JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)

VOL. 11. NO. 4 MAY 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i4.7800

PERFORMANCE EVALUATION OF TRANSFER LEARNING MODELS BASED ON OPTIMIZATION IN AGRICULTURAL PEST CLASSIFICATION

Attarik Mohammad[1*]; Sugiyarto Surono[1]; Aris Thobirin[1]

Mathematics[1]
Ahmad Dahlan University, Yogyakarta, Indonesia[1]
https://uad.ac.id/[1]
2200015011@webmail.uad.ac.id*, sugiyarto@math.uad.ac.id, aris.thobi@math.uad.ac.id

(*) Corresponding Author
(Responsible for the Quality of Paper Content)

Abstract— *Pests in agriculture lower crop yields and jeopardize the world's food security. Thus, quick and precise pest identification is crucial for successful pest management. Convolutional Neural Networks (CNN) and other deep learning techniques have made it possible to automatically classify pests thanks to developments in digital image processing and artificial intelligence (AI). Using three optimization algorithms, Adam, RMSprop, and SGD, this study assesses three transfer learning architectures, ResNet50V2, Xception, and EfficientNetB0. This study's primary contribution is a comparative analysis of CNN architectures and optimization techniques to determine the best configuration for classifying agricultural pests. The dataset, which includes 5494 pest photos from 12 classes, was acquired via Kaggle. A ratio of 80%, 10%, and 10% was used to separate the data into training, validation, and testing sets. The performance of feature extraction and classification was enhanced by applying transfer learning with fine-tuning. According to findings, Xception with Adam and RMSprop has the highest accuracy of 94%. Adam and EfficientNetB0 both achieved competitive results with the same precision. These results suggest that the performance of agricultural pest classification models is influenced by both optimizer and architecture choices.*

Keywords: *Agricultural Pests, Classification, CNN, Deep Learning, Optimization.*

Intisari— *Hama dalam pertanian menurunkan hasil panen dan mengancam ketahanan pangan dunia. Oleh karena itu, identifikasi hama yang cepat dan akurat sangat penting untuk keberhasilan pengendalian hama. Jaringan Saraf Konvolusional (CNN) dan teknik pembelajaran mendalam lainnya telah memungkinkan klasifikasi hama secara otomatis berkat kemajuan dalam pemrosesan gambar digital dan kecerdasan buatan (AI). Dengan menggunakan tiga algoritma optimasi, yaitu Adam, RMSprop, dan SGD, penelitian ini mengevaluasi tiga arsitektur pembelajaran transfer, yaitu ResNet50V2, Xception, dan EfficientNetB0. Kontribusi utama penelitian ini adalah analisis komparatif terhadap arsitektur CNN dan teknik optimasi untuk menentukan konfigurasi terbaik dalam mengklasifikasikan hama pertanian. Dataset yang terdiri dari 5.494 foto hama dari 12 kelas diperoleh melalui Kaggle. Perbandingan 80%, 10%, dan 10% digunakan untuk membagi data menjadi set pelatihan, validasi, dan pengujian. Kinerja ekstraksi fitur dan klasifikasi ditingkatkan dengan menerapkan transfer learning disertai fine-tuning. Berdasarkan temuan, Xception dengan Adam dan RMSprop memiliki akurasi tertinggi sebesar 94%. Adam dan EfficientNetB0 sama-sama mencapai hasil yang kompetitif dengan presisi yang sama. Hasil ini menunjukkan bahwa kinerja model klasifikasi hama pertanian dipengaruhi oleh pilihan optimizer dan arsitektur.*

Kata Kunci: *Hama Pertanian, Klasifikasi, CNN, Pembelajaran Mendalam, Optimasi*

INTRODUCTION

Agricultural pests are a major factor contributing to declining crop yields and pose a serious threat to global food security [1]. The Food and Agriculture Organization (FAO) reported in 2022 that insect and plant disease assaults cause 20-40% of the world's crop production to be lost each year [2]. In Indonesia, data from the Ministry of Agriculture in 2024 reported that pest attacks affected approximately 217,259 hectares of agriculture land, with 2,612 cases in rice crops remaining uncontrolled [3]. These conditions indicate that agricultural pests have a significant impact not only on local crop production but also on national and global food security.

Rapid and accurate pest identification plays an important role in pest management. Early detection enables farmers to implement appropriate control strategies more quickly and effectively, thereby minimizing potential crop losses [4]. However, conventional techniques for identifying pests are frequently laborious and highly dependent on specialized expertise. In order to facilitate quicker and more accurate pest detection, technical solutions are thus required.

Automated pest detection systems now have more options because to recent advancements in digital image processing and Artificial Intelligence (AI) [5]. Convolutional Neural Networks (CNN) are one type of AI approach that automatically learns hierarchical characteristics from visual input [6]. In agricultural pest classification, CNN can identify distinctive morphological and texture patterns in pest images, making them suitable for distinguishing different pest classes [7].

