Jurnal Komputer dan Elektro Sains, Vol.4 No.2, September 2026, pp 1-11
https://ejournal.sultanpublisher.com/komets

Received : July 29, 2026; Revision: August 12, 2026; Accepted: August 13, 2026; Publication: September 1, 2026.

E-ISSN: 3021-8462, DOI: https://doi.org/10.58291/komets.v4i2.654

Comparative Analysis of VGG16, VGG19, and EfficientNetB5 Architectures for Image-Based Banana Ripeness Classification

Didin Adri[1*], Rando[1], Wa Ode Vivi Murwati[1]
[1] Computer Systems Engineering, Faculty of Engineering, Muhammadiyah Buton University, Baubau, 93724, Indonesia
Corresponding author: Didin Adri (e-mail: adrididin@gmail.com)

ABSTRAK
Banana ripeness classification is an important task in agriculture and food processing that supports automatic and objective sorting processes. Advances in deep learning, particularly Convolutional Neural Networks (CNNs), have demonstrated high capability in image classification tasks. This research aims to analyze and compare the performance of three CNN architectures, namely VGG16, VGG19, and EfficientNetB5, for image-based banana ripeness classification, considering both classification performance and computational efficiency. The dataset used was the Banana Ripeness Classification Dataset, consisting of 1,080 images categorized into three classes: unripe, ripe, and overripe. The dataset was divided into training (70%), validation (15%), and testing (15%) sets. All models were implemented using a transfer learning approach with ImageNet pretrained weights and trained for 30 epochs using the Adam optimizer. Model performance was evaluated using accuracy, precision, recall, and F1-score, while computational performance was assessed based on the total number of parameters, model size, and average inference time per image. The results show that VGG16 and VGG19 achieved the best classification performance, obtaining an accuracy of 99.38%, precision of 0.99, recall of 0.99, and F1-score of 0.99. In contrast, EfficientNetB5 achieved an accuracy of 91.36%, precision of 0.92, recall of 0.91, and F1-score of 0.91. In terms of computational performance, VGG16 contained 15.24 million parameters, with a model size of 58.2 MB and an average inference time of 32.6 ms per image. VGG19 contained 20.29 million parameters, with a model size of 77.4 MB and an average inference time of 39.8 ms per image, while EfficientNetB5 contained 30.50 million parameters, with a model size of 116.8 MB and an average inference time of 45.7 ms per image. These findings demonstrate that VGG16 and VGG19 provide superior classification performance compared with EfficientNetB5 on the utilized dataset, while VGG16 also offers the lowest computational requirements among the evaluated architectures. Therefore, VGG16 represents a promising architecture for developing an accurate and computationally efficient image-based banana ripeness classification system for automated sorting applications.

KEYWORDS: Image Classification; Banana Ripeness; Deep Learning; Convolutional Neural Networks.

1. INTRODUCTION

Fruit ripeness classification plays a crucial role in modern agriculture because it directly influences harvesting decisions, product quality, storage management, and market value. Accurate assessment of fruit maturity enables farmers and distributors to minimize post-harvest losses while ensuring that fruits reach consumers at the optimal stage of ripeness. Traditionally, fruit ripeness has been evaluated through manual inspection based on color, texture, and appearance. However, this approach is subjective, labor-intensive, and highly dependent on the experience of human inspectors, resulting in inconsistent classification outcomes. Consequently, automatic fruit ripeness classification using digital image processing has attracted increasing attention as an effective alternative for improving consistency and efficiency [1], [2].

Recent advances in deep learning, particularly Convolutional Neural Networks (CNNs), have significantly improved image classification performance across various agricultural applications. CNN-based models have been successfully applied to plant disease identification, fruit recognition, and ripeness classification because of their ability to automatically extract hierarchical visual features from images. Among the widely adopted CNN architectures, VGG16 and VGG19 are recognized for their simple yet effective deep feature extraction capability, while EfficientNetB5 introduces a compound scaling strategy that aims to achieve better computational efficiency without sacrificing classification performance. Previous studies have demonstrated promising results using these architectures individually for agricultural image classification. However, most existing studies primarily evaluate a single architecture or compare only a limited number of models under different datasets and experimental settings, making direct performance comparisons difficult [3], [4].

Despite these advances, a comprehensive evaluation of representative CNN architectures for banana ripeness classification under identical experimental conditions remains limited. Differences in datasets, preprocessing techniques, data augmentation strategies, hyperparameter configurations, and evaluation metrics across previous studies hinder objective assessment of model performance. As a result, it remains unclear which architecture provides the most suitable balance between classification accuracy and computational efficiency for banana ripeness classification [5].

To address this limitation, this research performs a systematic comparative evaluation of three representative CNN architectures, namely VGG16, VGG19, and EfficientNetB5, using a publicly available banana ripeness image dataset. All models are implemented using *transfer learning* with the same preprocessing procedures, data augmentation techniques, training strategy, and evaluation protocol to ensure a fair comparison. Model performance is evaluated using accuracy, precision, recall, F1-score, and confusion matrix analysis, while training behavior and computational efficiency are also considered to provide a comprehensive assessment [6], [7].

The main objective of this research is to analyze and compare the classification performance of VGG16, VGG19, and EfficientNetB5 in recognizing three banana ripeness categories: unripe, ripe, and overripe. These three categories were selected based on the class structure of the utilized Banana Ripeness Classification Dataset, providing a coarse-grained representation of the major stages of banana ripeness. While this three-class configuration enables a simpler and more consistent classification task, it provides less detailed maturity information than fine-grained commercial grading systems, which may distinguish additional intermediate ripeness stages or quality levels. Therefore, the three-class approach represents a trade-off between dataset availability, classification simplicity, and grading granularity.The novelty of this research lies in establishing a consistent benchmarking framework that evaluates representative CNN architectures under identical experimental settings, enabling an objective comparison of their classification capability and computational characteristics. The findings are expected to provide practical guidance for selecting appropriate *deep learning* models for intelligent fruit ripeness classification systems and contribute to the development of *computer vision*-based technologies for smart agriculture [8], [9].

