Jurnal LOPI Selayar - Techno-Sociopreneur
Volume 1, No. 1, Edisi Pertama
https://ejournal.itsbmselayar.com/index.php/lopiselayar/en/

Comparative Performance Analysis of Deep Learning Models for Planthopper Classification in Rice Plant Ecosystems

Mursyid Ardiansyah [1*], Fathinul Syahir Bin Ahmad Sa'ad [2], Muhammad Imran Ahmad [3]
[1]Faculty of Intelligent Computing, Universiti Malaysia Perlis, Malaysia; mursyid@studentmail.unimap.edu.my
[2]Faculty of Electrical Engineering and Technology, Universiti Malaysia Perlis, Malaysia; fathinul@unimap.edu.my
[3]Agrotechnology Department, Universiti Malaysia Perlis, Malaysia; m.imran@unimap.edu.my

* Correspondence Author: mursyid@studentmail.unimap.edu.my

ARTICLE INFO

Kata kunci:
Deep Learning; Classification; Planthopper; EfficientNetB4; MobileNetV2; ResNet101; VGG16

Riwayat artikel:
Diterima: 19/08/2025
Direvisi: 24/08/2025
Disetujui: 28/08/2025

ABSTRACT

Planthoppers are among the most destructive pests in rice ecosystems, capable of causing severe yield losses if not detected early and accurately. Conventional identification methods are often time-consuming and dependent on expert knowledge, limiting their scalability in large agricultural areas. To address this challenge, this study investigates the application of deep learning for automatic planthopper classification using digital images. The objective is to evaluate and compare the performance of four widely used convolutional neural network (CNN) architectures EfficientNetB4, ResNet101, MobileNetV2, and VGG16 under a standardized experimental setup. All models were trained in PyTorch with CrossEntropyLoss, AdamW optimizer, and learning rate scheduling, using mixed precision to improve computational efficiency. The dataset consisted of three planthopper classes: Brown Planthopper (BPH), White-Backed Planthopper (WBPH), and Rice Leafhopper (RLM). Evaluation metrics included accuracy, precision, recall, F1-score, and confusion matrix analysis to assess per-class classification performance. The experimental results show that EfficientNetB4 achieved the best overall performance with 92.13% accuracy and 93.45% F1-score, demonstrating its strong generalization capability. ResNet101 followed closely with 92.80% accuracy and the highest recall of 94.46%, indicating effectiveness in minimizing false negatives. MobileNetV2 provided competitive results (91.85% accuracy) with high efficiency, making it suitable for lightweight, real-time deployment. In contrast, VGG16 achieved lower performance (87.51% accuracy) due to its shallower architecture and limited feature extraction capability. In conclusion, the findings highlight that modern architectures such as EfficientNetB4 and ResNet101 outperform older CNN designs in agricultural pest classification tasks. EfficientNetB4 is recommended for high-

		accuracy applications, ResNet101 for minimizing missed detections, and MobileNetV2 for resource-constrained environments. This study contributes to the development of intelligent pest monitoring systems, supporting precision agriculture and sustainable rice production.

1. INTRODUCTION

Rice is a staple crop that sustains more than half of the global population, and its productivity plays a vital role in ensuring food security, particularly in Asia(Asma, D, & D, 2023). However, rice production is severely threatened by planthoppers (family Delphacidae), which are among the most destructive insect pests in rice ecosystems(Iamba & Dono, 2021). Planthopper infestations not only cause direct damage through sap feeding, but also transmit viral diseases such as rice ragged stunt virus and rice grassy stunt virus, leading to significant yield losses if not detected and managed early. Conventional methods of planthopper identification rely heavily on manual observation by agricultural experts, which is labor-intensive, time-consuming, and prone to subjectivity. These limitations highlight the urgent need for an automated and reliable classification system that can support early pest monitoring and integrated pest management(Phatthalung & Tangkananond, 2021).

Recent advances in computer vision and deep learning have enabled breakthroughs in agricultural image analysis. Convolutional Neural Networks (CNNs) have demonstrated strong capability in learning discriminative visual features from digital images, enabling accurate classification of pests and diseases. While several studies have focused on plant disease detection and general insect recognition, specific research on automated planthopper classification within rice ecosystems remains relatively limited(Wang et al., 2025). Addressing this gap can significantly enhance pest monitoring systems, especially in smart agriculture applications where timely and accurate pest identification is essential(Lu, Tan, & Jiang, 2021).

This study presents a comparative performance analysis of four well-established deep learning models EfficientNetB4, MobileNetV2, ResNet101, and VGG16 for the classification of planthoppers in rice ecosystems. Each architecture embodies unique design principles: EfficientNetB4 applies compound scaling for balanced model efficiency; MobileNetV2 is designed for lightweight performance suitable for mobile deployment; ResNet101 introduces residual connections to facilitate deep feature learning; and VGG16, though relatively older, remains a widely used benchmark due to its straightforward architecture.

By conducting a rigorous comparison of these models, this research aims to evaluate their accuracy, computational efficiency, and generalization capability in planthopper image classification. The findings are expected to provide valuable insights into selecting optimal CNN architectures for pest monitoring systems and contribute to the development of scalable, real-time, and field-deployable solutions that support sustainable rice production.

2. RELATED WORKS

Recent studies have demonstrated the effectiveness of deep learning in agricultural pest monitoring, particularly for rice ecosystems where planthoppers (Delphacidae) remain one of the most destructive threats. Within the past five years, several works have applied convolutional neural networks (CNNs) for planthopper recognition and related tasks, providing a foundation for the comparative analysis conducted in this study.

(Ibrahim et al., 2023) presented a deep CNN-based planthopper classification using a high-density image dataset collected from sticky traps. The dataset included brown planthopper (BPH), green

leafhopper (GLH), white-backed planthopper (WBPH), and zigzag leafhopper classes. They compared ResNet-50, ResNet-101, ResNet-152, VGG-16, and VGG-19 architectures. Their results showed that ResNet-50 achieved the highest accuracy (97.28%), surpassing deeper networks such as ResNet-101, highlighting that residual connections are highly effective in distinguishing morphologically similar planthopper species.

In another study, (Khairunniza-Bejo et al., 2024) developed a model for automatic detection and counting of paddy planthoppers using a Faster R-CNN detector with a VGG-16 backbone. Their model achieved a mean average precision (mAP) of 97.69% and classification accuracy of 98.84%. While their focus was primarily on detection rather than classification, the study underscores the continued relevance of VGG-16 as a robust feature extractor for planthopper-related applications.

(Z. Zhang et al., 2024) introduced the RPH-Counter, a specialized deep learning system for field-level detection and counting of rice planthoppers. The model was evaluated under natural conditions and demonstrated strong performance in monitoring planthopper population dynamics. Although this approach emphasizes object detection and density estimation, it reflects the growing trend of integrating CNN-based solutions into practical field monitoring system.

A more recent contribution by (Ji et al., 2025) proposed a publicly available RGB image dataset for rice planthopper detection and counting. They integrated Swin Transformer with YOLOv8-p2 to enhance accuracy and robustness, achieving mAP50:95 of 0.835 and F1-score of 0.899. While transformer-based detectors differ from classical CNNs, this study highlights the evolution of architectures towards hybrid and efficient models, providing context for evaluating CNN-based classifiers such as EfficientNet and MobileNet in comparison.

Beyond planthopper-specific studies, there has been increasing attention to lightweight CNNs suitable for resource-constrained environments. (Rybczak & Kozakiewicz, 2024) compared MobileNet, EfficientNetB0, and InceptionV3 for image classification under limited computational resources. Their results confirmed that MobileNet provides the best trade-off between inference speed and accuracy, while EfficientNet offers balanced scalability. Although not focused on planthoppers, these findings reinforce the suitability of MobileNetV2 and EfficientNetB4 for deployment in mobile or edge-based pest monitoring systems.

In summary, the literature from the last five years emphasizes the strong performance of ResNet and VGG in planthopper tasks, highlights the practicality of MobileNet for lightweight deployment, and demonstrates the promising scalability of EfficientNet. However, few studies have conducted a direct head-to-head comparison of EfficientNetB4, MobileNetV2, ResNet101, and VGG16 specifically for planthopper classification. This research addresses that gap by systematically evaluating these four architectures on rice planthopper datasets, focusing on both accuracy and computational efficiency.

3. METHODOLOGY

3.1. Research Framework

The methodology of this study is structured to evaluate and compare the performance of four deep learning architectures EfficientNetB4, ResNet101, MobileNetV2, and VGG16 for the classification of planthoppers in rice ecosystems. The framework consists of sequential phases: dataset preparation, preprocessing and augmentation, model selection and modification, training and fine-tuning, evaluation, and result analysis. This structured pipeline ensures reproducibility and enables a fair comparison across models.

3.2. Dataset Preperation

The dataset employed in this study was sourced from the work of (Ji et al., 2025). This dataset contains annotated RGB images of rice pests collected under real field conditions. The dataset used in this study consists of three major rice insect pests that cause significant yield losses in rice ecosystems Brown Planthopper (Nilaparvata lugens), White-Backed Planthopper (Sogatella furcifera), and Rice Leaf Miner (Hydrellia griseola). For this research, three target classes were selected corresponding to class labels 0, 1, and 2, representing different pest categories within rice ecosystems.

Table 1. Dataset Summary

Subset	BPH (0)	WBPH (1)	RLM (2)	Total
Training	765	69	1,375	**2,209**
Validation	255	23	459	**737**
Testing	255	24	458	**737**
Total	**1,275**	**116**	**2,292**	**3,683**

Tabel 1 shows the dataset used in this study consists of 3,683 images representing three major rice insect pests Brown Planthopper (BPH, label 0) with 1,275 images, White-Backed Planthopper (WBPH, label 1) with 116 images, and Rice Leaf Miner (RLM, label 2) with 2,292 images. The dataset was stratified into three subsets with a ratio of 60% for training (2,209 images: 765 BPH, 69 WBPH, 1,375 RLM), 20% for validation (737 images: 255 BPH, 23 WBPH, 459 RLM), and 20% for testing (737 images: 255 BPH, 24 WBPH, 458 RLM). This distribution highlights a class imbalance, as WBPH accounts for only about 3% of the total dataset while RLM dominates with over 62%. To address this imbalance during training, a WeightedRandomSampler was employed to ensure that all classes contributed proportionally to the model learning process.

3.3. Data Preprocessing and Augmentation

To enhance model performance and ensure robustness under diverse field conditions, the dataset was subjected to both preprocessing and augmentation steps. Preprocessing was applied uniformly to all images to standardize their size and format before being fed into the models. Specifically, each image was resized to 256×256 pixels, center-cropped to 224×224 pixels, and normalized using the ImageNet mean and standard deviation, ensuring compatibility with pretrained CNN architectures such as EfficientNet, ResNet, MobileNet, and VGG(Beeravolu et al., 2021; Woerner, Jaques, & Baumgartner, 2024).

In addition, augmentation techniques were employed exclusively on the training set to artificially increase data diversity and reduce the risk of overfitting. These augmentations included random horizontal and vertical flips to simulate insect orientation changes, random rotations of 15°, 30°, and 45° to account for varying capture angles, random affine transformations (translation and scaling) to mimic positional variability, and color jittering to replicate variations in brightness, contrast, saturation, and hue caused by different environmental lighting. By simulating realistic field conditions, these steps improve the models' ability to generalize effectively when applied to unseen data.

3.4. Model Architectures

In this study, four convolutional neural network (CNN) architectures were selected due to their popularity, complementary design principles, and proven performance in various computer vision tasks. Each model represents a different generation of CNN development, allowing for a comprehensive comparison of accuracy, efficiency, and scalability in planthopper classification.

EfficientNetB4 is part of the EfficientNet family, which introduces a compound scaling method to balance model depth, width, and input resolution. This design enables the network to achieve high accuracy while maintaining computational efficiency. EfficientNetB4, being a mid-sized variant, provides an effective trade-off between performance and resource usage, making it

suitable for handling complex datasets with varying image conditions(Preetha, Priyadarsini, & Nisha, 2024).

Table 2. EfficientNetB4 Architectures

Stage	Input Resolution	Operator	Output Channels	Repeat (Layers)
Stem	380×380	Conv3×3	48	1
Block 1	190×190	MBConv1, k3×3	24	2
Block 2	190×190	MBConv6, k3×3	32	4
Block 3	95×95	MBConv6, k5×5	56	4
Block 4	48×48	MBConv6, k3×3	112	6
Block 5	24×24	MBConv6, k5×5	160	6
Block 6	12×12	MBConv6, k5×5	272	8
Block 7	12×12	MBConv6, k3×3	448	2
Head	12×12	Conv1×1	1792	1
Classifier	1×1 (Global Avg)	Fully Connected	3 (BPH, WBPH, RLM)	1

Table 2 above presents the detailed architecture of EfficientNetB4 used in this study. The process begins with the Stem layer, where the input image of 380×380 pixels is processed using a 3×3 convolution, producing 48 channels. The architecture then consists of seven main blocks built upon MBConv (Mobile Inverted Bottleneck Convolution) with varying kernel sizes (3×3 or 5×5) and expansion factors. Block 1 employs MBConv1 with a 3×3 kernel, producing 24 channels and repeated for 2 layers. In Block 2, MBConv6 with a 3×3 kernel is applied to extract more complex features, generating 32 channels with 4 repeated layers. Block 3 operates on a 95×95 resolution using MBConv6 with a 5×5 kernel, yielding 56 channels repeated 4 times. Block 4 processes the 48×48 resolution with MBConv6 (3×3), producing 112 channels with 6 layers. Block 5 applies MBConv6 (5×5) at a 24×24 resolution, generating 160 channels with 6 repetitions. Block 6 extracts feature at a 12×12 resolution using MBConv6 (5×5), producing 272 channels with the highest repetition count of 8 layers. Block 7 continues processing at a 12×12 resolution with MBConv6 (3×3), generating 448 channels with 2 layers. Afterward, the Head layer applies a 1×1 convolution to expand the feature representation to 1792 channels. Finally, the Classifier utilizes global average pooling (1×1), followed by a fully connected layer to output 3 neurons corresponding to the target classes: Brown Planthopper (BPH), White-Backed Planthopper (WBPH), and Rice Leaf Miner (RLM).

ResNet101 is a deep residual network that incorporates residual connections (skip connections) to address the vanishing gradient problem in very deep architectures. By allowing gradients to flow more effectively during backpropagation, ResNet101 can learn fine-grained features essential for distinguishing morphologically similar insects, such as different species of planthoppers. Its depth also allows it to capture more abstract and discriminative representations from input images(Q. Zhang, 2022).

Table 3. ResNet101 Architectures

Stage	Input Resolution	Operator / Block	Output Channels
Conv1	224×224	7×7 Conv, stride 2	64
MaxPool	112×112	3×3 MaxPool, stride 2	64
Conv2_x	56×56	Bottleneck (1×1, 3×3, 1×1)	256
Conv3_x	28×28	Bottleneck (1×1, 3×3, 1×1)	512
Conv4_x	14×14	Bottleneck (1×1, 3×3, 1×1)	1024
Conv5_x	7×7	Bottleneck (1×1, 3×3, 1×1)	2048
Pooling	1×1	Global Avg Pooling	2048
Classifier	1×1	Fully Connected	3 (BPH, WBPH, RLM)

The architecture of ResNet101 in Table 3 begins with a 7×7 convolutional layer (stride 2) followed by a 3×3 max pooling layer, reducing the input resolution from 224×224 to 112×112. The main feature extraction is divided into four stages of residual bottleneck blocks. The Conv2_x stage operates on a 56×56 resolution with three bottleneck blocks, each consisting of 1×1, 3×3, and 1×1 convolutions, producing 256 output channels. The Conv3_x stage processes 28×28 feature maps with four bottleneck blocks, expanding the depth to 512 channels. The Conv4_x stage is the deepest and most computationally intensive, containing 23 bottleneck blocks that output 1024 channels, enabling the model to capture highly abstract features. The Conv5_x stage further refines these representations with three bottleneck blocks at 7×7 resolution, generating 2048 channels. A global average pooling layer then reduces the spatial dimension to 1×1, and finally, a fully connected classifier outputs three neurons corresponding to the target classes: Brown Planthopper (BPH), White-Backed Planthopper (WBPH), and Rice Leaf Miner (RLM).

MobileNetV2 was designed with a focus on lightweight and efficient computation, making it ideal for mobile and edge devices. It employs depthwise separable convolutions and inverted residuals to reduce the number of parameters and floating-point operations while maintaining competitive accuracy. This architecture is particularly relevant for agricultural applications, as it supports deployment of resource-constrained devices used by farmers in the field(Rachburee & Punlumjeak, 2022).

Table 4. MobileNetV2 Architectures

Stage	Input Resolution	Operator / Block	Expansion Factor (t)	Output Channels	Stride	Repeat (Layers)
Conv1	224×224	3×3 Conv, stride 2	–	32	2	1
Bottleneck	112×112	Inverted Residual (t=1, k3×3)	1	16	1	1
Bottleneck	112×112	Inverted Residual (t=6, k3×3)	6	24	2	2
Bottleneck	56×56	Inverted Residual (t=6, k3×3)	6	32	2	3
Bottleneck	28×28	Inverted Residual (t=6, k3×3)	6	64	2	4
Bottleneck	14×14	Inverted Residual (t=6, k3×3)	6	96	1	3
Bottleneck	14×14	Inverted Residual (t=6, k3×3)	6	160	2	3
Bottleneck	7×7	Inverted Residual (t=6, k3×3)	6	320	1	1
Conv2	7×7	1×1 Conv	–	1280	1	1
Pooling	1×1	Global Average Pooling	–	1280	–	1
Classifier	1×1	Fully Connected	–	3 (BPH, WBPH, RLM)	–	1

In Table 4 MobileNetV2 architecture employed in this study is designed using inverted residual blocks and linear bottlenecks to balance accuracy with computational efficiency. The process begins with Conv1, where the input image of 224×224 pixels is passed through a 3×3 convolution with stride 2, producing 32 channels. The first bottleneck block applies an expansion factor of 1 with a 3×3 kernel, yielding 16 channels at a resolution of 112×112. Next, the network applies two inverted residual blocks with an expansion factor of 6, producing 24 channels. At 56×56 resolution, three bottleneck blocks with an expansion factor of 6 generate 32 channels. Moving to 28×28 resolution, four bottleneck blocks expand the features to 64 channels. At 14×14, three bottleneck layers output 96 channels, followed by three more bottleneck layers that produce 160 channels. At 7×7 resolution, a single bottleneck block increases the depth to 320 channels. This is followed by Conv2, a 1×1 convolution layer that expands the representation to 1280 channels.

The features are then summarized through global average pooling at 1×1 resolution, and finally, a fully connected layer produces three output neurons corresponding to the target classes: Brown Planthopper (BPH), White-Backed Planthopper (WBPH), and Rice Leaf Miner (RLM).

VGG16 is a classical deep CNN architecture composed of 16 sequential layers using small 3×3 convolution filters. Despite being relatively older compared to modern architectures, VGG16 remains widely adopted as a baseline model due to its simplicity and effectiveness in feature extraction. It has demonstrated strong performance in agricultural image analysis tasks and provides a valuable reference point in this comparative study(Siregar, Akbari, Poningsih, Wanto, & Solikhun, 2025).

Table 5. VGG16 Architectures

Stage	Input Resolution	Operator / Block	Output Channels	Repeat (Layers)
Conv1_x	224×224	3×3 Conv, stride 1	64	2
Conv2_x	112×112	3×3 Conv, stride 1	128	2
Conv3_x	56×56	3×3 Conv, stride 1	256	3
Conv4_x	28×28	3×3 Conv, stride 1	512	3
Conv5_x	14×14	3×3 Conv, stride 1	512	3
Pooling	7×7	Max Pooling + Flatten	512	1
FC1	1×1	Fully Connected	4096	1
FC2	1×1	Fully Connected	4096	1
Classifier	1×1	Fully Connected (Softmax)	3 (BPH, WBPH, RLM)	1

The VGG16 architecture in Table 5 is a classical convolutional neural network composed of 16 weight layers, designed with simplicity and uniformity in mind. The model begins with Conv1_x, where two convolutional layers with 64 filters process the 224×224 input image. The resolution is reduced by max pooling before entering Conv2_x, which consists of two convolutional layers with 128 filters. The network then deepens with Conv3_x, containing three convolutional layers of 256 filters, followed by Conv4_x and Conv5_x, each comprising three convolutional layers of 512 filters. At this stage, the feature map has been reduced to 7×7 resolution. A max pooling layer flattens the representation before passing through two fully connected layers (FC1 and FC2) with 4096 neurons each. Finally, the classifier layer maps the learned features to three output neurons corresponding to Brown Planthopper (BPH), White-Backed Planthopper (WBPH), and Rice Leaf Miner (RLM).

All four models were initialized with pretrained ImageNet weights to leverage transfer learning, enabling faster convergence and better generalization on limited agricultural datasets. For adaptation to the specific task of planthopper classification, the final fully connected (FC) layer of each model was replaced with a new classifier matching the number of target classes (BPH, WBPH, and RLM).

3.5. Training Strategy

All models were trained using a standardized training configuration. The loss function employed was CrossEntropyLoss, which is widely used for multi-class classification tasks. Model optimization was performed using the AdamW optimizer with an initial learning rate of 0.0001, chosen for its ability to handle sparse gradients and weight decay regularization effectively. A learning rate scheduler, ReduceLROnPlateau, was applied to dynamically adjust the learning rate when the validation accuracy plateaued, thereby improving convergence stability. Each model was trained for 20 epochs with a batch size of 32, ensuring a balance between computational efficiency and convergence speed. To accelerate training and reduce GPU memory consumption, Automatic Mixed Precision (AMP) with GradScaler was utilized, enabling faster computations

without sacrificing numerical stability(Liu, Li, & Zhang, 2025).

3.6. Evaluation Metrics

To comprehensively evaluate the classification models, multiple performance metrics and diagnostic tools were employed such as Accuracy, Precision, Recall, and F1-Score. Additionally, the confusion matrix was utilized to visualize detailed per-class performance, highlighting patterns of misclassification among Brown Planthopper (BPH), White-Backed Planthopper (WBPH), and Rice Leaf Miner (RLM). For consistency in reporting, classification results were presented in tabular format, while confusion matrices were stored as heatmap visualizations for intuitive interpretation(Düntsch & Gediga, 2020).

Accuracy measures the proportion of correct predictions compared to the total number of predictions. It provides a general overview of model performance but may be misleading in cases of class imbalance.

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

Precision evaluates the proportion of correctly predicted positive samples among all samples predicted as positive. It is important when minimizing false positives is crucial, such as avoiding misclassification of non-pests as pests.

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

Recall measures the proportion of actual positive samples that are correctly identified by the model. This metric is critical when the main goal is to capture as many positive cases as possible, such as ensuring all harmful pests are detected in the field.

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

F1-Score is the harmonic mean of precision and recall, providing a balanced evaluation of both metrics. It is particularly useful when dealing with imbalanced datasets, as it accounts for both false positives and false negatives.

$$F1 - Score \frac{Precision \text{ x } Recall}{Precision + Recall} \quad (4)$$

Explanation of Symbols

- TP (True Positive): the number of correctly predicted positive samples.
- TN (True Negative): the number of correctly predicted negative samples.
- FP (False Positive): the number of negative samples incorrectly predicted as positive.
- FN (False Negative): the number of positive samples incorrectly predicted as negative.

3.7. Implementation Details

All research in this study was implemented using the PyTorch deep learning framework. To ensure reproducibility, random seeds were fixed across NumPy, Torch, and Python's random library, thereby minimizing variation in results due to stochastic processes. Model training and

evaluation were executed on GPU hardware when available, leveraging CUDA acceleration for efficient computation. In addition, multiprocessing settings were optimized according to the operating system to achieve balanced performance between data loading and GPU utilization. These implementation strategies were adopted to guarantee consistency, efficiency, and reliability of the experimental outcomes.

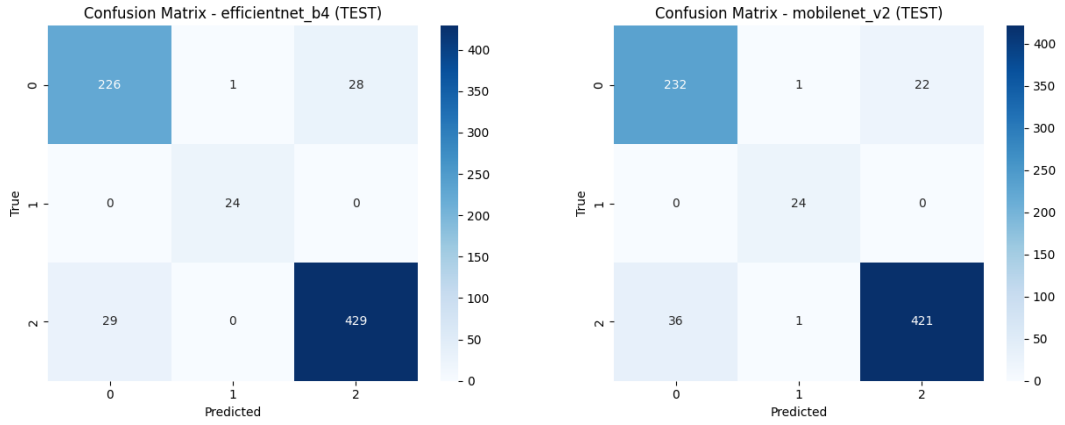
4. RESULTS

This chapter presents the experimental results obtained from training and evaluating four deep learning models EfficientNetB4, ResNet101, MobileNetV2, and VGG16 on the rice insect pest dataset consisting of Brown Planthopper (BPH), White-Backed Planthopper (WBPH), and Rice Leaf Miner (RLM). The analysis covers classification performance in terms of accuracy, precision, recall, and F1-score, as well as confusion matrix visualization to assess per-class recognition.

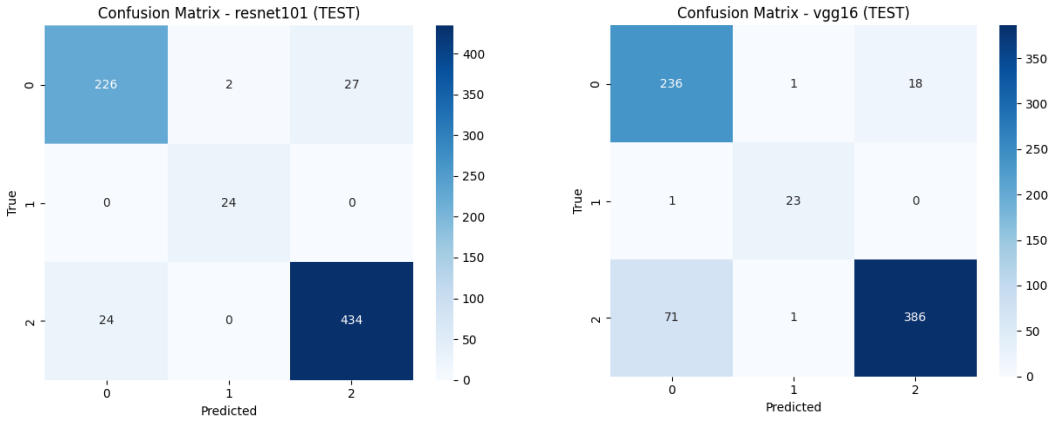
Table 6. The Performance Metrics

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
EfficientNetB4	92.13	92.83	94.09	93.45
ResNet101	92.80	92.28	94.46	93.31
MobileNetV2	91.85	91.30	94.30	92.72
VGG16	87.51	88.05	90.88	89.09

The performance metrics for each model are summarized in Table 6 ResNet101 achieved the highest accuracy (92.80%) and recall (94.46%), indicating its strength in capturing detailed features through deep residual learning. EfficientNetB4 closely followed with an accuracy of 92.13% but achieved the best F1-score (93.45%) and precision (92.83%), demonstrating balanced performance across all metrics. MobileNetV2 also performed competitively with 91.85% accuracy and strong recall (94.30%), making it a promising option for lightweight and real-time applications. In contrast, VGG16 recorded the lowest accuracy (87.51%) and F1-score (89.09%), reflecting its relatively limited feature extraction capabilities compared to modern architectures.



(a) (b)



(c) (d)

Figure 1. (a) Confusion Matrix EfficientNetB4 (b) Confusion Matrix MobileNetV2 (c) Confusion Matrix ResNet101 (d) Confusion Matrix VGG16

Figure 1 show the confusion matrices provide deeper insights into the classification performance of each model. EfficientNetB4 demonstrates strong performance, particularly in correctly classifying Rice Leaf Miner (class 2), with only minor misclassifications between Brown Planthopper (class 0) and Rice Leaf Miner. MobileNetV2 shows similar effectiveness, though with slightly higher misclassifications in class 2 compared to EfficientNetB4. ResNet101 delivers highly stable predictions across all classes, with minimal errors and especially strong accuracy in distinguishing class 2 from the others. VGG16, on the other hand, shows more frequent misclassifications, particularly with class 2 samples often being predicted as class 0, reflecting its relatively lower performance compared to the modern architecture. Overall, the confusion matrices confirm that EfficientNetB4 and ResNet101 excel in both accuracy and balanced per-class predictions, MobileNetV2 offers a practical trade-off between accuracy and efficiency, while VGG16 lags due to its limited representational depth.

The comparative evaluation of the four deep learning architectures EfficientNetB4, ResNet101, MobileNetV2, and VGG16 highlights both their strengths and limitations in classifying planthopper species. Based on the performance metrics and confusion matrices, EfficientNetB4 emerged as the most accurate model, achieving 92.13% accuracy and the highest F1-score of 93.45%. This confirms the effectiveness of compound scaling in balancing depth, width, and resolution to achieve robust generalization across classes.

ResNet101 also performed competitively, with an accuracy of 92.80% and recall of 94.46%, slightly outperforming EfficientNetB4 in recall. This indicates that ResNet101 is particularly effective in correctly identifying true positives, especially in the Rice Leaf Miner class. However, its deeper architecture demands higher computational resources, which may limit its practicality for real-time applications in field environments.

MobileNetV2, while slightly behind EfficientNetB4 and ResNet101 in terms of raw accuracy (91.85%), provided the best trade-off between efficiency and accuracy. Its lightweight design and inverted residual blocks enabled high recall (94.30%) with relatively low computational costs, making it highly suitable for deployment in mobile and resource-constrained devices.

VGG16, as the oldest architecture in comparison, achieved the lowest overall performance, with an accuracy of 87.51%. Its confusion matrix revealed frequent misclassifications, especially for the Rice Leaf Miner class being predicted as Brown Planthopper. This underlines the limitations of shallow architectures lacking modern optimization techniques such as depthwise separable convolutions or residual connections.

In summary, EfficientNetB4 is the most balanced and accurate model, ResNet101 excels in recall but is computationally demanding, MobileNetV2 offers the best efficiency for real-world deployment, and VGG16 demonstrates the constraints of older CNN designs when applied to complex agricultural image classification tasks.

5. CONSLUSION

This study conducted a comparative analysis of four deep learning architectures EfficientNetB4, ResNet101, MobileNetV2, and VGG16 for planthopper classification in rice plant ecosystems. The results demonstrate that modern architectures significantly outperform earlier convolutional neural networks in both accuracy and generalization.

EfficientNetB4 achieved the most balanced performance with the highest F1-score (93.45%) and strong accuracy (92.13%), highlighting the advantage of compound scaling for agricultural image classification. ResNet101 showed slightly higher recall (94.46%) and competitive accuracy (92.80%), making it particularly effective in reducing false negatives, though at the cost of greater computational demand. MobileNetV2 provided a strong trade-off between accuracy (91.85%) and efficiency, reinforcing its potential for mobile and edge-based deployment in real-time field applications. In contrast, VGG16 lagged behind (87.51% accuracy), frequently misclassifying among planthopper species, thus confirming the limitations of older architectures when compared to modern CNN designs.

Overall, the findings confirm that EfficientNetB4 is the most suitable architecture for high-accuracy applications, ResNet101 is preferable when minimizing missed detections is critical, and MobileNetV2 is the best option for lightweight deployment in precision agriculture. Future research should explore ensemble learning, cross-domain dataset validation, and integration with real-time detection frameworks such as YOLO or Faster R-CNN to enhance robustness and applicability in large-scale rice farming environments.

DAFTAR PUSTAKA

Asma, J., D, S., & D, K. (2023). The Global Lifeline: A Staple Crop Sustaining Two-Thirds of the World's Population. *Agriculture Archives*, *2*(3), 15–18. https://doi.org/10.51470/AGRI.2023.2.3.15

Beeravolu, A. R., Azam, S., Jonkman, M., Shanmugam, B., Kannoorpatti, K., & Anwar, A. (2021). Preprocessing of Breast Cancer Images to Create Datasets for Deep-CNN. *IEEE Access*, *9*, 33438–33463. https://doi.org/10.1109/ACCESS.2021.3058773

Düntsch, I., & Gediga, G. (2020). Indices for rough set approximation and the application to confusion matrices. *International Journal of Approximate Reasoning*, *118*, 155–172. https://doi.org/10.1016/j.ijar.2019.12.008

Iamba, K., & Dono, D. (2021). A Review on Brown Planthopper (Nilaparvata lugens Stål), a Major Pest of Rice in Asia and Pacific. *Asian Journal of Research in Crop Science*, 7–19. https://doi.org/10.9734/ajrcs/2021/v6i430122

Ibrahim, M. F., Khairunniza-Bejo, S., Hanafi, M., Jahari, M., Ahmad Saad, F. S., & Mhd Bookeri, M. A. (2023). Deep CNN-Based Planthopper Classification Using a High-Density Image Dataset. *Agriculture (Switzerland)*, *13*(6). https://doi.org/10.3390/agriculture13061155

Ji, X., Li, J., Cai, X., Ye, X., Gouda, M., He, Y., … Li, X. (2025). Driving by a Publicly Available RGB Image Dataset for Rice Planthopper Detection and Counting by Fusing Swin Transformer and YOLOv8-p2 Architectures in Field Landscapes. *Agriculture (Switzerland)*, *15*(13). https://doi.org/10.3390/agriculture15131366

Khairunniza-Bejo, S., Ibrahim, M. F., Hanafi, M., Jahari, M., Ahmad Saad, F. S., & Mhd Bookeri, M. A. (2024). Automatic Paddy Planthopper Detection and Counting Using Faster R-CNN. *Agriculture*, *14*(9), 1567. https://doi.org/10.3390/agriculture14091567

Liu, K., Li, L., & Zhang, X. (2025). Fast and Accurate Identification of Kiwifruit Diseases Using a Lightweight Convolutional Neural Network Architecture. *IEEE Access*, *13*, 84826–84843. https://doi.org/10.1109/ACCESS.2025.3564355

Lu, J., Tan, L., & Jiang, H. (2021, August 1). Review on convolutional neural network (CNN) applied to plant leaf disease classification. *Agriculture (Switzerland)*. MDPI AG. https://doi.org/10.3390/agriculture11080707

Phatthalung, T. N., & Tangkananond, W. (2021). Rice grassy stunt virus-free and pathogenic rice plants affect the brown planthopper (Nilaparvata lugens Stål) life cycle. *Agriculture and Natural Resources*, *55*(3), 331–340. https://doi.org/10.34044/j.anres.2021.55.3.02

Preetha, R., Priyadarsini, M. J. P., & Nisha, J. S. (2024). Automated Brain Tumor Detection From Magnetic Resonance Images Using Fine-Tuned EfficientNet-B4 Convolutional Neural Network. *IEEE Access*, *12*, 112181–112195. https://doi.org/10.1109/ACCESS.2024.3442979

Rachburee, N., & Punlumjeak, W. (2022). Lotus species classification using transfer learning based on VGG16, ResNet152V2, and MobileNetV2. *IAES International Journal of Artificial Intelligence*, *11*(4), 1344–1352. https://doi.org/10.11591/ijai.v11.i4.pp1344-1352

Rybczak, M., & Kozakiewicz, K. (2024). Deep Machine Learning of MobileNet, Efficient, and Inception Models. *Algorithms*, *17*(3). https://doi.org/10.3390/a17030096

Siregar, S. P., Akbari, I., Poningsih, P., Wanto, A., & Solikhun, S. (2025). Enhancing Tomato Leaf Disease Detection via Optimized VGG16 and Transfer Learning Techniques. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *9*(3), 500–510. https://doi.org/10.29207/resti.v9i3.6410

Wang, S., Xu, D., Liang, H., Bai, Y., Li, X., Zhou, J., … Wei, W. (2025, February 1). Advances in Deep Learning Applications for Plant Disease and Pest Detection: A Review. *Remote Sensing*. Multidisciplinary Digital Publishing Institute (MDPI). https://doi.org/10.3390/rs17040698

Woerner, S., Jaques, A., & Baumgartner, C. F. (2024). A comprehensive and easy-to-use multi-domain multi-task medical imaging meta-dataset (MedIMeta). Retrieved from http://arxiv.org/abs/2404.16000

Zhang, Q. (2022). A novel ResNet101 model based on dense dilated convolution for image classification. *SN Applied Sciences*, *4*(1). https://doi.org/10.1007/s42452-021-04897-7

Zhang, Z., Zhan, W., Sun, K., Zhang, Y., Guo, Y., He, Z., … Gui, L. (2024). RPH-Counter: Field detection and counting of rice planthoppers using a fully convolutional network with object-level supervision. *Computers and Electronics in Agriculture*, *225*, 109242. https://doi.org/10.1016/J.COMPAG.2024.109242

Copyright holder:
Mursyid (2025)
First publication right:
Lopi Selayar: Jurnal Techno-Sociopreneur
This article is licensed under:

