Era, "Performance Assessment of Advanced CNN and Transformer Architectures in Skin Cancer Detection," in *2024 11th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, Aug. 2024. doi: 10.1109/EECSI63442.2024.10776508.
[24] D. Verma, C. Bose, N. Tufchi, K. Pant, V. Tripathi, and A. Thapliyal, "An efficient framework for identification of Tuberculosis and Pneumonia in chest X-ray images using Neural Network," in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 217–224. doi: 10.1016/j.procs.2020.04.023.
[25] J. Park and J. Jeong, "Assessment of the effectiveness of a convolutional autoencoder for digital image-based automated core logging," *Geoenergy Science and Engineering*, vol. 227, 2023, doi: doi.org/10.1016/j.geoen.2023.211802.
[26] M. Idhom, D. A. Prasetya, P. A. Riyantoko, T. M. Fahrudin, and A. P. Sari, "Pneumonia Classification Utilizing VGG-16 Architecture and Convolutional Neural Network Algorithm for Imbalanced Datasets," *TIERS Information Technology Journal*, vol. 4, no. 1, pp. 73–82, Jun. 2023, doi: 10.38043/tiers.v4i1.4380.
[27] A. A. Elngar *et al.*, "Image Classification Based On CNN: A Survey," *Journal of Cybersecurity and Information Management (JCIM)*, vol. 6, no. 1, pp. 18–50, 2021, doi: 10.5281/zenodo.4897990.
[28] A. Karunanithi, A. S. Singh, and T. Kannapiran, "Enhanced Hybrid Neural Networks (CoAtNet) for Paddy Crops Disease Detection and Classification," *Revue d'Intelligence Artificielle*, vol. 36, no. 5, pp. 671–679, Oct. 2022, doi: 10.18280/ria.360503.
[29] M. N. Azmi, M. A. Rifqi, and I. G. P. S. Wijaya, "Detection System for Rice Leaf Diseases Using Convolutional Neural Network Method and Android Application-based to Improve Rice Production Efficiency," *accepted for publication in The 9th Internacional Conference on Science and Technology (ICST)*, 2024.
[30] E. Pintelas, I. E. Livieris, and P. E. Pintelas, "A convolutional autoencoder topology for classification in high-dimensional noisy image datasets," *Sensors*, vol. 21, no. 22, Nov. 2021, doi: 10.3390/s21227731.
[31] C. Li, H. Li, Z. Liu, B. Li, and Y. Huang, "SeedSortNet: a rapid and highly effificient lightweight CNN based on visual attention for seed sorting," *PeerJ Comput Sci*, vol. 7, pp. 1–21, 2021, doi: 10.7717/peerj-cs.639.
[32] K. Saddami, Y. Nurdin, M. Zahramita, and M. Shahreeza Safiruz, "Advancing Green AI: Efficient and Accurate Lightweight CNNs for Rice Leaf Disease Identification," 2024. doi: https://doi.org/10.48550/arXiv.2408.01752.
[33] P. Bedi and P. Gole, "Plant disease detection using hybrid model based on convolutional autoencoder and convolutional neural network," *Artificial Intelligence in Agriculture*, vol. 5, pp. 90–101, Jan. 2021, doi: 10.1016/j.aiia.2021.05.002.
[34] J. Hua, J. Zou, J. Rao, H. Yin, and J. Chen, "ECG Signals Deep Compressive Sensing Framework Based on Multiscale Feature Fusion and SE Block," *IEEE Access*, vol. 11, pp. 104359–104372, 2023, doi: 10.1109/ACCESS.2023.3316487.
[35] R. Deng *et al.*, "Automatic Diagnosis of Rice Diseases Using Deep Learning," *Front Plant Sci*, vol. 12, Aug. 2021, doi: 10.3389/fpls.2021.701038.
[36] Y. Dogan, "A New Global Pooling Method for Deep Neural Networks: Global Average of Top-K Max- Pooling," *Traitement du Signal*, vol. 40, no. 2, pp. 577–587, Apr. 2023, doi: 10.18280/ts.400216.
[37] M. S. I. Prottasha, Z. Tasnim, S. M. S. Reza, and D. A. Hossain, "A Lightweight CNN Architecture to Identify Various Rice Plant Diseases in Bangladesh," in *2021 International Conference on Information and Communication Technology for Sustainable Development, ICICT4SD 2021 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., Feb. 2021, pp. 316–320. doi: 10.1109/ICICT4SD50815.2021.9396927.
[38] S. I. Prottasha and S. M. S. Reza, "A classification model based on depthwise separable convolutional neural network to identify rice plant diseases," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 4, pp. 3642–3654, Aug. 2022, doi: 10.11591/ijece.v12i4.pp3642-3654.
[39] S. Watanabe, "Tree-Structured Parzen Estimator: Understanding Its Algorithm Components and Their Roles for Better Empirical Performance," May 2023, [Online]. Available: http://arxiv.org/abs/2304.11127
[40] J. Parra-Ullauri, X. Zhang, A. Bravalheri, R. Nejabati, and D. Simeonidou, "Federated Hyperparameter Optimisation with Flower and Optuna," in *Proceedings of the ACM Symposium on Applied Computing*, Association for Computing Machinery, Mar. 2023, pp. 1209–1216. doi: 10.1145/3555776.3577847.