2. METHODS

2.1. Research Design

This research employed an experimental research design to compare the performance of three Convolutional Neural Network (CNN) architectures, namely VGG16, VGG19, and EfficientNetB5, for banana ripeness classification. The experiments were conducted under a controlled and consistent experimental framework to facilitate a fair comparison among the three models. All models were implemented using transfer learning and trained using the same dataset, data split, preprocessing techniques, data augmentation strategies, batch size, number of training epochs, optimizer, loss function, and evaluation metrics. However, the learning rate was treated as an architecture-specific hyperparameter and was adjusted according to the optimization characteristics of each pretrained architecture. The overall workflow consisted of dataset acquisition, image preprocessing, data augmentation, model development, model training, performance evaluation, and comparative analysis[10], [11].

2.2. Dataset Acquisition

The dataset used in this research was obtained from the publicly available Banana Ripeness Classification Dataset on Kaggle. The dataset contains RGB images of bananas categorized into three ripeness levels: unripe, ripe, and overripe. Representative samples of each ripeness class are presented in Table 1.

Table 1. Distribution of Banana Ripeness Dataset

Image	Class	Number of Images
	*Unripe*	360
	*Ripe*	360
	*Overripe*	360
Total		**1080**

A total of 1,080 images were used in the experiments. The dataset was divided into training, validation, and testing subsets with a ratio of 70%, 15%, and 15%, respectively, resulting in 756 training images, 162 validation images, and 162 testing images, respectively. The three subsets were maintained separately to prevent data leakage between training and evaluation. Although the dataset size is relatively modest for deep learning experiments, it provides a consistent benchmark for the three-class classification task considered in this study. The use of ImageNet-pretrained models through transfer learning reduces the dependence on training a deep network entirely from scratch, while data augmentation was applied to the training set to increase the variability of training samples. The fixed 70:15:15 split was retained across all three architectures to ensure a consistent and fair comparison under identical experimental conditions. Nevertheless, the relatively limited dataset size and the use of a single hold-out split constitute limitations of this study, and future work should employ repeated or k-fold cross-validation on larger and more diverse banana image datasets to further assess the statistical robustness and generalizability of the results. The training set was used to learn feature representations, the validation set was used to monitor model performance during training and reduce overfitting, while the testingset was reserved exclusively for final model evaluation[12], [13].

2.3. Image Preprocessing and Data Augmentation

Prior to model training, all images were preprocessed to match the input dimensions required by each CNN architecture. Images for VGG16 and VGG19 were resized to 224 × 224 pixels, whereas EfficientNetB5 required an input size of 456 × 456 pixels. Pixel values were normalized according to the preprocessing function corresponding to each pretrained model. To improve model generalization and reduce overfitting, data augmentation was applied to the training dataset. The augmentation process included random rotation, horizontal flipping, zooming, width shifting, height shifting, shearing, and brightness variation. Validation and testing datasets were not augmented to ensure an unbiased evaluation of model performance.

2.4. Model Development

Three pretrained CNN architectures were implemented using transfer learning: VGG16, VGG19, and EfficientNetB5. The convolutional base of each pretrained network, initialized with ImageNet weights, three pretrained CNN architectures were implemented using transfer learning: VGG16, VGG19, and EfficientNetB5. The convolutional base of each pretrained network was initialized with ImageNet weights, while the original classification layers were removed and replaced with task-specific classification layers. For VGG16 and VGG19, the extracted feature maps were connected to a Flatten layer followed by a fully connected Dense layer with ReLU activation. Dropout layers with rates of 0.5 and 0.3 were incorporated to reduce overfitting, and the final output layer consisted of three neurons with Softmax activation corresponding to the three banana ripeness classes. EfficientNetB5 employed a GlobalAveragePooling2D layer instead of Flatten to reduce the number of parameters in the classification head. A Dense layer with 512 units and ReLU activation, followed by a Dropout layer with a rate of 0.5, was subsequently added before the final three-neuron Softmax layer.

Fine-tuning was performed by freezing the initial layers of each pretrained network while keeping the remaining layers trainable. Specifically, 15 layers were frozen and 4 layers were kept trainable in VGG16, whereas 18 layers were frozen and 4 layers were kept trainable in VGG19. For EfficientNetB5, the first 400 layers were frozen, while the remaining layers were kept trainable. Based on these configurations, VGG16 contained 2.36 million trainable parameters and 12.88 million non-trainable parameters, VGG19 contained 2.36 million trainable parameters and 17.93 million non-trainable parameters, and EfficientNetB5 contained 12.76 million trainable parameters and 17.74 million non-trainable parameters. Thus, the corresponding total parameter counts were 15.24 million, 20.29 million, and 30.50 million for VGG16, VGG19, and EfficientNetB5, respectively.

2.5. Model Training

All models were trained using the Adam optimizer with the categorical cross-entropy loss function. A batch size of 16 and 30 training epochs were used for all experiments.The learning rate was treated as an architecture-specific hyperparameter and was set to 0.00001 for VGG16 and VGG19 and 0.0001 for EfficientNetB5. The different learning rates were selected to accommodate differences in the optimization behavior and architectural characteristics of the pretrained networks, particularly their feature-extraction structures and fine-tuning configurations. Therefore, although the core experimental conditions were kept consistent across models, the learning rate was intentionally adjusted as a model-specific training hyperparameter. In addition, the ReduceLROnPlateau callback was employed to automatically decrease the learning rate when the validation performance stopped improving, thereby improving training stability and convergence. [14], [15]

2.6. Performance Evaluation

Model performance was evaluated using the independent testing dataset that was not involved during model training. A confusion matrix was first generated to summarize the classification results for each ripeness category. Subsequently, four evaluation metrics were calculated, namely accuracy, precision, recall, and F1-score. Accuracy was considered the primary performance indicator because it measures the proportion of correctly classified images among all testing samples. Precision, recall, and F1-score were used to provide complementary information regarding class-wise classification performance.

