Vol. 4 No. (1) (June 2026) 239-255

Journal of Data Insights

http://journalnew.unimus.ac.id/index.php/jodi

EfficientNetV2-B0 Hyperparameter Tuning for Rice Leaf Disease Classification in Malang Regency

Mohammad Idhom*[1], Trimono Trimono[2], Indah Manfaati Nur[3]
[1,2]Data Science Study Program, UPN Veteran Jawa Timur, Indonesia
[3]Statistics Study Program, Universitas Muhammadiyah Semarang, Indonesia

DOI: https://doi.org/10.26714/jodi.v4i1.1349

Article Info

Article history:
Received June 04, 2026
Revised June 25, 2026
Accepted June 27, 2026

Keywords:
Deep learning; EfficientNetV2-B0; Hyperparameter tuning; Image classification; Rice leaf diseases; Smart farming

Abstract

Rice is a strategic food commodity that plays a crucial role in maintaining food security; however, leaf diseases can reduce productivity and are difficult to identify manually. This study aims to develop a rice leaf disease classification model using EfficientNetV2-B0 through transfer learning and hyperparameter optimization. The research dataset consists of Kaggle images and images collected directly from rice fields in Malang Regency, East Java, especially in Lawang region. The model was developed to classify six categories: blast, brown spot, leaf blight, tungro, healthy leaves, and non-rice images. The research process included image preprocessing, data augmentation, model development, comparison of five optimizers, and classification performance evaluation. Experimental results show that Adam achieved the highest validation accuracy (94.24%), outperforming Adamax, Nadam, SGD, and RMSprop. In the final evaluation, the model achieved 95% accuracy. We also used Grad-CAM visualization to identify the specific image regions that contributed to the model's predictions. The findings suggest that EfficientNetV2-B0 can be applied for automated classification of rice leaf diseases. Nevertheless, the model's ability to recognize non-rice images requires further improvement to support more reliable implementation in smart agriculture.

✉ Correspondence Address:
E-mail: idhom@upnjatim.ac.id

e-ISSN: 2988 - 2109

*This work is an open access article licensed under a **CC BY 4.0** International License.*

INTRODUCTION

Rice (Oryza sativa) is one of the world's major grain commodities, ranking third after maize and wheat (1). Rice accounts for approximately 20% of the global caloric supply, making it an essential food source for a large proportion of the world's population (2). Rice demand is expected to continue rising as the global population is projected to reach 9.7 billion by 2050 (3). Meeting this growing demand poses a significant challenge for agricultural production, especially because shrinking agricultural land, climate change, and pest and disease outbreaks constrain rice cultivation(4). These challenges underscore the importance of maintaining rice productivity to support food security and socioeconomic stability, especially in countries where rice is a staple food (5). Therefore, improving rice production requires appropriate cultivation practices and effective strategies to identify and manage factors that may reduce crop yields.

In Indonesia, rice production plays an important role in meeting domestic food needs and supporting the livelihoods of farming communities. Maintaining rice productivity requires attention not only to harvest quantity but also to quality. However, rice cultivation faces several challenges, including changing environmental conditions, pest infestations, plant diseases, and limited access to modern agricultural technologies (6). Among these challenges, plant diseases are a major concern because they can interfere with plant growth and reduce potential yields. Rice leaf diseases may cause symptoms such as spots, discoloration, and lesions. However, different diseases can produce similar visual symptoms, making identification through direct observation difficult, particularly for individuals with limited experience in plant disease diagnosis (7). Inaccurate identification may result in inappropriate disease management, potentially increasing the risk of crop losses. Consequently, developing technology-assisted approaches to identify rice leaf diseases is important to support more consistent crop monitoring and informed agricultural decision-making. Adopting precision agriculture and smart farming technologies offers opportunities to improve agricultural monitoring and disease management (8).

Advances in artificial intelligence (AI) have created opportunities to automate image analysis and support plant disease identification. One approach within this field is deep learning, a branch of machine learning that uses multilayer neural networks to learn complex patterns and representations from data (9). Unlike approaches that rely heavily on manually engineered features, deep learning models can learn relevant representations directly from training data. This capability is particularly useful for image classification, where category differences may involve variations in color, texture, shape, and spatial patterns. Deep learning has been applied to various classification tasks, demonstrating its potential to extract meaningful features from complex data (10). In agriculture, image-based classification can help identify healthy leaves and leaves showing disease symptoms. However, the effectiveness of these systems depends on several factors, including dataset characteristics, model architecture, training configuration, and evaluation procedures.

Convolutional neural networks (CNNs) are widely used for image classification because convolutional operations let models learn spatial features from images. Through successive layers, CNNs can represent visual information at different levels of complexity, from basic patterns to more distinctive features. This capability makes CNN-based approaches relevant to rice leaf disease classification, where differences between disease categories may be subtle. However, selecting an appropriate architecture matters because classification performance and computational requirements vary across models. EfficientNetV2 is a candidate architecture for this task. It incorporates Fused-MBConv blocks in selected network stages and employs a progressive learning strategy that adjusts image resolution and regularization during training. These characteristics support investigating EfficientNetV2 for rice leaf disease classification. Nevertheless, its suitability for this task must be established empirically by evaluating its performance on the target dataset.

Beyond architecture selection, training configuration can also influence deep learning model performance. Hyperparameters are settings specified before or during training, rather than parameters learned directly as model weights. Common examples include the learning rate, batch

size, number of epochs, and optimizer selection. Inappropriate configurations may affect a model's ability to learn useful image representations and achieve satisfactory classification performance. Hyperparameter tuning therefore identifies suitable setting combinations for a given learning task (11). However, the effects of tuning depend on the dataset, model architecture, and optimization strategy. A systematic comparison between baseline and tuned configurations is therefore necessary to determine whether hyperparameter tuning measurably improves classification performance. To ensure a fair comparison, evaluate both configurations using consistent dataset partitions and evaluation metrics.