[41] Y. Lin, R. Palaniappan, P. De Wilde, and L. Li, "Reliability Analysis for Finger Movement Recognition With Raw Electromyographic Signal by Evidential Convolutional Networks," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 30, pp. 96–107, 2022, doi: 10.1109/TNSRE.2022.3141593.
[42] S. M. H. Mirsadeghi, H. Bahsi, R. Vaarandi, and W. Inoubli, "Learning From Few Cyber-Attacks: Addressing the Class Imbalance Problem in Machine Learning-Based Intrusion Detection in Software-Defined Networking," *IEEE Access*, vol. 11, pp. 140428–140442, 2023, doi: 10.1109/ACCESS.2023.3341755.
[43] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design," in *Proceedings of the European conference on computer vision (ECCV)*, 2018.
[44] S. Sheikholeslami, M. Meister, T. Wang, A. H. Payberah, V. Vlassov, and J. Dowling, "AutoAblation: Automated Parallel Ablation Studies for Deep Learning," in *Proceedings of the 1st Workshop on Machine Learning and Systems, EuroMLSys 2021*, Association for Computing Machinery, Inc, Apr. 2021, pp. 55–61. doi: 10.1145/3437984.3458834.
[45] Md. S. I. Prottasha, A. B. M. K. Hossain, Md. Z. Rahman, S. M. S. Reza, and D. A. Hossain, "Identification of Various Rice Plant Diseases Using Optimized Convolutional Neural Network," *International Journal of Computing and Digital Systems*, vol. 12, no. 7, pp. 1539–1551, Dec. 2022, doi: 10.12785/ijcds/1201124.
[46] M. H. Bijoy *et al.*, "Towards Sustainable Agriculture: A Novel Approach for Rice Leaf Disease Detection Using dCNN and Enhanced Dataset," *IEEE Access*, vol. 12, pp. 34174–34191, 2024, doi: 10.1109/ACCESS.2024.3371511.
[47] R. Dubey, R. Kaur, A. Anand, and S. Saxena, "Rice Crop Disease Forecasting Using Tuned Deep Learning Approach," in *Seventh International Conference on Recent Trends in Image Processing and Pattern Recognition (RTIP2R-2024)*, Elsevier B.V., 2025, pp. 325–331. doi: 10.1016/j.procs.2025.03.208.
[48] N. Thai-Nghe, N. T. Tri, and N. H. Hoa, "Deep Learning for Rice Leaf Disease Detection in Smart Agriculture," in *Lecture Notes on Data Engineering and Communications Technologies*, vol. 124, Springer Science and Business Media Deutschland GmbH, 2022, pp. 659–670. doi: 10.1007/978-3-030-97610-1_52.
[49] Y. Xu, D. Li, C. Li, Z. Yuan, and Z. Dai, "LiSA-MobileNetV2: an extremely lightweight deep learning model with Swish activation and attention mechanism for accurate rice disease classification," *Front Plant Sci*, vol. 16, 2025, doi: 10.3389/fpls.2025.1619365.

.

BIOGRAPHIES OF AUTHORS

Destia Suhada is currently pursuing a Master's degree in Information Technology at the Faculty of Engineering, University of Mataram, Indonesia. Her academic and research interests include artificial intelligence, computer vision, information systems, and educational technology. She has been actively involved in research and academic conferences, including The 3rd MIMSE and The 9th ICST, where she received the Best Presenter Award. Her current research focuses on developing a lightweight CAE–CNN model for efficient rice disease classification, highlighting her dedication to advancing intelligent and sustainable digital solutions. She can be contacted at email: destiasuhada96@gmail.com.

I Gede Pasek Suta Wijaya is a full-time lecturer in the Informatics Engineering Study Program, Faculty of Engineering, University of Mataram. He earned his Master's degree in Information Engineering from Gadjah Mada University and his Doctorate from Kumamoto University, Japan. His research interests include artificial intelligence, digital image processing, machine learning, and intelligent systems. In addition to teaching core informatics courses, he actively supervises students, engages in applied research, and contributes to the development of outcome-based curricula. He is also actively involved in national and international research collaborations and scientific publications. He can be contacted at email: gpsutawijaya@unram.ac.id.

Ida Bagus Ketut Widiartha is a Lecturer in Informatics Engineering at the University of Mataram. He earned his Ph.D. from Seoul National University in Technology Management, Economics and Policy with an integrated focus on Smart City Digital Convergence, an M.T. in Information Technology from Universitas Gadjah Mada, and a B.Eng. in Electrical Engineering from Universitas Udayana. His teaching and research span smart city systems, IoT, machine learning for image recognition, information systems, and software quality. Earlier Scopus-indexed studies addressed building-crack classification and pornographic image recognition (2020–2019). He can be contacted at email: widi@unram.ac.id.

Prof. Minho Jo is a Full Professor at Korea University, Sejong Campus, and the Director of the IoT.AI Laboratory. He is internationally recognized as an expert in the fields of IoT, AI, blockchain, and intelligent wireless communications. His research focuses on network security, edge and cloud computing, autonomous vehicles, and sustainable communication systems powered by wireless energy harvesting. Prof. Jo has published over 120 papers in prestigious international journals and actively serves as an editor and reviewer for several IEEE publications. His work continues to shape innovation toward the realization of sustainable, AI-driven connectivity in the emerging 5G/6G era and the future smart society. He can be contacted at email: minhojo@korea.ac.kr.