**Indonesian Journal of Electronics, Electromedical Engineering, and Medical Informatics** e-ISSN: 2656-8624
Homepage: https://ijeeemi.org/; Vol. 8, No. 2, pp. 233-247, May 2026

**RESEARCH PAPER** **OPEN ACCES**

# Detection of Rice Diseases: Leaf Blast, Bacterial Leaf Light, and Brown Spot Using Image Enhancement and Faster Region-Based Convolutional Neural Network

**Monika Faswia Fahmi[1], Deni Tri Laksono[1], Achmad Fiqhi Ibadillah[1] , and Dedi Tri Laksono[2]**

[1] Department of Electrical Engineering, Universitas Trunodjoyo Madura, Madura, Indonesia
[2] Department of Electrical Engineering, Politeknik Negeri Padang, Padang, Indonesia

## Abstract

**Rice diseases such as leaf blight, blast, and brown spot remain major constraints on food security and rural livelihoods across Southeast Asia, causing significant yield losses each year. In Indonesia, particularly in Lamongan, East Java, these pathogens threaten smallholder productivity and disrupt national rice supply chains. This study aims to enhance automated rice disease detection under real agricultural conditions by integrating image preprocessing techniques with a deep learning-based detection framework. The main contribution lies in developing a hybrid pipeline that combines RGB-to-grayscale conversion and contrast stretching prior to model training, effectively mitigating low-contrast conditions and noise commonly found in field-acquired image datasets. The enhanced images are subsequently processed using the Faster Region-Based Convolutional Neural Network (Faster R-CNN) with a ResNet-50 backbone to localize and classify disease symptoms. Experiments conducted on a dataset of 1,500 annotated rice leaf images achieved high detection performance, with accuracies of 97.37% for leaf blight, 94.12% for blast, and 95.24% for brown spot. Compared with the baseline Faster R-CNN model, the proposed approach improved classification accuracy from 0.8906 to 0.9297, reduced false negatives from 0.439 to 0.1998, increased foreground classification accuracy from 0.55 to 0.78, and descreased total loss from 0.839 to 0.6493. These results demonstrate that integrating RGB-to-grayscale conversion and contrast stretching significantly enhances feature representation, leading to improved detection accuracy, reduced error rates, and more stable training behavior. Overall, the proposed framework provides a robust and reliable approach for rice disease identification and offers strong potential for practical deployment in precision agriculture systems.**

## Paper History

Received Nov. 17, 2025
Revised Feb 20, 2026
Accepted March 24, 2026
Published May 13, 2026

## Keywords

Faster R-CNN;
RGB to Grayscale Conversion;
Contrast Stretching;
Rice Disease Detection

## Author Email

monika.faswiaf@trunojoyo.ac.id
deni.laksono@trunojoyo.ac.id
fiqhi.ibadillah@trunojoyo.ac.id
deditrilaksono@pnp.ac.id

## I. Introduction

Rice, one of the world's most essential staple crops, sustains more than 3.5 billion people globally and contributes nearly 20% of daily caloric intake [1]. In Southeast Asia, where rice production underpins food security and rural livelihoods, diseases such as leaf blight, blast, and brown spot continue to cause substantial yield losses each year [2], [3], and [4]. In Indonesia, national statistics report that bacterial leaf blight and blast can cause yield reductions ranging from 20–40% under moderate infection and up to 50% in severe outbreaks, particularly in humid lowland regions such as East Java. In Lamongan Regency, fluctuations in rice productivity in recent years have been partly associated with recurrent disease incidence reported by local agricultural agencies [5]. These quantitative impacts highlight the urgency of developing robust, field-adaptable disease detection systems tailored to Indonesian agricultural conditions.

More broadly, plant diseases remain one of the primary constraints to sustainable agriculture, with early detection and accurate diagnosis increasingly recognized as critical strategies for mitigating yield loss and strengthening food security [6]. These conditions underscore the urgent need for reliable rice disease detection systems to support timely and effective disease management. Over the past decade, diverse approaches have been developed for plant disease detection, ranging from manual inspection and disease maps to computer-assisted techniques. Manual inspection, while traditional, is inefficient and prone to subjective errors, whereas disease maps may lead to misclassification due to visual similarity of symptoms. More recently, computer vision and deep learning have emerged as dominant

**Corresponding author:** Monika Faswia Fahmi, monika.faswiaf@trunojoyo.ac.id, Department of Electrical Engineering, Universitas Trunodjoyo Madura, Madura, Indonesia
**DOI:** https://doi.org/10.35882/ijeeemi.v8i2.287

approaches, with convolutional neural networks (CNNs) and region-based convolutional neural networks (R-CNNs) achieving notable success [7], [8], [9], [10], and [11]. For instance, the literature [8] applied a CNN-based model to detect *Helminthosporium* leaf spot in wheat with 91.43% accuracy, while studies [10], [11] explored hybrid CNNs for diseases in corn and alfalfa, achieving accuracies exceeding 97%. Focusing on rice, hybrid FCM-KM and Faster R-CNN achieved over 96% accuracy for multiple diseases [12], and Faster R-CNN has been tested for rice false smut [13]. Other classifiers, such as Bayes, SVM, and optimized deep neural networks [14], [15], have shown competitive results, with CNN and Faster R-CNN variants reporting accuracies between 90% and 99% [16], [17], [18], [19], and [20].

Despite these promising results, current models exhibit several critical limitations. First, most rely on datasets of inconsistent quality, where issues such as noise, low contrast, and blurred edges are prevalent. These factors significantly degrade the precision of detection models like Faster R-CNN, increasing false negatives and reducing training stability [21], [22], and [23]. Second, many prior works emphasize model architecture but devote comparatively little attention to preprocessing, despite the fact that real-world agricultural images are frequently captured under uncontrolled field conditions (e.g., varying illumination, motion blur, or cluttered backgrounds), which degrades detection robustness [24], [25]. Third, although technical accuracy has markedly improved in lab settings, far fewer studies focus on usability and scalable deployment in agricultural contexts, limiting the delivery of reliable, field-ready tools for large-scale farmer adoption [26], [27].

To address these challenges, this study proposes an enhanced detection framework that integrates image preprocessing and deep learning. The approach introduces a hybrid pipeline that combines RGB-to-grayscale conversion and contrast stretching before training the Faster Region-Based Convolutional Neural Network (Faster R-CNN). Grayscale conversion minimizes color variance caused by illumination changes [28], thereby mitigating data quality inconsistencies commonly observed in heterogeneous field conditions. Contrast stretching enhances lesion visibility and edge definition [29], directly addressing the limited emphasis on preprocessing in prior studies while improving feature extraction and region proposal stages. Furthermore, by incorporating lightweight preprocessing rather than complex architectural modifications, the proposed strategy preserves computational efficiency and supports scalable, farmer-oriented deployment scenarios. This preprocessing-driven approach therefore strengthens detection robustness while maintaining practical usability in real-world agricultural environments.

The primary aim of this study is to evaluate the effectiveness of integrating image preprocessing techniques (RGB to grayscale conversion and contrast stretching) with Faster R-CNN in improving detection accuracy, training stability, and convergence speed for rice disease identification using field-acquired image datasets.

The contributions of this study can be summarized as follows:

1. Problem-oriented framing: We explicitly address the limitations of existing rice disease detection methods, particularly their sensitivity to noise, blur, and low contrast commonly found in real-world agricultural images.
2. Methodological novelty: We propose a hybrid pipeline that integrates RGB-to-grayscale conversion and linear contrast stretching within a Faster R-CNN framework optimized for field-acquired rice disease images. This integration empirically improves detection stability, sensitivity, and convergence performance, demonstrating the practical value of preprocessing-aware detector design.
3. Empirical and practical significance: Experiments on 1,500 rice leaf images demonstrate improved accuracy and stability over the baseline Faster R-CNN. The model supports near-real-time deployment within edge-assisted or cloud-based frameworks, demonstrating practical scalability for farmer-oriented diagnostic applications.
4. Interdisciplinary contribution: By linking advances in computer vision with the needs of precision agriculture, this study contributes to the broader agenda of food security and smart farming systems.

This paper is organized as follows: Section 2 reviews relevant studies on rice disease detection and outlines the research structure; Section 3 details the proposed preprocessing and detection methodology; Section 4 describes the dataset and experimental setup; Section 5 presents and discusses the results, and Section 6 concludes the paper and highlights future research directions.

## II. Rice Disease Performance and Research Structure

This research focuses on detecting rice diseases, particularly leaf blight, blast, and brown spot, which are among the most prevalent and destructive pathogens affecting rice cultivation. To support the training and evaluation processes, a dataset comprising 500 images for each disease category was prepared. These images were sourced from the publicly available Kaggle dataset and supplemented with real field images captured in the Lamongan area of East Java using a Logitech C270 camera to ensure variability under real agricultural conditions. For each class, approximately 300 images (60%) were collected from Lamongan field conditions, and 200 images (40%) were sourced from publicly available repositories. This consistent distribution across

**Corresponding author:** Monika Faswia Fahmi, monika.faswiaf@trunojoyo.ac.id, Department of Electrical Engineering, Universitas Trunodjoyo Madura, Madura, Indonesia
**DOI:** https://doi.org/10.35882/ijeeemi.v8i2.287

categories was maintained to minimize source-induced bias and enhance dataset representativeness.

Among the three diseases, leaf blight poses a serious threat to rice production. It is caused by the Gram-negative bacterium Xanthomonas oryzae pv. Oryzae (Xoo), which was first reported in Indonesia in 1950. This pathogen can lead to yield reductions ranging from 21–29% during the rainy season and 18–28% during the dry season [30]. Morphologically, Xoo is a short, rod-shaped bacterium that moves via a single flagellum and does not form spores, contributing to its rapid spread under favorable conditions. In addition to blight, blast disease represents a major fungal threat caused by Pyricularia grisea. It affects various parts of the rice plant, including leaves, leaf necks, stems, panicles, and grains, and can occur throughout all stages of growth from seedling to pre-harvest. The disease typically manifests as diamond-shaped lesions with gray or whitish centers encircled by brown borders, which serve as key diagnostic features [31]. Its aggressive nature and broad host range make blast one of the most challenging rice diseases to manage. Equally important is brown spot disease, which is caused by the fungal pathogen Cercospora oryzae. This disease is particularly severe in potassium-deficient paddy fields and is known for reducing both grain quality and yield. Early symptoms include small, uniformly distributed brown lesions, which eventually develop into larger spots with gray centers and brown margins. If not treated promptly, brown spots can significantly compromise the photosynthetic capacity of the plant.

To address the need for accurate and efficient diagnosis of these diseases, this study proposes a method that integrates image preprocessing techniques (RGB to grayscale conversion and contrast stretching) with a deep learning-based detection framework, Faster R-CNN. The overall methodology is illustrated in Fig. 1. The process begins with dataset preparation, including image annotation for all three disease categories. This is followed by an image enhancement stage that involves RGB to grayscale conversion and contrast stretching to enhance visual clarity and highlight disease-related features, thereby improving the quality of input data for model training. The enhanced images are then processed using the Faster Region-Based Convolutional Neural Network (Faster R-CNN), which consists of two main components: a Region Proposal Network (RPN) that identifies candidate regions of interest, and a Fast R-CNN module that performs object classification and bounding box regression. The output is presented as bounding boxes with associated confidence scores, allowing precise localization and identification of diseased regions.

## III. Materials And Methods

### A. Datasets

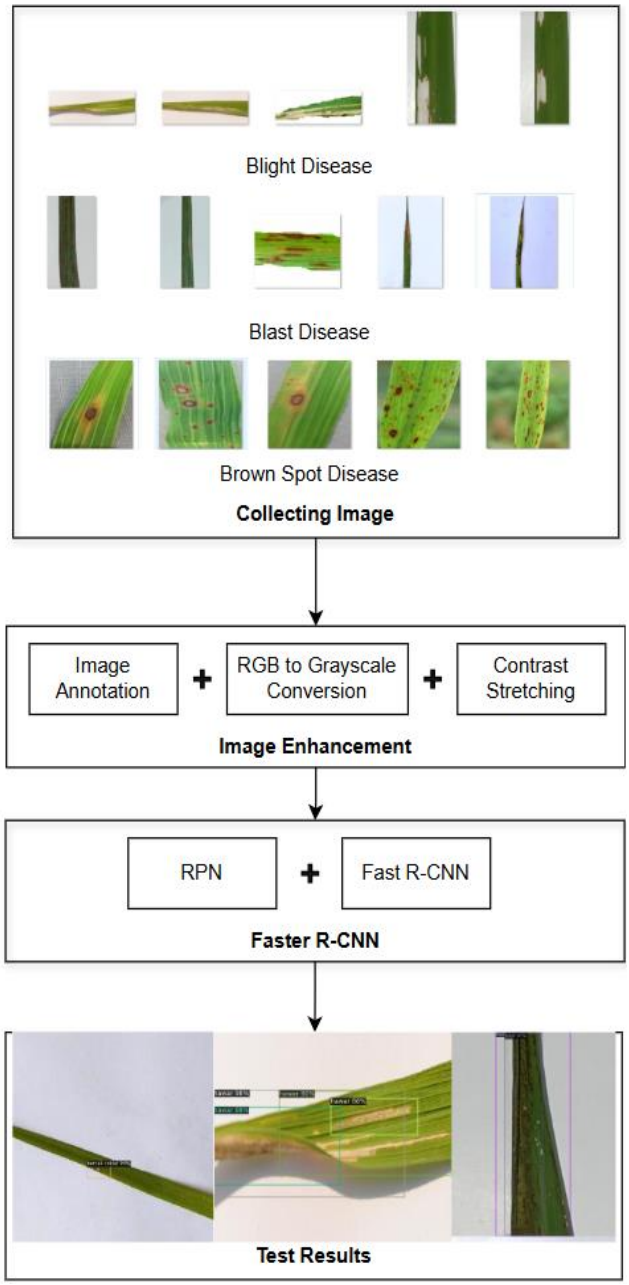



**Fig. 1. Rice disease identification structure**

The dataset used in this study consisted of 1,500 images of rice leaves, collected from multiple sources to ensure variability and reduce dataset bias. Approximately 60% of the images were captured directly in the field across different plots in the Lamongan area, East Java, using varying lighting and background conditions. The remaining 40% were obtained from publicly available repositories such as Kaggle, which provided standardized samples for cross-reference. To maintain balance across classes, the dataset was distributed evenly into three categories: blast (500 images), brown spot (500 images), and bacterial leaf blight (500 images), as illustrated in Fig. 2. This balanced distribution was designed to prevent class imbalance during model training, which can otherwise bias predictions toward the majority class.

**Corresponding author:** Monika Faswia Fahmi, monika.faswiaf@trunojoyo.ac.id, Department of Electrical Engineering, Universitas Trunodjoyo Madura, Madura, Indonesia
**DOI:** https://doi.org/10.35882/ijeeemi.v8i2.287

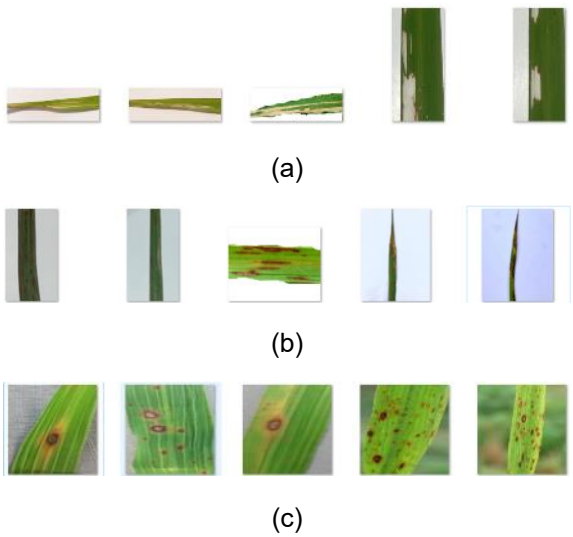



**Fig. 2. Part of the picture of rice disease: (a) blight; (b) blast; (c) brown spot**

Potential sources of bias, such as over-representation of a single rice variety or environmental condition, were addressed by including samples from different stages of plant growth, multiple lighting scenarios (morning, noon, cloudy), and various image resolutions. While this strategy improves diversity, the dataset remains limited to three major classes of rice conditions. Therefore, generalization to unseen diseases or extreme environmental variations remains a challenge and is noted as a limitation of this study.

### B. Image Annotations

Before starting the training process, image annotation was conducted to ensure that objects could be properly

**Fig. 3. Image annotation process using labelimg**

recognized in the dataset. In this study, the annotation was performed using LabelImg software, with the output stored in XML format in accordance with the PASCAL VOC standard, as illustrated in Fig. 3. The annotation process focused on identifying the region of interest (ROI) corresponding to visible disease symptoms on rice leaves, including blight, blast, and brown spot lesions. To ensure reliability, clear annotation criteria were defined: (1) only lesions occupying at least 5% of the leaf area were labeled, (2) ambiguous or overlapping regions were excluded, and (3) bounding boxes were drawn tightly around symptomatic areas without including excessive background. The exclusion of highly ambiguous or heavily overlapping lesions improves annotation consistency but may limit the model's exposure to complex real-world scenarios. Consequently, detection performance under severe co-infection conditions may be reduced, representing a direction for future dataset expansion. After completing the annotation stage, the dataset, including both images and XML files from LabelImg, was uploaded to Roboflow, a platform designed for dataset management and preprocessing. Within Roboflow, several preprocessing steps were applied, including data augmentation to increase variability and the splitting of the dataset into training, validation, and test sets. The dataset was then exported in COCO format, which generated a shareable link for seamless integration with external training environments such as Google Colab.