Although CNN models have achieved promising result, they usually require large datasets and substantial computational resources during training. Transfer learning provides an efficient approach by using parameters from models pre-trained on extensive datasets like ImageNet [8]. This approach can improve feature extraction and classification performance, especially when the available agricultural pest dataset is limited. Previous studies have shown that transfer learning based CNN models are effective for agricultural pest identification and can reduce the limitations of conventional identification methods that are often subjective and biased [9], [10].

Numerous prior studies have investigated the application of deep learning models for the categorization of agricultural pests. For instance, research conducted by [11] evaluated multiple CNN architectures using the IP102 dataset, which contains 102 classes of insect pests. The study compared DenseNet101, MobileNetV2, InceptionV3, and Xception models, achieving accuracies of 70%, 66%, 67%, and 69%, respectively, using the Adam optimizer.

Using the same dataset, [12] divided the pest classes into 5, 10, and 15 categories and applied Faster R-CNN with EfficientNetB4 and EfficientNetB7 architectures using the SGD optimizer. Their results showed that EfficientNetB4 achieved accuracies of 98%, 95%, and 90%, while EfficientNetB7 achieved 99%, 96%, and 93%, respectively.

Other studies have also investigated pest detection in crops such as corn, jatropha, rice, and wheat using deep learning models, including ResNet18, ResNet34, VGG16, and ResNet50. The AdamW optimizer was used to train these models, using 0.0001 as the learning rate and 0.00001 as the weight decay, producing classification accuracies below 80% [13].

The studies show that the choice of CNN architecture and the optimization technique employed during training are two important factors that affect model performance in pest classification tasks.

ResNet50V2 has become more well-known among CNN designs due to its enhanced residual training process, which addresses the vanishing gradient issue and facilitates the effective training of deeper networks [14], [15]. Another widely used architecture is Xception, which applies depthwise separable convolution to reduce computational cost while maintaining strong feature extraction capability [16], [17]. Additionally, EfficientNetB0 balances input resolution, network depth, and breadth via compound scaling, which enables the model to perform well with fewer parameters and reduced computational cost [18].

In addition to model architecture, the optimizer plays an important role in determining training performance. Optimizers update network parameters based on gradients and loss values, which affects convergence speed and training stability [19]. Common optimizers with distinct features include Root Mean Square Propagation (RMSprop), Adaptive Moment Estimation (Adam), and Stochastic Gradient Descent (SGD). RMSprop modifies the learning rate according to gradient magnitude, Adam integrates momentum and adaptive learning rate mechanisms, and SGD is renowned for its ease of use and steady convergence [20], [21,][22].

The combination of transfer learning and an appropriate optimizer can improve prediction performance [10]. However, inappropriate learning rate selection and limited dataset variation may

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 11. NO. 4 MAY 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i4.7800

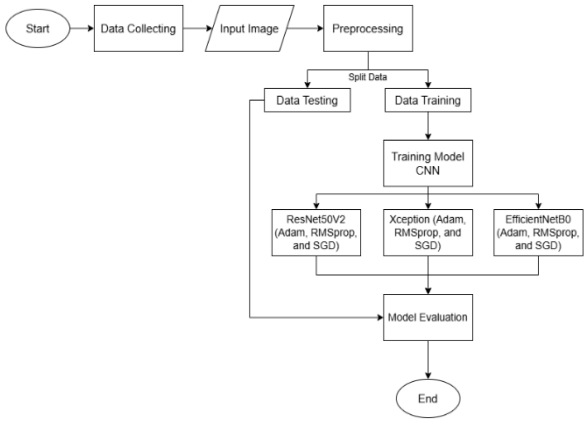
increase the risk of overfitting [23]. To address this issue, fine-tuning can be applied to pre-trained models by adjusting selected final layers so that the model can better learn task spesific features. This strategy has been used to optimize architecture such as ResNet50V2, Xception, and EfficientNetB0, improve classification accuracy, and reduce computational requirements [24], [25].

MATERIALS AND METHODS

This study used transfer learning approach based on three CNN architectures, namely ResNet50V2, Xception, and EfficientNetB0. Adam, RMSprop, and SGD were the three optimizers used to train each architecture. Comparing the effectiveness of various architecture and optimizers in the categorization of agricultural pests was the goal.

Research Stages

Data collection, data preprocessing, model training, and model assessment were the phase of the study procedure. Accuracy, precision, recall, and F1-score were used to evaluate the model's performance. Figure 1 shows the complete research procedure.



Source: (Research Results, 2025)
Figure 1. Research Workflow

Data Collection

The Kaggle website provided the dataset for this study, which included 5494 pictures of agricultural pests. Ants, bees, beetles, Caterpillars, earthworms, earwigs, grasshoppers, moths, slugs, snails, wasps, and weevils are among the twelve classes into which the pictures are divided. The suggested classification models were trained, validated, and tested using dataset, which offers a varied collection of pests photos. Tabel 1 displays the quantity of pictures in each class.

Table 1. Number of Images per Class

No	Class	Number of Images
1.	Ants	499
2.	Bees	500
3.	Beetle	416
4.	Caterpillar	434
5.	Earthworms	323
6.	Earwig	466
7.	Grasshopper	485
8.	Moth	497
9.	Slug	391
10.	Snail	500
11.	Wasp	498
12.	Weevil	485
Total		5494

(Research Results, 2025)

Table 1 displays pictures from the dataset that were obtained from the Kaggle website (https://www.kaggle.com/datasets/vencerlanz09/agricultural-pests-image-dataset).

Data Splitting and Preprocessing

This stage was an important step before the image data entered the training process. Image resizing was performed to ensure that all input image had consistent dimensions suitable for the CNN models. Next, normalization was applied to scale pixel values into a uniform range, which helped improve training stability.

The dataset distribution was first analyzed identify potential class imbalance. With 500 photos in the largest class and 323 in the lowest, the imbalance ratio was around 1.55, this indicates that the dataset was relatively balanced. During training, data augmentation techniques such rescaling, rotation, width shift, height shift, shear, zoom, and horizontal flipping were used to further increase data variety and lessen the possible consequences of class imbalance.

The dataset was divided into training, validation, and testing sets at ratios of 80%, 10%, and 10%, respectively, after augmentation and preprocessing. Table 2 displays the final data distribution for each subgroup.

Table 2. Image Data Splitting

No	Data Spliting	Class	Amount
1.	Training	Ants	399
2.		Bees	400
3.		Beetle	333
4.		Caterpillar	347
5.		Earthworms	258
6.		Earwig	373
7.		Grasshopper	388
8.		Moth	398
9.		Slug	313
10.		Snail	400

No	Data Spliting	Class	Amount
11.		Wasp	398
12.		Weevil	388
	Total Training		4395
13.	Testing	Ants	50
14.		Bees	50
15.		Beetle	41
16.		Caterpillar	43
17.		Earthworms	32
18.		Earwig	47
19.		Grasshopper	49
20.		Moth	50
21.		Slug	39
22.		Snail	50
23.		Wasp	50
24.		Weevil	49
	Total Testing		550
25.	Validation	Ants	50
26.		Bees	50
27.		Beetle	42
28.		Caterpillar	44
29.		Earthworms	33
30.		Earwig	46
31.		Grasshopper	48
32.		Moth	49
33		Slug	39
34.		Snail	50
35		Wasp	50
36.		Weevil	48
	Total Validation		549

Source: (Research Results, 2025)

Where the data has been categorized based on class. The following are sample images for each class of agricultural pests, presented in table 3.

Table 3. Image per Class

Class	Image Example
Ants	
Bees	
Beetle	
Caterpillar	
Earthworms	
Earwig	
Grasshopper	
Moth	
Slug	
Snail	
Wasp	
Weevil	

Source: (Research Results, 2025)

VOL. 11. NO. 4 MAY 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i4.7800

Architecture Model CNN

Residual Network (ResNet) is a CNN architecture intended to solve the issue of the vanishing gradient through the use of residual connection. Convolutional, pooling, and fully connected layers make up the network, with skip connections that enable more effective training of deep networks. An improved version, ResNet50V2, introduces modifications of the residual block structure that enchance training performance on large dataset such as ImageNet [14], [24].

An expansion of the Inception architecture, Xception uses depthwise separable convolution in favor of normal convolution to increase computational efficiency while preserving robust features extraction capabilities. Entry flow, middle flow, and exit flow are the three primary flows that make up the architecture and incorporates residual connections to facilitate deeper network training. This design enables Xception to capture complex image features with reduced computational cost [27].

EfficientNetB0 is a CNN architecture that introduces a compound scaling technique that balance the simultaneous scaling of network breadth, depth, and input resolution. This approach allows EfficientNet to achieve strong classification performance while maintaining computational efficiency and a relatively small number of parameters. EfficientNetB0, the foundational model of the EfficientNet family, successfully balances computational cost and accuracy [18].

All models were trained for 50 epochs with a batch size of 32 and learning rate of 0.0001 using Adam, RMSprop, and SGD optimizers with the categorical cross-entropy loss function. The final 30 layers of the pre-trained models were unfrozen while the preceding layers were frozen in order to do fine-tuning. ReduceLROnPlateau was utilized with a patience value of three epochs to lower the learning rate by a factor of 0.5.

Optimizer

Adam is a well-liked deep learning optimization method that combines the advantage of Adaptive Gradient (AdaGrad) and RMSprop. Adam computes adaptive learning rates for each parameter using estimates of the gradient's first and second moments. This mechanism allows Adam to achieve faster convergence and stable training performance, particularly when working with big datasets and intricate neural network architecture [22]. The following is a definition of Adam update rules.

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \quad (1)$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \quad (2)$$

$$\theta_{t+1} = \theta_t - \eta \frac{m_t}{\sqrt{v_t}+\epsilon} \quad (3)$$

where η is the learning rate, g_t is the gradient at step t, m_t is the first moment estimate, v_t is the second moment estimate, and ϵ is a tiny constant to avoid division by zero.

RMSprop is an optimization technique that accelerates neural network training convergence by using adaptive learning rate modifications for each parameter. It divides the gradient by the average root while monitoring a moving average of the squared gradient. This approach helps stabilize the learning process and prevents the learning rate from becoming excessively large, making RMSprop effective for handling nonstationary objectives in deep learning models [28]. The RMSprop update rule defined as.

$$S_t = \beta_1 \cdot S_{t-1} + (1-\beta_1) g_t^2 \quad (4)$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{s_t}+\epsilon} g_t \quad (5)$$

where η is the learning rate, β_1 is the decay rate, g_t is the gradient at step t, and ϵ is a tiny constant to avoid division by zero.

One of the most straightforward and popular optimization algorithms in machine learning is SGD. The model parameters are updated iteratively based on gradients calculated from individual samples or small bacthes of training data. Although SGD may converge more slowly compared to adaptive optimizer, it is known for its simplicity, computational efficiency, and strong generalization performance when properly tuned [21]. The SGD formula is given as follows.

$$v_t = \beta v_{t-1} + \eta \nabla_w f(w) \quad (6)$$

$$w_{t+1} = w_t - v_t \quad (7)$$

Where η is the learning rate, w is the model weight, v_t is the momentum at iteration t, and $\nabla_w f(w)$ is the gradient of the loss function with respect to w.

Mathematical Formulation

Next, we will explain the important components that make up the CNN model architecture, as follows.

1. Convolution Layer

The convolution layer generates feature maps by applying a kernel (filter) to the input image matrix. This operation multiplies each element of the kernel with the corresponding region of the input image and sums the results. The convolution operation is represented by the symbol (∗) [29]. The formula can be follows as.

$$Y_{(i,j)} = \sum_{m=1}^{k} \sum_{n=1}^{k} I_{(i-1)\cdot s+m,(j-i)\cdot s+n} * K_{(m,n)} \quad (8)$$

From equation (8), where K is the kernel, I is the input image matrix, and $Y_{i,j}$ is the value of the generated features at location (i, j).

2. Rectified Linear Unit (ReLU)

Following the convolution procedure, the network is given nonlinearity using an activation function known as ReLU. It removes negative values from the feature map and retains only positive values. The definition of the ReLU function is.

$$\sigma(x) = \max(0, x) \quad (9)$$

where x is the input value and $\sigma(x)$ is the output of the activation function.

3. Max Pooling

Max pooling decreases the spatial size of feature maps by selecting the biggest value inside a constrained region of the input feature map. This operation helps reduce computational complexity while preserving important features [30]. The expression for the max pooling operation is.

$$Pool_{(i,j)} = \max\left(C_{(i-1)s+m,(j-1)s+n}\right) \quad (10)$$

Where $Pool_{(i,j)}$ is the final pooled feature at location (i, j) and C is the input feature map.

4. Global Average Pooling (GAP)

GAP is a pooling method that creates a single representative value by calculating the average value of every element in a feature map. At the conclusion of CNN architecture, this procedure is frequently employed to replace completely connected layer [30], with the following formula.

$$GAP_{(i,j)} = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{(i,j)} \quad (11)$$

Where $I_{(i,j)}$ represents the input feature map and $m \times n$ denotes its spatial dimensions.

5. Softmax

An activation function called Softmax is utilized in the neural network last layer for multiclass categorization. It convert the output values into probability distribution for each class [28]. The softmax function is defined as.

$$softmax(z_i) = \frac{\exp(z_i)}{\sum_j \exp(z_j)} \quad (12)$$

Model Evaluation

The last step is model assessment, which is carried out using the test dataset. Accuracy, precision, recall, and F1-score are among the evaluation metrics that are derived from the categorization report. Precision is the accuracy of the projected positive occurrences, recall assesses the model's ability to correctly identify samples of each class, and F1-score is the harmonic mean of precision and recall. Accuracy displays the model's overall ability to correctly classify each test sample [31]. The following is the formula for accuracy, precision, recall, and F1-score.

$$Precision = \frac{TP}{TP+FP} \quad (13)$$

$$Recall = \frac{TP}{TP+FN} \quad (14)$$

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision+Recall} \quad (15)$$

$$Accuracy = \frac{TP+TN}{N} \quad (16)$$

True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) are explained.

RESULTS AND DISCUSSION

The experimental outcomes of the transfer learning models employed in this investigation are shown in this section. Training accuracy, training loss, confusion matrices, and classification metrics were used to assess the model's performance.