In addition to classification performance, computational characteristics were evaluated using three metrics: total model parameters, model disk size, and average inference time per image. The total number of parameters was reported in millions (M) to represent model complexity, while model disk size was measured in megabytes (MB) to characterize storage requirements. Average inference time was measured in milliseconds (ms) per image using repeated inference runs under the same computational environment. The same input image resolution, preprocessing procedure, and inference protocol were applied to all three architectures to ensure a fair comparison.

The inference-time measurements were conducted using an Intel Xeon CPU and an NVIDIA Tesla T4 GPU with 15 GB of VRAM. The same hardware configuration was used for VGG16, VGG19, and EfficientNetB5. The measured average inference times were 32.6 ms/image for VGG16, 39.8 ms/image for VGG19, and 45.7 ms/image for EfficientNetB5. The three CNN architectures were subsequently compared based on accuracy, precision, recall, F1-score, total model parameters, model size, and inference time to identify the most suitable model for automatic banana ripeness classification.[16], [17].

3. RESULTS AND DISCUSSIONS

3.1. Experimental Dataset and Training Configuration

The dataset used in this research is the Banana Ripeness Classification Dataset, which was obtained from Kaggle. The dataset consists of RGB images of banana fruit categorized into three classes: unripe, ripe, and overripe. A total of 1,080 images were used in the experiment. The dataset was divided into training, validation, and testing subsets to ensure fair evaluation and to avoid data leakage. The distribution of the dataset is shown in Table 2.

Table 2. Distribution of Dataset

Dataset Split	Number of Images	Percentage
Train	756	70%
Validation	162	15%
Test	162	15%
Total	**1080**	**100%**

This partition ensures that the model is trained on sufficient data while maintaining an unbiased evaluation using unseen test data. To ensure reproducibility, all experiments were conducted under identical experimental settings. The training configuration used in this research is summarized in Table 3.

Table 3. The training configuration

No	Parameter	Description
1	Image size	224×224 (VGG16, VGG19), 456×456 (EfficientNetB5)
2	Number of classes	3 (unripe, ripe, overripe)
3	Batch size	16
4	Epochs	30
5	Learning rate	0.0001 and 0.00001
6	Optimizer	Adam
7	Loss function	Categorical cross-entropy
8	Callback	ReduceLROnPlateau

In addition, all models were trained under the same software environment to ensure fair comparison. The experiments were implemented using Python with TensorFlow.

The experiments were conducted using Google Colab with GPU acceleration. The computational environment used an NVIDIA Tesla T4 GPU with 15 GB of VRAM, while the local computer used to access the environment was equipped with an Intel Core i5 13th-generation processor and 16 GB of RAM running Windows. The CNN training and inference processes were executed in the Google Colab cloud environment using the NVIDIA Tesla T4 GPU, see Table 4 below.

Table 4. The hardware configuration

Component	Specification
Execution environment	Google Colab
GPU	NVIDIA Tesla T4
GPU VRAM	15 GB
Local CPU	Intel Core i5 13th Generation
Local RAM	16 GB
Local Operating System	Windows

All experiments were executed under identical hardware and software environments to guarantee the validity of the comparative analysis among VGG16, VGG19, and EfficientNetB5. The image size was adjusted according to model architecture requirements. VGG16 and VGG19 used input dimensions of 224×224 pixels, while EfficientNetB5 used 456×456 pixels. The batch size was set to 16 to balance computational efficiency and learning stability. The models were trained for 30 epochs to ensure convergence. The learning rate was adjusted based on model complexity, where VGG-based models used a smaller learning rate (1e-5) to maintain stable convergence, while EfficientNetB5 used a higher learning rate (1e-4) to accelerate learning due to its deeper architecture. The Adam optimizer was selected because of its adaptive learning capability, and categorical cross-entropy was used as the loss function for multi-class classification problems.

3.2 Training Performance

The training performance of the three CNN architectures VGG16, VGG19, and EfficientNetB5 were analyzed based on the learning behavior of training and validation accuracy as well as training and validation loss over 30 epochs. The learning curves of each model are presented in Figure 1, Figure 2 and Figure 2 respectively.

3.2.1. VGG16 Training Performance

As shown in Figure 3.1, VGG16 demonstrates a rapid and stable convergence behavior during training. Both training and validation accuracy increase sharply in the early epochs and reach near-optimal performance after approximately 10 epochs. The corresponding loss curves decrease consistently and stabilize close to zero.

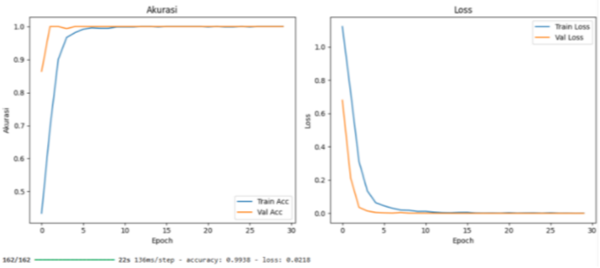


Figure 1. Training and Validation Accuracy and Loss Curves of VGG16 Model for Banana Ripeness Classification

As shown in Figure 3.1, VGG16 demonstrated rapid convergence during training. The training accuracy reached 99.38%, with a corresponding training loss of 0.0218 at the end of the training process. The training accuracy increased rapidly during the initial epochs and subsequently stabilized close to 1.00. Similarly, the validation accuracy rapidly increased and remained close to 1.00 during the later epochs, while the validation loss decreased substantially and stabilized at a low level. The small difference between the training and validation accuracy curves indicates that the model achieved consistent performance on the validation data. These results indicate that VGG16 converged effectively under the applied training configuration.