Previous studies have explored applying deep learning to plant leaf disease classification. Krishnaveni and colleagues examined rice leaf disease classification using a convolutional neural network (CNN) and compared its performance with that of a support vector machine (SVM). According to the findings summarized in this study's background, the CNN approach achieved higher classification accuracy than the SVM, indicating the potential of deep learning for rice leaf disease classification. Yasid and colleagues investigated rice leaf disease classification using EfficientNetB3 with transfer learning, considering disease categories such as brown spot, hispa, and blast. Their work demonstrates how EfficientNet can be applied to rice leaf image classification.

Meanwhile, Singla and Gupta investigated EfficientNetV2-S for classifying bean leaf images into three categories: healthy, angular leaf spot, and bean rust. Although their study focused on a different plant species, it provides a relevant example of EfficientNetV2-S applied to leaf image classification. These studies demonstrate the potential of CNNs and EfficientNet architectures for plant disease classification. However, differences in plant species, disease categories, datasets, and experimental configurations mean that their findings cannot directly establish the effect of hyperparameter tuning on EfficientNetV2 for the rice leaf classification task investigated in this study.

Despite previous research on CNN-based rice leaf disease classification and EfficientNet architectures, further evaluation is needed to determine how hyperparameter tuning affects EfficientNetV2 performance for classifying five rice leaf categories. Therefore, this study implements EfficientNetV2 using transfer learning and compares its classification performance before and after hyperparameter tuning. The dataset consists of primary images collected from rice fields in Malang Regency, East Java, and secondary images obtained from Kaggle. The classification task includes five categories: healthy rice leaves, blast, bacterial leaf blight, brown spot, and tungro. The images are represented in RGB format and undergo preprocessing to ensure a consistent input size before model training. The study then evaluates the baseline and tuned configurations using classification metrics to assess performance. This comparison empirically assesses the effect of hyperparameter tuning on EfficientNetV2 for the specified rice leaf classification task. The findings are expected to contribute to research on deep learning-based plant disease classification and provide a basis for the further development of image-based systems to support rice crop monitoring.

METHOD

This study applies EfficientNetV2, with hyperparameter tuning, to rice leaf disease classification. The proposed methodology includes dataset acquisition, image labeling, data cleaning, image preprocessing, data augmentation, dataset splitting, EfficientNetV2 model training, optimizer selection, model fine-tuning, and model evaluation. The dataset comprises primary images collected from rice fields in Malang and a secondary dataset from Kaggle. The image preparation stage includes image resizing and the preprocessing procedures implemented in the study. We then train EfficientNetV2 using transfer learning, followed by an optimizer comparison to identify the most suitable configuration. We then train the selected model and evaluate it using classification performance metrics. Figure 1 illustrates the research workflow.

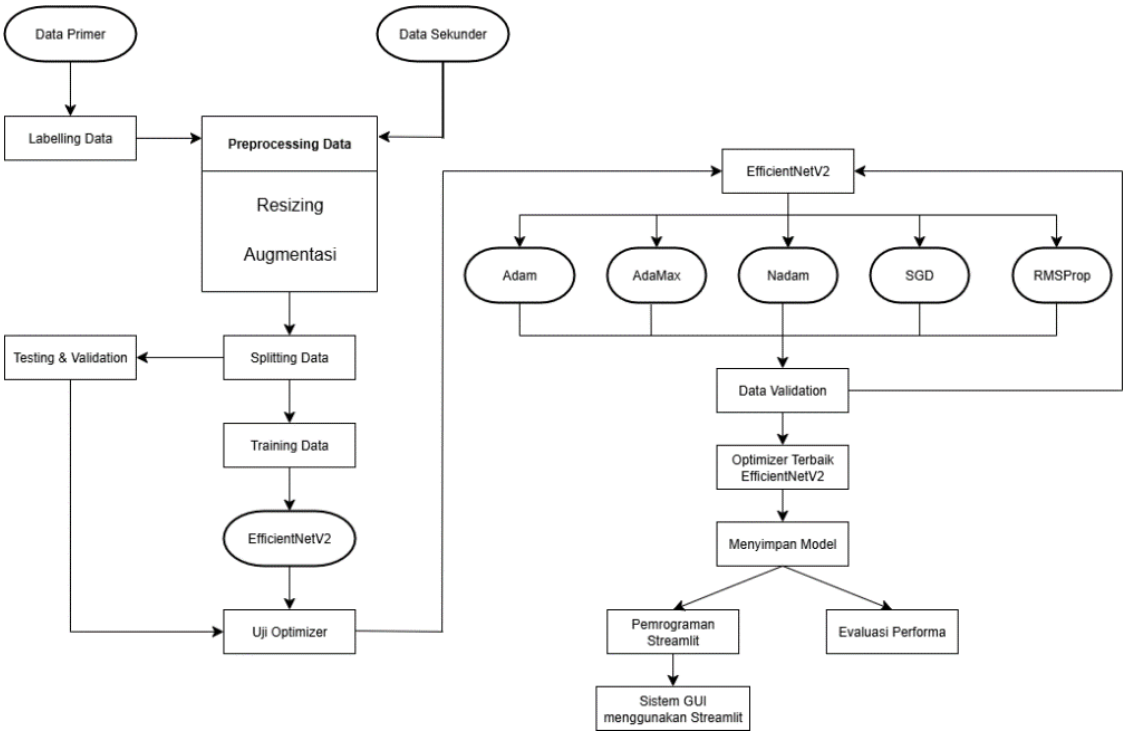

Figure 1. Research Workflow for Rice Leaf Disease Classification

2.1. Dataset

This study uses two primary data sources: a secondary dataset obtained from Kaggle and primary images collected directly from rice fields in Malang Regency, East Java, Indonesia. The secondary dataset was obtained from the *Rice Leafs* dataset available at https://www.kaggle.com/datasets/shayanriyaz/riceleafs. It consists of 3,855 rice leaf images in JPG format with RGB color channels and a resolution of 224 × 224 pixels. The dataset comprises five classes: blast (523 images), bacterial leaf blight (565 images), brown spot (779 images), tungro (500 images), and healthy leaves (1,488 images). We collected the primary dataset from rice fields in Sidokumpul, Malang District, Malang Regency, where we photographed rice plants exhibiting the relevant disease symptoms and healthy conditions. We used these images to evaluate and validate the model trained on the secondary dataset. We defined five classification categories based on the conditions represented in the dataset: blast, bacterial leaf blight, brown spot, tungro, and healthy leaves.