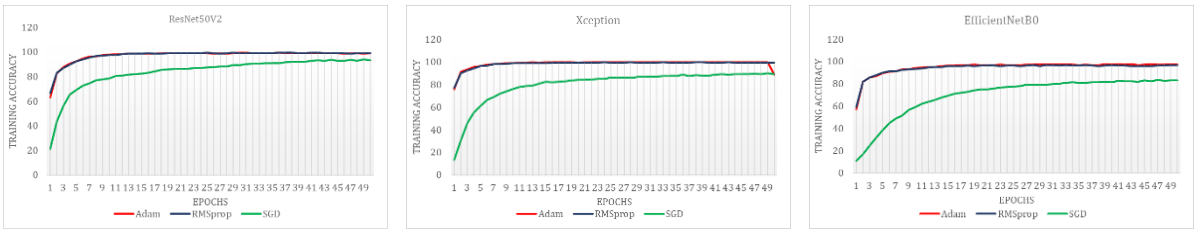
Training Accuracy and Loss Curve

The training accuracy and training loss curves of ResNet50V2, Xception, and

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 11. NO. 4 MAY 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i4.7800

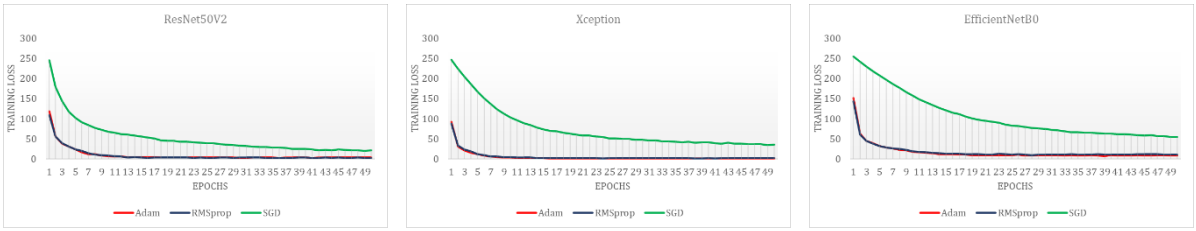
EfficientNetB0, optimized across 50 training epochs using Adam, RMSprop, and SGD, are displayed in Figures 2 and 3.



Source: (Research Results, 2025)

Figure 2. Training accuracy curves of ResNet50V2, Xception, and EfficientNetB0 using Adam, RMSprop, and SGD optimizers.

Figure 2 illustrates how all models training accuracy rose quickly in the early epochs and stabilized in the later ones. Models trained with Adam and RMSprop achieved higher training accuracy and converged faster than those trained with SGD. In contrast, SGD showed a slower increase in accuracy and reached lower final accuracy values.



Source: (Research Results, 2025)

Figure 3. Training loss curves of ResNet50V2, Xception, and EfficientNetB0 using Adam, RMSprop, and SGD optimizers.

Figure 3 demonstrates how the training loss rapidly dropped in the early epochs before steadily stabilizing as training went on. Adam and RMSprop reduced the loss more quickly and reached values close to zero in earlier epochs.

Meanwhile, SGD showed higher initial loss values and a more gradual decrease throughout the training process.

Classification Performance

Precision, recall, and F1-score are the three main metrics used to evaluate the performance of classification models.

Table 4. Classification performance of transfer learning models using different optimizers.

Model	Optimizer	Accuracy	Precision	Recall	F1-Score
ResNet50V2	Adam	0.90	0.90	0.90	0.90
	RMSprop	0.89	0.89	0.89	0.89
	SGD	0.89	0.89	0.89	0.89
Xception	Adam	0.94	0.94	0.94	0.94
	RMSprop	0.94	0.94	0.94	0.94
	SGD	0.91	0.91	0.91	0.91
EfficientNetB0	Adam	0.94	0.94	0.94	0.94
	RMSprop	0.93	0.93	0.93	0.93
	SGD	0.89	0.89	0.89	0.88

Source: (Research Results, 2025)

Based on Table 4, Xception with Adam and RMSprop achieved the highest performance, with accuracy, precision, recall, and F1-score values of 0.94. EfficientNetB0 with Adam also achieved

VOL. 11. NO. 4 MAY 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i4.7800

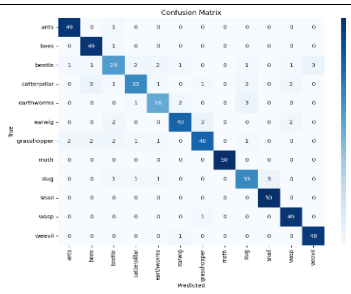
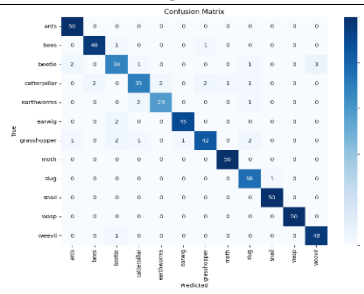
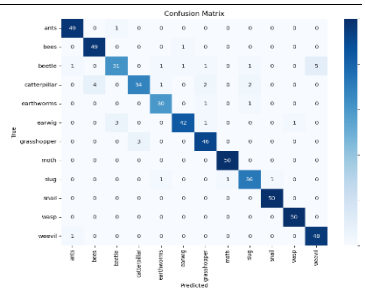
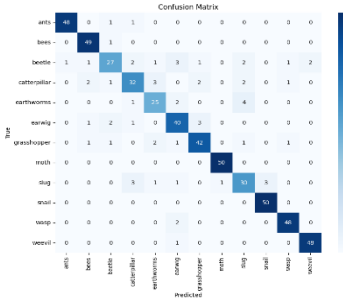
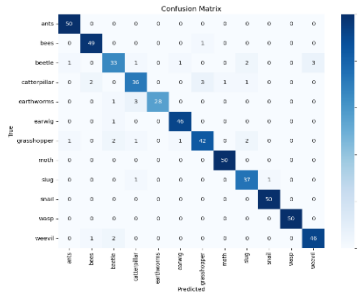
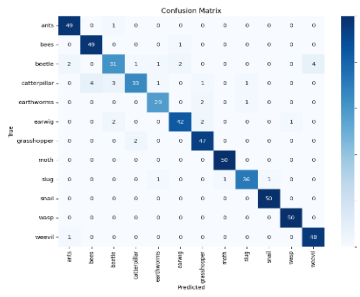
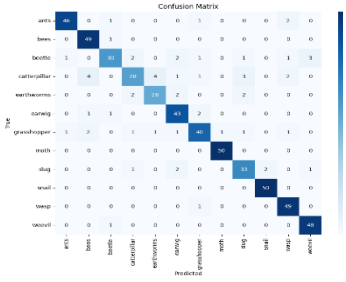
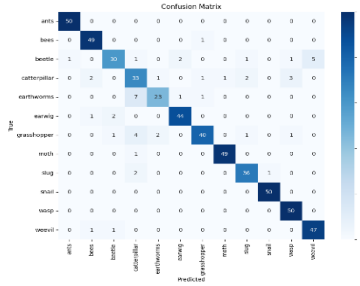
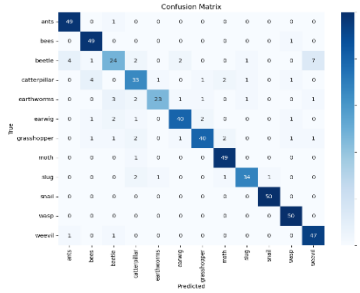
JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)

comparable performance, with an accuracy of 0.94. Meanwhile, ResNet50V2 showed relatively stable results across the three optimizers, with accuracy values ranging from 0.89 to 0.90. Overall, the results indicate that adaptive optimizers, particularly Adam and RMSprop, produced better performance than SGD in most model combinations.

Confusion Matrix

A matrix that offers crucial details regarding a classification model's performance is called confusion matrix. A confusion matrix can show locations where the model might make prediction errors in addition to a classification report.

Table 5. Result of Confusion Matrices of transfer learning models using Adam, RMSprop, and SGD optimizers.

	ResNet50V2	Xception	EfficientNetB0
Adam			
RMSprop			
SGD			

Source: (Research Results, 2025)

Table 5 presents the confusion matrix results of the ResNet50V2, Xception, and EfficientNetB0 models optimized using Adam, RMSprop, and SGD. The confusion matrices illustrate the distribution of predicted classes compared to the actual classes for each model and optimizer combination. In general, most predictions were concentrated along the diagonal elements of the confusion matrices. This indicates that most samples were correctly classified. The Xception and EfficientNetB0 models showed stronger classification patterns, particularly when trained with Adam and RMSprop. Misclassifications appeared in several classes, but the number of incorrect predictions was relatively small compared with the correctly classified samples.

Discussion

The experimental results show that both model architecture and optimizer selection affected classification performance. Xception achieved the best overall results when combined with Adam and RMSprop. This performance may be related to its depthwise separable convolution mechanism, which supports efficient feature extraction while reducing computational complexity. EfficientNetB0 also produced competitive results, especially when trained with Adam. Its compound scaling

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 11. NO. 4 MAY 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i4.7800

mechanism allows the model to balance network depth, width, and input resolution, which may contribute to its strong classification performance with relatively efficient computation. ResNet50V2 produced slightly lower results than Xception and EfficientNetB0, but its performance remained stable across all optimizers. This indicates that the residual connection mechanism still provides reliable feature extraction for agricultural pest classification, although it did not achieve the highest performance in this experiment. The optimizer comparison shows that Adam and RMSprop generally provided faster convergence and better classification performance than SGD. This pattern is also reflected in the training accuracy and loss curves, where Adam and RMSprop reached stable performance earlier than SGD. Overall, the results suggest that Xception with Adam or RMSprop provides the most effective configuration for agricultural pest classification in this study.