The small gap between training and validation accuracy indicates that VGG16 achieves strong generalization ability with minimal overfitting. This suggests that the feature representations learned by VGG16 are highly effective for distinguishing banana ripeness categories under the given dataset characteristics.

3.2.2. VGG19 Training Performance

The training behavior of VGG19, illustrated in Figure 2, shows a pattern similar to VGG16. The model exhibits a slightly slower convergence in the early training phase due to its deeper architecture. However, as training progresses, both training and validation accuracy steadily increase and eventually reach a level comparable to VGG16.

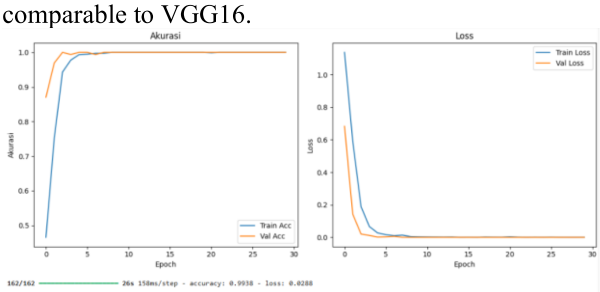


Figure 2. Training and Validation Accuracy and Loss Curves of VGG19 Model for Banana Ripeness Classification

The loss curves also show a consistent downward trend, indicating stable optimization. Despite having a higher number of layers, VGG19 does not significantly outperform VGG16, suggesting that additional depth does not necessarily translate into improved performance for this dataset.

As illustrated in Figure 2, VGG19 demonstrated rapid convergence during the training process. At the end of the 30 training epochs, the model achieved a training accuracy of 99.38% with a training loss of 0.0288. The validation accuracy increased rapidly during the initial epochs and reached approximately 99,5 %, while the validation loss decreased substantially and approached zero. The training and validation accuracy curves remained close to each other during the later epochs, indicating consistent performance on the validation dataset. The results demonstrate that VGG19 achieved stable convergence under the applied training configuration. Although VGG19 has a deeper architecture than VGG16, its training behavior was comparable to that of VGG16, indicating that the additional convolutional depth did not produce a substantial difference in the observed training performanc

3.2.3. EfficientNetB5 Training Performance

In contrast, Figure 3.3 shows that EfficientNetB5 follows a more gradual learning curve compared to VGG-based models. The increase in both training and validation accuracy is slower, and convergence is achieved at later epochs. Although the model continues to improve throughout training, the validation accuracy remains lower than that of VGG16 and VGG19. The loss curves also exhibit higher fluctuations, indicating less stable convergence behavior.

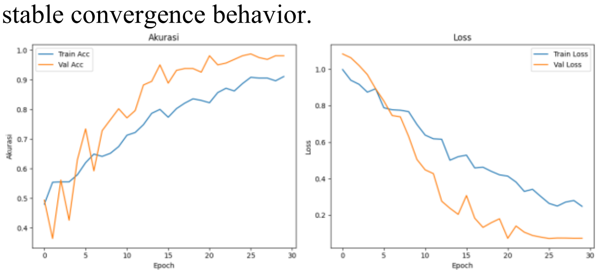


Figure 3. Training and Validation Accuracy and Loss Curves of EfficientNetB5 Model for Banana Ripeness Classification

Figure 3 presents the training and validation performance of EfficientNetB5 over 30 epochs. The model showed a gradual improvement in training accuracy, increasing from approximately 46% in the initial epoch to approximately 90% at the end of training. Validation accuracy exhibited greater fluctuations during the early training epochs but subsequently improved and reached approximately 98% toward the final epochs. The training loss decreased progressively from approximately 1.0 to around 0.25, while the validation loss decreased more substantially and approached approximately 0.05. The relatively large difference between the final training and validation accuracy indicates that the validation performance was higher than the training performance at the end of training. This behavior may be associated with the use of data augmentation during training, which introduces greater variability into the training samples. Overall, EfficientNetB5 exhibited slower and more fluctuating convergence than VGG16 and VGG19 under the applied training configuration.

3.2.4. Comparative Analysis of Training Behavior

Based on Figures 1 – Figure 3, several key observations can be made:

1. Convergence Speed
   VGG16 converges the fastest, followed by VGG19, while EfficientNetB5 requires more epochs to stabilize.
2. Stability
   VGG16 and VGG19 show smooth and stable convergence patterns, whereas EfficientNetB5 exhibits higher variability in validation performance.
3. Generalization Ability
   VGG-based models demonstrate a smaller gap between training and validation curves, indicating better generalization compared to EfficientNetB5.

3.2.5. Summary of Training Performance

In summary, the training curves presented in Figures 1– Figure 3 indicate that all models successfully learn discriminative features for banana ripeness classification. However, VGG16 and VGG19 outperform EfficientNetB5 in terms of convergence speed, stability, and generalization behavior. The similarity between VGG16 and VGG19 further suggests that increasing network depth does not necessarily improve performance for this dataset.

3.3. Classification Performance

This section presents the classification performance of VGG16, VGG19, and EfficientNetB5 based on the test dataset. The evaluation results are presented using confusion matrix analysis and visualization of classification outcomes.

3.3.1. VGG16 Classification Performance

The classification performance was evaluated using an independent testing dataset consisting of 162 images, with 54 images for each banana ripeness category. As presented in Figure 4, the model achieved an overall accuracy of 99.38%, corresponding to 161 correctly classified images out of 162 test images. Only one image was incorrectly classified. The class-level results show that the matang class achieved a precision of 1.00, recall of 0.98, and F1-score of 0.99. The mentah class obtained a precision of 0.98, recall of 1.00, and F1-score of 0.99, while the overripe class achieved a precision, recall, and F1-score of 1.00. The macro-average and weighted-average precision, recall, and F1-score were approximately 0.99. These results quantitatively demonstrate the high classification performance of the model on the independent test dataset.

The confusion matrix in Figure 5 provides a detailed representation of the classification results for the three banana ripeness categories. Of the 54 matang images, 53 were correctly classified as matang, while 1 image was incorrectly classified as mentah. All 54 mentah images were correctly classified, and all 54 overripe images were also correctly classified. Therefore, the model produced 161 correct predictions and only 1 incorrect prediction among the 162 test images. The resulting accuracy was 99.38%. The only observed classification error occurred between the matang and mentah categories, while no misclassification occurred for the overripe category. The confusion matrix therefore confirms the high classification performance indicated by the precision, recall, and F1-score results.

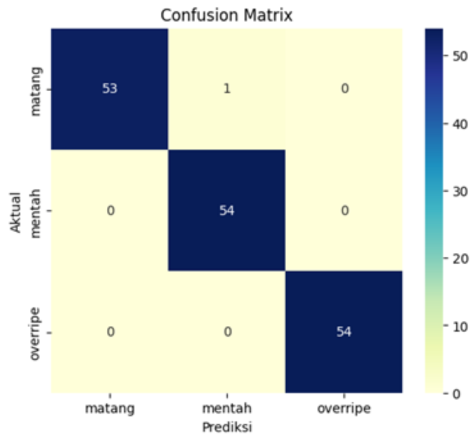


Figure 4. Confusion Matrix of VGG16 Model for Banana Ripeness Classification

```
              precision    recall  f1-score   support

      matang       1.00      0.98      0.99        54
      mentah       0.98      1.00      0.99        54
    overripe       1.00      1.00      1.00        54

    accuracy                           0.99       162
   macro avg       0.99      0.99      0.99       162
weighted avg       0.99      0.99      0.99       162

1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 443ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 221ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 233ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 223ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 204ms/step
```

Figure 5. Classification Results of VGG16 Model on Test Dataset

3.3.2. VGG19 Classification Performance

Similar to VGG16, the VGG19 model also demonstrates excellent classification performance. The confusion matrix shows that the model correctly classifies nearly all test samples across the three classes. The classification results and confusion matrix for VGG19 are shown in Figure 6 and Figure 7. The Figures 6 and 7 present the confusion matrices of VGG16 and VGG19, respectively. Both models produced an identical classification pattern on the 162-image testing dataset. For the matang class, 53 of 54 images were correctly classified, while 1 image was incorrectly classified as mentah. All 54 mentah images and all 54 overripe images were correctly classified. Consequently, both VGG16 and VGG19 correctly classified 161 of 162 test images and produced only 1 misclassification, resulting in an accuracy of 99.38%.

In contrast, EfficientNetB5 correctly classified 148 of 162 test images and misclassified 14 images, resulting in an accuracy of 91.36%. Therefore, VGG16 and

VGG19 produced 13 fewer classification errors than EfficientNetB5 on the same testing dataset. The confusion matrix results confirm that VGG16 and VGG19 provided substantially more accurate classification than EfficientNetB5 under the evaluated experimental conditions.

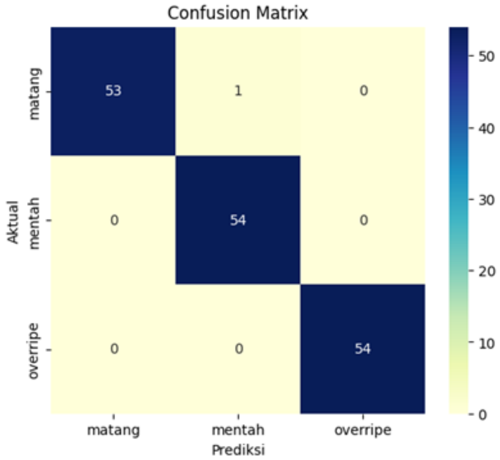


Figure 6. Confusion Matrix of VGG19 Model for Banana Ripeness Classification

	precision	recall	f1-score	support
matang	1.00	0.98	0.99	54
mentah	0.98	1.00	0.99	54
overripe	1.00	1.00	1.00	54
accuracy			0.99	162
macro avg	0.99	0.99	0.99	162
weighted avg	0.99	0.99	0.99	162

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 703ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 245ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 247ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 238ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 241ms/step
```

Figure 7. Classification Results of VGG19 Model on Test Dataset

3.3.3. EfficientNetB5 Classification Performance

The EfficientNetB5 model shows lower classification performance compared to VGG-based models. The confusion matrix reveals a higher number of misclassified samples across all classes, particularly between unripe and ripe bananas. The classification results and confusion matrix for EfficientNetB5 are presented in Figure 3.8 and Figure 3.9.

Figure 3.8 presents the confusion matrix and classification report of EfficientNetB5 on the independent testing dataset consisting of 162 images. The model correctly classified 148 images and misclassified 14 images, resulting in an overall accuracy of 91.36%.

At the class level, the matang category achieved a precision of 0.98, recall of 0.91, and F1-score of 0.94, based on 54 test images. The mentah category achieved a precision of 0.92, recall of 0.85, and F1-score of 0.88, also based on 54 test images. The overripe category achieved a precision of 0.85, recall of 0.98, and F1-score of 0.91, with 54 test images. The macro-average precision, recall, and F1-score were 0.92, 0.91, and 0.91, respectively, while the corresponding weighted averages were also 0.92, 0.91, and 0.91.

The confusion matrix shows that 49 of 54 matang images were correctly classified, while 3 images were classified as mentah and 2 as overripe. For the mentah category, 46 of 54 images were correctly classified, while 1 image was classified as matang and 7 as overripe. For the overripe category, 53 of 54 images were correctly classified, while 1 image was classified as mentah. Therefore, the largest classification error occurred between the mentah and overripe categories, with 7 mentah images incorrectly classified as overripe.

These results indicate that EfficientNetB5 achieved lower classification performance than VGG16 and VGG19 on the same testing dataset. While EfficientNetB5 correctly classified 148 of 162 images, VGG16 and VGG19 each correctly classified 161 of 162 images. Thus, EfficientNetB5 produced 13 more classification errors than each VGG-based model.

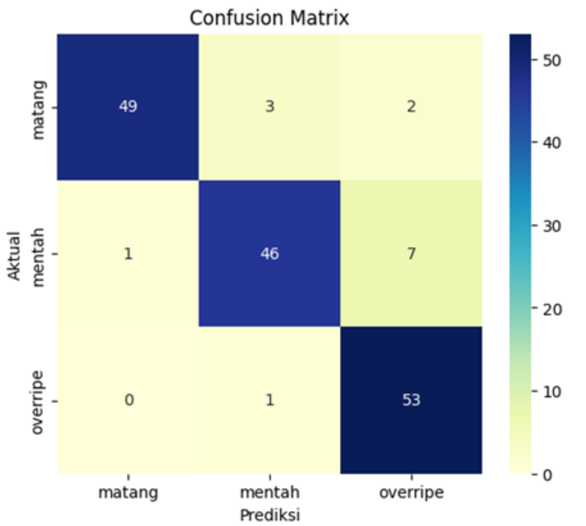


Figure 8. Confusion Matrix of EfficientNetB5 Model for Banana Ripeness Classification

	precision	recall	f1-score	support
matang	0.98	0.91	0.94	54
mentah	0.92	0.85	0.88	54
overripe	0.85	0.98	0.91	54
accuracy			0.91	162
macro avg	0.92	0.91	0.91	162
weighted avg	0.92	0.91	0.91	162

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 15s 15s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 79ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 77ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 79ms/step
```

Figure 9. Classification Results of EfficientNetB5 Model on Test Dataset

The higher misclassification rate indicates that EfficientNetB5 has more difficulty in distinguishing subtle differences between classes under the same experimental conditions. Although EfficientNetB5 is designed for efficiency and scalability, its performance in this research is lower compared to VGG16 and VGG19. This may be attributed to dataset size limitations and the sensitivity of EfficientNet architecture to hyperparameter tuning and training configuration.

3.3.4. Summary of Classification Performance

Based on the overall evaluation results, the following conclusions can be drawn:

1. VGG16 and VGG19 achieve nearly perfect and consistent classification performance.
2. Both VGG-based models show similar confusion matrix patterns and error distributions.
3. EfficientNetB5 shows lower performance with higher misclassification rates, especially in similar-class discrimination.
4. The results confirm that VGG-based architectures are more robust for banana ripeness classification under the given dataset conditions.

3.4. Model Prediction Results

The prediction performance of the trained CNN models—VGG16, VGG19, and EfficientNetB5—was evaluated using unseen test images. The results of model predictions are presented in Figure 10 (a), (b), and (c).

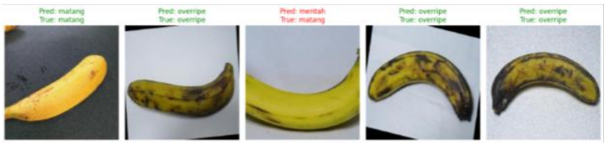
Based on Figure 10(a), the VGG16 model is able to correctly classify banana images according to their ground truth labels. All displayed test samples are predicted accurately, particularly for the ripe and overripe (overripe) classes. These results indicate that VGG16 has successfully learned discriminative visual features such as color, texture, and shape variations that characterize banana ripeness levels. The model demonstrates strong generalization capability on unseen data, consistent with its high classification accuracy reported earlier.

As shown in Figure 10(b), the VGG19 model also produces highly accurate prediction results. All test images are correctly classified into their respective categories, including unripe, ripe, and overripe (overripe). This performance indicates that VGG19 is capable of effectively learning visual patterns across different ripeness levels. However, despite having a deeper architecture than VGG16, the prediction outcomes do not show a significant improvement, suggesting that additional network depth does not necessarily enhance classification performance for this dataset.

In contrast, Figure 10(c) shows that EfficientNetB5 correctly classifies most of the test images, but one misclassification is observed. Specifically, an image with a true label of ripe is incorrectly predicted as unripe. This misclassification suggests that EfficientNetB5 faces challenges in distinguishing between visually similar classes. Although the model is designed for efficient scaling and improved performance, its behavior in this research indicates that it may require further tuning or a larger dataset to achieve more stable classification results in fine-grained visual tasks. Nevertheless, most of the test images are still correctly classified, indicating that the model retains a reasonable level of discriminative capability.

(a)

(b)



(c)

Figure 10. Results of Model Predictions: (a) VGG16, (b) VGG19, and (c) EfficientNetB5

3.5. Comparative Performance of CNN Models

The three CNN architectures were compared not only in terms of classification performance but also in terms of their computational characteristics. The computational benchmarking considered the total number of model parameters, trainable and non-trainable parameters, model storage size, and average inference latency per image. The inference measurements were conducted using Google Colab with an NVIDIA Tesla T4 GPU with 15 GB of VRAM under the same testing environment for all three models.

Table 5 summarizes the computational characteristics and classification performance of VGG16, VGG19, and EfficientNetB5. VGG16 contained 15.24 million total parameters, of which 2.36 million were trainable and 12.88 million were non-trainable. Its model storage size was 58.2 MB, and the measured average inference latency was 32.6 ms per image. VGG19 contained 20.29 million total parameters, including 2.36 million trainable parameters and 17.93 million non-trainable parameters. Its storage size was 77.4 MB, with an average inference latency of 39.8 ms per image.

Table 5. Comparative Classification and Computational Performance of the CNN Models

Model	Total Parameters (M)	Trainable Parameters (M)	Non-trainable Parameters (M)	Model Size (MB)	Inference Time (ms/image)	Accuracy
VGG16	15.24	2.36	12.88	58.2	**32.6**	**99.38%**
VGG19	20.29	2.36	17.93	77.4	**39.8**	**99.38%**
EfficientNetB5	30.50	12.76	17.74	116.8	**45.7**	**91.36%**

EfficientNetB5 contained the largest total number of parameters among the three models, with 30.50 million parameters, consisting of 12.76 million trainable parameters and 17.74 million non-trainable parameters. Its model storage size was 116.8 MB, while the average inference latency was 45.7 ms per image. Thus, under the evaluated hardware configuration, EfficientNetB5 required the largest storage capacity and exhibited the highest inference latency among the three architectures.

In terms of classification performance, VGG16 and VGG19 both achieved an accuracy of 99.38%, whereas EfficientNetB5 achieved 91.36%. Therefore, VGG16 provided the highest classification accuracy while also requiring the smallest model storage size and the lowest measured inference latency among the three models. VGG19 achieved the same classification accuracy as VGG16 but required more parameters, larger storage capacity, and longer inference time. EfficientNetB5, despite its compound-scaled architecture, showed lower classification accuracy and higher inference latency under the experimental conditions used in this study.

3.6. Discussion

The experimental results demonstrate that VGG16 and VGG19 consistently outperformed EfficientNetB5 in banana ripeness classification, achieving an accuracy of 99.38% compared to 91.36% for EfficientNetB5. Several technical factors may explain these performance differences.

First, the relatively small dataset size (1,080 images) likely favored the VGG-based architectures. Although transfer learning enables effective feature reuse from ImageNet, deeper and more sophisticated models such as EfficientNetB5 generally require larger and more diverse datasets to fully exploit their representational capacity. With a limited dataset, the simpler feature extraction strategy of VGG16 and VGG19 appears sufficient to capture the dominant visual characteristics of banana ripeness, such as color transition, texture variation, and surface appearance, while reducing the risk of unstable optimization.

Second, the transfer learning behavior of the models also contributes to the observed results. VGG16 and VGG19 have straightforward sequential convolutional structures that adapt effectively during fine-tuning, especially when pretrained ImageNet weights are employed. Their learned low-level and mid-level visual features, including edges, color gradients, and texture patterns, closely correspond to the discriminative characteristics required for banana ripeness classification. In contrast, EfficientNetB5 employs compound scaling that jointly increases network depth, width, and input resolution. While this design generally improves performance on large-scale datasets, it also increases sensitivity to dataset size and fine-tuning strategies.

Another important factor is model complexity. EfficientNetB5 contains substantially more sophisticated architectural components than VGG16 and VGG19, making it more dependent on carefully optimized hyperparameters. Although all three models were trained under identical experimental settings to ensure a fair comparison, the same training configuration may not represent the optimal setting for EfficientNetB5. Consequently, the model exhibited slower convergence and greater fluctuations during training, indicating that additional hyperparameter tuning, such as learning rate scheduling, batch size adjustment, or longer training, may be required to fully realize its potential.

Furthermore, optimization characteristics may also explain the observed performance gap. VGG16 and VGG19 demonstrated faster and more stable convergence, suggesting that their optimization landscape is more suitable for the current dataset. EfficientNetB5, despite using a higher learning rate to accelerate training, showed less stable validation performance, indicating that its optimization process may require more careful parameter adjustment. This behavior is consistent with previous studies reporting that EfficientNet architectures generally benefit from larger datasets and more extensive hyperparameter optimization than conventional VGG-based networks.

Overall, these findings suggest that higher architectural complexity does not necessarily lead to superior classification performance. For relatively small image datasets with limited visual variability, simpler transfer learning architectures such as VGG16 and VGG19 may provide better generalization, more stable optimization, and higher classification accuracy. Therefore, VGG16 and VGG19 represent practical and reliable solutions for automated banana ripeness classification in smart agriculture applications, particularly for post-harvest quality assessment, fruit grading, and decision support systems.

4. CONCLUSION

This study compared the performance of VGG16, VGG19, and EfficientNetB5 for banana ripeness classification using a transfer learning approach. The results show that VGG16 and VGG19 achieved the highest classification performance, both reaching an accuracy of 99.38%, while EfficientNetB5 achieved 91.36%. These findings indicate that VGG-based architectures are more suitable for the dataset used in this study. Since VGG16 achieved the same performance as VGG19 with a simpler architecture, it can be considered the more practical model for banana ripeness classification.

This study contributes to the application of deep learning in agricultural image classification by providing a comparative evaluation of three widely used CNN architectures under identical experimental settings. The findings may serve as a reference for selecting appropriate deep learning models for automated fruit ripeness classification. Several limitations should be acknowledged. The experiments were conducted using a single public dataset with only three ripeness categories, and only three CNN architectures were evaluated without extensive hyperparameter optimization. These methodological limitations may have influenced the model performance, particularly for EfficientNetB5, which is sensitive to dataset characteristics and training configuration.

Future research should evaluate the proposed approach using larger and more diverse datasets, investigate additional state-of-the-art architectures, and apply more comprehensive hyperparameter optimization to improve model robustness and generalization for real-world agricultural applications.

6. REFERENCES

[1] Weny Indah Kusumawati and Adisaputra Zidha Noorizki, "Perbandingan Performa Algoritma VGG16 Dan VGG19 Melalui Metode CNN Untuk Klasifikasi Varietas Beras," *J. Comput. Electron. Telecommun.*, vol. 4, no. 2, 2023, doi: 10.52435/complete.v4i2.387.

[2] R. T. Wahyuningrum, R. Erfian, A. Kusumaningsih, and H. P. A. Tjahyaningtijas, "Klasifikasi Penyakit Daun Jagung Menggunakan Model Deep Learning EfficientNetB5," *J. Pekommas*, vol. 10, no. 1, pp. 1–8, 2025, doi: 10.56873/jpkm.v9i1.5322.

[3] S. Y. Sutanto and D. Udjulawa, "Klasifikasi Penyakit Mata pada Citra Fundus Menggunakan VGG-16," *JATISI*, vol. 11, no. 4, pp. 2407–4322, Dec. 2024, doi: 10.35957/JATISI.V11I4.9165.

[4] H. R. J. Sinaga, "Penerapan Model Arsitektur Efficientnetb5 Pada Klasifikasi Multi-Kelas Penyakit Tanaman Padi," Mar. 2025, Accessed: Jul. 31, 2026. [Online]. Available: https://repositori.uma.ac.id/handle/123456789/27911

[5] N. Rumui, A. Mualo, J. Rahayaan, L. Batjo, and M. Mokansi, "Analisis Komparasi Model Deep Learning CNN dengan VGG16 dalam Klasifikasi Jenis Bunga," *Inform. J. Ilmu Komput.*, vol. 21, no. 1, pp. 35–44, Apr. 2025, doi: 10.52958/IFTK.V21I1.11105.

[6] I. G. Perwati, N. Suarna, and T. Suprapti, "Analisis Klasifikasi Gambar Bunga Lily Menggunakan Metode Convolutional Neural Network (Cnn) Dalam Pengolahan Citra," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 3, pp. 2908–2915, May 2024, doi: 10.36040/JATI.V8I3.9193.

[7] M. Nawaz, A. Javed, and A. K. J. Saudagar, "Residual-SwishNet: a deep learning-based approach for reliable lung cancer classification," *Front. Oncol.*, vol. 15, p. 1729021, Jan. 2026, doi: 10.3389/FONC.2025.1729021/TEXT.

[8] N. Nafiiyah and T. Dwi Lestari, "Perbandingan Vgg16, Vgg19, Mobilenet Dan Mobilenetv2 Untuk Klasifikasi Beras Berdasarkan Citra," *Rabit J. Teknol. dan Sist. Inf. Univrab*, vol. 11, no. 1, pp. 1368–1380, Jan. 2026, doi: 10.36341/RABIT.V11I1.7338.

[9] T. K. Mazdavilaya, F. Yanto, E. Budianita, S. Sanjaya, and F. Syafria, "Implementasi VGG 16 dan Augmentasi Zoom Untuk Klasifikasi Kematangan Sawit," *KLIK Kaji. Ilm. Inform. dan Komput.*, vol. 4, no. 6, pp. 2881–2891, Jun. 2024, doi: 10.30865/KLIK.V4I6.1940.

[10] T. W. Isna Wulandari, Hasbi Yasin, "Klasifikasi Citra Digital Bumbu dan Rempah Dengan Algoritma Convolutional Neural Network (CNN)," *J. GAUSSIAN*, vol. 9, no. 3, pp. 273–282, 2020, doi: 10.14710/j.gauss.9.3.273-282.

[11] D. Adri, "Sistem Pendukung Keputusan Penentuan Kelulusan Mahasiswa Menggunakan Metode Fuzzy Mamdani | GENETIKA: Jurnal Generasi Teknologi Informasi dan Informatika," *Genet. J. Gener. Teknol. Inf. dan Inform.*, vol. 1, no. 1, pp. 1–8, 2026, Accessed: Jul. 31, 2026. [Online]. Available: https://baituljurnal.umbuton.ac.id/index.php/genetika/article/view/3

[12] R. A. Ines Heidiani Ikasari, Perani Rosyani, "Klasifikasi Jenis Buah Menggunakan Metode CNN," *RIGGS J. Artif. Intell. Digit. Bus.*, vol. 4, no. 2, pp. 5451–5458, Jul. 2025, doi: 10.31004/riggs.v4i2.1271.

[13] J. Halim and A. N. Fajar, "Klasifikasi Pisang Berbasis Algoritma VGG16 Melalui Metode CNN Deep Learning," *Inf. (Jurnal Inform. dan Sist. Informasi)*, vol. 15, no. 1, pp. 1–17, May 2023, doi: 10.37424/INFORMASI.V15I1.190.

[14] A. Garg, S. Salehi, M. La Rocca, R. Garner, and D. Duncan, "Efficient and visualizable convolutional neural networks for COVID-19 classification using Chest CT," *Expert Syst. Appl.*, vol. 195, p. 116540, Jun. 2022, doi: 10.1016/J.ESWA.2022.116540.

[15] G. D. Cahyo, R. Helilintar, and I. N. Farida, "Pengolahan Citra Kematangan Pisang Menggunakan Convolutional Neural Network VGG19," *Pros. SEMNAS INOTEK (Seminar Nas. Inov. Teknol.*, vol. 9, no. 2, pp. 1507–1515, Jul. 2025, doi: 10.29407/D079Q671.

[16] E. Bima Evansyah, C. Sri Kusuma, U. Muhammadiyah Malang Jl Raya Tlogomas No, K. Lowokwaru, K. Malang, and J. Timur, "Comparison of VGG16 and VGG19 Models in the Classification of Down Syndrome in the European Region with Transfer Learning," *J. Innov. Technol. Polbeng Ser. Informatics (INOVTEK Polbeng - Seri Inform.*, vol. 10, no. 2, pp. 922–933, Jun. 2025, doi: 10.35314/PZ35E881.

[17] F. A. H. M. K. B. S. A. W. P. S. Alala, "Jurnal

tika," *J. TIKA Fak. Ilmu …*, vol. xx, no. 1, pp. 85–91, 2024, [Online]. Available: http://download.garuda.kemdikbud.go.id/article.php?article=3285185&val=21707&title=Game Edukasi Pengenalan Nama-Nama Sayuran Menggunakan Metode Development Life Cycle