The five classes exhibit distinct visual characteristics. Dark brown or black lesions generally characterize blast, whereas bacterial leaf blight typically produces elongated brown lesions along the leaf margins. Brown spot is associated with oval or circular brown lesions, while tungro may cause yellowish leaf discoloration. In contrast, healthy leaves generally remain green without visible disease symptoms. EfficientNetV2 learns these visual patterns during image classification. We used the secondary dataset for model training, while reserving the primary images for validation and evaluation, as described in the subsequent methodology sections. Figure 2 presents the distribution of the secondary dataset and examples of the five classification categories.

Rice Leaf Disease Type Variable				
Blast	**Leaf Blight**	**Leaf Spot**	**Tungro**	**Normal Leaf**
				

Figure 2. Representative Images of the Five Rice Leaf Classification Classes

2.2. Image Preprocessing and Augmentation

Before training the model, we labeled and organized the rice leaf images into five categories: blast, leaf blight, leaf spot, tungro, and normal leaf. We assigned each image to its corresponding class and stored it in a separate directory to organize the dataset and support subsequent model training. After labeling, we applied data augmentation to increase dataset diversity. The augmentation techniques included 90°, 180°, and 270° rotations, as well as horizontal and vertical flips. These transformations varied image orientation while preserving visual characteristics relevant to rice leaf classification. Examples of the augmented images are presented in Figure 3.

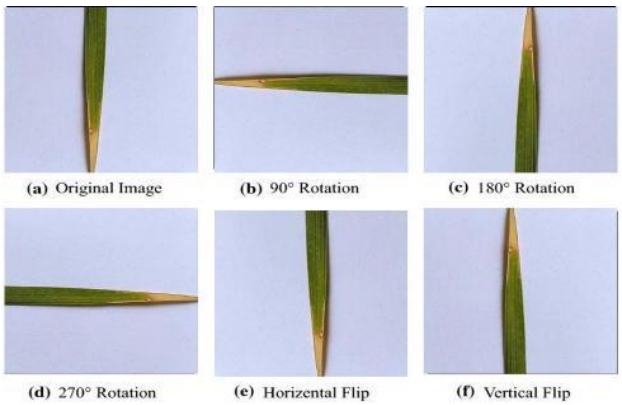

Figure 3. Example of Data Augmentation Results

After augmentation, we resized the images to a uniform resolution of 224 × 224 pixels. This step ensured consistent input dimensions for the EfficientNetV2 model. Resizing transformed the original images into a standardized format for subsequent feature extraction and classification. Figure 4 illustrates the image resizing process.

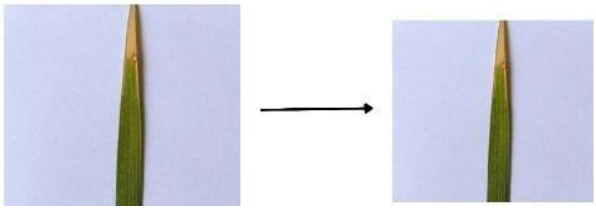

Figure 4. Image Resizing Process

These stages prepared the rice leaf images for model development by organizing class labels, introducing variations through augmentation, and standardizing image dimensions before training.

2.3. EfficientNetV2 Model Development

EfficientNetV2 is a Convolutional Neural Network (CNN) architecture developed to improve training efficiency and image classification performance. This architecture combines network design strategies with structural convolution optimizations to balance feature extraction capabilities with computational requirements (12). One core principle underlying EfficientNet is compound scaling, which proportionally scales a network based on its depth, width, and input image resolution. This scaling relationship is formulated as follows:

$$F(\alpha, \beta, \gamma) = \alpha^d x\, \beta^w x\, \gamma^r \quad (1)$$

In this equation, α, β, and γ are scaling factors related to network depth, network width, and input image resolution, respectively. Meanwhile, d, w, and r represent the dimensions of depth, width, and resolution used in network scaling. The function $F(\alpha, \beta, \gamma)$ describes the relationship between these three factors within the architectural scaling process. This approach balances model capacity with computational requirements.

EfficientNetV2 also integrates Fused-MBConv and MBConv blocks to support efficient feature extraction. Fused-MBConv blocks combine convolution operations in the early stages of the network, while MBConv blocks utilize an inverted bottleneck structure to process features. Some MBConv blocks include a Squeeze-and-Excitation (SE) mechanism that adjusts channel weights to highlight relevant information better. This combination of blocks enables the network to learn an image's visual characteristics incrementally, progressing from initial features to more complex representations.

In this research, we use EfficientNetV2 as the main model to classify rice leaf images into specified categories. The 224 × 224 pixel input image passes through an initial 3 × 3 convolution layer, then through a series of Fused-MBConv and MBConv blocks to extract features. In the final stage, a 1 × 1 convolution layer processes the feature representation, which is then summarized via global average pooling. The results are passed to the classification layer to produce a class prediction for the rice leaf image. Figure 5 shows the model architecture used in this research.

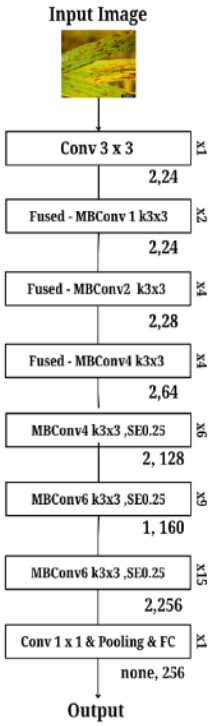

Figure 5. EfficientNetV2 Architecture Model

2.4. Hyperparameter Tuning and Model Training

Hyperparameter tuning selects suitable hyperparameter values to improve model performance. In deep learning, hyperparameters such as the learning rate, batch size, number of epochs, and optimizer influence the training process and the model's ability to learn patterns from data (13). Optimizer selection matters because different algorithms use different strategies to update model parameters and minimize the loss function.

Adam uses adaptive estimates of the first and second moments of gradients to adjust parameter updates during training (14). Adamax is a variant of Adam that uses the infinity norm to regulate parameter updates (15). Nadam combines adaptive moment estimation with Nesterov momentum to guide the optimization process (16). In contrast, Stochastic Gradient Descent (SGD) updates model parameters using gradients computed from individual samples or mini-batches (17). RMSprop adjusts parameter updates using an exponentially decaying average of squared gradients (18).

In this study, we evaluated five optimizers, Adam, Adamax, Nadam, SGD, and RMSprop, to identify a suitable training configuration for the EfficientNetV2 model. Each experiment used a batch size of 32 and was conducted for 15 epochs. We varied the optimizer between experiments while keeping the batch size and number of epochs constant to enable comparison under consistent training conditions. Table 1 presents the experimental configurations.

Table 1. Model Training Testing Scenarios

EfficientNetV2 Model			
No.	Parameter Type	Parameter Value Test	Epoch
1	Adam	Batch Size 32	15
2	Adamax	Batch Size 32	15
3	Nadam	Batch Size 32	15
4	SGD	Batch Size 32	15
5	RMSprop	Batch Size 32	15

2.5. Model Evaluation

We evaluated the EfficientNetV2 model using a confusion matrix and four classification metrics: accuracy, precision, recall, and F1-score. These metrics assess the model's ability to classify rice leaf images into their respective disease categories correctly. The confusion matrix compares predicted class labels with actual labels, showing correct classifications and misclassifications (18).

For multiclass classification, the confusion matrix records how many images are assigned to each predicted class versus their actual class. You can then calculate evaluation metrics for each class by treating the class being evaluated as the positive class and the remaining classes as the negative class. The four components are defined as follows:

1) True Positive (TP): images correctly classified as belonging to the evaluated class.
2) True Negative (TN): images correctly classified as not belonging to the evaluated class.
3) False Positive (FP): images incorrectly classified as belonging to the evaluated class.
4) False Negative (FN): images belonging to the evaluated class but incorrectly classified as another class.

Accuracy measures the proportion of correctly classified images among all evaluated images. It is calculated as follows:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

Precision measures the proportion of images correctly predicted as a particular class among all images predicted as that class. It is expressed as:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (3)$$

Recall measures the proportion of images in a class that the model correctly identifies. It is calculated as:

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

The F1-score combines precision and recall through their harmonic mean, providing a balanced measure of both metrics. It is calculated as:

$$\text{F1-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5)$$

These metrics provide complementary information about the model's classification performance. Accuracy reflects overall classification correctness, while precision and recall assess the model's ability to identify individual classes. The F1-score balances precision and recall, making it useful when evaluating classification performance across classes with different sample sizes.

RESULTS AND DISCUSSION

This section presents the results of classifying rice leaf diseases using EfficientNetV2 with hyperparameter tuning. The discussion covers data collection, image preprocessing, model development, performance evaluation, and prediction results, outlining the research stages and findings.

3.1. Data Collection

This study uses secondary data comprising 6,564 images from the Kaggle platform and primary data comprising 4,376 images collected directly from several regions in East Java Province. Consequently, the total number of images collected is 10,940. However, after loading the dataset into the program, the system detected 10,946 images categorized into six classes: blast, normal, leaf spot, leaf blight, tungro, and non-rice. Figure 6 shows the distribution of images in each class.

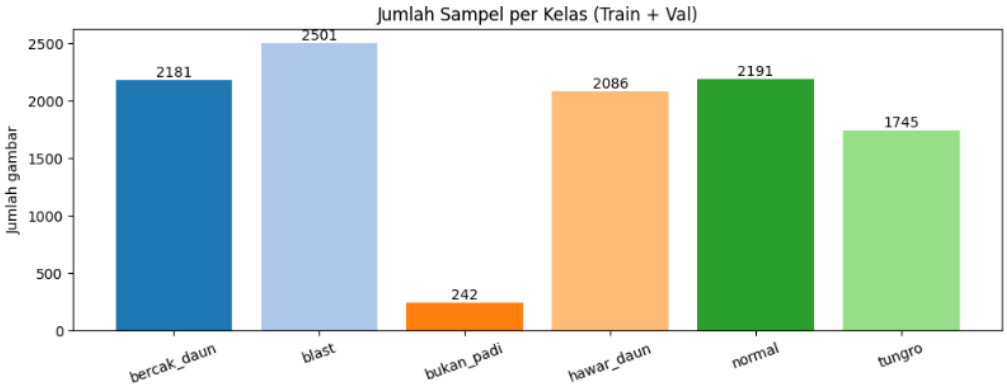

Figure 6. Distribution of Data Quantity

Based on this distribution, the blast class has the most images (2,501), followed by the normal class (2,191), leaf spot (2,181), leaf blight (2,086), tungro (1,745), and non-rice (242). This distribution is unbalanced, with a significant disparity between the blast class and the non-rice class.

