Accredited SINTA 2 Ranking
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online at: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 9 No. 1 (2025) 10 - 19 e-ISSN: 2580-0760

Identifying Rice Plant Damage Due to Pest Attacks Using Convolutional Neural Networks

Andi Tenriola[1*], Putri Alysia Azis[2], Andi Baso Kaswar[3], Fhatiah Adiba[4], Dyah Darma Andayani[5]
[1,2,3,4,5] Department of Informatics and Computer Science, Faculty of Engineering, Universitas Negeri Makassar, Makassar, Indonesia
[1]andtnriola@gmail.com, [2]putrialysia.djarre@gmail.com, [3]a.baso.kaswar@unm.ac.id, [4]adibafhatiah@unm.ac.id, [5]dyahdarma@unm.ac.id

Abstract

Rice (Oryza Sativa) is an important crop for meeting global food needs; however, one of the main challenges in its cultivation is the attack of stem borer pests, which can cause significant damage. This study aims to identify the damage caused by these pest attacks using Convolutional Neural Networks (CNN) methods. We developed and trained several CNN architectures, including the proposed architecture, MobileNet, and EfficientNetB0, to detect pest attacks on rice. The dataset used consists of 700 images per class taken directly from the field, where the images depict rice plants that have been peeled or opened to inspect for the presence of pests, specifically stem borer pests. To enhance the quality and diversity of the dataset, we applied a rigorous selection process, ensuring that only high-quality images were used. Additionally, augmentation techniques such as rotation were employed to expand the dataset to 2000 images per class. Labeling was carried out carefully to ensure that each image accurately reflected the condition of the pest attack. The results of the study indicate that the proposed CNN model can identify damage with high accuracy, thereby contributing to efforts to increase rice production through early detection of pest attacks using computer vision technology.

Keywords: *convolutional neural networks; rice; stem borer pest*

How to Cite: A. Tenriola, P. A. Azis, A. B. Kaswar, F. Adiba, and D. D. Andayani, "Identifying Rice Plant Damage Due to Pest Attacks Using Convolutional Neural Networks", *J. RESTI (Rekayasa Sist. Teknol. Inf.)* , vol. 9, no. 1, pp. 10 - 19, Jan. 2025.
DOI: https://doi.org/10.29207/resti.v9i1.6125

1. Introduction

Rice (Oryza Sativa) is a plant native to India and Indochina [1]. It has become one of the world's staple food sources and plays a vital role in meeting global food demands, reaching up to 33% of the population in various countries [2], [3]. In Indonesia, rice has been known since around 1500 BC and the country is recognized as having the highest rice consumption rate in Asia [1]. According to available data, rice production in 2021 was recorded at 54.42 million tons, showing a significant decrease compared to rice production in 2020, which amounted to 54.65 million tons [4]. In 2022, rice production increased by 0.61% to 54.75 million tons [5], but again decreased in 2023 to 53.98 million tons [6]. One of the main factors contributing to the decline in rice production is pest and disease attacks, which account for 24.6%–40.9% [7].

Pests and plant diseases are among the organisms that can interfere with crops, causing damage and leading to economic losses [8]. Pest attacks on rice plants can result in disturbances in plant growth, delays or failures in flower formation, disrupted panicle development, reduced yields, and even failures to achieve satisfactory harvests [9].

One of the common pest attacks on rice plants is from the rice stem borer [10]. This type of pest attacks rice plants from the early seedling phase until just before harvest. During the growth stage, the larvae cut through the plant's central stem, disrupting nutrient flow and causing wilting and death of the plant (sundep). During grain formation, the larvae damage the stem, preventing nutrients from reaching the rice grains (beluk) [11]. Therefore, prevention and management efforts against this pest attack are crucial to ensure stable and high-quality rice production.

Received: 21-10-2024 | Accepted: 23-12-2024 | Published Online: 19-01-2024

At present, technology is crucial in managing pest infestations in rice plants, particularly through the use of computer vision. Computer vision is a field within Artificial Intelligence (AI) that allows computers to interpret and extract information from images, videos, and other visual inputs, enabling them to make decisions or offer recommendations based on that data. This technology relies on large datasets and powerful Machine Learning (ML) algorithms, especially Deep Learning methods like Convolutional Neural Networks (CNN) [12]. CNN is a type of artificial neural network designed to process data in the form of grids, such as images, with neurons that have learnable weights and biases. Image inputs are passed through these neurons, which perform mathematical operations using weights and biases, and are then processed with a non-linear activation function. The three-dimensional structure of CNN, with layers such as convolutional layers, pooling layers, and fully connected layers, allows CNN to automatically extract features from images [13].

Several research efforts have explored this area. In 2023, a study presented a CNN model that integrates elements from both Inception and ResNet architectures to detect disease characteristics in rice plants [14]. The primary goal of this research was to reduce the error rate in identifying diseases by integrating disease characteristics with contextual information. The enhanced CNN model incorporated the CBAM module for more precise feature extraction. Moreover, BiGRU was employed to capture relationships between image features, improving the model’s ability to understand disease structures in rice plants. Experimental findings demonstrated the model's effectiveness, achieving higher accuracy and lower costs compared to other models. In a different study, an algorithm was developed to automatically detect various stages of rice panicle development (AI-MDSRS), transforming the identification process into detecting rice panicles at different maturity stages. This model applied several optimization techniques, such as replacing VGG16 with Inception_ResNet-v2 for feature extraction, using a feature pyramid network (FPN) instead of a single-scale feature map, and incorporating Distance Intersection over Union (DIoU) as a non-maximum suppression (NMS) criterion. The results showed an accuracy of 92.47%, significantly outperforming the original Faster R-CNN and YOLOv4 models [15].

Additionally, another study centered on identifying rice leaf diseases emphasized the effectiveness of Deep Learning, especially CNN, in detecting and classifying leaf diseases in rice plants. The proposed custom VGG16 model exhibited outstanding performance in recognizing and categorizing nine types of rice leaf diseases, achieving an impressive accuracy of 99.94%. Among six models that were chosen and retrained, VGG16 excelled in precision, recall, and overall accuracy [16]. Furthermore, a study examined various CNN-based architectures for classifying diseases in rice. This research assessed the effectiveness of the original CNN, transfer learning, and ensemble models in identifying rice leaf diseases. Six distinct CNN models DenseNet121, Inceptionv3, MobileNetV2, ResNext101, ResNet152V, and SeresNext101 were tested across nine categories of rice diseases using a dataset of 14,118 images of rice leaves. DenseNet121 achieved the highest classification results, reinforcing previous findings that emphasize its capability to learn from all preceding layers [17].

Additionally, recent research in 2024 explored the use of the ResNet101-SE-LSTM model for identifying the nutritional levels of rice by integrating spatial and temporal aspects. The findings of this study indicated that the model successfully achieved the highest accuracies of 85.88% and 88.38% for HHZ and XS134 in 2021. Furthermore, the model demonstrated good generalization capabilities, achieving accuracies of 81.25% and 82.50% for HHZ and XS134 in 2022. This approach proved to be more efficient than traditional methods that rely on destructive detection and can provide real-time information to field workers during the rice fertilization process to enhance yields [18].

Based on the research findings outlined earlier, it is evident that numerous studies have successfully implemented computer vision technology and demonstrated its significant potential in the agricultural sector. However, no research has specifically addressed rice plant damage caused by stem borer pests, focusing on the specific characteristics of rice plant stems. Therefore, this study aims to bridge the existing gap in detecting rice plant damage caused by stem borer infestations by utilizing a specially developed Convolutional Neural Network (CNN) method.

Although models such as YOLO (You Only Look Once) and Faster R-CNN have demonstrated success in general tasks like disease identification and maturity stage detection, they are less effective in detecting localized and specific damage. For example, YOLO struggles with identifying small objects with irregular shapes, often producing inaccurate bounding boxes [19]. On the other hand, while Faster R-CNN excels in detecting small details, its slow inference speed and high computational demands make it unsuitable for real-time applications in the field. Damage caused by rice stem borer pests has unique characteristics, such as small holes or subtle discoloration, which conventional object detection models often fail to recognize. To address this issue, this study proposes an optimized Convolutional Neural Network (CNN) method designed to capture these specific visual patterns. The proposed CNN architecture offers several advantages, including optimized layers that gradually enhance convolutional filters to extract complex features from stem damage, reduction of overfitting through the use of dropout layers and batch normalization techniques to improve the model's generalization, and computational efficiency that supports real-time applications in field settings.

The integration of this CNN model into pest management strategies enables early detection of stem borer infestations, providing significant advantages for farmers. In addition to its high accuracy, the computational efficiency it offers ensures that this method can be widely adopted, even in resource-constrained agricultural environments, delivering a scalable and cost-effective solution to support sustainable farming.

2. Research Methods

To identify the damage caused to rice plants as a result of stem borer pest attacks, this study undergoes a series of detailed stages can be seen in Figure 1.

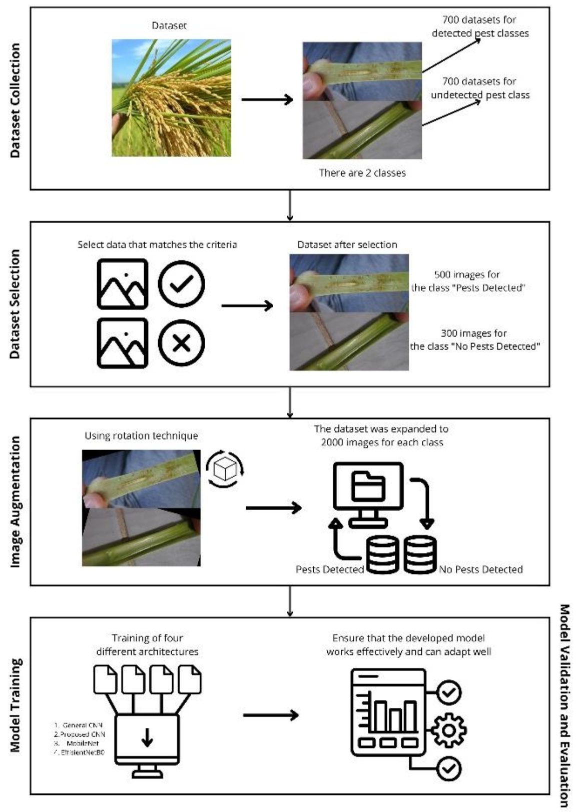

Figure 1. Research Method Diagram

2.1 Dataset Collection

At this stage, a dataset was collected that includes approximately 700 images for each class: "Detected Pests" (Fig. 1) and "Not Detected Pests" (Fig. 2). All of these images were obtained through a series of image-capturing processes conducted directly in the field using cameras and smartphones. This image acquisition process involved physical presence at agricultural sites to conduct in-depth observations and thorough inspections of the rice plants.

The purpose of this activity is to identify and classify rice plants that have been infected by stem borer pests, as well as to distinguish them from rice plants that are in normal condition or not affected by pests. This classification is crucial for developing effective pest management strategies and improving overall crop health.

Figure 2. Pest Detected Dataset

Figure 3. Pest Not Detected Dataset

2.2 Dataset Selection

During the selection phase, it was found that approximately 500 images met the criteria to be retained as representatives of the "Detected Pests" class, while around 300 images remained to represent the "Not Detected Pests" class. This reduction resulted from a series of selection processes that involved careful evaluation of each collected image. Images assessed to be of low quality or not meeting the research criteria were discarded, including those with inadequate lighting, blurry images, and images that were irrelevant to the intended subject.

The selection process aims to ensure the integrity and accuracy of the images used in the subsequent analysis. By retaining only high-quality images, it is expected that the analysis results can provide more accurate and reliable information for pest detection. The use of a clean and standardized dataset will strengthen the developed model, thereby enhancing its effectiveness in identifying and classifying rice plants based on the presence of pests.

2.3. Image Augmentation

During the image augmentation stage, the total dataset was expanded to 2000 images for each class: "Detected Pests" and "Not Detected Pests." This augmentation process included image rotation techniques, allowing images to be altered at various angles to simulate different viewpoints that may occur in real-world scenarios. The primary goal of this augmentation is to ensure a representation that more closely resembles actual field conditions, as images of rice plants can vary significantly due to environmental factors such as sunlight, humidity, and different growth stages.

Increasing the number of samples in the dataset not only enhances the variety of inputs but also reduces the risk of overfitting, ultimately resulting in a more robust and generalized model. With this enhanced dataset, the developed models are expected to achieve improved performance and higher accuracy in detecting the presence of pests. This is crucial for effective pest management, enabling timely interventions to protect rice crops from significant damage.

2.4 Training the Model

This study focuses on training four different architectures to detect stem borer pests in rice crops: a standard Convolutional Neural Network (CNN), a CNN with the proposed architecture, the MobileNet architecture an efficient, lightweight CNN model optimized for use on smartphones and other edge devices and the EfficientNetB0 architecture, a scalable and efficient series of CNN models [20].

In the general CNN architecture, the image dataset is resized to 216x288 pixels, and the labels are transformed into one-hot encoding. The dataset is then split into 80% for training and 20% for testing, ensuring the model is evaluated on data separate from what it was trained on. This basic CNN design includes multiple convolutional layers, which utilize filters and the ReLU activation function to extract key features from the images.

These convolutional layers are followed by MaxPooling layers, which aim to reduce the feature dimensions and minimize overfitting by decreasing the spatial resolution of the extracted features. This process is repeated with several additional convolutional and MaxPooling layers to deepen and enrich the features extracted from the images. Figure 4 is an overview of the CNN architecture:

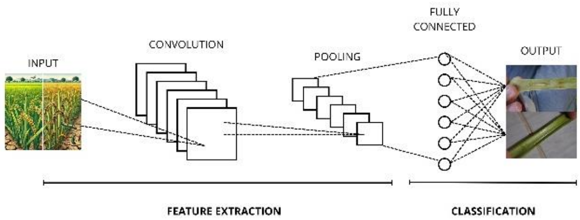

Figure 4. General CNN architecture

After extracting features, a flattened layer is applied to convert the feature matrix generated by the earlier layers into a one-dimensional vector, making it suitable for processing by the fully connected layer. This fully connected layer is composed of multiple neurons, each connected to all neurons in the preceding layer. The softmax activation function is then employed in this layer to classify the output into two categories: "Pest Detected" and "Pest Not Detected." The specifics of these layers are outlined in Table 1.

Table 1. Layers of the General CNN Architecture

Layer	Layer Type	Output Dimensions	Number of Filters	Kernel Size	Activation Function
1	Conv2D	(None, 214, 286, 10)	10	3x3	ReLU
2	Conv2D	(None, 212, 284, 10)	10	3x3	ReLU
3	MaxPooling2D	(None, 106, 142, 10)	-	2x2	-
4	Conv2D	(None, 104, 140, 10)	10	3x3	ReLU
5	Conv2D	(None, 102, 138, 10)	10	3x3	ReLU
6	MaxPooling2D	(None, 51, 69, 10)	-	2x2	-
7	Flatten	(None, 35190)	-	-	-
8	Dense	(None, 2)	-	-	Softmax

The proposed Convolutional Neural Network (CNN) architecture is designed with several significant enhancements to address the complexity of image processing, particularly in detecting subtle damage to rice stems caused by stem borer pests. One key aspect of this architecture is the use of a 216x288-pixel image resolution. This resolution was selected to maintain the aspect ratio and ensure optimal visualization of the structural details of stem damage. Preliminary experiments indicate that this resolution delivers superior performance in detecting damage with subtle nuances.

The proposed CNN architecture employs a progressive approach by increasing the number of convolutional filters from 32 to 128. This enhancement allows the model to gradually extract increasingly complex and abstract features, thereby enriching the resulting data representation. These additional filters enhance the model's ability to capture intricate features that are often overlooked by conventional architectures.

To optimize efficiency and prevent overfitting, the architecture integrates 2x2 pooling layers, which efficiently reduce data dimensions and computational load. This step accelerates the training process by enhancing convergence and decreasing the number of parameters the model needs to learn. Additionally, a dropout layer with a 50% ratio is applied after fully connected (Dense) layers with 256 and 128 units. This dropout layer functions to minimize the risk of overfitting and improve the model's generalization, particularly when dealing with variations in the training data.

The Dense layers in this architecture play a crucial role in learning more complex and abstract features, enabling the model to understand deeper patterns within the data. At the output layer, a softmax activation function is used to produce reliable class probabilities for classification tasks. With this approach, the proposed CNN not only achieves high accuracy in predicting target classes but also provides strong confidence levels for each prediction, making it highly effective in precisely detecting pest damage in agricultural settings. Table 2 outlines the specific layers used in the proposed CNN architecture:

Table 2. Layers of the Proposed CNN Architecture

Layer	Layer Type	Output Dimensions	Number of Filters	Kernel Size	Activation Function
1	Conv2D	(None, 214, 286, 32)	32	3x3	ReLU
2	Conv2D	(None, 212, 284, 32)	32	3x3	ReLU
3	MaxPooling2D	(None, 106, 142, 32)	-	2x2	-
4	Conv2D	(None, 104, 140, 64)	64	3x3	ReLU
5	Conv2D	(None, 102, 138, 64)	64	3x3	ReLU
6	MaxPooling2D	(None, 51, 69, 64)	-	2x2	-
7	Conv2D	(None, 49, 67, 128)	128	3x3	ReLU
8	Conv2D	(None, 47, 65, 128)	128	3x3	ReLU
9	MaxPooling2D	(None, 23, 32, 128)	-	2x2	-
10	Flatten	(None, 94208)	-	-	-
11	Dense	(None, 256)	-	-	ReLU
12	Dropout	(None, 256)	-	-	-
13	Dense	(None, 128)	-	-	ReLU
14	Dropout	(None, 128)	-	-	-
15	Dense	(None, 2)	-	-	Softmax

The MobileNet architecture is designed for efficiency on resource-constrained devices, such as smartphones and edge devices. In its implementation, input images are resized to 224x224 pixels. This resolution strikes an optimal balance between preserving image details and reducing computational complexity, ensuring efficiency without compromising the model's ability to extract critical features. Additionally, converting labels to one-hot encoding facilitates a more effective training process.

The dataset is divided into two parts, with 80% allocated for training and 20% for testing, ensuring that the model is evaluated on unseen data. This is crucial for accurately assessing the model's generalization capability. MobileNet features several convolutional layers with progressively increasing filter sizes of 32, 64, and 128, followed by pooling layers to reduce feature dimensions. This reduction enhances efficiency in capturing essential characteristics from the images. After the feature extraction process, a flattening layer is applied to convert the feature matrix into a one-dimensional vector, which is then processed by a dense layer with 128 units and a ReLU activation function. The output layer employs a softmax activation function for classification based on the number of classes in the dataset.

The lightweight and optimized MobileNet architecture, designed for 224x224 input size, enables fast inference, making it ideal for real-time applications and mobile devices. Despite its efficiency, the model strikes a good balance between speed and accuracy, allowing for practical deployment in various resource-constrained environments. Table 3 summarises the details of the layers in the MobileNet architecture:

Table 3. Layers of the MobileNet Architecture

Layer	Layer Type	Output Dimensions	Number of Filters	Kernel Size	Activation Function
1	Conv2D	(None, 222, 222, 32)	32	3x3	ReLU
2	MaxPooling2D	(None, 111, 111, 32)	-	2x2	-
3	Conv2D	(None, 109, 109, 64)	64	3x3	ReLU
4	MaxPooling2D	(None, 54, 54, 64)	-	2x2	-
5	Conv2D	(None, 52, 52, 128)	128	3x3	ReLU
6	MaxPooling2D	(None, 26, 26, 128)	-	2x2	-
7	Flatten	(None, 86528)	-	-	-
8	Dense	(None, 128)	-	-	ReLU
9	Dense	(None, 128)	-	-	Softmax

In implementing the EfficientNetB0 architecture, input images are resized to 100x100 pixels and converted from grayscale to RGB. This resolution is designed to maximize computational efficiency while preserving the model's capability to extract critical features, leveraging pre-trained base layers from the ImageNet dataset. This approach allows for a significant reduction in computational load while maintaining high accuracy levels.

The dataset is split into three parts: 60% for training, 20% for validation, and 20% for testing. This division is crucial to evaluate the model on previously unseen data, supporting an accurate assessment of its generalization ability. Additionally, the validation process during training ensures optimal model performance by fine-tuning parameters, achieving the best results.

The EfficientNetB0 architecture utilizes a scalable base model with compound scaling designed to balance computational efficiency and accuracy. It incorporates additional layers such as GlobalAveragePooling2D to reduce feature dimensions, BatchNormalization to stabilize and accelerate training, and Dense layers with varying units and activation functions. To mitigate overfitting, Dropout layers are also employed, creating a sophisticated and efficient architecture. This combination of elements makes EfficientNetB0 an optimal choice for resource-constrained environments, maintaining high performance in feature extraction even at smaller resolutions. Table 4 outlines the layers of the EfficientNetB0 architecture.

Table 4. Layers of the EfficientNetB0 Architecture

Layer	Layer Type	Output Dimensions	Number of Filters	Kernel Size	Activation Function
1	EfficientNetB0 (Base)	(None, 4, 4, 1280)	-	-	ReLU
2	GlobalAveragePooling2D	(None, 1280)	-	-	ReLU
3	BatchNormalization	(None, 1280)	-	-	-
4	Dense	(None, 256)	-	-	ReLU
5	Dropout	(None, 256)	-	-	ReLU
6	Dense	(None, 128)	-	-	-
7	Dropout	(None, 128)	-	-	-
8	Dense	(None, 1)	-	-	Softmax

2.5 Model Validation and Evaluation

After training four different architectures and determining which one achieved the highest accuracy, the next step is to validate and evaluate these models. This phase aims to ensure that the developed models perform effectively and can adapt well when applied to new, unseen data. During the model validation and evaluation process, a Confusion Matrix is utilized to gain a deeper insight into the model's performance, particularly in classification tasks. This matrix provides information on precision, recall, and accuracy values. Here, precision indicates how accurately the model identifies the positive class, while recall reflects its ability to capture instances of the positive class. Accuracy measures how correctly the model predicts all classes. Typically, the values in the Confusion Matrix are presented as percentages (%) [21] to provide a clearer picture of the model's predictive quality. The calculations are as seen in Formulas 1 – 3.

The formula for determining accuracy:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}\, x\ 100\% \quad (1)$$

The formula for determining precision:

$$\text{Precision} = \frac{TP}{TP+FP}\ x\ 100\% \quad (2)$$

The formula for determining recall:

$$\text{Recall} = \frac{TP}{TP+FN}\ x\ 100\% \quad (3)$$

Next, the comparison between the average precision and recall values obtained is referred to as the F1-Score. The calculation is shown in Formula 4.

The formula for determining F1-Score:

$$\text{F1 – Score} = \frac{2\ X\ Precision\ X\ Recall}{Precision\ X\ Recall} \quad (4)$$

TP (True Positive) represents the count of positive images that the model accurately identified; TN (True Negative) denotes the count of negative images that were correctly recognized by the model; FP (False Positive) refers to the number of images that are actually negative but were mistakenly classified as positive by the model; and FN (False Negative) indicates the number of images that are truly positive but were incorrectly categorized as negative by the model [22].

3. Results and Discussions

In this study, rice plant images were analyzed to detect the presence of stem borer pests using four architectures: a general Convolutional Neural Network (CNN), a proposed CNN, MobileNet, and EfficientNetB0. The performance of these architectures was evaluated based on several metrics, including accuracy, precision, recall, F1 score, and computation time. This evaluation aimed to provide a comprehensive overview of the effectiveness of each architecture in detecting pests, which is crucial for sustainable rice crop management.

The results of the performance comparison between these four architectures using 100 epochs with a batch size of 32 are presented below. These results demonstrate how each architecture contributes to the model's performance in pest detection and helps identify the most effective architecture for practical field applications. The data from this evaluation will serve as a foundation for further development and refinement of the pest detection methods used in this research.

Based on Table 5 and Figure 5, the MobileNet architecture demonstrated significantly lower performance compared to the other three architectures, with an accuracy of 47.2%, precision of 47.2%, recall of 100%, and an F1 score of 64.1%. Its computation time was only 6.63 seconds. Although this computation time is relatively fast compared to the other architectures, its overall performance is considerably poor, making it unsuitable for this research. The low performance of MobileNet can be attributed to its design philosophy and inherent limitations. MobileNet is optimized for power-constrained environments, prioritizing computational efficiency over high accuracy. The use of depthwise separable convolutions reduces the number of parameters and computational load, but it also limits the model's ability to extract complex features, such as the intricate structural damage caused by stem borer pests on rice plants

Table 5. Results of Performance Comparison Among Architectures

Type of Architecture	Accuracy	Precision	Recall	F1 Score	Computation Time (seconds)
General CNN	0.961	0.944	0.976	0.960	395.85976672172546
Proposed CNN	0.998	0.997	1.0	0.998	14.193211078643799
MobileNet	0.472	0.472	1.0	0.641	6.655308246612549
EfficientNetB0	0.996	0.992	1.0	0.996	688.3367702960968

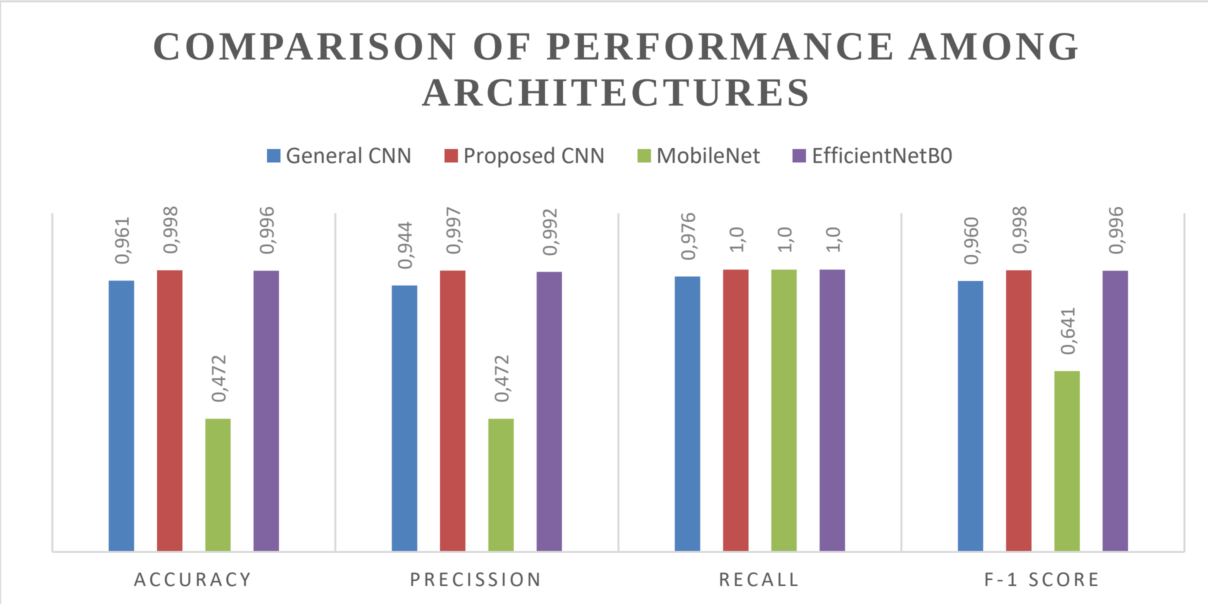

Figure 5. Performance Comparison Among Architectures

The subtle visual differences between damaged and healthy rice stems require deep and comprehensive feature extraction. The shallow architecture of MobileNet struggles with this complexity, resulting in high recall (100%) but low precision, as it tends to overclassify images as pest-damaged to maximize sensitivity. Additionally, MobileNet's reliance on a 224x224 resolution, while efficient, may miss finer details critical for detecting pest damage, further diminishing its performance in this specific application.

In contrast, the general CNN architecture demonstrated commendable performance with an accuracy of 96.1%, precision of 94.4%, recall of 97.6%, and a computation time of 395.85 seconds. However, a notable drawback of this architecture is its long computation time, which may limit its practicality in time-sensitive or resource-constrained applications.

The EfficientNetB0 architecture demonstrated strong performance with an accuracy of 99.6%, precision of 99.2%, recall of 100%, and an F1 score of 99.6%. However, its training time was significantly longer than the other three architectures, at 688.33 seconds. The slow computation time of EfficientNetB0 can be attributed to several factors. First, the model employs compound scaling, which simultaneously increases image resolution, model depth, and network width in a balanced manner. While this scaling enhances accuracy, it also adds computational complexity. Additionally, the use of pre-trained base layers from ImageNet requires careful fine-tuning, which can slow down the training process. Although the input image size was reduced to 100x100 pixels, smaller than the standard for the model, this still required additional adaptation, further extending computation time. These factors suggest that while EfficientNetB0 delivers exceptional accuracy, its application may be limited in scenarios where computational time is a critical factor.

On the other hand, the proposed CNN architecture achieved the best performance, with an accuracy of 99.8%, precision of 99.7%, recall of 100%, and an F1 score of 99.8%. Furthermore, the training time for this model was relatively short, at just 14.19 seconds, demonstrating that the model is both highly accurate in classification and efficient in training duration. The proposed CNN achieved these impressive results due to several key design improvements. The increase in the number of convolutional filters (from 32 to 128) allowed the model to capture increasingly complex and abstract features, which proved highly effective for detecting subtle structural differences in rice stems.

Additionally, the dropout layer (50%) mitigated overfitting by randomly deactivating neurons during training, ensuring the model generalized well to unseen data. Unlike MobileNet, the proposed CNN maintained a resolution that preserved critical structural details in the images. This resolution was empirically determined to provide a balance between computational efficiency and feature retention. The architecture also incorporated pooling layers to reduce dimensionality while preserving essential features, resulting in efficient computation (training time: 14.19 seconds) and superior accuracy. Figure 6 is a graph illustrating the variation in accuracy across given epochs, along with the loss metrics of the proposed CNN architecture:

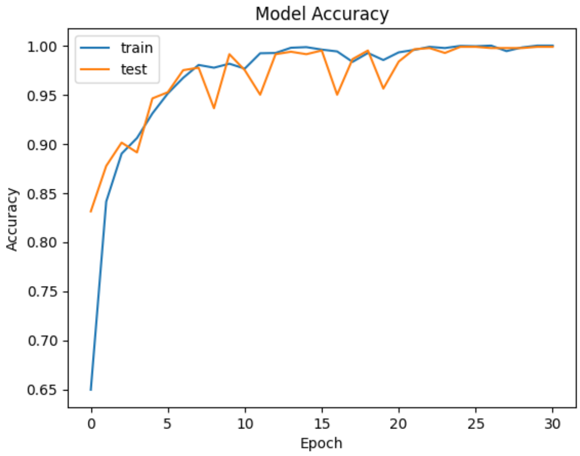

Figure 6. Visualization of Model Accuracy During Training Process

The visualization results indicate that the proposed Convolutional Neural Network (CNN) performs effectively throughout the training process. During the initial epochs (1-5), accuracy rises sharply, with training accuracy increasing from approximately 0.65 to 0.85 and testing accuracy from 0.70 to 0.85. This upward trend continues until epoch 10, where training accuracy reaches about 0.95 and testing accuracy hovers around 0.90, despite some minor fluctuations. After epoch 10 and up to around epoch 30, the model's accuracy remains highly stable, with training accuracy fluctuating between 0.95 and 1.00, while testing accuracy stabilizes at about 0.95, albeit with consistent small variations. From epochs 30 to 100, both training and testing accuracy continue to stabilize in a similar pattern, suggesting that the model retains its performance without significant overfitting. The minimal difference in accuracy between training and testing indicates strong and consistent performance on unseen data.

The visualization results in Figure 7 illustrate the graph depicting the changes in the model's loss for both training and testing data. In the first epoch, the training data loss began at a high value of approximately 2.5 and dropped sharply to nearly 0.3. Similarly, the loss of the testing data also experienced a significant reduction during the first epoch. Following this initial phase, the loss continued to decrease gradually and eventually stabilized at a very low level. From epoch 5 onward, both training and testing loss remained around 0.1 or lower, with minor fluctuations. These fluctuations reflect slight variations in the model's performance from one epoch to the next; however, overall, the loss remained consistently low. The substantial decrease in loss during the early epochs, along with its subsequent stability, suggests that the architecture is learning effectively and maintaining performance without evident overfitting. Additionally, the low and stable loss values for the testing data indicate that the architecture can deliver reliable and consistent results on unseen data (testing data), demonstrating that the model not only learns well from the training data but also effectively applies its acquired knowledge to new situations.

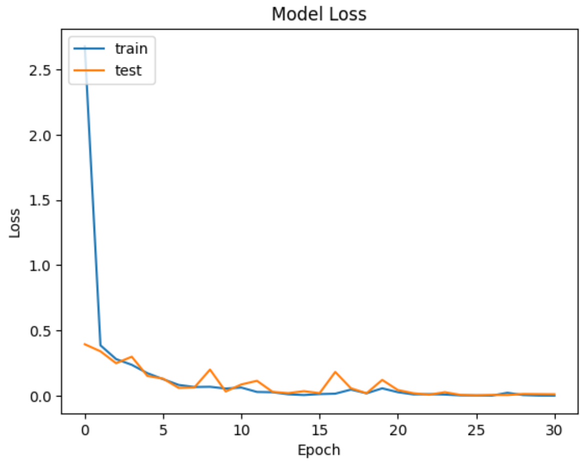

Figure 7. Visualization of Model Loss During the Training Process

Next, predictions were made on the test data and compared with the original labels to create a Confusion Matrix. This matrix offers a clear overview of the correct and incorrect predictions, which is then visualized for additional analysis. Figure 8 shows the results from the visualization of the Confusion Matrix:

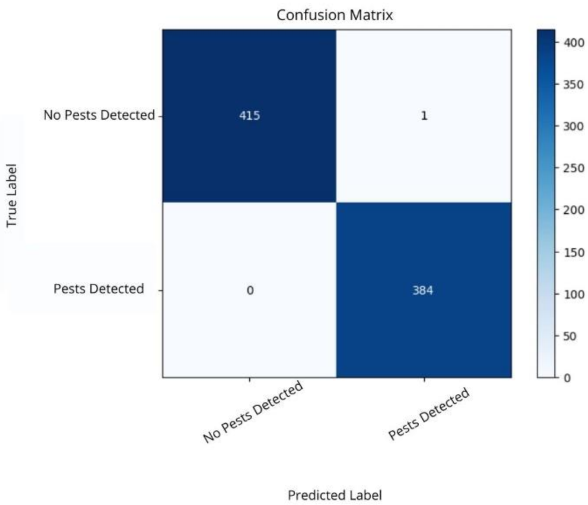

Figure 8. Visualization of the Confusion Matrix Results

The confusion matrix visualization highlights the excellent performance of the proposed Convolutional

Neural Network (CNN) architecture. A high number of true positives and true negatives indicates the model's strong capability to accurately detect pests in rice crops, as well as to correctly identify pest-free plants. The low occurrence of false positives and false negatives further shows that the model rarely makes errors in its predictions, minimizing both Type I errors (FP) and Type II errors (FN).

However, despite the proposed CNN model achieving an almost perfect accuracy of 99.8%, further examination of misclassified images revealed several key causes. One notable error was false positives, where healthy rice stems were classified as damaged by pests. This mistake often stemmed from environmental factors such as shadows, soil patches, or water reflections that resembled pest damage patterns. In field applications, false positives could lead to unnecessary pest control actions, increasing costs and the risk of excessive pesticide use, which could harm the environment. To address this, several mitigation strategies can be employed. Expanding the dataset by adding more images captured under similar environmental conditions can help the model distinguish between actual pest damage and environmental artifacts. Additionally, post-processing techniques, such as secondary validation using temporal image sequences or cross-referencing with sensor data, can improve the accuracy of pest detection by ensuring more reliable pest presence verification.

Another type of error observed was false negatives, where rice stems damaged by pests were classified as healthy or pest-free. This error primarily occurred when pest damage was in its early stages or minimal, making the visual signs difficult to detect. The consequence of failing to detect pest attacks early can delay necessary intervention, potentially leading to more widespread crop damage and significant economic losses. To mitigate this issue, the model architecture needs to be refined by enhancing feature extraction layers to become more sensitive to subtle changes, such as minor discoloration or small anomalies on the rice stems. Additionally, incorporating multi-modal inputs, such as combining image data with supporting information like temperature or humidity levels that correlate with pest activity, could further improve detection accuracy and reduce the likelihood of false negatives.

The practical application of the proposed CNN architecture in the field also depends on its ability to address real-world challenges. Environmental variability, such as changes in lighting, weather conditions, and different growth stages of rice, can introduce variability that affects the model's performance. Additionally, power and connectivity constraints in remote agricultural areas, where access to electricity and internet may be limited, pose significant challenges. Deploying the model on battery-powered devices with offline inference capabilities ensures consistent functionality, even in areas with limited infrastructure, making it suitable for field applications.

4. Conclusions

This study demonstrates that the proposed Convolutional Neural Network (CNN) architecture outperforms other models, such as general CNN, MobileNet, and EfficientNetB0, in detecting stem borer damage in rice plants. With an accuracy of 99.8%, precision of 99.7%, recall of 100%, an almost perfect F1 score, and a training time of only 14.19 seconds, this model is highly efficient and suitable for low-power devices like smartphones, even in remote areas with limited internet access. However, error analysis highlights challenges such as false positives caused by environmental factors like shadows or water reflections, and false negatives where damage goes undetected. These issues can be addressed by expanding the dataset and refining the model architecture to detect more subtle signs of damage. This model not only supports more effective pest control but also contributes to sustainable agricultural practices by reducing excessive pesticide use. Its flexibility allows adaptation for other pests or crops through transfer learning. Furthermore, integration with IoT technologies, such as drones and remote sensors, opens opportunities for automated and large-scale field monitoring, enhancing precision agriculture practices. By enabling early pest detection and timely control, this model is expected to improve crop yields, minimize losses from infestations, and revolutionize pest management with an efficient and sustainable approach.

Acknowledgements

We would like to extend our heartfelt gratitude to the farming community in Takalar Regency, South Sulawesi Province, for their invaluable cooperation and generous permission granted during our field data collection efforts. Their willingness to engage with us and share their insights played a crucial role in the success of this research.

Without the support and active participation of the farmers, we would not have been able to gather the necessary data or gain a deeper understanding of the challenges and practices in rice cultivation within the region. The farmers' contributions ranging from their knowledge about local agricultural techniques to their firsthand experiences with pest management are essential to the integrity and success of our study.

References

[1] N. Sunandar and J. Sutopo, "Pemanfaatan Artificial Neural Network Terhadap Klasifikasi Penyakit Tanaman Padi Menggunakan Citra Daun," *Jurnal Sains Komputer & Informatika*, vol. 8, Mar. 2024, doi: http://dx.doi.org/10.30645/j-sakti.v8i1.

[2] N. Dominic, T. W. Cenggoro, and B. Pardamean, "Systematic Literature Review on Statistics and Machine Learning Predictive Models for Rice Phenotypes," in *Procedia Computer Science*, Elsevier B.V., 2023, pp. 1054–1061. doi: 10.1016/j.procs.2023.10.615.

[3] X.-R. Li *et al.*, "Journal of Integrative Agriculture Strategies for improving crop comprehensive benefits via a decision-

making system based on machine learning of rice-rape, rice-wheat and rice-garlic rotation systems in Southwest China", doi: https://doi.org/10.1016/j.jia.2023.10.005.

[4] Badan Pusat Statistik, "Produksi Padi Tahun 2021 Turun 0,43 persen (Angka Tetap)," Badan Pusat Statistik. Accessed: Mar. 29, 2024. [Online]. Available: https://www.bps.go.id/id/pressrelease/2022/03/01/1909/produksi-padi-tahun2021-turun-0-43-persen--angka-tetap-.html

[5] Badan Pusat Statistik, "Pada 2022, luas panen padi mencapai sekitar 10,45 juta hektar dengan produksi sebesar 54,75 juta ton GKG.," Badan Pusat Statistik. Accessed: May 30, 2024. [Online]. Available: https://www.bps.go.id/id/pressrelease/2023/03/01/2036/pada-2022--luas-panen-padi-mencapai-sekitar-10-45-juta-hektar-dengan-produksi-sebesar-54-75-juta-ton-gkg-.html

[6] Badan Pusat Statistik, "LUAS PANEN DAN PRODUKSI PADI DI INDONESIA 2023 (ANGKA TETAP) PADDY HARVESTED AREA AND PRODUCTION IN INDONESIA 2023 FINAL FIGURESS R I N G K A S A N," 2024. Accessed: May 30, 2024. [Online]. Available: https://www.bps.go.id/id/publication/2024/05/06/69834d72f7ef1c32eee5c4b6/ringkasan-eksekutif-luas-panen-dan-produksi-padi-di-indonesia-2023-angka-tetap-.html

[7] 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, no. 3, pp. 430–439, Feb. 2019, doi: 10.1038/s41559-018-0793-y.

[8] C. G. Simhadri, H. K. Kondaveeti, V. K. Vatsavayi, A. Mitra, and P. Ananthachari, "Deep learning for rice leaf disease detection: A systematic literature review on emerging trends, methodologies and techniques," 2024, *China Agricultural University*. doi: 10.1016/j.inpa.2024.04.006.

[9] Nurhadiah, S. N. Kartana, and Sanjan, "STUDI JENIS HAMAPENTING TANAMAN PADI LOKAL LAHAN KERINGDESATANJUNG SARI SINTANG ," vol. 19, Apr. 2023, doi: https://doi.org/10.51826.

[10] Heriandi, E. Syahputra, and F. Rianto, "TINGKAT SERANGAN HAMA PENGGEREK BATANG PADI DI KABUPATEN KAYONG UTARA," *Jurnal Pertanian Agros* , vol. 25, 2023, doi: http://dx.doi.org/10.37159/jpa.v25i1.

[11] "Hama Utama Padi," Dinas Pertanian dan Pangan Kabupaten Kulon Progo. Accessed: Apr. 05, 2024. [Online]. Available: https://pertanian.kulonprogokab.go.id/detil/1206/hama-utama-padi#

[12] A. Ettalibi, A. Elouadi, and A. Mansour, "AI and Computer Vision-based Real-time Quality Control: A Review of Industrial Applications," in *Procedia Computer Science*, Elsevier B.V., 2024, pp. 212–220. doi: 10.1016/j.procs.2023.12.195.

[13] G. Habib and S. Qureshi, "Optimization and acceleration of convolutional neural networks: A survey," Jul. 01, 2022, *King Saud bin Abdulaziz University*. doi: 10.1016/j.jksuci.2020.10.004.

[14] Y. Lu, X. Wu, P. Liu, H. Li, and W. Liu, "Rice disease identification method based on improved CNN-BiGRU," *Artificial Intelligence in Agriculture*, vol. 9, pp. 100–109, Sep. 2023, doi: 10.1016/j.aiia.2023.08.005.

[15] Y. Zhang, D. Xiao, Y. Liu, and H. Wu, "An algorithm for automatic identification of multiple developmental stages of rice spikes based on improved Faster R-CNN," *Crop Journal*, vol. 10, no. 5, pp. 1323–1333, Oct. 2022, doi: 10.1016/j.cj.2022.06.004.

[16] P. I. Ritharson, K. Raimond, X. A. Mary, J. E. Robert, and A. J, "DeepRice: A deep learning and deep feature based classification of Rice leaf disease subtypes," *Artificial Intelligence in Agriculture*, vol. 11, pp. 34–49, Mar. 2024, doi: 10.1016/j.aiia.2023.11.001.

[17] M. T. Ahad, Y. Li, B. Song, and T. Bhuiyan, "Comparison of CNN-based deep learning architectures for rice diseases classification," *Artificial Intelligence in Agriculture*, vol. 9, pp. 22–35, Sep. 2023, doi: 10.1016/j.aiia.2023.07.001.

[18] F. Liao *et al.*, "A hybrid CNN-LSTM model for diagnosing rice nutrient levels at the rice panicle initiation stage," *J Integr Agric*, vol. 23, no. 2, pp. 711–723, Feb. 2024, doi: 10.1016/j.jia.2023.05.032.

[19] M. Fauzan Arif, A. Nurkholis, S. Laia, and P. Rosyani, "Deteksi Kendaraan Dengan Metode YOLO," *Jurnal Artificial Inteligent dan Sistem Penunjang Keputusan*, vol. 01, no. 01, 2023, [Online]. Available: https://jurnalmahasiswa.com/index.php/aidanspk

[20] P. A. Arjun, S. Suryanarayan, R. S. Viswamanav, S. Abhishek, and T. Anjali, "Unveiling Underwater Structures: MobileNet vs. EfficientNet in Sonar Image Detection," *Procedia Comput Sci*, vol. 233, pp. 518–527, 2024, doi: 10.1016/j.procs.2024.03.241.

[21] B. Firmanto, H. Soekotjo, and H. Suyono, "PERBANDINGAN KINERJA ALGORITMA PROMETHEE DAN TOPSIS UNTUK PEMILIHAN GURU TELADAN," vol. 2, [Online]. Available: http://jurnal.unram.ac.id/index.php/jpp-ipa

[22] Q. Liu *et al.*, "Development of Machine Learning Methods for Accurate Prediction of Plant Disease Resistance," *Engineering*, Apr. 2024, doi: 10.1016/j.eng.2024.03.014.