CONCLUSION

Based on the research results, this study primary contribution is the comparative analysis of three CNN architectures, ResNet50V2, Xception, and EfficientNetB0, for agricultural pest classification utilizing three optimization algorithms, Adam, RMSprop, and SGD. The experimental findings demonstrate that classification performance and convergence behavior during training are influenced by the choice of both optimizer and architecture. With accuracy, precision, recall, and F1-score values of 0.94, Xception optimized using Adam and RMSprop performed the best among the assessed models. EfficientNetB0 optimized using Adam also achieved comparable performance with an accuracy of 0.94, while ResNet50V2 produced stable but slightly lower results across the evaluated optimizers. In general, adaptive optimizers such as Adam and RMSprop provided better performance than SGD, which showed slower convergence and lower overall results. These findings indicate that selecting an appropriate combination of CNN architecture and optimization algorithm is important for improving the performance of deep learning models in agricultural pest classification. Based on the presented results, Xception combined with Adam or RMSprop can be considered the most effective configuration among the evaluated models.

REFERENCE

[1] S. Zhao, J. Liu, Z. Bai, C. Hu, and Y. Jin, "Crop Pest Recognition in Real Agricultural Environment Using Convolutional Neural Networks by a Parallel Attention Mechanism," *Front. Plant Sci.*, vol. 13, no. February, pp. 1–14, 2022, doi: 10.3389/fpls.2022.839572.

[2] FAO, "FAO ' s Plant Production and Protection Division," Rome, 2022. doi: https://doi.org/10.4060/cc2447en.

[3] R. I. Kementerian Pertanian, "LAKIN Kementerian Pertanian 2024," 2024.

[4] B. Hossain, U. Kanti, S. Ali, A. Kawser, and S. M. A. Al, "Image-based Pest Identification Using Support Vector Machine for Agricultural Crop Protection," *Asian J. Res. Crop Sci.*, vol. 10, no. 3, pp. 13–22, 2025, doi: https://doi.org/10.9734/ajrcs/2025/v10i3368.

[5] S. Khalid, H. M. Oqaibi, and M. Aqib, "Small Pests Detection in Field Crops Using Deep Learning Object Detection," *Sustainability*, vol. 15, no. 8, pp. 1–19, 2023, doi: https://doi.org/10.3390/su15086815.

[6] C. Li, T. Zhen, and Z. Li, "Image Classification of Pests with Residual Neural Network Based on Transfer Learning," *Appl. Sci.*, vol. 12, no. 9, May 2022, doi: 10.3390/app12094356.

[7] Z. Alshingiti, R. Alaqel, J. Al-Muhtadi, Q. E. U. Haq, K. Saleem, and M. H. Faheem, "A Deep Learning-Based Phishing Detection System Using CNN, LSTM, and LSTM-CNN," *Electron.*, vol. 12, no. 1, Jan. 2023, doi: 10.3390/electronics12010232.

[8] Y. Chen, M. Chen, M. Guo, J. Wang, and N. Zheng, "Pest recognition based on multi-image feature localization and adaptive filtering fusion," *Front. Plant Sci.*, vol. 14, 2023, doi: 10.3389/fpls.2023.1282212.

[9] C. H. Praharsha, A. Poulose, and C. Badgujar, "Comprehensive Investigation of Machine Learning and Deep Learning Networks for Identifying Multispecies Tomato Insect Images," *Sensors*, vol. 24, no. 23, Dec. 2024, doi: 10.3390/s24237858.

[10] D. Popescu, A. Dinca, L. Ichim, and N. Angelescu, "New trends in detection of harmful insects and pests in modern agriculture using artificial neural networks. a review," 2023, *Frontiers Media SA*. doi: 10.3389/fpls.2023.1268167.

[11] A. P. Syahputra, A. C. Siregar, and R. W. S. Insani, "Comparison of CNN Models With Transfer Learning in the Classification of Insect Pests," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 17, no. 1, p. 103, Feb. 2023, doi: 10.22146/ijccs.80956.

[12] P. B. Mallikarjuna, "Insect Pest Image Detection and Classification using Deep Learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 9, pp. 411–421, 2022, doi: 10.14569/IJACSA.2022.0130947.

[13] A. Amrani, D. Diepeveen, D. Murray, M. G. K. Jones, and F. Sohel, "Multi-task learning model for agricultural pest detection from crop-plant imagery : A Bayesian approach," *Comput. Electron. Agric.*, vol. 218, no. July 2023, p. 108719, 2024, doi: 10.1016/j.compag.2024.108719.