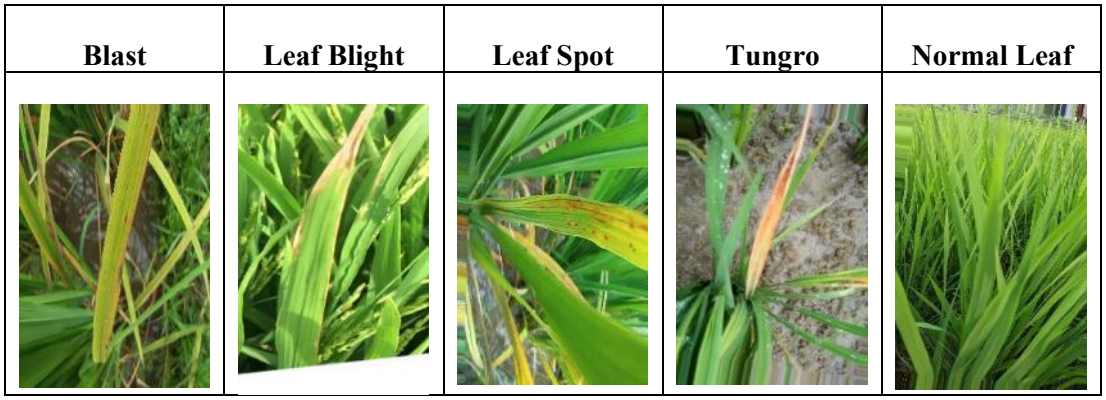

Figure 7. Types of Rice Leaf Diseases

Figure 7 shows example images representing each class in the dataset. The first five classes represent rice leaf conditions, whether infected with a disease or healthy, while the non-rice class represents images that do not depict rice leaves. Including this class allows the model to learn to distinguish rice leaf images from objects outside the targeted classification categories.

3.2. Data Preprocessing

The preprocessing stage prepares the image dataset before training the EfficientNetV2 model. This stage includes resizing images, applying data augmentation, and splitting the dataset into training and validation sets. Each step aligns the images with the model's requirements and supports training and evaluation.

3.2.1. Image Resizing

All images in the dataset are standardized to 224 × 224 pixels to match the EfficientNetV2-B0 model's input dimensions. This happens when the dataset is loaded with TensorFlow's `image_dataset_from_directory` function. Standardizing the size ensures that all images are processed with consistent input dimensions.

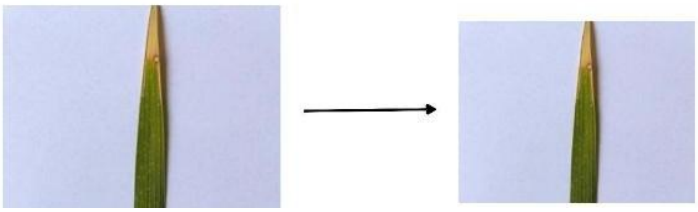

Figure 8. Resizing Results

Figure 8 shows a rice leaf image after its size was standardized to 224 × 224 pixels. This standardization ensures the image dimensions meet the model's input requirements. Relevant visual information, such as spot patterns and color changes on the leaves, remains the basis for the model to learn each class's characteristics.

3.2.2. Data Augmentation

Data augmentation increases image variety during training without changing class labels. The techniques include horizontal flipping, random rotation (up to ±10%), horizontal and vertical translation (up to 10%), resizing via zooming (up to 10%), and brightness and contrast adjustments (up to approximately 15%). These transformations are implemented using TensorFlow augmentation layers integrated into the model architecture. As a result, the model

generates image variations through random transformations during training. This augmentation helps the model recognize rice leaf disease characteristics across varying orientations, positions, and lighting conditions.

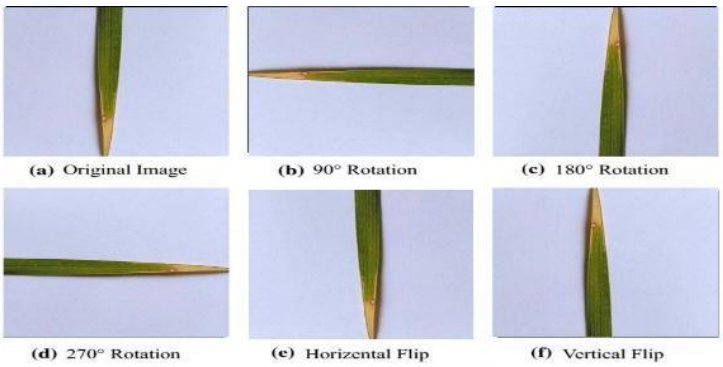

Figure 9. Augmentation Output

Figure 9 illustrates examples of image modifications produced through the augmentation process. These variations encompass changes in image orientation, position, size, brightness, and contrast. Applying these transformations introduces visual diversity during training, enabling the model to learn disease characteristics that do not depend on any single image condition.

3.2.3. Dataset Splitting

We split the dataset, consisting of 10,946 images, into training and validation sets at an 80:20 ratio. We performed this split using the `image_dataset_from_directory` function, setting `validation_split` to 0.2 and a seed of 123 to ensure a consistent split. After loading the dataset, 8,757 images were allocated for training, and 2,189 images were designated for validation.

The training data teaches the model each class's visual characteristics, while the validation data monitors performance during training and compares hyperparameter configurations. We further optimized data loading through caching, shuffling the training data, and prefetching to improve processing efficiency. This split yields training and validation sets, with no separate test set. Consequently, interpret the validation results as validation performance, not as independent test results from data not used in model development.

3.3. Modeling

We used the EfficientNetV2-B0 architecture to classify images into six categories: blast, normal, leaf spot, leaf blight, tungro, and non-rice. We developed the model using transfer learning, leveraging pre-trained ImageNet weights. We then compared five optimizers to determine the training configuration that yielded the best validation accuracy. We then used the selected optimizer for final model training and fine-tuning.

3.3.1. Development of the EfficientNetV2-B0 Model

The classification model is built using the EfficientNetV2-B0 architecture, initialized with pre-trained ImageNet weights and an input size of 224 × 224 pixels. In the initial phase, the backbone layers are frozen to preserve the ImageNet weights learned during early training.

Input images first pass through an augmentation layer, then through the EfficientNetV2-B0 backbone to extract visual features. The resulting features then pass through a Global Average Pooling layer to summarize the feature information, followed by a Dropout layer (rate 0.2) to help mitigate overfitting. Finally, a Dense layer with a softmax activation function generates prediction probabilities for six classes. The model is compiled with sparse categorical crossentropy loss and accuracy as the metric. A transfer learning approach is employed, allowing the model to leverage previously learned visual features and adapt them to the specific characteristics of rice leaf disease images.

3.3.2. Comparison of Optimizers

We tested five optimizers: Adam, Adamax, Nadam, SGD, and RMSprop. Each configuration utilized an initial learning rate of 0.001 and a batch size of 32. We rebuilt the model for each experiment to ensure the weights were re-initialized before training. We then compared each configuration based on its validation accuracy.

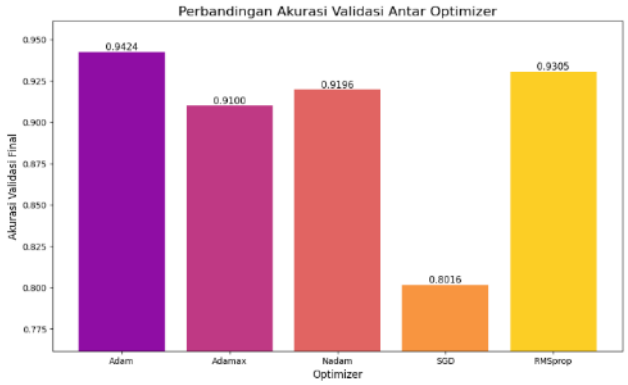

Figure 10. Comparison of Validation Accuracy Across Five Optimizers

Based on the experimental results shown in Figure 10, Adam achieved the highest validation accuracy at 94.24%. RMSprop followed at 93.05%, followed by Nadam at 91.96%, Adamax at 91.00%, and SGD at 80.16%. Adam achieved a validation accuracy 1.19 percentage points higher than RMSprop, which ranked second. Meanwhile, SGD recorded the lowest validation accuracy among the five configurations tested. Given these results, we selected Adam to train the final model. These findings indicate that Adam provided the best validation performance under the experimental configuration used in this study. Nevertheless, these results do not imply that Adam is invariably superior across all datasets or training conditions.

3.3.3. Final Model Training and Fine-Tuning

After selecting Adam as the optimal optimizer, we rebuilt the model with an initial learning rate of 0.001. During the initial training phase, we kept the EfficientNetV2-B0 backbone frozen, so learning focused on the added classification layers. We then fine-tuned the model by unfreezing the backbone layers, allowing the previously learned features to adapt to the research dataset. In our implementation, we set the final 20 backbone layers to be trainable, while keeping the preceding layers frozen. We then reduced the learning rate to 0.00001 to allow more gradual weight adjustment.

```
Epoch 25/25
274/274 ━━━━━━━━━━━━━━━━━━━━ 188s 685ms/step - accuracy: 0.9423 - loss: 0.1705 - val_accuracy: 0.9479 - val_loss: 0.1471
```

Figure 11. Training Results for the EfficientNetV2-B0 Model Using Adam and Fine-Tuning

Based on the training output reported in the thesis, the model achieved a training accuracy of 94.23% with a loss value of 0.1705 at the 25th epoch. On the validation data, the model achieved 94.79% accuracy with a validation loss of 0.1471.

3.4. Model Evaluation Results

3.4.1. Evaluation of Accuracy and Loss Before and After Fine-Tuning

We evaluated the training process by observing changes in accuracy and loss on the training and validation datasets during the initial training phase and the subsequent fine-tuning phase. The accuracy and loss graphs are presented in Figure 11.



Figure 12. Accuracy and Loss Graphs Before and After Fine-Tuning

Figure 12 illustrates the EfficientNetV2-B0 model's performance progression throughout training. Overall, both training and validation accuracy increased as epochs progressed, indicating the model learned the visual characteristics of the rice leaf images. At the start of fine-tuning, the learning curves shifted before stabilizing again.

In the loss graphs, both training and validation loss values generally trended downward throughout training. This indicates that the model's prediction errors decreased as the optimization progressed. The shift in the curves at the onset of fine-tuning reflects weight adjustments in the backbone layers, which were retrained to better align with the research dataset's specific characteristics.

Overall, these graphs provide an overview of the model's learning process on the training and validation data. However, these graphs alone are insufficient to confirm the model's generalization capability on new, independent data, as this evaluation utilized a validation dataset rather than a separate, dedicated test dataset.

3.4.2. Confusion Matrix Analysis

A confusion matrix identifies how many images are correctly or incorrectly classified for each class. This analysis covers six classes: leaf spot, blast, non-rice, leaf blight, normal, and tungro.

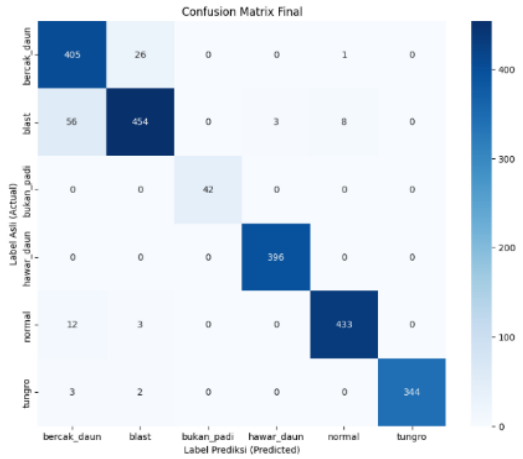

Figure 13. Final Confusion Matrix

Based on the confusion matrix results presented in the thesis, the model correctly classified 405 leaf spot images, 454 blast images, 396 leaf blight images, 433 normal images, and 344 tungro images. Nevertheless, some misclassifications persist, particularly between the leaf spot and blast classes. Specifically, 26 leaf spot images were predicted as blast, while 56 blast images were predicted as leaf spot. These results indicate that the model still struggles to distinguish the visual characteristics between these two classes.

The most significant weakness lies in the non-rice class. According to the results, the model incorrectly classified all 42 images in this category as leaf blight. This finding suggests the model cannot yet effectively differentiate images outside the rice leaf category from images of diseased rice leaves. Consequently, the model needs further improvement to recognize the non-rice class. The confusion matrix shows that evaluating model performance should consider each class's classification outcomes, rather than relying solely on overall accuracy. Furthermore, the identified misclassifications can inform which classes require refinement in future iterations of the model.

3.4.3. Evaluation of Classification Metrics

The model's performance is evaluated using precision, recall, F1-score, and accuracy. Precision indicates the accuracy of predictions for a given class relative to all predictions made for that class. Recall measures the model's ability to identify all images that actually belong to a specific class. The F1-score is the harmonic mean of precision and recall, while accuracy represents the proportion of all images that were correctly classified.

Table 2. Classification Metric Evaluation Results

Class	Precision	Recall	F1-score
Leaf Spot	0.85	0.94	0.89
Blast	0.94	0.87	0.90
Leaf Blight	0.99	1.00	1.00
Normal Leaf	0.98	0.97	0.97
Tungro	1.00	0.99	0.99

Based on the reported values, the leaf spot class has a precision of 0.85 and a recall of 0.94. These results indicate that the model identified most leaf spot images, although it misclassified some images from other classes as leaf spot. Meanwhile, the blast class achieved a precision of 0.94 and a recall of 0.87. The lower recall suggests that the model failed to identify some blast images.

The EfficientNetV2-B0 model achieved an overall accuracy of 95% in classifying rice leaf images into five categories. This result indicates that the model correctly classified the majority of the evaluated images. However, the model's performance varied across classes, as reflected in the precision, recall, and F1-score values. Therefore, interpret overall accuracy alongside the class-specific metrics for a more comprehensive assessment of the model's classification performance.

3.5. Prediction Results

3.5.1. Image Classification Results

The prediction results provide an overview of the EfficientNetV2-B0 model's ability to classify rice leaf images into predefined categories. The visualization compares the actual and predicted labels for nine sample images from the validation dataset, as shown in Figure 14.



Figure 14. Rice Leaf Image Classification Prediction Results

Based on the predictions for the nine sample images, the model correctly classified eight according to their actual labels, while misclassifying one. This error occurred with an image of a healthy (normal) rice leaf, which was incorrectly predicted as having leaf spot disease. These results indicate that the model can correctly identify most samples, although it still struggles to distinguish healthy leaves from images that show disease-like visual characteristics. On these nine samples, the model achieved 88.9% prediction accuracy. This figure reflects only the nine visualized samples and does not represent the model's overall accuracy. The comprehensive performance evaluation remains based on the classification metrics discussed in the previous subsection.

3.5.2. Prediction Visualization Using Grad-CAM

To provide insight into the image areas that contribute to classification decisions, this study employs Gradient-weighted Class Activation Mapping (Grad-CAM). This visualization generates a heatmap overlaid on the original image to highlight regions the model attended to most when making a prediction.

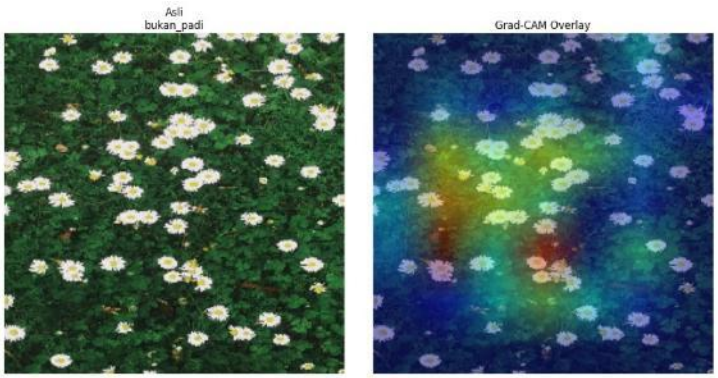

Figure 15. Prediction Results and Grad-CAM Visualization for the Non-Rice Class

Figure 15 displays the prediction results for an image of a flower, which belongs to the non-rice class. The Grad-CAM visualization highlights the flower area that contributed to the classification. This indicates which parts of the image the model used to distinguish the object from the rice leaf category. However, Grad-CAM alone does not prove that the model semantically understands the object's characteristics.



Figure 16. Prediction Results and Grad-CAM Visualization for the Tungro Class

For an image of a rice leaf labeled as tungro, the Grad-CAM visualization highlights areas of the leaf that have shifted in color toward yellow or orange. These highlighted regions indicate which parts of the image contributed to the model's prediction. This helps you see how the areas the model focused on align with the image's visual characteristics.

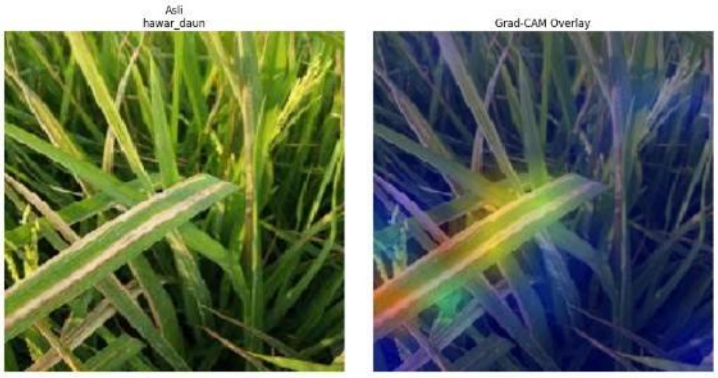

Figure 17. Prediction Results and Grad-CAM Visualization for the Leaf Blight Class

For an image labeled as leaf blight, the Grad-CAM visualization highlights elongated, brownish discoloration on the leaf. These areas correspond to a visual characteristic commonly observed in leaf blight images. While this visualization helps interpret which parts of the image contributed to the model's prediction, it cannot alone confirm the accuracy of a disease diagnosis. Overall, the prediction results and Grad-CAM visualizations provide complementary information. The prediction results show how well the model's labels align with the actual labels, while Grad-CAM helps identify the image regions that contribute to the classification decision. Nevertheless, these interpretations should be considered alongside classification metric results to provide a more comprehensive view of the model's performance.

CONCLUSION

This study develops a classification model for rice leaf diseases using the EfficientNetV2-B0 architecture, incorporating transfer learning and hyperparameter tuning. The dataset combines secondary data from Kaggle with primary images collected directly from rice fields in Malang Regency, East Java. The model classifies six categories: blast, leaf spot, leaf blight, tungro, healthy leaves, and non-rice images. Preprocessing steps include resizing images to 224 × 224

pixels and applying data augmentation to increase image variety during training. A comparison of five optimizers showed that Adam achieved the highest validation accuracy (94.24%) and was selected for subsequent model training. The evaluation results indicate that the model achieved 95% accuracy.Furthermore, Grad-CAM visualization was used to identify the image regions that contributed most to the model's predictions. The findings suggest that EfficientNetV2-B0 has potential for classifying rice leaf diseases, although the model's ability to identify non-rice images requires further improvement. Additional evaluation on an independent test dataset is necessary to assess the model's ability to generalize to new, unseen images.

REFERENCES

1. TAGLIAPIETRA BL, SOARES CF, CLERICI MTPS. Rice (Oryza sativa L.) and its products for human consumption: general characteristics, nutritional properties, and types of processing. Food Science and Technology. 13 Juni 2024;44. doi:10.5327/fst.00292
2. van Dam RM. A Global Perspective on White Rice Consumption and Risk of Type 2 Diabetes. Diabetes Care. 1 November 2020;43(11):2625–7. doi:10.2337/dci20-0042
3. Shi J, An G, Weber APM, Zhang D. Prospects for rice in 2050. Plant Cell Environ. April 2023;46(4):1037–45. doi:10.1111/pce.14565
4. Asma J, D S, D K. The Global Lifeline: A Staple Crop Sustaining Two-Thirds of the World's Population. Agriculture Archives. 22 Oktober 2023;2(3):15–8. doi:10.51470/AGRI.2023.2.3.15
5. Yusuf MA, Sundari S, Purwanto ID, Kurnia RR. Rice Production, Policies, and Their Implications on Food Security Indonesia - Thailand Perspective. Jurnal Multidisiplin Madani. 29 Januari 2024;4(1):174–9. doi:10.55927/mudima.v4i1.7927
6. Widyanto S, Subanu LP. The Factors of Rice Farmers' Poverty in Indonesia: The Perspective of Land Conversion, Land Ownership Area, and Agriculture Technologi. Jurnal Kawistara. 30 April 2023;13(1):121. doi:10.22146/kawistara.84368
7. Meghana C. Rice Leaf Disease Recognition using CNN. Int J Res Appl Sci Eng Technol. 31 Juli 2024;12(7):593–9. doi:10.22214/ijraset.2024.63609
8. Li Y, Chen X, Yin L, Hu Y. Deep Learning-Based Methods for Multiclass Rice Disease Detection Using Plant Images. Agronomy. 23 Agustus 2024;14(9):1879. doi:10.3390/agronomy14091879
9. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 28 Mei 2015;521(7553):436–44. doi:10.1038/nature14539
10. Rachman FP. Perbandingan Model Deep Learning untuk Klasifikasi Sentiment Analysis dengan Teknik Natural Languange Processing. Jurnal Teknologi dan Manajemen Informatika. 22 Desember 2021;7(2):113–21. doi:10.26905/jtmi.v7i2.6506
11. Amalia N. Optimasi Algoritma Random Forest dengan Hyperparameter Tuning Menggunakan GridSearchCV untuk Prediksi Nasabah Churn pada Industri Perbankan.
12. Adisaputra Sinaga N, Rosnelly R, kunci-CNN K, Kelamin J. Analisis Penggunaan Model EfficientNetV2 Dalam Memprediksi Jenis Kelamin Pada Wajah Pengguna Masker. Jurnal Teknik Informatika dan Sistem Informasi [Internet]. 2022;9(3). Tersedia pada: http://jurnal.mdp.ac.id
13. Nugraha W, Sasongko A. Hyperparameter Tuning on Classification Algorithm with Grid Search. SISTEMASI. 21 Mei 2022;11(2):391. doi:10.32520/stmsi.v11i2.1750
14. Gu J, Zeng Z, Li S, Jing W, Huang F. The Application of Cusp Catastrophe Theory to Analyze Road Traffic Potential Characteristics, Considering Headway, with Urban Data. Mathematics. 16 Februari 2025;13(4):652. doi:10.3390/math13040652
15. Yang L. Theoretical Analysis of Adam Optimizer in the Presence of Gradient Skewness. International Journal of Applied Science. 4 Oktober 2024;7(2):p27. doi:10.30560/ijas.v7n2p27

16. Shim DS, Shim J. A Modified Stochastic Gradient Descent Optimization Algorithm With Random Learning Rate for Machine Learning and Deep Learning. Int J Control Autom Syst. 25 November 2023;21(11):3825–31. doi:10.1007/s12555-022-0947-1
17. Elshamy R, Abu-Elnasr O, Elhoseny M, Elmougy S. Improving the efficiency of RMSProp optimizer by utilizing Nestrove in deep learning. Sci Rep. 31 Mei 2023;13(1):8814. doi:10.1038/s41598-023-35663-x
18. Markoulidakis I, Markoulidakis G. Probabilistic Confusion Matrix: A Novel Method for Machine Learning Algorithm Generalized Performance Analysis. Technologies (Basel). 13 Juli 2024;12(7):113. doi:10.3390/technologies12070113