### C. RGB to Grayscale Conversion

Image preprocessing plays a crucial role in improving the visibility of disease symptoms prior to training. In this study, grayscale conversion was applied to reduce the three RGB channels into a single intensity channel. This choice was motivated by two factors: (1) computational efficiency, as reducing channels lowers training complexity, and (2) disease relevance, since rice diseases are often better distinguished by lesion texture, boundary irregularity, and spot density than by absolute color values. Grayscale conversion also helps suppress background color variations caused by illumination differences or soil reflections, ensuring that leaf lesions remain the primary discriminative feature. The RGB-to-grayscale conversion process transforms a color image, which contains three color channels (Red, Green, and Blue), into a single channel representing intensity in shades of gray. RGB to Grayscale steps:

**Step 1. Extract RGB values for each pixel: Red, Green, and Blue** (R, G, B).

**Step 2. Calculate the Grayscale Intensity: Use** a formula to compute the grayscale intensity based on the RGB values.

**Step 3. Replace The Pixel Values**, assign the computed grayscale value to all three channels (R, G, and B) to create a uniform grayscale pixel

The most common formula for RGB to Grayscale conversion is a weighted sum of the RGB components, which is shown in Eq. (1) [28] as follows:

$$Gray = 0.2989R + 0.5870G + 0.1140B \quad (1)$$

In this formulation, R, G, and B represent the red, green, and blue components of a pixel, respectively, while Gray denotes the resulting grayscale intensity. After applying this transformation to every pixel, the output is a single-channel grayscale image with intensity values ranging from 0 (black) to 255 (white). The conversion results show that the original RGB rice disease images are successfully

**Corresponding author:** Monika Faswia Fahmi, monika.faswiaf@trunojoyo.ac.id, Department of Electrical Engineering, Universitas Trunodjoyo Madura, Madura, Indonesia
**DOI:** https://doi.org/10.35882/ijeeemi.v8i2.287

transformed into grayscale images that emphasize lesion texture, boundary irregularity, and spot patterns, as illustrated in Fig. 4.

(a) (b)

**Fig. 4. Image enhancement effect: (a) original image; (b) RGB to grayscale conversion**

### D. Contrast Stretching

Following grayscale conversion, contrast stretching was applied to enhance visibility in low-contrast images. Compared with alternative methods such as global histogram equalization or Contrast Limited Adaptive Histogram Equalization (CLAHE), contrast stretching provides a linear and controlled enhancement that avoids over-amplification of noise, which is common in field-captured images. Preliminary trials indicated that this method yielded more consistent detection, particularly for small brown spot symptoms. Thus, the combination of grayscale and contrast stretching was selected as a balanced trade-off between feature clarity, noise robustness, and computational efficiency. Contrast stretching is a technique used in image processing to enhance contrast by expanding the range of intensity levels. It aims to make dark areas darker and light areas lighter, improving the visibility of details in an image. Contrast Stretching steps:

**Step 1. Identify Intensity Range**: The first step is to identify the minimum and maximum intensity values in the original image. These values represent the dark and light extremes of the image.

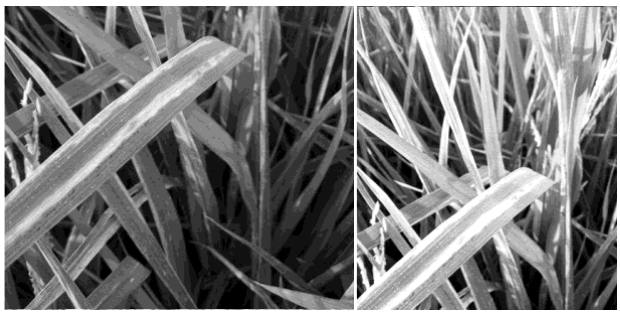
**Step 2. Define New Intensity Range**, set a new desired range of intensity values, typically spanning from the minimum to the maximum intensity

**Step 3. Apply Contrast Stretching Formula**, every pixel intensity in the original image is transformed according to a linear function to stretch the contrast to the new range.

The formula for contrast stretching using Eq. (2) [29] as follows

$$I'(x,y) = \frac{I(x,y) - I_{min}}{I_{max} - I_{min}} \times L_{max} - L_{min} + L_{min} \quad (2)$$

In this formulation, $I'(x,y)$ represents the new pixel value after contrast stretching, while $I(x,y)$ denotes the original pixel value. The terms $I\_min$ and $I\_max$ correspond to the minimum and maximum intensity values in the original image, respectively, whereas $L\_min$ and $L\_max$ represent the desired minimum and maximum intensity values of the output image. The results of applying contrast stretching to the grayscale rice disease images show that the intensity distribution is effectively expanded, producing images with a wider dynamic range. This enhancement improves the visibility of lesion features, particularly in regions that were previously too dark or too bright. As a result, important characteristics such as lesion boundaries and texture become more distinguishable, as illustrated in Fig. 5.



(a) (b)

**Fig. 5. Image enhancement effect: (a) original image; (b) contrast stretching**

### E. Faster R-CNN

Ross Girshick and the target detection research community introduced Faster R-CNN [32] in 2015, following the earlier releases of R-CNN [33] and Fast R-CNN [34]. Faster R-CNN can be simply viewed as a model of "the regional generation network + Fast R-CNN", which applies the Region Proposal Network (RPN) instead of Selective Search in Fast R-CNN. RPN is a recommendation algorithm for the proposal. The convolution layer/ full connection layer processing is performed on the feature map, and then the detected target is subjected to position regression and classification, and finally, the region recommendation is used to obtain a more accurate disease location. In this study, Faster R-CNN was selected as the detection framework due to its proven effectiveness in object detection tasks involving complex backgrounds and small objects, such as leaf diseases. The backbone network used was ResNet-50 pretrained on ImageNet, which balances accuracy and computational efficiency. The Region Proposal Network (RPN) was configured with anchor sizes of [32, 64, 128, 256, 512] and aspect ratios of [0.5, 1, 2], while the Intersection over Union (IoU) threshold for positive proposals was set to ≤0.5 and 0.7.

**Corresponding author:** Monika Faswia Fahmi, monika.faswiaf@trunojoyo.ac.id, Department of Electrical Engineering, Universitas Trunodjoyo Madura, Madura, Indonesia
**DOI:** https://doi.org/10.35882/ijeeemi.v8i2.287

Although recent architectures such as YOLOv7 [35], EfficientDet [36], and Vision Transformers (ViT) [37] have demonstrated state-of-the-art performance in object detection tasks, Faster R-CNN was chosen in this study for three reasons. First, its two-stage detection pipeline provides higher localization accuracy, particularly for small and irregular lesions on rice leaves. The region refinement after proposal generation allows more precise localization compared to many single-stage detectors that rely on dense prediction grids, which is especially beneficial when lesion boundaries are subtle and heterogeneous. Second, Faster R-CNN offers greater interpretability and stability during training, particularly with a moderate dataset size, as in this study. Third, the architecture has been widely adopted in prior research on plant disease detection, enabling meaningful benchmarking and reproducibility. Future work may explore YOLO-based detectors or transformer-based models to further improve inference speed and scalability.

**F. Faster R-CNN steps:**

**Step 1. Backbone Network (Feature Extraction**

Faster R-CNN extracts features from the input image using a pre-trained convolutional network, such as ResNet. This backbone generates a Feature Map from the original image.

**Step 2. Generate RPN for Candidate Regions**

The RPN uses anchor boxes with various sizes and aspect ratios to generate region proposals. The anchor configuration used in this study is:

Anchor Sizes: [32, 64, 128, 256, 512]

Aspect Ratios: [0.5, 1.0, 2.0]

The total number of anchor boxes can be calculated using the formula Eq. (3) [32]:

$$\sum anchor\ Box = Nz \times Nr \quad (3)$$

$Nz$ represents the number of anchor sizes, while $Nr$ represents the number of aspect ratios. With this configuration, the total number of anchor boxes is 15 per position on the feature map.

**Step 3. Bounding Box Regression**

Bounding box regression is a process used to refine the initial region proposals (anchor boxes) generated by the Region Proposal Network (RPN). It adjusts the coordinates of these proposals to better match the ground truth bounding boxes of the objects in the image.