[14] M. S. A. M. Al-Gaashani *et al.*, "Deep transfer learning with gravitational search algorithm for enhanced plant disease classification," *Heliyon*, vol. 10, no. 7, Apr. 2024, doi: 10.1016/j.heliyon.2024.e28967.

[15] M. K. U. Ahamed *et al.*, "DTLCx: An Improved ResNet Architecture to Classify Normal and Conventional Pneumonia Cases from COVID-19 Instances with Grad-CAM-Based Superimposed Visualization Utilizing Chest X-ray Images," *Diagnostics*, vol. 13, no. 3, Feb. 2023, doi: 10.3390/diagnostics13030551.

[16] M. A. Hasan and K. Dey, "Depthwise Separable Convolutions with Deep Residual Convolutions," arXiv preprint arXiv:2411.07544, pp. 1–9, Nov. 2024, doi: 10.48550/arXiv.2411.07544.

[17] E. R. Agustina, H. Marcos, and U. A. Purwokerto, "PARKINSONS DISEASE DETECTION USING INCEPTION AND X-CEPTION," *JITK (JURNAL ILMU Pengetah. DAN Teknol. KOMPUTER)*, vol. 11, no. 1, pp. 1–7, 2025, doi: 10.33480/jitk.v11i1.6277

[18] P. Dollár, M. Singh, and R. Girshick, "Fast and Accurate Model Scaling," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 924–932, 2021, doi: 10.1109/CVPR46437.2021.00098.

[19] A. Ly and P. Gong, "Optimization on multifractal loss landscapes explains a diverse range of geometrical and dynamical properties of deep learning," *Nat. Commun.*, vol. 16, no. 3252, pp. 1–13, 2025, doi: 10.1038/s41467-025-58532-9.

[20] M. Aljebreen, H. A. Mengash, F. Kouki, and A. Motwakel, "Improved Artificial Ecosystem Optimizer with Deep-Learning-Based Insect Detection and Classification for Agricultural Sector," *Sustainability*, vol. 15, no. 20, p. 14770, 2023, doi: 10.3390/su152014770.

[21] Y. Tian and Y. Zhang, "Recent Advances in Stochastic Gradient Descent in Deep Learning," *Mathematics*, vol. 11, no. 3, pp. 1–23, 2023, doi: 10.3390/math11030682.

[22] M. Reyad, A. M. Sarhan, and M. Arafa, "A modified Adam algorithm for deep neural network optimization," *Neural Comput. Appl.*, vol. 35, no. 23, pp. 17095–17112, 2023, doi: 10.1007/s00521-023-08568-z.

[23] Y. Yuan, J. Sun, and Q. Zhang, "An Enhanced Deep Learning Model for Effective Crop Pest and Disease Detection," *J. Imaging*, vol. 10, no. 11, 2024, doi: 10.3390/jimaging10110279.

[24] Y. H. Gu, H. Yin, D. Jin, J. Park, and S. J. Yoo, "Image-Based Hot Pepper Disease and Pest Diagnosis Using Transfer Learning and Fine-Tuning," *Front. Plant Sci.*, vol. 12, no. December, pp. 1–11, 2021, doi: 10.3389/fpls.2021.724487.

[25] J. Yao, J. Liu, Y. Zhang, and H. Wang, "Identi fi cation of winter wheat pests and diseases based on improved convolutional neural network," *Open Life Sci.*, vol. 18, pp. 1–11, 2023, doi: doi.org/10.1515/biol-2022-0632.

[26] Y. F. Riti, R. P. Kristianto, I. Science, U. Katolik, and D. Cendika, "PERFORM COMPARATION OF DEEP LEARNING METHODS IN GENDER CLASSIFICATION FROM FACIAL IMAGES," vol. 10, no. 4, pp. 926–936, 2025, doi: 10.33480/jitk.v10i4.4717.

[27] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," Apr. 2017, [Online]. Available: http://arxiv.org/abs/1610.02357

[28] P. Arafin and A. H. M. M. Billah, "Deep learning-based concrete defects classification and detection using semantic segmentation," *Struct. Heal. Monit.*, vol. 23, no. 1, pp. 383–409, 2024, doi: 10.1177/14759217231168212.

[29] S. Cb and R. Jothilakshmi, "Image Extraction using Convolution Kernel," *J. Comput. Math.*, vol. 5, no. August, pp. 41–47, 2021.

[30] A. Zafar *et al.*, "A Comparison of Pooling Methods for Convolutional Neural Networks," *Appl. Sci.*, vol. 12, pp. 1–21, 2022.

[31] K. M. Sujon, R. Hassan, K. Choi, and A. Samad, "empirical evidence from advanced statistics , ML , and XAI for evaluating business predictive models," *J. Big Data*, vol. 12, no. 268, pp. 1–45, 2025, doi: https://doi.org/10.1186/s40537-025-01313-4.

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri