INTERNATIONAL JOURNAL OF APPLIED ARTIFICIAL INTELLIGENCE AND ROBOTICS
E-ISSN: 3124-1212, DOI: 10.67745/ijaic.v2i2.30
Vol. 2 No. 2 2026 pp. 65-73

Comparative Performance Analysis of CNN Architectures for Coffee Bean Image Classification Based on Accuracy and Training Time

Nazwa Firyal Cherry[1], Nurwahida Zakia[2], Hani Uliya Pasya[3*]

[1,2] Informatics Engineering, Departement of Information Technology and Computer, Politeknik Negeri Lhokseumawe
[3] Faculty of Medicine, UIN Syarif Hidayatullah Jakarta, Indonesia

*Corresponding Author: haniuliyapasya@mhs.uin-syarif.com

Article info: Received 01/06/2026, Revised 07/06/2026, Accepted 04/11/2026

This is an open access article under the CC BY-SA license.

Abstract

Coffee bean image classification is an application of computer vision that can facilitate the automatic identification and quality-based categorization of coffee beans. This study aims to compare the performance of several Convolutional Neural Network (CNN) models, namely MobileNetV2, ResNet50, EfficientNetB0, VGG16, and DenseNet121, for coffee bean image classification. The dataset consists of coffee bean images that underwent preprocessing and data splitting for model training and testing. Model performance was evaluated based on classification accuracy and training time. The results show that MobileNetV2 achieved the best performance, with an accuracy of 98.75% and the shortest training time of 97.69 seconds. DenseNet121 also achieved high accuracy of 98.25%, with a training time of 230.96 seconds. Meanwhile, VGG16 attained an accuracy of 89.00% but required the longest training time of 491.79 seconds. ResNet50 and EfficientNetB0 demonstrated less optimal performance, achieving accuracies of 63.75% and 25.00%, respectively. Based on these findings, MobileNetV2 is recommended as the most effective model for coffee bean image classification, as it provides an excellent balance between high classification accuracy and training efficiency.

Keywords: coffee bean classification, deep learning, transfer learning, model performance evaluation, training time efficiency

1. Introduction

Coffee is one of the major plantation commodities with high economic value and is considered one of Indonesia's leading agricultural products. Quantitative data from Statistics Indonesia (Badan Pusat Statistik, BPS) indicate that Indonesia's coffee production and export volumes continue to make significant contributions to the national economy each year. However, farmers and industry stakeholders face considerable challenges in maintaining consistent product quality, particularly during the coffee bean quality classification process. To date, most industry practitioners still rely on conventional visual inspection by experts to identify the physical characteristics of coffee beans. This manual approach requires considerable time and operational costs and is susceptible to errors caused by human subjectivity and fatigue.

To address the limitations of conventional methods, previous studies have applied computer vision and deep learning technologies to automate the classification process. For example, Raharjo (2023) [1] employed a basic Convolutional Neural Network (CNN) model to classify coffee bean maturity levels and achieved an accuracy of 88%.Furthermore, Pratama et al. (2024) [2] implemented the VGG16 architecture to detect physical defects in coffee beans, achieving consistent performance. Meanwhile, Santoso (2025) [3] conducted experiments on coffee variety classification using the MobileNet architecture and demonstrated that lightweight models can operate efficiently on devices with limited hardware resources, although this efficiency may involve a slight trade-off in accuracy compared with deeper networks.

Although various CNN models have demonstrated their effectiveness in digital image processing, each architecture has distinct characteristics, network depth, and parameter complexity when processing specific object features. Therefore, this study conducts an in-depth comparative analysis to evaluate the performance of five widely used CNN architectures—MobileNetV2, ResNet50, EfficientNetB0, VGG16, and DenseNet121—for coffee bean image classification. These architectures were selected based on their different structural advantages. MobileNetV2 offers high computational efficiency for lightweight applications; ResNet50 addresses the vanishing gradient problem through residual learning; VGG16 provides a straightforward and reliable architecture; DenseNet121 enhances information flow between network layers; and EfficientNetB0 optimally balances network depth and input resolution.

This study evaluates model performance using two primary metrics: classification accuracy and training time. Through this comparative approach, the performance of all five architectures is assessed using the same coffee bean image dataset. The experimental results are expected to identify the most effective and efficient deep learning model and provide a valuable scientific reference for developing automated classification systems in Indonesia's agricultural and coffee plantation industries.

2. Methods

This section describes the systematic steps undertaken in this study to classify coffee bean images. Overall, the research adopts an experimental comparative approach using deep learning methods. The research workflow is structured into several stages, beginning with dataset collection and image preprocessing, followed by the implementation of five Convolutional Neural Network (CNN) architectures using transfer learning, and concluding with model performance evaluation. The primary objective of this methodology is to identify the architecture that provides the best balance between classification accuracy and computational efficiency. The research methodology is illustrated in Fig. 1, which presents the workflow from dataset collection and image preprocessing to transfer learning-based CNN model training, followed by testing and coffee bean image classification.

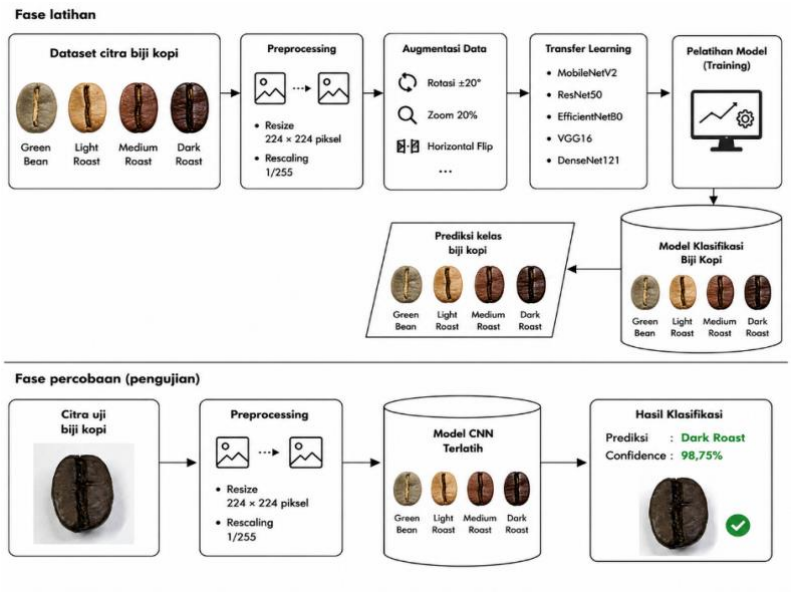


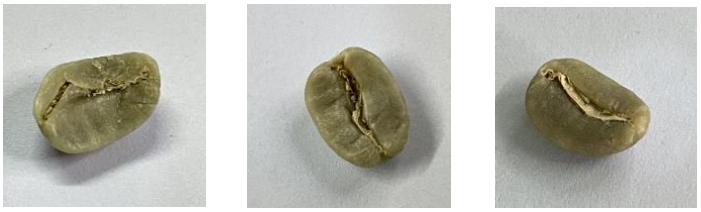
Figure 1. Research Methodology

2.1. Dataset

The dataset used in this study consists of coffee bean images categorized into four classes: dark, green, light, and medium. Each class exhibits distinct visual characteristics, including color, surface texture, shape, and observable defect patterns. These variations serve as the basis for the Convolutional Neural Network (CNN) models to learn relevant features and distinguish between the coffee bean categories. Before being used for model training, all images undergo preprocessing to ensure data quality and consistency. This process includes resizing the images to 224 x 224 pixels to meet the input requirements of the CNN architectures and normalizing pixel values to optimize the model training process. In addition, data augmentation is applied to the training set to increase dataset variability and reduce the risk of overfitting during model training.

The dataset is then organized into a structured directory based on the corresponding class labels to facilitate data processing and model training. Subsequently, the dataset is divided into two subsets: a training set and a testing set. The training set is used to develop the models by enabling the CNN architectures to automatically learn relevant image features, whereas the testing set is used to evaluate their ability to classify previously unseen images.

Table 1. Representative Images of Each Coffee Bean Class in the Dataset

No	Images	Class
1		Dark
2		Green
3		Light
4		medium

2.2. Prepocessing

Preprocessing is the initial stage performed before the images are used for deep learning model training. This stage aims to prepare the data in a consistent format so that it can be processed optimally by the Convolutional Neural Network (CNN) models. The quality of the input data significantly influences the model's ability to learn object characteristics from images. Therefore, preprocessing is performed to reduce irrelevant data variations and improve the quality of the data used for classification. The dataset used in this study consists of 1,600 coffee bean images categorized into four classes: Dark, Green, Light, and Medium. Before being used for model training and testing, all images undergo several preprocessing steps, including image resizing, pixel value normalization, and data augmentation. These steps ensure that the images have consistent characteristics that meet the requirements of the CNN architectures employed in this study. Furthermore, preprocessing helps improve computational efficiency during model training. The preprocessing process is implemented using the ImageDataGenerator class from TensorFlow Keras. Through this process, images with varying dimensions and characteristics are transformed into a format suitable for the MobileNetV2, ResNet50, EfficientNetB0, VGG16, and DenseNet121 models. Preprocessing not only helps the models learn coffee bean visual patterns more consistently but also supports their generalization to previously unseen data, potentially improving overall classification performance.

2.2.1 Resize

Resizing is the process of adjusting an image to a predefined dimension. In the coffee bean image dataset, each image may have a different resolution depending on the device used for image acquisition or the dataset collection process. These variations in image dimensions may cause inconsistencies in the input data and therefore must be addressed before the images are processed by the CNN models. In this study, all images are resized to 224x224 pixels using the target_size=(224,224) parameter during dataset loading. This dimension is selected because it corresponds to the standard input size commonly used by the MobileNetV2, ResNet50, EfficientNetB0, VGG16, and DenseNet121 architectures. By standardizing the image dimensions, all models receive input data with consistent spatial dimensions, ensuring consistency throughout the training and testing processes.

In addition to meeting the input requirements of the CNN architectures, resizing aims to reduce computational complexity. Larger images increase the number of operations required during convolution, resulting in longer training times. Conversely, the 224x224-pixel resolution is considered sufficient to preserve essential visual information in coffee bean images while maintaining computational efficiency. As illustrated in Fig. 2, the images exhibit differences in dimensions before and after resizing. Therefore, resizing is an important step in preparing the dataset before model training.



Before resize After Resize

Figure 2. Coffee Bean Images Before and After Resizing

2.2.2 Pixel Normalization

After standardizing the image dimensions, the next preprocessing step is pixel normalization. In RGB digital images, each pixel has an intensity value ranging from 0 to 255. This relatively wide range of values may affect optimization stability and slow model convergence during training. To address this issue, normalization is performed using the rescaling method with a scaling factor of $1/255$. This process transforms all pixel values from the range of 0–255 to the range of 0–1. Mathematically, the normalization process can be expressed as follows:

$$x' = \frac{x}{255} \tag{1}$$

where:
- x represents the normalized pixel intensity.
- x' represents the original pixel intensity, ranging from 0 to 255.
- 255 is the maximum intensity value for an 8-bit RGB image.

This normalization process ensures that pixel values fall within a consistent range, helping improve numerical stability during model training. Note: If the models use their architecture-specific preprocessing functions, such as those provided for MobileNetV2 and ResNet50 in TensorFlow Keras, the normalization procedure should be adjusted accordingly to match the input requirements of each pretrained model.

2.3. Augmentation

Data augmentation is a technique used to increase dataset diversity through specific transformations without changing the class labels of the images. This technique is widely used in deep learning research because it can improve model generalization and reduce the risk of overfitting. In this study, the dataset consists of 1,600 coffee bean images, divided into 1,200 training images (75%) and 400 testing images (25%). Data augmentation is applied exclusively to the training set, while the testing images remain unchanged to ensure that the evaluation results objectively reflect model performance. Data augmentation is implemented using the ImageDataGenerator class from TensorFlow Keras. The process is performed automatically during model training, generating new image variations in each epoch without requiring additional images to be created manually. Consequently, the models can learn from more diverse data rather than relying solely on patterns present in the original images. Augmentation also helps the models recognize coffee bean characteristics under varying image acquisition conditions, such as changes in object orientation and scale.

Three augmentation techniques are employed in this study: rotation, zoom, and horizontal flipping. Rotation is applied using the rotation_range=20 parameter, allowing images to be rotated by up to $\pm 20^{\circ}$ from their original orientation. Zoom is implemented using the zoom_range=0.2 parameter, enabling images to be enlarged or reduced by up to 20% relative to their original scale. In addition, horizontal flipping is performed using the horizontal_flip=True parameter, which mirrors images horizontally to introduce variations in object orientation. These techniques are selected because they introduce image diversity while preserving the essential visual characteristics of coffee beans in each class. Applying augmentation to the 1,200 training images is expected to increase the diversity of the data used to train MobileNetV2, ResNet50, EfficientNetB0, VGG16, and DenseNet121. By exposing the models to diverse image variations during training, they can learn coffee bean visual features more effectively and potentially achieve better generalization to unseen data. Furthermore, augmentation helps reduce the likelihood of overfitting and improve overall classification performance.

The complete dataset comprises 1,600 images, divided into 75% for training and 25% for testing. This proportion was selected to provide sufficient data for model training while maintaining a representative test set

for performance evaluation. Table 1 presents representative images from each coffee bean class used in this study. All experiments are conducted using five CNN architectures: MobileNetV2, ResNet50, EfficientNetB0, VGG16, and DenseNet121. This experimental setup enables a comparative analysis of the classification accuracy and training time of each model.

2.4. Convolutional Neural Network (CNN)

This study employs five Convolutional Neural Network (CNN) architectures: MobileNetV2, ResNet50, EfficientNetB0, VGG16, and DenseNet121. All five models are implemented using a transfer learning approach with pretrained weights obtained from the ImageNet dataset. Transfer learning leverages the feature extraction capabilities of previously trained models, enabling more efficient training and potentially better performance even when the available dataset is relatively limited. All models use input images with dimensions of 224 x 224 x 3 pixels, as specified by the input_shape=(224,224,3) parameter during model development. In this study, the base layers of each architecture are frozen using the base_model.trainable=False parameter, ensuring that the weights learned from ImageNet remain unchanged during training. Subsequently, additional layers are incorporated, including a GlobalAveragePooling2D layer, a Dense layer with 128 neurons and a ReLU activation function, a Dropout layer with a rate of 0.5, and an output layer with a Softmax activation function configured according to the number of classes in the dataset.

In general, CNNs operate by extracting features through convolutional and pooling layers, followed by classification through fully connected layers. The convolution operation used to extract image features can be expressed mathematically as shown in Equation (2):

$$S(i,j) = \sum_{m} \sum_{n} I(i+m, j+n) \times K(m,n). \quad (2)$$

Where:
- $S(i,j)$= convolution output.
- $I(i,j)$= input image pixel values.
- $K(m,n)$= convolution kernel or filter.

Through this process, CNNs can automatically extract important features, such as color, texture, shape, and visual patterns, from coffee bean images without requiring manual feature extraction.

2.4.1 MobileNetV2

MobileNetV2 is a CNN architecture designed to achieve high performance with relatively low computational requirements. It employs the concept of depthwise separable convolution, which divides the convolution operation into two stages, significantly reducing the number of parameters and computational operations compared with standard convolution. In addition, MobileNetV2 incorporates inverted residual blocks and linear bottlenecks to preserve important information during feature extraction. These characteristics make MobileNetV2 more lightweight than many other CNN architectures, making it suitable for deployment on devices with limited computational resources.

2.4.2 ResNet50

ResNet50 is a 50-layer CNN architecture based on the concept of residual learning. This architecture introduces skip connections, which allow information from earlier layers to be directly passed to subsequent layers. This mechanism helps mitigate the vanishing gradient problem, which commonly occurs in very deep neural networks. Compared with VGG16, ResNet50 employs a residual learning mechanism that enables the construction of deeper networks and facilitates the learning of complex image features. Consequently, ResNet50 has been widely used in various image classification studies.

2.4.3 EfficientNetB0

EfficientNetB0 is a CNN architecture developed using the compound scaling method, an approach that simultaneously balances increases in network depth, network width, and input image resolution. This strategy enables the model to achieve high performance without excessively increasing the number of parameters. Compared with several conventional CNN architectures, EfficientNetB0 provides a balance between classification accuracy and computational efficiency, making it widely used in modern image classification research

2.4.3 EfficientNetB0

VGG16 is a CNN architecture developed by the Visual Geometry Group (VGG) and consists of 16 trainable

layers. It uses 3x3 convolutional filters arranged sequentially to extract features from images. Its simple architectural structure makes VGG16 relatively easy to understand and implement. However, compared with MobileNetV2 and EfficientNetB0, VGG16 has a relatively large number of parameters, requiring greater memory capacity and longer training times.

2.4.5 DenseNet121

DenseNet121 is a CNN architecture that employs the concept of dense connectivity, in which each layer is directly connected to all preceding layers within a network block. This approach facilitates the flow of information and gradients during training, thereby reducing information loss between layers. DenseNet121 consists of 121 layers and can generate richer feature representations because each layer utilizes information from all preceding layers. In addition to improving feature extraction, this approach enables more parameter-efficient learning than some CNN architectures of comparable depth.

Based on their architectural characteristics, the five CNN models employed in this study exhibit different levels of complexity and learning mechanisms. MobileNetV2 and EfficientNetB0 prioritize computational efficiency through relatively lightweight architectures, whereas ResNet50 and DenseNet121 are designed to enhance feature extraction capabilities through deeper networks and advanced connectivity mechanisms. VGG16 offers a simple and straightforward architecture but requires greater computational resources. These differences in network depth, connectivity mechanisms, and feature extraction strategies may lead to variations in classification performance. Therefore, this study compares the five architectures to identify the most effective CNN model for coffee bean image classification based on classification accuracy and training time.

2.5. Model testing and evaluation

The evaluation was conducted to assess the performance of five Convolutional Neural Network (CNN) models, namely MobileNetV2, ResNet50, EfficientNetB0, VGG16, and DenseNet121, in classifying coffee bean images into four classes: Dark, Green, Light, and Medium. All models were evaluated using 400 test images that were not used during the training process. Separating the test data from the training data aimed to measure the generalization ability of each model in recognizing previously unseen images. In this study, model performance was evaluated based on accuracy and training time. Accuracy was used to measure the proportion of coffee bean images correctly classified by each model, whereas training time was used to assess the time efficiency of each architecture during the training process. These two metrics were selected because this study focuses not only on classification accuracy but also on the computational efficiency of the CNN architectures under comparison.

2.5.1 Accuracy

Accuracy is an evaluation metric used to measure the proportion of correctly classified predictions relative to the total number of test samples. A higher accuracy value indicates that the model has a better ability to recognize and distinguish the visual characteristics of each coffee bean class.

Accuracy is calculated using Equation (3).

$$Accuracy = \frac{Jumlah\ Prediksi\ Benar}{Jumlah\ Seluruh\ Data\ Uji} \tag{3}$$

Alternatively, accuracy can be expressed as:

$$Accuracy = \frac{\sum_{i=1}^{n} \quad Prediksi\ Benar_i}{N} \tag{4}$$

Where:
- Accuracy = classification accuracy.
- N = total number of test samples.
- Correct Predictions = number of images correctly classified according to their actual classes.

In this study, accuracy was calculated using the accuracy_score() function from the Scikit-learn library, which compares the actual labels with the labels predicted by the model.

2.5.2 Training Time

In addition to accuracy, this study evaluates the training time of each CNN model. Training time is measured to determine the computational efficiency of each architecture in completing the learning process.

Training time is calculated as the difference between the timestamps recorded before and after model training using Python's time() function. The training time is expressed mathematically in Equation (5).

$$Training\ Time = T_{end} - T_{start} \tag{5}$$

Where:
- Training Time = total model training time (seconds).
- T_{end} = timestamp recorded after the training process is completed.
- T_{start} = timestamp recorded before the training process begins.

Training time serves as an indicator of computational efficiency. A model with a shorter training time completes the learning process more quickly under the given experimental conditions, whereas a model with a longer training time generally requires more computational resources or processing time for feature extraction and parameter optimization. The evaluation results based on accuracy and training time are subsequently used to compare the performance of MobileNetV2, ResNet50, EfficientNetB0, VGG16, and DenseNet121 in coffee bean image classification. The comparison considers both classification accuracy and training efficiency to identify the most effective architecture for this study. The results also provide a basis for analyzing the relationship between classification performance and computational efficiency across the five CNN architectures.

This analysis is expected to clarify the strengths and limitations of each model and provide recommendations for selecting an optimal transfer learning architecture based on the balance between classification accuracy and training time. The evaluation results will also inform the selection of a suitable model for implementation in a coffee bean image classification system. Thus, model selection considers not only high classification accuracy but also training efficiency, supporting a more effective and computationally efficient development process.

3. Result and Discussions

This study evaluates five Convolutional Neural Network (CNN) architectures to identify the most effective model for coffee bean image classification. All models are trained and tested using the same dataset to ensure that differences in performance reflect the capabilities of each architecture in learning the visual characteristics of coffee beans. The training parameters include an image size of 224x224 pixels, a batch size of 16, and 5 training epochs. The model evaluation results are presented in Table 2.

Table 2. CNN Model Evaluation Results

Model	Accuracy (%)	Training Time (s)
MobileNetV2	98,75	97,69
ResNet50	63,75	284,08
EfficientNetB0	25,00	122,46
VGG16	89,00	491,79
DenseNet121	98,25	230,96

Based on the evaluation results presented in Table 2, MobileNetV2 achieved the best overall performance, with an accuracy of 98.75% and a training time of 97.69 seconds. These results indicate that MobileNetV2 can classify coffee bean images into the Dark, Green, Light, and Medium classes with a very high level of accuracy. In addition to achieving the highest accuracy, MobileNetV2 required the shortest training time among all evaluated models, demonstrating its computational efficiency. DenseNet121 achieved an accuracy of 98.25% with a training time of 230.96 seconds. Although its accuracy was close to that of MobileNetV2, with a difference of only 0.50 percentage points, its training time was more than twice as long. This finding indicates that DenseNet121 provides strong classification performance but requires substantially more training time under the experimental conditions.

Meanwhile, VGG16 achieved an accuracy of 89.00% with a training time of 491.79 seconds, the longest among all evaluated models. ResNet50 achieved an accuracy of 63.75% with a training time of 284.08 seconds, whereas EfficientNetB0 obtained an accuracy of 25.00% with a training time of 122.46 seconds. These results suggest that ResNet50 and EfficientNetB0 did not effectively learn the visual characteristics of coffee beans under the training configuration used in this study. Overall, MobileNetV2 was the most effective model for coffee bean image classification, achieving the highest accuracy and the shortest training time among the five CNN architectures. Its combination of high classification performance and computational efficiency makes it a promising candidate for implementation in a coffee bean image classification system.

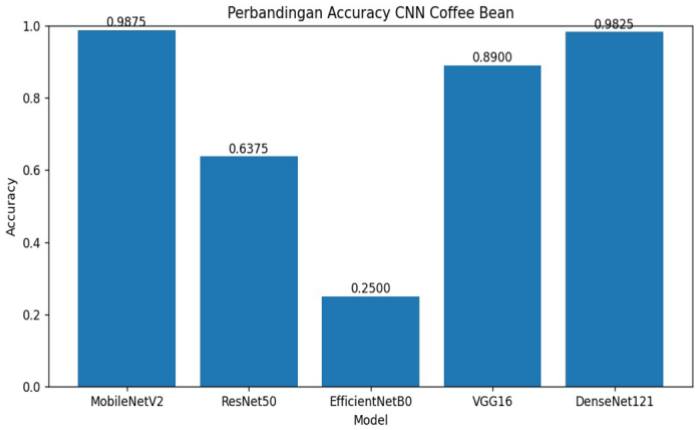


Figure 3. Comparison of CNN Model Accuracy

As shown in Fig. 3, MobileNetV2 achieved the highest accuracy of 98.75%, followed by DenseNet121 at 98.25% and VGG16 at 89.00%. Meanwhile, ResNet50 and EfficientNetB0 achieved accuracies of 63.75% and 25.00%, respectively. These results indicate that MobileNetV2 and DenseNet121 demonstrate excellent classification capabilities in recognizing the visual characteristics of coffee bean images. However, MobileNetV2 outperforms the other models by achieving the highest accuracy with a shorter training time.

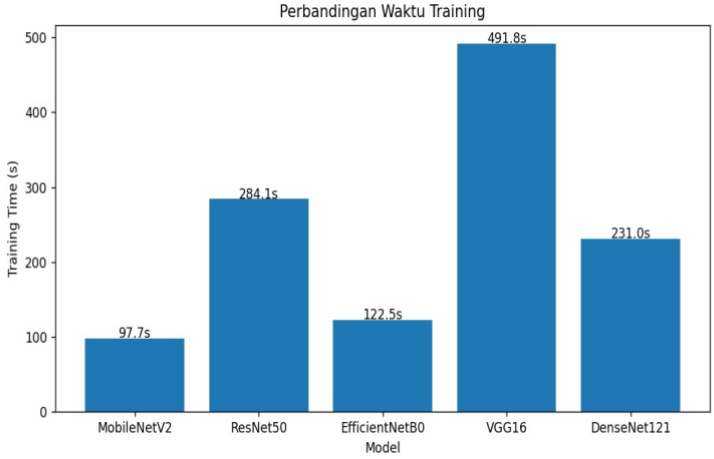


Figure 4. Comparison of CNN Model Training Times

As shown in Fig. 4, there are significant differences in training time among the CNN architectures. MobileNetV2 achieved the shortest training time at 97.69 seconds, followed by EfficientNetB0 at 122.46 seconds, DenseNet121 at 230.96 seconds, and ResNet50 at 284.08 seconds. Meanwhile, VGG16 required the longest training time, at 491.79 seconds. Overall, the results indicate that differences in CNN architectures affect coffee bean image classification performance. MobileNetV2 achieved the best combination of classification accuracy and computational efficiency, with an accuracy of 98.75% and a training time of 97.69 seconds, making it the most effective model in this study. DenseNet121 also achieved high accuracy at 98.25%, but required a longer training time of 230.96 seconds. These findings demonstrate that selecting an appropriate CNN architecture can improve classification performance while maintaining training efficiency.

4. Conclusion

Based on the results of this study, it can be concluded that the Convolutional Neural Network (CNN) method was successfully applied to classify coffee bean images into four classes: Dark, Green, Light, and Medium. The classification process employed five CNN architectures—MobileNetV2, ResNet50, EfficientNetB0, VGG16, and DenseNet121—using a transfer learning approach. The evaluation results indicate that MobileNetV2 was the best-performing model, achieving an accuracy of 98.75% and the shortest training time of 97.69 seconds. DenseNet121 also demonstrated excellent performance, with an accuracy of 98.25%, but required a longer training time of 230.96 seconds. Meanwhile, VGG16 achieved an accuracy of 89.00% with a training time of 491.79 seconds, ResNet50 achieved an accuracy of 63.75% with a training time of 284.08 seconds, and EfficientNetB0 achieved

an accuracy of 25.00% with a training time of 122.46 seconds. These findings demonstrate that CNN architecture selection influences coffee bean image classification performance. MobileNetV2 provided the best balance between classification accuracy and computational efficiency, making it the most effective architecture for the dataset used in this study.

REFERENCES

[1] Pratama, G. A., Puspaningrum, E. Y., & Maulana, H. (2024). "Convolutional Neural Network dan Faster Region Convolutional Neural Network untuk Klasifikasi Kualitas Biji Kopi Arabika." Jurnal Informatika dan Teknik Elektro Terapan (JITET), 12(3).

[2] Murinto, M., Rosyda, M., & Melany, M. (2023). "Klasifikasi Jenis Biji Kopi Menggunakan Convolutional Neural Network dan Transfer Learning pada Model VGG16 dan MobileNetV2." JRST (Jurnal Riset Sains dan Teknologi), 7(2), 183-189.

[3] Hibatullah, A. A., Asriyanik, A., & Apriandari, W. (2024). "Klasifikasi Kualitas Jenis Kopi Halus Robusta Menggunakan Convolutional Neural Network (CNN) dan Mobilenet-V2." JATI (Jurnal Mahasiswa Teknik Informatika), 8(5), 8650-8657.

[4] Putra, K. U., et al. (2024). "Identifikasi Varietas Kopi Berdasarkan Analisis Warna dan Tekstur Menggunakan Metode Convolutional Neural Network." Bulletin of Computer Science Research.

[5] Putra, F. D. B. S. (2023). "Klasifikasi Tingkat Roasting Biji Kopi Berbasis Citra dengan CNN." Jurnal Teknologi Informasi, Universitas Teknologi Digital Indonesia.

[6] Rahmawati, A., et al. (2024). "Penerapan Arsitektur U-Net pada Segmentasi Cacat Biji Kopi untuk Optimalisasi Inspeksi Kualitas." Jurnal Ilmu Komputer dan Sistem Informasi, 5(2).

[7] Yuliyanti, S., et al. (2024). "Multi-Class Real-Time Color Classification of Coffee Beans via Fine-Tuned EfficientNetB0 and Post-Training Quantization." Jurnal Teknik Informatika (Jutif).

[8] Juarto, B., et al. (2024). "Mobile Deep Learning-Based Coffee Bean Quality Classification and Smartphone Integration Using Transfer Learning." SISFO.

[9] Ardian, Y., et al. (2024). "Comparative Performance Analysis of Modified VGG16 and Slim-CNN for Arabica Coffee Bean Defect Classification." ILKOM Jurnal Ilmiah.

[10] Peryanto, A., Yudhana, A., & Umar, R. (2019). "Rancang Bangun Klasifikasi Citra Dengan Teknologi Deep Learning Berbasis Metode Convolutional Neural Network." Jurnal Ilmiah Teknik Informatika (FORMAT), 8(2), 139-140.

[11] Santoso, B. R., Sari, C. A., & Rachmawanto, E. H. (2025). "Coffee Beans Classification Using Convolutional Neural Networks Based On Extraction Value Analysis In Grayscale Color Space." Journal of Applied Informatics and Computing, 9(1), 31-37.

[12] Tamayo-Monsalve, M. A., et al. (2022). "Coffee Maturity Classification Using Convolutional Neural Networks and Transfer Learning." IEEE Access, 10, 42971-42982.

[13] Chang, S., & Huang, C. (2021). "Deep Learning Model for the Inspection of Coffee Bean Defects." Applied Sciences, 11(17), 8226.

[14] Chou, Y., et al. (2019). "Deep Learning Model for Defective Bean Inspection with GAN-Structured Automated Labeled Data Augmentation in Coffee Industry." Applied Sciences, 9(19), 4166.

[15] Arwatchananukul, S., et al. (2024). "Implementing a Deep Learning Model for Defect Classification in Thai Arabica Green Coffee Beans." Smart Agricultural Technology, 9, 100680.

[16] Muchtar, M., et al. (2025). "Edge AI-Based Detection for Defective Coffee Beans Using Deep Learning and Streamlit Framework." IEEE Access, 13, 67977-67992.

[17] Hsia, C. H. (2021). "Deep Convolutional Neural Network for Coffee Bean Inspection." Sensors and Materials, 33(9).

[18] Putra, B. T. W., Amirudin, R., & Marhaenanto, B. (2022). "The Evaluation of Deep Learning Using Convolutional Neural Network (CNN) Approach for Identifying Arabica and Robusta Coffee Plants." Journal of Biosystems Engineering, 47(2), 118-129.