1) Anchor Box Calculation

The author calculates the value of anchor boxes with a size of 32 and an aspect ratio of 0.5. Based on this configuration, the resulting anchor box dimensions are $w_a = 22.6$ and $h_a = 45.3$

2) Ground Truth Bounding Box Calculation

From the image enhancement data, one dataset is selected with coordinates $X_{min} = 1$, $X_{max} = 614$, $Y_{min} = 249$, and $Y_{max} = 488$, with image dimensions of 640 × 640. Based on this data, the ground truth bounding box has center coordinates $x = 307.5$and $y = 368.5$, with width $w = 613$and height $h = 239$.

3) Bounding Box Regression Calculation

Using the anchor box values $w_a = 22.6$ and $h_a = 45.3$, along with the anchor box center coordinates $(x_a, y_a) = (4,4)$, the bounding box regression produces offset values of $\text{Offset}_x = 13.4$, $\text{Offset}_y = 8.05$, $\text{Offset}_w = 1.4$, and $\text{Offset}_h = 0.7$. These offsets represent the spatial differences between the anchor box and the ground-truth bounding box and are used during model training to refine bounding-box predictions and improve accuracy.

4) Apply Bounding Box Regression Offsets

After obtaining the regression offsets from the bounding box regression process, these offsets are applied to the anchor box to produce a more accurate bounding box, referred to as the final bounding box. Using the anchor box parameters $(x_a, y_a) = (4,4)$, $w_a = 22.6$, and $h_a = 45.3$, along with the regression offsets $\text{Offset}_x = 13.4$, $\text{Offset}_y = 8.05$, $\text{Offset}_w = 1.4$, and $\text{Offset}_h = 0.7$, the resulting final bounding box has coordinates $x_{min} = 260.99$, $y_{min} = 323.1$, $x_{max} = 352.69$, and $y_{max} = 414.3$.

5) Intersection of Union (IoU) Calculation

The Intersection over Union (IoU) is a metric used to evaluate the overlap between two bounding boxes, namely the predicted box and the ground truth box. It is calculated as shown in Eq. (4) [32], where IoU represents the ratio between the area of overlap and the area of union. In this context, the area of overlap refers to the shared region between the predicted and ground truth boxes, while the area of union represents the total combined area covered by both boxes.

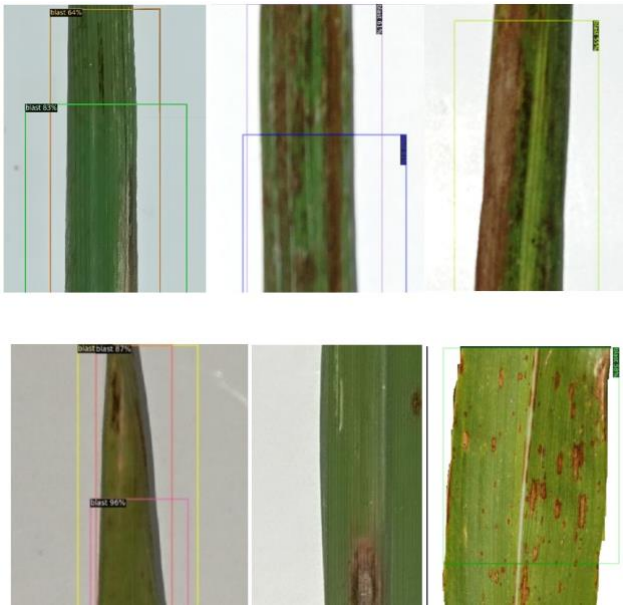


**Fig. 6. Inference process for the leaf blast class: the bounding box highlighted in red indicates an instance that was not successfully detected by the model**

**Corresponding author:** Monika Faswia Fahmi, monika.faswiaf@trunojoyo.ac.id, Department of Electrical Engineering, Universitas Trunodjoyo Madura, Madura, Indonesia
**DOI:** https://doi.org/10.35882/ijeeemi.v8i2.287

$$IoU = \frac{area\ overlap}{area\ union} \quad (4)$$

To determine whether a predicted bounding box is classified as an object or not based on the IoU (Intersection over Union), thresholds are commonly used as follows [24]:

a) **IoU ≥ 0.5 or 0.7:** The proposal is considered a positive match, indicating that the predicted box corresponds to an object.
b) **IoU ≤ 0.3:** The proposal is considered a negative match, indicating that it is background or does not correspond to an object.
c) **0.3 < IoU < 0.5 (or threshold used):** These proposals are often ignored during training to avoid ambiguity.

The result shows that the predicted bounding box achieves an IoU value of approximately 0.57, indicating that about 57% of the predicted and ground-truth areas overlap. Based on the defined threshold, this value is classified as a positive match, indicating that the model accurately localizes the object.

## IV. Implementation And Results

To verify the effectiveness of the algorithm, the simulation environment is Windows 11; the CPU is a 11th Gen Intel(R) Core™ i3-115G4 @ 3.00 GHz, the SSD is 512 GB, and the memory is 8 GB. The experimental platform is Ubuntu 14.04, and the PyTorch framework is chosen as the deep learning framework. The dataset used comprises three categories (rice blight, blast, and brown spot), totaling 1,500 images, with 83% used for training, 12% for validation, and 6% for testing. All reported accuracy, precision, recall, F1-score, and confusion matrices represent performance on the independent test set, which was strictly separated from training and validation data to ensure unbiased generalization evaluation. No test images were used during training or hyperparameter tuning. The training configuration was as follows: learning rate of 0.001 with a step decay scheduler, batch size of 16, and training for 50 epochs. Stochastic Gradient Descent (SGD) with momentum 0.9 and weight decay 1e-4 was used as the optimizer. Input images were resized to 512 × 512 pixels, and data augmentation included random horizontal flipping and contrast adjustment to enhance generalization.The following is Fig. 6- Fig. 9 showing the results of the rice disease detection experiments: leaf blight, blast, and brown spot. The confusion matrix is used to calculate accuracy, loss, precision, recall, and F1 score from rice disease detection test results. Accuracy is calculated as the ratio of correct predictions (True Positives + True Negatives) to the total number of predictions (True Positives + True Negatives + False Positives + False Negatives).

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (5)$$

Precision is the ratio of True Positives to the sum of True Positives and False Positives, reflecting the model's ability to correctly identify positive cases.

$$precission = \frac{TP}{TP + FP} \quad (6)$$

Recall is the ratio of True Positives to the sum of True Positives and False Negatives, showing the model's ability to capture all relevant positive cases.

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

F1-Score is the harmonic mean of Precision and Recall, providing a balance between the two metrics.

$$F1 - Score = 2 \times \frac{Precission \times Recall}{Precission + Recall} \times 100\% \quad (8)$$

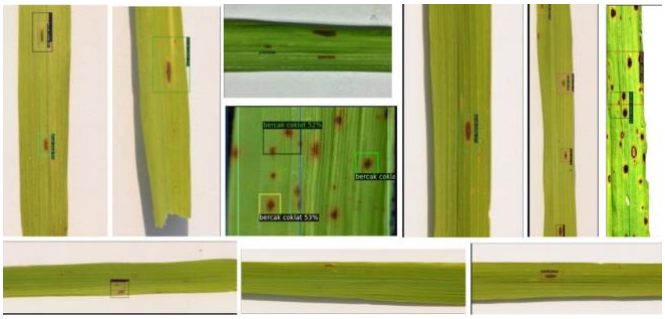


**Fig. 8. Results of inference for the brown spot class: the red box denotes a false negative produced by the model**


**Fig. 7. Inference process for the blight class:** the **red bounding box denotes a detection failure by the model**

Loss Rate is the ratio of incorrect predictions (False Positives + False Negatives) to the total number of predictions. Where True Positives (TP) represent the number of instances where the model correctly predicts the positive class, False Positives (FP) indicate the number of instances where the model incorrectly predicts the positive class, True Negatives (TN) correspond to the number of instances where the model correctly predicts the negative class, and False Negatives (FN) denote the number of instances where the model incorrectly predicts the negative class.

### A. Leaf Blight

**Corresponding author:** Monika Faswia Fahmi, monika.faswiaf@trunojoyo.ac.id, Department of Electrical Engineering, Universitas Trunodjoyo Madura, Madura, Indonesia
**DOI:** https://doi.org/10.35882/ijeeemi.v8i2.287

Based on Table 1, the performance evaluation metrics for rice leaf blight detection are obtained as follows: the model achieves an accuracy of 97.37%, a precision of 1.00, a recall of 0.97, and an F1-score of 98.48%. These metrics indicate that the model performs well at detecting

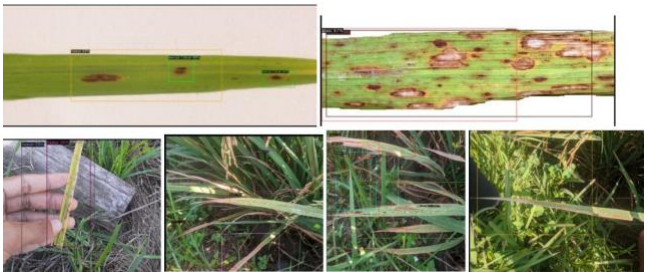


**Fig. 9. Multi-class inference on a single plant instance**

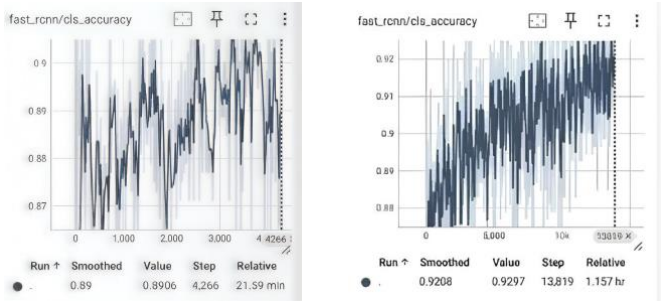
leaf blight, with accuracy reflecting a high level of overall correctness across predictions. The precision value of 1.00 indicates that all predicted positive instances are correctly classified, suggesting the absence of false positives in the evaluated data, while the recall of 0.97 demonstrates the model's strong ability to identify most of the actual positive cases, with only a small proportion of missed detections. Furthermore, the F1-score of 98.48% confirms a well-balanced trade-off between precision and recall, highlighting the model's reliability and consistency in classification performance. Overall, these results suggest that the model is highly effective at detecting leaf blight, although additional evaluation on larger, more diverse datasets would further strengthen confidence in its generalizability.

### B. Rice Blast

Based on Table 1, the performance evaluation metrics for rice blast detection are obtained as follows: the model achieves an accuracy of 94.12%, a precision of 1.00, a recall of 0.94, and an F1-score of 96.91%. These evaluation metrics show that the model achieves strong performance in detecting blast disease, with the accuracy indicating that the majority of cases are correctly classified. A precision of 1.00 suggests that all detected blast cases are accurate, implying no false positives in the evaluated dataset, while a recall of 0.94 reflects the model's ability to identify 94% of the actual blast cases, with a small portion remaining undetected. The F1-score of 96.91% highlights a good balance between precision and recall, demonstrating the model's effectiveness and reliability in classification. Overall, these results confirm that the model performs consistently well in detecting blast disease, although further evaluation on larger and more diverse datasets would help to better assess its robustness and generalization capability.

### C. Brown Spot

Based on Table 1, the performance evaluation metrics for rice brown spot detection are obtained as follows. The model achieves an accuracy of 95.24%, a precision of 1.00, a recall of 0.95, and an F1-score of 97.44%. The results indicate that the model performs effectively in detecting brown spot disease. An accuracy of 95.24% demonstrates that most predictions are correct. A precision of 1 signifies that all positive predictions for




(a) (b)

**Fig. 10. Classification Accuracy: (a) Faster R-CNN (b) Image Enhancement + Faster R-CNN**

brown spot are accurate, while a recall of 0.95 shows that the model successfully identifies 95% of actual cases. With an F1 Score of 97.44%, the model achieves an excellent balance between precision and recall, confirming its high reliability in detecting brown spot disease. Table 1 shows the confusion matrices for each disease class. Based on these, the performance results were as follows:

**Table 1. Summary confusion matrix (accuracy, precision, recall, and F1-score)**

| Disease | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| Blight | 97.37% | 1.0 | 0.97 | 98.48% |
| Blast | 94.12% | 1.0 | 0.94 | 96.91% |
| Brown Spot | 95.24% | 1.0 | 0.95 | 97.44% |

These results indicate that the model consistently achieves high precision (minimizing false positives) while maintaining recall above 0.94, ensuring that most diseased cases are detected. The strong F1 scores confirm a robust balance between precision and recall. The precision value of 1.00 across all classes is attributable to the absence of false positives in the independent test set (n = 90), influenced by the limited test size and the 0.7 confidence threshold applied during inference. While this threshold reduces false alarms, it may increase false negatives, reflecting a precision–recall trade-off. Therefore, the perfect precision should be interpreted as threshold-dependent rather than indicative of universal robustness. Although recall values above 0.94 demonstrate strong sensitivity, the relatively small test set may introduce statistical variability. Future work should validate the model on larger, multi-location datasets and explore adaptive threshold tuning to optimize deployment performance. The results indicate conservative classification behavior, where the applied confidence threshold minimizes false positives, achieving

**Corresponding author:** Monika Faswia Fahmi, monika.faswiaf@trunojoyo.ac.id, Department of Electrical Engineering, Universitas Trunodjoyo Madura, Madura, Indonesia
**DOI:** https://doi.org/10.35882/ijeeemi.v8i2.287

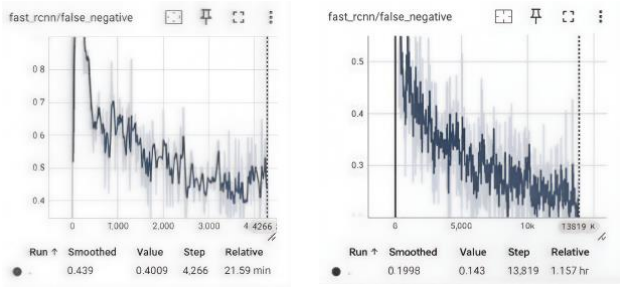
perfect precision but slightly reducing recall due to potential false negatives. While performance appears consistent across classes, the small sample size may limit the reliability of these metrics and lead to an overestimate of stability. Therefore, further validation on larger, more diverse datasets, along with threshold tuning, is necessary to confirm generalizability and achieve more balanced performance.

### D. Classification Accuracy

In the Faster R-CNN framework, *cls_accuracy* denotes the ROI head classification accuracy, measuring how accurately regions of interest (ROIs) are classified into object classes (blight, blast, brown spot) or background proposals. The classification accuracy results are depicted in Fig. 10. In Fig. 10, Graph (a) shows a step count of approximately 4,266 with a relative time of 21.59 minutes. Graph (b) shows a much higher step count, around 13,819, with a relative time of 1.157 hours (or approximately 70 minutes). Additionally, in graph (a), the smoothed accuracy value is 0.8906, with the highest value approaching 0.90, and it appears more fluctuating with some declines. While graph (b) shows a higher smoothed accuracy of 0.9297, indicating that the model at this stage performs better in classification compared to graph (a). Its accuracy is more stable.

### E. False Negative

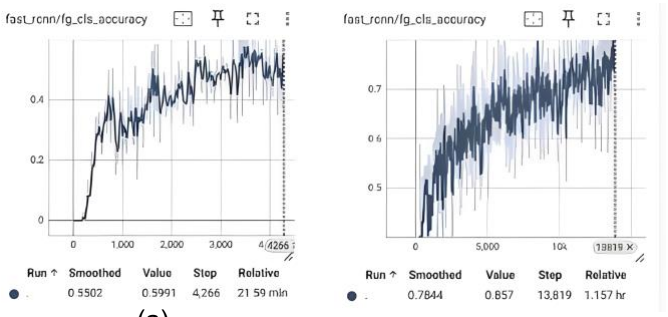
In Faster R-CNN training, false_negative refers to the number of instances where the model incorrectly predicts the absence of an object (foreground) when it is actually present in the image. The results of false negatives are shown in Fig. 11. Based on Fig. 11, Figure 11 indicates that the incorporation of image enhancement leads to a more stable and lower false negative rate throughout training, as evidenced by the consistently reduced smoothed values in graph (b) compared to graph (a). Specifically, graph (a) shows a smoothed false negative value of approximately 0.439, with an initial peak approaching 0.90 that gradually decreases over time, whereas graph (b) presents a lower smoothed value of




(a) (b)

**Fig. 11. False Negative: (a) Faster R-CNN; (b) Image Enhancement + Faster R-CNN**

around 0.1998, indicating a reduced proportion of missed detections. Although both models exhibit a declining trend, suggesting improved detection capability over iterations, the enhanced model converges to a lower error



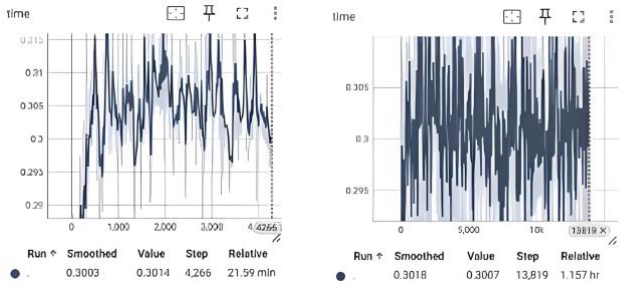

(a) (b)

**Fig. 12. Foreground Classification Accuracy: (a) Faster R-CNN; (b) Image Enhancement + Faster R-CNN**

level, implying better sensitivity in identifying true objects, likely due to improved feature representation, despite minor fluctuations that indicate residual training instability.Figure 11 indicates that incorporating image enhancement leads to a more stable, lower false-negative rate throughout training, as evidenced by the consistently reduced smoothed values in graph (b) compared to graph (a). While both models show a decreasing trend, suggesting improved detection capability over iterations, the enhanced model converges to a lower error level, implying better sensitivity in identifying true objects. This improvement likely stems from clearer feature representations after the enhancement, enabling the detector to reduce missed detections more effectively, although some fluctuations still suggest residual learning instability in later stages.

### F. Foreground Classification Accuracy

In the Faster R-CNN framework, *fg_cls_accuracy* (foreground classification accuracy) reflects the model's ability to correctly classify object proposals identified as




(a) (b)

**Fig. 13. Time Iterations: (a) Faster R-CNN; (b) Image Enhancement + Faster R-CNN**

foreground (positive anchors after IoU thresholding), thereby providing a focused assessment of classification performance independent of background noise. As illustrated in Fig. 12, graph (a) exhibits relatively modest performance, with a smoothed accuracy of approximately 0.55 and peak values nearing 0.60, suggesting that the model is still in an early learning phase where feature representations and class boundaries are not yet well optimized. In contrast, graph (b) demonstrates a

**Corresponding author:** Monika Faswia Fahmi, monika.faswiaf@trunojoyo.ac.id, Department of Electrical Engineering, Universitas Trunodjoyo Madura, Madura, Indonesia
**DOI:** https://doi.org/10.35882/ijeeemi.v8i2.287

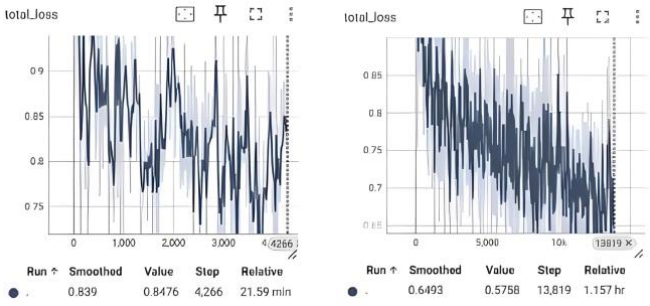
substantial improvement, with a smoothed accuracy of around 0.78 and peak values reaching up to 0.85, indicating that the model has developed a stronger discriminative capability for foreground objects. This improvement suggests that integrating image enhancement leads to clearer, more informative feature extraction, enabling the classifier to better distinguish between object classes. Moreover, the higher and more stable accuracy in graph (b) suggests improved convergence behavior and reduced misclassification among positive samples. Nevertheless, the remaining gap from perfect accuracy indicates that some degree of intra-class similarity or feature ambiguity persists, highlighting the need for further optimization, such as enhanced feature refinement or data diversity, to achieve more robust classification performance in complex real-world scenarios.

### G. Time Iterations

The time-iteration results presented in Fig. 13 highlight notable differences in training dynamics between the baseline Faster R-CNN and the image enhancement–augmented model. As shown in graph (a), the baseline model completes training in a relatively shorter duration (21.59 minutes) with 4,266 steps, indicating a faster but potentially less exhaustive optimization process. In contrast, graph (b) shows a substantially longer training time (1.157 hours) and a significantly higher number of steps (13,819), indicating a more prolonged, iterative learning process. This extended training enables the model to perform more gradient updates, facilitating finer adjustments of network parameters and improved convergence toward a better solution. Consequently, the longer optimization process contributes to better feature learning, reduced loss, and enhanced detection and classification performance, as observed in previous metrics. However, the increased computational cost also entails a trade-off between performance and efficiency, with improved accuracy achieved at the expense of longer training time and greater resource consumption. Therefore, while the enhanced model demonstrates superior learning capability, practical deployment considerations may require balancing training duration with available computational resources. This finding underscores the importance of optimizing training strategies to achieve an effective balance between model performance and computational efficiency.

### H. Total Loss

In Faster R-CNN training , total_loss is the sum of all loss components used to train the model. It represents the overall error or discrepancy between the predicted results and the ground truth. The total-loss results are shown in Fig. 14. From Fig. 14, graph (a) shows that the total loss is relatively high throughout training (in the range of 0.75–0.9), with a smoothed loss of 0.839. The total loss in graph (a) tends to indicate greater fluctuations, with no significant downward trend compared to graph (b). While in graph (b), the loss value was initially high (approaching 0.85–0.9) and gradually decreased to around 0.65. The smoothed loss value is 0.6493, indicating that the model has improved its performance with a significant reduction in loss.




(a) (b)

**Fig. 14. Total Loss: (a) Faster R-CNN; (b) Image Enhancement + Faster R-CNN**

## V. Discussion

### A. Confusion Matrices

The experimental results demonstrate that the proposed Image Enhancement + Faster R-CNN framework achieves strong detection performance across all rice disease classes. The model achieves high accuracy of 97.37% for blight, 94.12% for blast, and 95.24% for brown spot, indicating reliable classification performance. The consistently high precision of 1.00 across all classes confirms that the model effectively minimizes false-positive predictions. Meanwhile, recall values above 0.94 indicate that the model can detect most diseased instances, although a small number of false negatives remain. The F1-scores, all exceeding 96%, further demonstrate a strong balance between precision and recall, confirming the robustness of the detection framework. These results indicate that the proposed preprocessing approach enhances feature clarity, allowing the model to better capture lesion characteristics such as texture and boundary irregularity.

### B. Comparison of the proposed model with only Faster R-CNN without pre-processing

The comparative analysis of the proposed Image Enhancement + Faster R-CNN pipeline and the baseline Faster R-CNN clearly demonstrates substantial, quantifiable improvements across all performance metrics. From a data standpoint, the enhanced model achieves a notable increase in classification accuracy from 0.8906 to 0.9297, accompanied by a significant reduction in false negatives from 0.439 to 0.1998, reflecting stronger sensitivity to subtle disease indicators. These improvements are consistent with prior findings, where preprocessing-assisted CNN frameworks demonstrated enhanced discriminative performance for

**Corresponding author:** Monika Faswia Fahmi, monika.faswiaf@trunojoyo.ac.id, Department of Electrical Engineering, Universitas Trunodjoyo Madura, Madura, Indonesia
**DOI:** https://doi.org/10.35882/ijeeemi.v8i2.287

rice disease datasets captured under controlled imaging scenarios [38], [39].

From a value perspective, the reduction in total loss from 0.839 to 0.6493 highlights the role of grayscale conversion and contrast stretching in improving feature separability and stabilizing the learning process. This result aligns with [40] and [41], who reported that illumination normalization and contrast-based enhancements facilitate more effective convergence in deep learning pipelines by emphasizing fine-grained texture patterns vital for disease identification. Although these improvements come with increased computational load, as evidenced by the rise in training iterations from 4,266 to 13,819, such trade-offs are commonin enhancement-based detection models and remain acceptable for high-precision analysis scenarios [40].

From a performance perspective, foreground classification accuracy increases from 0.55 to 0.78, accompanied by smoother and more stable learning curves. These observations echo those of [42] and [43], who demonstrated that multiscale representation enhancement, whether through preprocessing or attention-based modules, is critical for distinguishing foreground lesion patterns from complex backgrounds. The improved stability also reflects the model's strengthened ability to capture morphological properties of rice diseases, consistent with the robustness enhancements discussed by [44] and [45].

Although the enhanced pipeline improves detection accuracy and stability, it increases training time from 21.59 minutes to approximately 70 minutes due to additional preprocessing and extended optimization steps. This reflects a computational trade-off between accuracy and efficiency. Notably, preprocessing is applied during both training and inference; however, it does not significantly increase inference latency (<12% additional processing time per image). Therefore, while training complexity increases, deployment feasibility remains practical for offline or semi-real-time agricultural diagnostic systems.

Despite these improvements, several limitations must be acknowledged. First, the controlled acquisition conditions of the dataset, stable lighting, minimal background variation, and uniform distance, may artificially inflate performance. The risk of performance degradation under natural field conditions is supported by prior studies, where CNN-based rice disease detectors exhibited reduced generalization when exposed to heterogeneous illumination and complex vegetation backgrounds [38], [39]. Second, while inference latency remains within acceptable limits for semi-real-time applications, the increased overall computational demand may constrain deployment in fully real-time, resource-limited environments [40], [41].

Furthermore, although performance gains are evident, the enhanced Faster R-CNN still exhibits limitations in detecting early-stage symptoms, fine-grained lesions, and overlapping disease regions. Early-stage infections often present low texture contrast and subtle morphological variations, limiting feature separability even after contrast enhancement. Additionally, as an anchor-based detector, Faster R-CNN may inadequately capture very small or irregular lesion regions. Overlapping disease areas further introduce ambiguity during IoU-based proposal matching, potentially affecting bounding box regression accuracy. These constraints are widely recognized in two-stage detection frameworks. Recent studies indicate that anchor-free detectors or transformer-based attention mechanisms may better capture fine-grained pathological patterns, particularly for small or overlapping targets [42], [43], [46].

Finally, the study only explores grayscale and contrast stretching as preprocessing approaches. Prior research has shown substantial benefit from alternative methods such as CLAHE, Retinex-based illumination correction, GAN-assisted augmentation, and hybrid CNN Transformer integrations [44], [45]. The absence of comparative preprocessing baselines limits the ability to make broader claims about the optimality of the proposed strategy. Overall, the improved accuracy, reduced loss, enhanced stability, and lower false negatives validate the effectiveness of the proposed pipeline, while the observed limitations provide a clear direction for future enhancements involving multi-condition datasets, advanced preprocessing algorithms, and next-generation detection architectures. These findings have important practical and methodological implications. From a practical perspective, the high detection accuracy and robustness suggest that the proposed framework has strong potential for application in agricultural monitoring systems to support early diagnosis of rice diseases, thereby enabling timely intervention and reducing potential crop losses. From a methodological standpoint, the results demonstrate that relatively simple preprocessing techniques can significantly enhance deep learning performance without requiring complex architectural modifications. This highlights the critical role of data quality and feature enhancement in computer vision pipelines, particularly for tasks involving subtle texture variations such as plant disease detection.

## VI. Conclusion

This study aimed to evaluate the effectiveness of integrating image preprocessing techniques, namely RGB to grayscale conversion and contrast stretching, with the Faster R-CNN framework for rice disease detection using field-acquired image datasets. The experimental results demonstrate that the proposed approach achieves strong detection performance across three disease classes, with accuracies of 97.37% for blight, 94.12% for blast, and 95.24% for brown spot. In addition, compared to the baseline Faster R-CNN, the enhanced model improves

**Corresponding author:** Monika Faswia Fahmi, monika.faswiaf@trunojoyo.ac.id, Department of Electrical Engineering, Universitas Trunodjoyo Madura, Madura, Indonesia
**DOI:** https://doi.org/10.35882/ijeeemi.v8i2.287

classification accuracy from 0.8906 to 0.9297, reduces false negatives from 0.439 to 0.1998, increases foreground classification accuracy from 0.55 to 0.78, and decreases total loss from 0.839 to 0.6493. These findings indicate that the applied preprocessing techniques improve feature clarity and contribute to better model performance and stability. For future work, further evaluation is required using larger and more diverse datasets captured under varying environmental conditions to assess generalization capability. In addition, integrating the proposed preprocessing strategy with more advanced detection architectures, such as YOLO-based or transformer-based models, may further improve detection accuracy, robustness, and computational efficiency.

## Acknowledgment

The authors gratefully acknowledge financial support from the Directorate of Research and Community Service, Directorate General of Research and Development, Ministry of Higher Education, Science, and Technology of the Republic of Indonesia, under the 2026 research funding scheme.

## References

[1] Food and Agriculture Organization of the United Nations, *World Food and Agriculture – Statistical Yearbook 2021*. Rome, Italy: FAO, 2021.

[2] S. Savary, L. Willocquet, S. J. Pethybridge, P. Esker, N. McRoberts, and A. Nelson, "The global burden of pathogens and pests on major food crops," *Nat. Ecol. Evol.*, vol. 3, pp. 430–439, 2019, doi: 10.1038/s41559-018-0793-y

[3] P. Skamnioti and S. J. Gurr, "Against the grain: Safeguarding rice from rice blast disease," *Trends Biotechnol.*, vol. 27, no. 3, pp. 141–150, 2009, doi: 10.1016/j.tibtech.2008.12.002

[4] D. O. Niño-Liu, P. C. Ronald, and A. J. Bogdanove, "Xanthomonas oryzae pathovars: Model pathogens of a model crop," *Mol. Plant Pathol.*, vol. 7, no. 5, pp. 303–324, 2006, doi: 10.1111/j.1364-3703.2006.00344.x

[5] Ministry of Agriculture, *Plant Protection Annual Report*, 2022.

[6] A.-K. Mahlein, "Plant Disease Detection by Imaging Sensors - Parallels and Specific Demands for Precision Agriculture and Plant Phenotyping," *Plant Dis.*, vol. 100, no. 2, pp. 1–11, 2016, doi: 10.1094/PDIS-03-15-0340-FE

[7] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep neural networks based recognition of plant diseases by leaf image classification," *Comput. Intell. Neurosci.*, vol. 2016, 2016, doi: 10.1155/2016/3289801

[8] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Front. Plant Sci.*, vol. 7, 2016, doi: 10.3389/fpls.2016.01419

[9] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," *Comput. Electron. Agric.*, vol. 161, pp. 272–279, 2019, doi: 10.1016/j.compag.2018.03.032

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

[11] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, "Identification of rice diseases using deep convolutional neural networks," *Neurocomputing*, vol. 267, pp. 378–384, 2017, doi: 10.1016/j.neucom.2017.06.023

[12] G. Zhou, W. Zhang, A. Chen, M. He, and X. Ma, "Rapid detection of rice disease based on FCM-KM and Faster R-CNN fusion," *IEEE Access*, vol. 7, pp. 143190–143206, 2019, doi: 10.1109/ACCESS.2019.2943454

[13] P. K. Sethy, N. K. Barpanda, A. K. Rath, and S. K. Behera, "Rice false smut detection based on Faster R-CNN," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 19, no. 3, pp. 1590–1595, 2020, doi: 10.11591/ijeecs.v19.i3.pp1590-1595

[14] S. P. and J. Sil, "Rice disease identification using pattern recognition techniques," *11th Int. Conf. Comput. Inf. Technol. Khulna, Bangladesh*, pp. 420–423, 2008, doi: doi: 10.1109/ICCITECHN.2008.4803079

[15] S. Ramesh and D. Vydeki, "Recognition and classification of paddy leaf diseases using optimized deep neural network with Jaya algorithm," *Inf. Process. Agric.*, vol. 7, no. 2, pp. 249–260, 2020, doi: 10.1016/j.inpa.2019.09.002

[16] A. Fuentes, S. Yoon, S. C. Kim, and D. S. Park, "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition," *Sensors*, vol. 17, no. 9, 2017, doi: 10.3390/s17092022

[17] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Comput. Electron. Agric.*, vol. 145, pp. 311–318, 2018, doi: 10.1016/j.compag.2018.01.009

[18] J. G. A. Barbedo, "Plant disease identification from individual lesions and spots using deep learning," *Biosyst. Eng.*, vol. 180, pp. 96–107, 2019, doi: 10.1016/j.biosystemseng.2019.02.002

[19] S. Coulibaly, B. Kamsu-Foguem, D. Kamissoko, and D. Traore, "Deep neural networks with transfer learning in millet crop images," *Comput. Ind.*, vol. 108, pp. 115–120, 2019, doi: 10.1016/j.compind.2019.02.003

[20] A. Picon *et al.*, "Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild," *Comput. Electron. Agric.*, vol. 161, pp. 280–290, 2019, doi: 10.1016/j.compag.2018.04.002

[21] W. H. Zeng *et al.*, "Identification of maize leaf diseases using SKPSNet-50 Convolutional Neural

**Corresponding author:** Monika Faswia Fahmi, monika.faswiaf@trunojoyo.ac.id, Department of Electrical Engineering, Universitas Trunodjoyo Madura, Madura, Indonesia
**DOI:** https://doi.org/10.35882/ijeeemi.v8i2.287

Network Model," *Sustain. Comput. Informatics Syst.*, vol. 35, 2022, doi: 10.1016/j.suscom.2022.100695
[22] G. O. and E. Mwebaze, "Machine learning for plant disease incidence and severity measurements from leaf images," in *Proc. IEEE ICMLA*, 2016, pp. 158–163, doi: 10.1109/ICMLA.2016.0034
[23] G. Polder *et al.*, "Automatic detection of tulip breaking virus (TBV) using a deep convolutional neural network," *IFAC-PapersOnLine*, vol. 52, no. 30, pp. 12–17, 2019, doi: 10.1016/j.ifacol.2019.12.482
[24] M. Shafay *et al.*, "Recent advances in plant disease detection: Challenges and opportunities," *Plant Methods*, vol. 21, no. 1, 2025, doi: 10.1186/s13007-025-01450-0
[25] A. Y. Ashurov *et al.*, "Enhancing plant disease detection through deep learning," *Front. Plant Sci.*, vol. 15, 2024, doi: 10.3389/fpls.2024.1505857
[26] S. U. Khan *et al.*, "A review on automated plant disease detection: motivation, limitations, challenges, and recent advancements for future research," 2025, doi: 10.1007/s44443-025-00040-3
[27] A. Upadhyay *et al.*, "Deep learning and computer vision in plant disease detection," *Artif. Intell. Rev.*, vol. 58, 2025, doi: 10.1007/s10462-024-11100-x
[28] L. C. Ngugi, M. Abelwahab, and M. Abo-Zahhad, "Recent advances in image processing techniques for automated leaf pest and disease recognition," *Inf. Process. Agric.*, vol. 8, no. 1, pp. 27–51, 2021, doi: 10.1016/j.inpa.2020.04.004
[29] P. Sridhar and P. Angamuthu, "Enhancing image based classification for crop disease detection using a multiclass SVM approach with kernel comparison," *Scientific Reports*, vol. 15, art. no. 40055, 2025, doi: 10.1038/s41598-025-23568-w
[30] T. W. Mew, "Current status and future prospects of research on bacterial blight of rice," *Annual Review of Phytopathology*, vol. 25, pp. 359–382, 1987, doi: 10.1146/annurev.py.25.090187.002043
[31] D. W. Utami *et al.*, "The pathogenicity and genetic diversity of Indonesian blast pathogen from wide host ranges of rice sub-species," *J. Plant Pathol.*, vol. 107, pp. 661–673, 2025, doi: 10.1007/s42161-024-01815-9
[32] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031
[33] B. Liu, W. Zhao and Q. Sun, "Study of object detection based on Faster R-CNN," *2017 Chinese Automation Congress (CAC)*, Jinan, China, 2017, pp. 6233-6236, doi: 10.1109/CAC.2017.8243900
[34] Z. Zou, K. Chen, Z. Shi, Y. Guo and J. Ye, "Object Detection in 20 Years: A Survey," in *Proceedings of the IEEE*, vol. 111, no. 3, pp. 257-276, March 2023, doi: 10.1109/JPROC.2023.3238524
[35] Y. Amit, P. Felzenszwalb, and R. Girshick, "Object detection," in *Computer Vision: A Reference Guide*, Cham, Switzerland: Springer International Publishing, 2021, pp. 875–883, doi: 10.1007/978-3-030-63416-2_660
[36] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Machine Learning (ICML)*, 2019, pp. 6105–6114, doi: 10.48550/arXiv.1905.11946
[37] Z. Liu *et al*., "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada, 2021, pp. 9992-10002, doi: 10.1109/ICCV48922.2021.00986
[38] T. Ahad *et al.*, "Comparison of CNN-based architectures for rice disease classification," *Artif. Intell. Agric.*, vol. 9, pp. 22–35, 2023, doi: 10.1016/j.aiia.2023.07.001
[39] C. K. Sunil, C. D. Jaidhar, and N. Patil, "Systematic study on deep learning-based plant disease detection or classification," *Artificial Intelligence Review*, vol. 56, pp. 14955–15052, 2023, doi: 10.1007/s10462-023-10517-0
[40] M. H. Saleem, J. Potgieter, and K. M. Arif, "Plant disease detection and classification by deep learning," *Plants*, vol. 8, no. 11, p. 468, 2019, doi: 10.3390/plants8110468
[41] A. G. Sebastián *et al.*, "Enhancing plant disease detection: Incorporating Advanced CNN Architectures for Better Accuracy and Interpretability," *Int. J. Comput. Intell. Syst.*, 2025, doi: 10.1007/s44196-025-00835-2
[42] F. Martinelli, R. Scalenghe, S. Davino *et al*., "Advanced methods of plant disease detection: A review," *Agronomy for Sustainable Development*, vol. 35, pp. 1–25, 2015, doi: 10.1007/s13593-014-0246-1
[43] C. Zhang *et al.*, "Lightweight multi-scale CNN for rice disease recognition," 2023, doi: 10.32604/cmc.2023.027269
[44] A. S. V. and S. F. Sayyad, "Enhancing agricultural sorting systems through image-based classification and machine learning," *Multimed. Tools Appl.*, 2026, doi: 10.1007/s11042-026-21155-3
[45] Y. Xu *et al*., "FDViT: Improve the Hierarchical Architecture of Vision Transformer," *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, Paris, France, 2023, pp. 5927-5937, doi: 10.1109/ICCV51070.2023.00547
[46] A. Dosovitskiy *et al.*, "An image is worth 16×16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020, doi: 10.48550/arXiv.2010.11929

**Corresponding author:** Monika Faswia Fahmi, monika.faswiaf@trunojoyo.ac.id, Department of Electrical Engineering, Universitas Trunodjoyo Madura, Madura, Indonesia
**DOI:** https://doi.org/10.35882/ijeeemi.v8i2.287

## Author Biography

**Monika Faswia Fahmi** was born in Tulungagung, East Java, on July 28, 1994. She received her Bachelor's degree from the Department of Electrical Engineering, Faculty of Engineering, Universitas Negeri Malang in 2016, and completed her Master's degree at Institut Teknologi Bandung in 2019. Since 2022, she has been serving as an Assistant Lecturer in the Department of Electrical Engineering at Universitas Trunojoyo Madura. Her main research interests include control systems, intelligent systems, robotics, machine learning, and their applications in automation and sensor-based monitoring. In addition to her teaching responsibilities, she has actively contributed to research projects and publications in computational intelligence, digital signal processing, renewable energy systems, and precision agriculture. She has participated in national research grants, community service programs, and collaborative innovation initiatives to support smart farming and the adoption of sustainable technologies. She has published several articles in journals and conferences and continues to expand her work in intelligent automation and applied machine learning. She can be contacted by email at monika.faswiaf@trunojoyo.ac.id.

**Deni Tri Laksono** was born in Banyuwangi, East Java, on November 29, 1992. He has been a lecturer at the Department of Electrical Engineering, Faculty of Engineering, Universitas Trunojoyo Madura, since 2019. He received his Bachelor's degree in Electrical Engineering Education from Universitas Negeri Malang and completed his Master's degree in Electrical Engineering from Institut Teknologi Bandung. His primary research interests include power electronics and renewable energy. He has been actively involved in research and development projects related to partial discharge measurement, people count detection using convolutional neural networks, off-grid solar energy systems, and energy optimization for sustainable engineering applications. He can be contacted at email: deni.laksono@trunojoyo.ac.id

**Achmad Fiqhi Ibadillah** is an Assistant Lecturer in the Department of Electrical Engineering at Universitas Trunojoyo Madura. He obtained his B.Eng. (S.T.) in Electrical Engineering and his M.Sc. degree in Electronic and Computer Engineering from National Taiwan University of Science and Technology (NTUST). His research interests include sensor and actuator systems, data acquisition, machine learning, and embedded systems. In addition to teaching, he has been actively involved in applied research projects in smart instrumentation, automation, IoT-based monitoring, and intelligent sensing technologies. His work focuses on bridging theoretical engineering concepts with practical, real-world system implementation to support advancements in modern automation and intelligent computing. He can be contacted at email: fiqhi.ibadillah@trunojoyo.ac.id

**Dedi Tri Laksono** was born in Banyuwangi, East Java, on November 29, 1992. He has been a lecturer in the Department of Electrical Engineering Diploma Program at the Politeknik Negeri Padang, where he is actively engaged in teaching and applied research. His research domain includes electrical power systems and renewable energy technologies. He has contributed to studies on energy monitoring systems, off-grid solar power development, automatic transfer switch design, and the use of TEV and RC-based techniques for partial discharge detection. His academic and professional work focuses on developing reliable and efficient solutions for advancing modern power and energy systems. He can be contacted at email: deditrilaksono@pnp.ac.id

**Corresponding author:** Monika Faswia Fahmi, monika.faswiaf@trunojoyo.ac.id, Department of Electrical Engineering, Universitas Trunodjoyo Madura, Madura, Indonesia
**DOI:** https://doi.org/10.35882/ijeeemi.v8i2.287