Jurnal Teknik Informatika (JUTIF) Vol. 7, No. 3, June 2026, Page. 2517-2533
P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5338

Detection of Coffee Leaf Diseases Using Deep Learning to Support Digitalization and Smart Agriculture

Siti Mutmainah[*1], **Zumhur Alamin**[2]

[1,2]Department of computer science, Universitas Muhammadiyah Bima, Indonesia

Email: [1]siti.mutmainah.id19@gmail.com

Received : Nov 21, 2025; Revised : Dec 20, 2025; Accepted : Jan 19, 2026; Published : Jun 15, 2026

Abstract

Coffee is one of Indonesia's main commodities and an important agricultural sector for the economy. However, one of the challenges in coffee cultivation is disease. Improper and delayed treatment can cause leaf damage and death of coffee plants. This study aims to detect disease types on coffee leaves using a deep learning CNN approach and lightweight CNN architectures such as MobileNet and EfficientNet variants. This study also applies traditional image augmentation and OpenCV. The results of EfficientNetV2S and EfficientNetV2L achieve 95–98% accuracy with stable precision and recall in almost all classes, although minority classes remain challenging. The MobileNetV3 architecture showed optimal results with 99% accuracy in all variants (Small, Large, and Small with OpenCV augmentation). The research model was verified using local coffee leaf images Bumi Pajo. These findings confirm that MobileNetV3 not only excels in terms of accuracy but also has the potential to be applied to mobile device-based or Internet of Things (IoT) coffee leaf disease monitoring systems. With high accuracy and low computational requirements, this model can support real-time disease detection in the field, helping farmers and agricultural practitioners make quick and accurate decisions in disease control.

Keywords : *Agricultural Digitalization, Coffee Leaf Disease, Deep Learning, Disease Detection, MobileNet.*

1. INTRODUCTION

Coffee is a cultivated plant in more than 80 countries and considered a high economic value plant [1]. In additional, coffee contains compounds rich in antioxidants, anti-inflammatory and has anti-cancer potential [2] and decreases the risk of diabetes, cardiovascular and liver [3] [4]. Coffee is the most consumed drink in the world, recorded consumption in 2021-2022 reached nearly 176 million 60-kilogram bags [5]. Every day at least 2.25 billion cups are consumed [6]. Coffee culture with production is the second largest world commodity after oil [7]. Coffee is an agricultural commodity and superior product that contributes greatly to Indonesia's foreign exchange [8]. Indonesia exported 342 thousand tons from January to September 2024 [9].

Bumi Pajo, Bima Regency is a coffee producer in West Nusa Tenggara. Coffee produced has the potential to be developed on a national and international scale. However, coffee quality and productivity is influenced by various factors [10], one of the main challenges in coffee cultivation is disease attack. Diseases on coffee leaves often pose a threat to Bumi Pajo coffee farmers, Some types of diseases are Miner (insect borer attack), Phoma (Phoma fungal infection), Red Spider Mite (red mite), and Rust (leaf rust caused by Hemileia vastatrix). Understanding the factors is critical to producing quality coffee and optimizing sensory and potency [11]. Inappropriate and untimely treatment can lead to leaf damage and death of coffee plants. Identifying and quantifying diseases in plants is important in phytology, this is to understand the cause and manage the disease effectively. A technology-based system is needed to detect diseases on coffee leaves quickly and efficiently so that appropriate actions can be taken.

The utilization of technology in plant disease detection can be an innovative solution [12]. Deep learning is an artificial intelligence method that can be used to solve various problems and handle image detection [13]. Deep learning results accurate and effective in various applications and diagnosis such as classification, object detection, segmentation and enhancement [14],[15]. Previous research explored various approaches for detection in health fields such as malaria [16], eye [17], cardiovascular [18] and agriculture [19] such as chili [20], tomato [21], [22]. One of the widely used deep learning approaches is CNN [14]-[21]. Some types of CNN models that are commonly used are c Another approach used to detect objects or diseases is YOLO [31],[32]. Lightweight CNN architectures such as MobileNet and EfficientNet are highly relevant in the development of image-based automated disease detection systems, CNNs have been proven to improve automated diagnosis capabilities [33].

Detection is error-prone, so in-depth knowledge of disease characteristics is required. Diseases on coffee leaves can be similar but caused by different pathogens, such as spotting caused by Phoma and Rust. In addition, problems such as limited datasets and limited computing power and memory will cause errors in identification. Incorrect identification can lead to inappropriate treatment. Feature extraction techniques can help distinguish the unique characteristics of each disease by utilizing analytics such as color, texture and leaf edges. Feature extraction can simplify input data and improve model efficiency. Feature extraction is an important aspect in image processing to aid recognition and detection tasks [34]. Previous research applied feature extraction to improve the model's ability to detect [23]-[26].

Standard augmentation methods with low variability values cannot represent the complexity of image variation in real field conditions. Therefore, it is necessary to compare it with the OpenCV-based augmentation approach, which is closer to field conditions with model generalization capabilities. OpenCV is a popular library that can be used in various applications in image processing [39],[40]. OpenCV has optimized image processing to be faster and more efficient [41], besides allowing normalization and reducing the risk of overfitting with data augmentation, thus improving the quality of the dataset for model training. OpenCV enables real-time quality monitoring, thereby preventing material waste [42]. Previous research applied OpenCV because it has lightweight computing and is important in mobile applications with hardware limitations [43],[44],[45].

This study utilizes deep learning to detect types of coffee leaf diseases based on images of infected coffee leaves. The contribution of this study is the use of a local dataset of Bumi Pajo coffee leaves. In addition, this study utilizes OpenCV Feature Extraction to improve the accuracy of deep learning models in detecting diseases on coffee leaves. This study also implements and compares several CNN models with architectures to determine the best model for detecting diseases on coffee leaves. The combination of deep learning with OpenCV augmentation ensures that this research can be developed into a highly efficient mobile application prototype. This is expected to help farmers quickly and accurately identify diseases on coffee leaves, thereby supporting digitalization and sustainable smart agriculture.

2. METHOD

This study uses an experimental research design that focuses on detecting types of coffee leaf diseases as input for computational processing. This study uses a deep learning approach with CNN, MobileNet and EfficientNet. The computational process uses Google Colab with cloud-based GPU. This study utilizes preprocessing libraries such as OpenCV to process coffee leaf images. The proposed methodological stages in this study can be seen in Figure 1, starting from the collection of coffee leaf images to model evaluation.

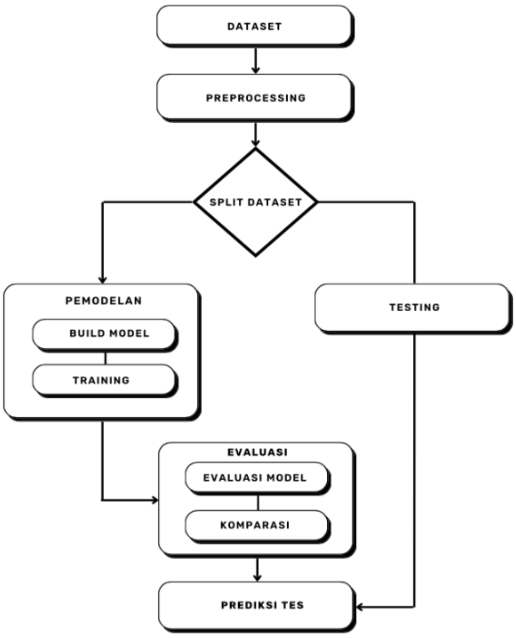


Figure 1. Proposed Method

2.1. Dataset

This study uses a dataset of coffee leaf images, this dataset consists of about 3,766 which come from open-source data. This dataset includes images of healthy coffee leaves and leaves affected by Miner's disease, Phoma, Red Spider Mite, Rust. 400 test data, 368 validation data and data of local coffee leaf images from Bumi Pajo plantation in Bima Regency. At this stage, the image parameters are initialized with a size of 96x96 pixels, number of channels = 3, color mode = RGB, batch size = 16. Next, prepare the dataset using the tf.keras.preprocessing.image_dataset_from_directory library. Image files from the folder will automatically be labelled according to the subfolder name, then the data order will be randomized. Figure 2 shows the input image subfolders used, Figure 3 presents a visual representation of several classes in the coffee leaf disease dataset. It shows the unique characteristics that form the basis of the classification process, such as yellow-orange spots on leaves for the Rust class and green leaves without damage for the Healthy class. This image visualization aims to provide an initial overview of the condition of coffee leaves used as training data.

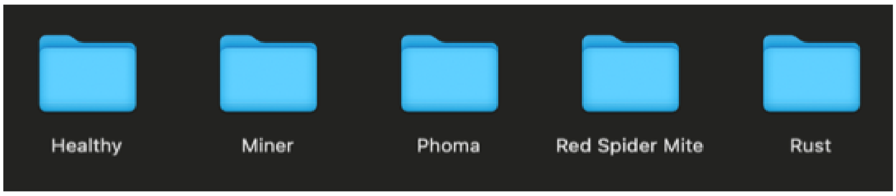


Figure 2. Coffee Leaf Image

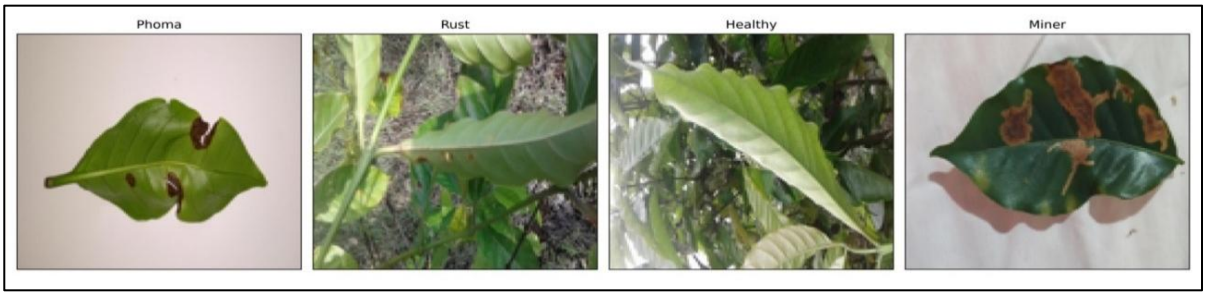


Figure 3. Sample Coffee Leaf Image

2.2. Preprocessing

Preprocessing is performed to improve the diversity and quality of deep learning model training [46], [47]. In the preprocessing stage image resized, the input image size becomes 224 × 224 pixels, and all pixel values (0–255) are divided into 0–1. This size is widely used as a standard input model in image processing research [48], [49], [50], as it is considered to be a good balance between preserving important details and remaining computationally efficient without requiring excessive memory. This step is important for more stable training to avoid large numbers that can cause gradient explosions, augmentation enriches the data so that it can improve accuracy by creating variations of images [13], [46], [47], [51]. This aims to allow data variation and reduce overfitting by applying augmentation techniques such as rotation, flipping, zooming, and lighting changes Using the *TensorFlow.Keras ImageDataGenerator* library. Figure 4 shows the application of augmentation to images of coffee leaf disease.

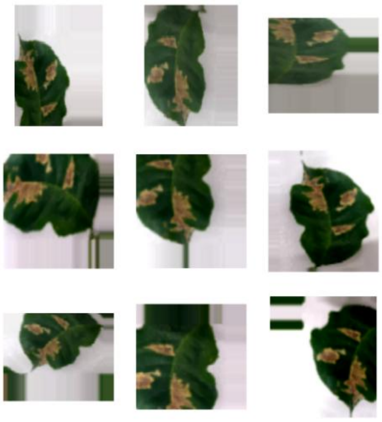


Figure 4. Dataset Augmentation

The process begins from randomly rotating the image horizontally and vertically using *RandomFlip("horizontal_and_vertical"),* an augmentation strategy Figure 4. Next, the image is randomly rotated up to ±23 degrees, enlarged or reduced up to ±30%, and shifted along the vertical and horizontal axes with the same shift limits. Additionally, the image height is randomly adjusted by ±30%, and the image contrast level is randomly adjusted to 90% of its original value. Empty areas resulting from the transformation are filled with nearby pixels to maintain image integrity. The augmentation process is performed dynamically each time a data batch is processed, so that at each training epoch, the model sees new variations of the same image, which ultimately helps improve the model's generalization ability to previously unseen data.

This study compares the results of image augmentation using the OpenCV method with conventional augmentation using the deep learning library. OpenCV one of the free libraries widely used in various image and video processing applications [52]. OpenCV is also effective for text extraction from natural scene images when combined with Convolutional Neural Networks, providing better performance and time complexity compared to traditional multi-stage approaches [53]. OpenCV provides a variety of more adaptive image preprocessing techniques such as Gaussian Blur, thresholding, edge detection, and morphological transformation. The quality of leaf images is improved by reducing noise, sharpening leaf edges, and normalizing lighting. Color space conversion from RGB to HSV/YCbCr helps highlight color differences so that disease symptoms such as spots or discoloration of leaves can be better identified.

Figure 5 represents dataset augmentation with OpenCV, OpenCV generates more diverse transformation variations such as random angle rotation, translation, color intensity changes, and blur effects, which directly increase the model's robustness against image conditions.

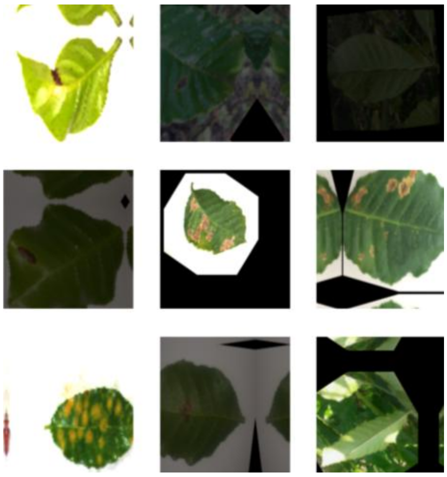


Figure 5. Dataset Augmentation OpenCV

2.3. Modelling

Previously, the dataset was split into three sections for CNN method comparison purposes. CNN is one of the artificial neural network algorithms that is often used in various applications, such as object classification and recognizing patterns and features from input data such as images [49]. In addition, previous studies have used CNN to support smart agriculture [54]. Figure 6 shows the CNN architecture. CNN consists of several layers, including convolution layers that extract hierarchical spatial features, pooling layers that reduce dimensions while retaining important features, and fully connected layers that synthesize these features into final predictions [33].

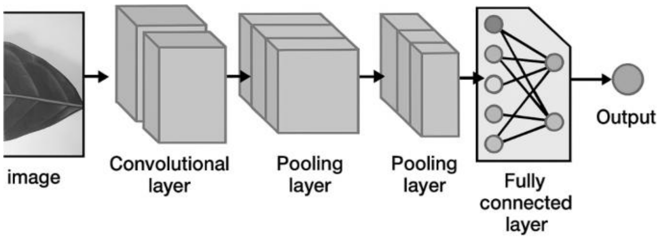


Figure 6. CNN Arsitektur

As development progresses, several CNN architectures have been proposed with the aim of improving accuracy, efficiency, and computation, such as lighter architectures like MobileNet and EfficientNet. Due to their high computational efficiency, MobileNet and EfficientNet enable real-time inference on devices with limited resources [55]. In recent studies, MobileNet and EfficientNet have become popular choices as CNN backbones for several tasks.

MobileNet is a family of lightweight CNNs designed specifically for mobile and embedded devices with limited resources. The main idea is depthwise separable convolution, which separates standard convolution into two stages: depthwise and pointwise operations [56]. MobileNetV2 refines this design by introducing an inverted residual structure and linear bottleneck, which improves representation capabilities while maintaining efficiency [55]. MobileNetV3 is the result of optimization through Neural Architecture Search (NAS) and architecture adaptation algorithms (NetAdapt), adding squeeze-and-excitation modules and new activation functions to further improve performance on limited devices [55]. This combination of techniques makes MobileNet ideal for medical imaging applications where lightweight and fast models are required.

EfficientNet is one of the CNN architecture families introduced by Tan and Le (2019) with an advanced model scaling approach. Its main innovation is the concept of compound scaling, which scales depth, width, and input resolution simultaneously with optimized coefficients. This model can balance all these dimensions to achieve optimal performance. In addition, Tan et al. used Neural Architecture Search to find the optimal base architecture, then scaled the model into a series of EfficientNets. EfficientNet is capable of achieving very high accuracy with a relatively small number of parameters and computations [57] [55].

This split was intended to ensure that the model could be trained optimally, validated to prevent overfitting, and tested to evaluate performance objectively. The data was divided into 80% training data, 10% testing data, and 10% validation data [49], [58], [59]. Modelling using *Tensorflow.keras*, a pre-trained Convolutional Neural Network model using the *ImageNet* dataset. The model is loaded without its top classification layer (include_top=False) so that it can be used as a feature extractor, and the input size is set to (96, 96, 3) according to the image dimensions of the dataset. All weights are locked (trainable=False) so that during training only the newly added layers are updated, while the original weights are retained to leverage the knowledge from pre-training. The base model output is processed through *Global Average Pooling 2D*, followed by three Dense layers (512, 256, 128 units, ReLU) and a Softmax output layer with 5 neurons for five-class classification. The model was trained using the Adam optimizer with an initial learning rate of 0.001. In addition, a callback with Early Stopping At Max Accuracy stopped training when the accuracy reached 99% to control training adaptively, while verbose = 1 displayed training progress at each epoch. Training uses the Exponential Decay learning rate schedule decay_steps=10000 and decay_rate=0.9, the Sparse Categorical Crossentropy loss function, and the accuracy metric. The model is trained for 25 epochs with a batch size of 16 using training and validation data to monitor model performance.

2.4. Model Evaluation

At this stage, model performance is evaluated using a Confusion Matrix. Several parameters are used accuracy, precision, recall, and F1 score [58]. This evaluation aims to compare the performance of CNN, MobileNet, and EfficientNet architectures in classifying coffee leaf diseases. These evaluation metrics provide a diverse approach to assessing model performance [60]. Accuracy describes the proportion of correct predictions overall but tends to be sensitive to imbalances in the amount of data in each class. Recall assesses the model's ability to recognize all samples from a class, so it is important to minimize omission errors. Precision evaluates the accuracy of positive predictions, thereby reducing false positive errors. Meanwhile, the F1 score is the harmonic mean of precision and recall, which is ideal for use on datasets with imbalanced class distributions. components of the confusion matrix:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$F1 - score = 2x\,\frac{Precision + Recall}{Precision + Recall} \tag{4}$$

Where:

- TP (True Positives): Correctly predicted positive cases
- FP (False Positives): Negative cases incorrectly predicted as positive
- TN (True Negatives): Correctly predicted negative cases
- FN (False Negatives): Positive cases incorrectly predicted as negative

3. RESULT

3.1. Inisialisasi Parameter dan Dataset

Dataset used is a collection of coffee leaf images with five class categories, namely Healthy, Miner, Phoma, Red Spider Mite, and Rust. Each category represents a specific condition of coffee leaves, whether healthy or infected with certain diseases or pests. Labeling was performed by assigning class labels to each image based on the condition of the leaf. The data distribution shown in Figure 7 indicates that the Rust class has the highest number of data points, with 1,207 images (32.0%), followed by the Healthy class with 1,191 images (31.6%). The Miner class has 717 images (19%), the Phoma class has 484 images (12.9%), while the class with the least amount of data is Red Spider Mite with 167 images (4.4%).

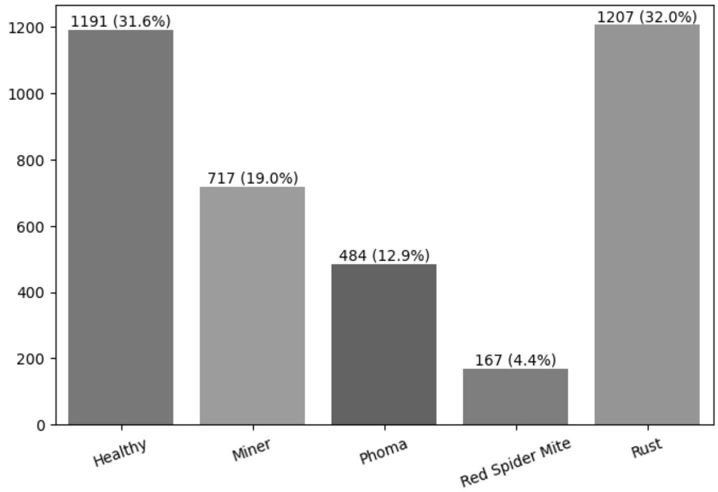


Figure 7. Dataset Augmentation

3.2. Modelling

The modeling process was carried out in several stages to test several model architectures, such as image preprocessing techniques and data augmentation strategies. The architectures used included conventional CNN, MobileNet and EfficientNet, each of which was selected to evaluate the performance differences between the baseline model, a lightweight model with high efficiency, and a model with compound scaling for performance optimization. Data augmentation techniques such as rotation, flipping, and zooming were applied to expand the diversity of the dataset. Additionally, preprocessing methods like image resizing and normalization were applied to ensure consistent input across all models. This approach enables a comprehensive evaluation of the contribution of each factor architecture, augmentation, parameter configuration, and the same test data proportion to the model's performance in detecting coffee leaf disease.

3.2.1. Convolutional Neural Network (CNN)

The CNN model training process shown in Figure 8 shows a loss trend curve that indicates a consistent downward trend in both training loss and validation loss. At the beginning of training, the loss value is quite high, but it gradually decreases, indicating that the model is gradually minimizing prediction errors. The best epoch for loss is achieved at epoch 25, indicating the optimal performance point in minimizing errors in the validation data. Meanwhile, the accuracy graph shows a steady improvement in model performance, with validation accuracy generally exceeding training accuracy. The best accuracy value is achieved at epoch 24, exceeding 90%. Overall, the CNN model can learn effectively and produce good performance on validation data.

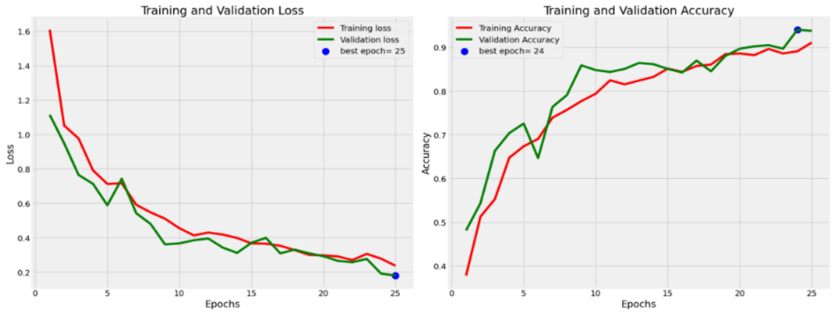
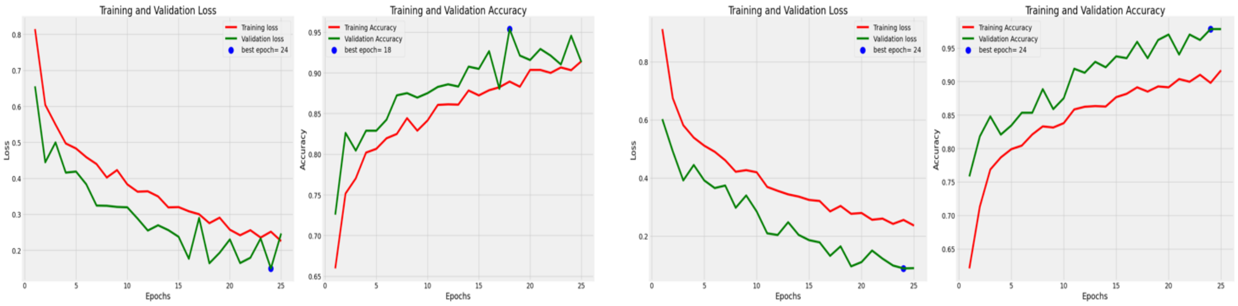


Figure 8. Training History CNN

3.2.2. EfficientNet

Based on the training results presented in Figure 9, EfficientNetV2L shows a steady increase in accuracy until it reaches an optimal point of 97% at epoch 24, with very high validation accuracy and consistent loss reduction. Meanwhile, EfficientNetV2S also showed good performance, but the optimal point was reached earlier, namely 95% in the epoch 18. Although the validation accuracy of EfficientNetV2S is relatively high, EfficientNetV2L is slightly superior in maintaining performance stability in the final epochs, as seen from the smaller gap between training and validation loss. This indicates that EfficientNetV2L has better generalization than EfficientNetV2S, especially on validation data. Thus, in this case, EfficientNetV2L can be considered more optimal in terms of overall stability and accuracy, even though EfficientNetV2S has a faster convergence speed.



EfficientNetV2S

EfficientNetV2L

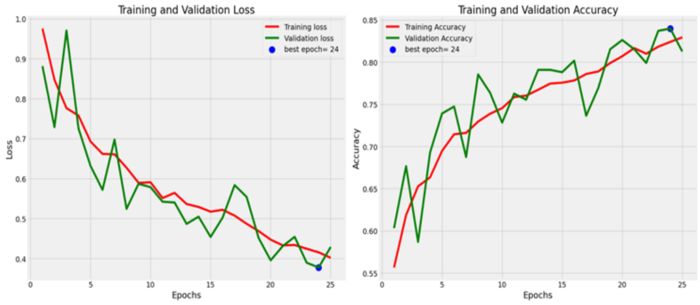
Figure 9. Training History EfficientNet

3.2.3. MobileNet

The results of MobileNetV2 training in Figure 10 show a relatively consistent decrease in training loss and validation loss until the best epoch at epoch 24. Validation accuracy reached a value of around 84%, lower than MobileNetV3-S and MobileNetV3-L, both of which were able to reach close to 99%. The curve pattern shows significant fluctuations in validation loss, particularly in the early epochs, which may indicate that the model is more sensitive to variations in the validation data. In MobileNetV3-S, training loss and validation loss decrease consistently until they approach zero at epoch 16, which is the best epoch.

Validation accuracy reached a peak value of around 99% at that epoch, with a relatively smooth loss curve and a slight difference between training and validation, indicating that the model learned efficiently without significant overfitting. Meanwhile, MobileNetV3-L also showed a good loss reduction trend, with the best epoch for the loss metric at epoch 21 and the best epoch for accuracy at epoch 15. Validation accuracy also reached around 99%, but the loss curve in the early training phase showed slightly higher fluctuations compared to MobileNetV3-S and required more epochs to achieve full stability.

MobileNetV2



MobileNetV3-S **MobileNetV3-L**

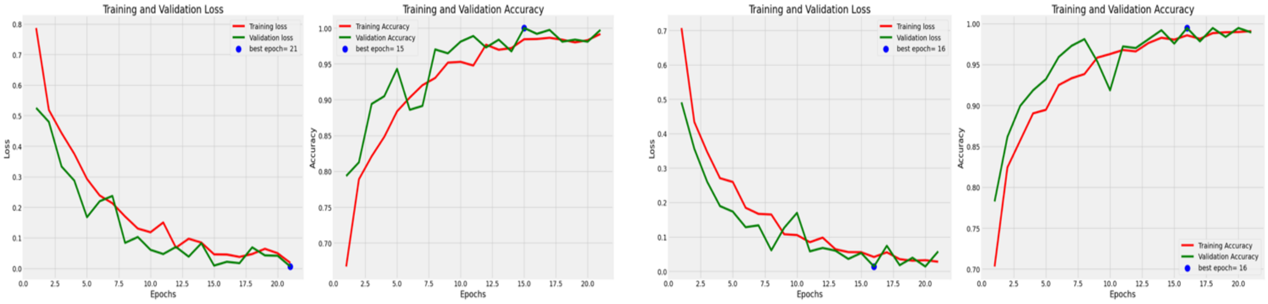


Figure 10. Training History MobileNet

Based on the training results, all three models showed a trend of increasing accuracy and decreasing loss as the number of epochs increased, but with different characteristics. MobileNetV3-S reached its optimal point faster at epoch 16 with a validation accuracy of 99% and a smooth loss curve, reflecting high training efficiency. MobileNetV3-L takes longer to stabilize, with the best accuracy at epoch 15 and loss at epoch 21, but it is still able to achieve accuracy comparable to MobileNetV3-Small. Meanwhile, MobileNetV2 shows lower performance with a maximum validation accuracy of 84% at epoch 24, as well as relatively large fluctuations in validation loss, especially in the early stages of training. MobileNetV3-S offers faster training efficiency, while MobileNetV3-Ltends to require more resources and training time.

3.2.4. MobileNet With OpenCV

The experiment in this study also applied OpenCV for image augmentation. Figure 11 shows the training model using MobileNetV3-S. The training model diagram results were excellent, as indicated by a decrease in training loss and a significant increase in accuracy. At epoch 18, the model achieved its best performance with the lowest validation loss and highest validation accuracy, which remained stable until the end of training. The balanced curve pattern between the training and validation data also indicates that the model did not experience overfitting, so it can be concluded that the model's performance is highly optimal and capable of generalizing well to new data.

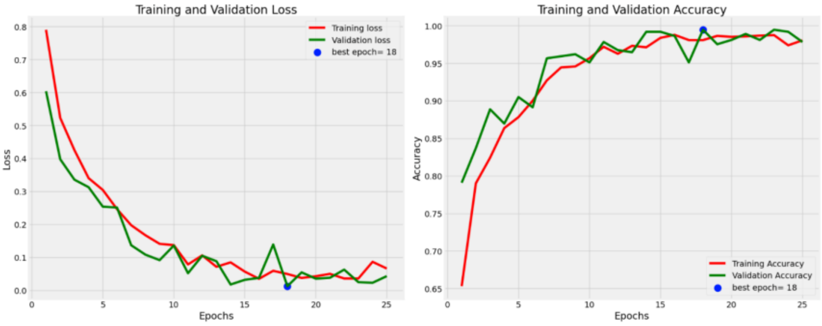


Figure 11. Training MobileNet with OpenCV

3.3. Evaluation

The results of the evaluation of several models are presented in Table 1, with the main metrics used being accuracy, precision, recall, and F1-score. In the table, the Class and Support columns show the distribution of test data for each class. The accuracy value describes the level of prediction accuracy, while precision assesses the extent to which positive predictions are truly relevant. Recall indicates the model's ability to detect all positive samples, while the F1-score represents the balance between precision and recall. This comparison provides an overview of which models perform well in recognizing data patterns, while also highlighting the varying effectiveness of each algorithm across the available class distributions. By analyzing these metrics, the model's ability to accurately detect objects can be evaluated, and classification errors can be identified.

Table 1. Model Evaluation

Method	Class	Support	Accuracy	Precision	Recall	F1 score
CNN	Healthy	137	93	91	99	94
	Miner	74		99	100	99
	Phoma	52		100	100	100
	Red Spider Mite	16		80	25	38
	Rust	121		90	88	89
EfficientNetV2S	Healthy	129	95	93	100	97
	Miner	77		95	100	97
	Phoma	59		100	95	97
	Red Spider Mite	13		90	69	78
	Rust	122		97	91	94
EfficientNetV2L	Healthy	121	98	97	100	98
	Miner	93		100	100	100
	Phoma	50		100	100	100
	Red Spider Mite	22		83	91	87
	Rust	114		100	95	97
MobileNetV2	Healthy	111	82	73	81	77
	Miner	90		87	90	89
	Phoma	45		92	98	95
	Red Spider Mite	20		100	20	33
	Rust	134		83	81	82
MobileNetV3S	Healthy	138	99	100	100	100
	Miner	73		100	100	100
	Phoma	46		100	100	100
	Red Spider Mite	12		100	92	96
	Rust	131		99	100	100
MobileNetV3L	Healthy	125	99	100	99	100
	Miner	71		100	100	100
	Phoma	51		100	100	100
	Red Spider Mite	14		100	100	100
	Rust	139		99	100	100
Method	**Class**	**Support**	**Accuracy**	**Precision**	**Recall**	**F1 score**
MobileNet OpenCV	Healthy	125	99	100	100	100
	Miner	70		100	97	99
	Phoma	53		98	100	99
	Red Spider Mite	17		100	100	100
	Rust	137		99	100	100

The evaluation results in Table 1 show that model performance varies between methods. The CNN and MobileNetV2 models still show limitations, especially in the Red Spider Mite class, with low recall and F1-score values due to limited data. Improved performance is observed in EfficientNetV2S and EfficientNetV2L, achieving accuracy of 95–98% with stable precision and recall across nearly all classes, though minority classes remain challenging. Meanwhile, the MobileNetV3S and MobileNetV3L models showed the best performance with 99% accuracy and nearly perfect precision, recall, and F1-score values across all classes, concluding that MobileNetV3 is the best architecture in this study.

3.4. Coffee Leaf Disease Prediction

In addition, to strengthen the validation of the results, this study also presents examples of visualizations of predictions from several models on local datasets. Figure 12 shows several samples of coffee leaves with actual labels and prediction results using the MobileNetV3 architecture, which successfully classified the condition of the leaves with high accuracy. In general, the models were able to identify leaf disease classes well, as indicated by the agreement between the ground truth labels and the prediction results in most samples. For example, healthy leaves and those infected with Rust, Miner, and Phoma were accurately classified by almost all models. However, some prediction errors also occurred, such as the EfficientNetV2S model incorrectly identifying Rust-class leaves as Miner. This visualization shows that while the performance of the models, particularly MobileNetV3S and MobileNetV3L, is already very good with high accuracy, challenges still arise for classes with similar visual characteristics or limited data, such as Red Spider Mite.

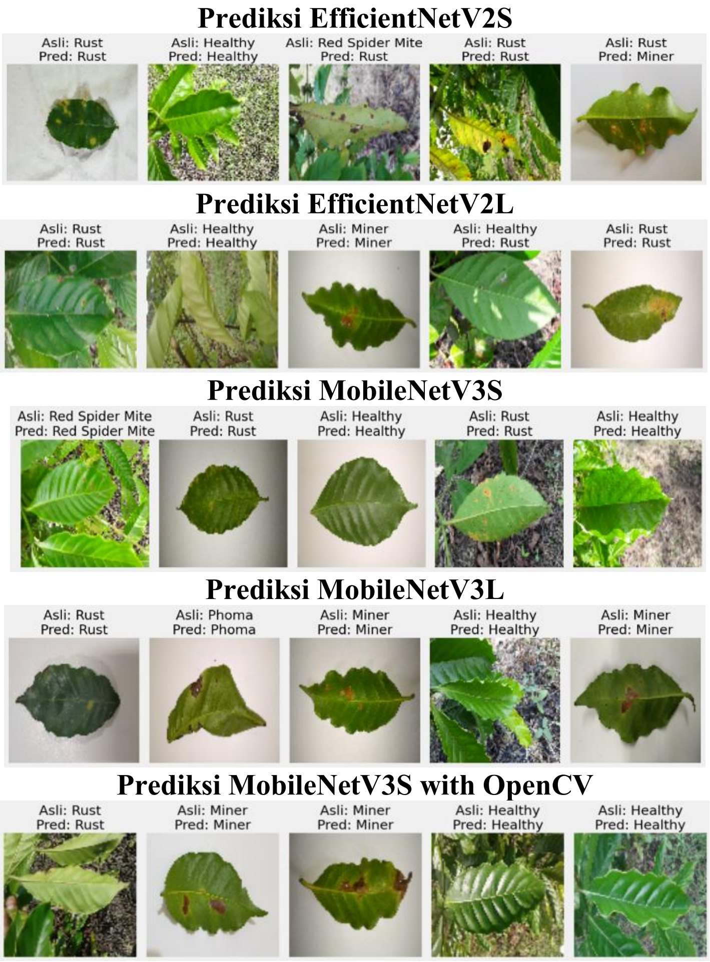


Figure 12. Coffee Leaf Disease Prediction

4. DISCUSSIONS

This study developed an efficient model for detecting coffee leaf disease by utilizing CNN architecture, specifically the EfficientNet and MobileNet variants. In addition, this study also compared the performance of several modern architectures and applied image augmentation with OpenCV to improve the model's generalization capabilities. Figure 13 presents a comparison, the results of the experiment showed that the use of a lighter but optimal architecture, such as MobileNetV3, was able to produce very high accuracy while being computationally efficient [30], [61]. Making it ideal for implementation on devices with limited resources, such as IoT-based monitoring systems or mobile applications. The comparison of the research shown in Table 2:

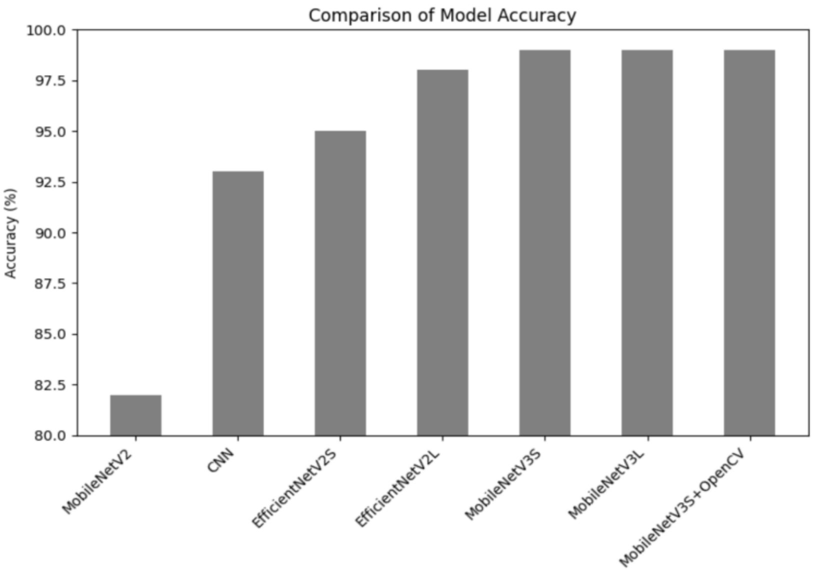


Figure 13. Comparison Model

Table 2. Model Comparison Evaluation

Model	Accuracy
Resnet-50 [62]	96%
CNN [63]	88%
CNN [64] [24]	97%
MobileNet [65]	97%
MobileNetV3	**99%**

Table 2 presents a comparison of the research results with previous studies. The results show that the conventional CNN model has an accuracy of 88% to 97%, The results of the experiment show a trade-off between accuracy and computation time. While the MobileNetV3 architecture shows the most optimal results with an accuracy of 99% across all variants (Small, Large, and Small with OpenCV augmentation), the main advantage of this architecture lies in its lightweight and efficient design, which requires shorter training and inference times compared to EfficientNetV2L. Conversely, EfficientNetV2L has a much larger number of parameters and network complexity, which results in longer computation time, both in the training and inference stages, despite its potential to deliver competitive performance.

These findings confirm that MobileNetV3 offers a more optimal balance between accuracy and computational efficiency, especially for real-time and mobile-based implementations, while EfficientNetV2L is more suitable for computing environments with large resources and looser time constraints. These findings confirm that MobileNetV3 is not only superior in terms of accuracy but also computationally efficient, making it more suitable for real-world implementation.

5. CONCLUSION

This study developed an efficient model for classifying images of coffee leaf diseases to support Smart Agriculture by utilizing Convolutional Neural Networks (CNN) with EfficientNet and MobileNet architectures. This study also compared the application of augmentation and OpenCV. In addition, this model achieved the highest accuracy compared to traditional CNN architectures. Experimental results show that the performance of EfficientNetV2S and EfficientNetV2L achieves stable accuracy, precision, and recall in almost all classes. Meanwhile, the best models observed, MobileNetV3S and MobileNetV3L, show the best performance with 99% accuracy and near-perfect precision, recall, and F1-score values in all classes. Further research can be expanded to explore data augmentation methods and make them more suitable for real-world implementation. MobileNetV3 has the potential to be applied to real-time coffee leaf disease detection systems based on Android mobile devices or the Internet of Things (IoT). With high accuracy and low computational requirements, this model can support real-time disease detection in the field, thereby helping farmers and agricultural practitioners make quick and accurate decisions for disease control.

ACKNOWLEDGEMENT

This research was funded by the 2025 Beginner Lecturer Research grant, number 129/C3/DT.05.00/PL/2025 from the Ministry of Education, Culture, Research, and Technology, Indonesia.

REFERENCES

[1] A. A. Moreira-Morrillo, J. P. Vélez-Zambrano, S. I. Moreira, dan F. R. Garcés-Fiallos, “Diseases affecting the coffee crop: Elucidating the life cycle of Rust, Thread Blight and Cercospora Leaf Spot.,” *Scientia Agropecuaria*, vol. 14, no. 3, 2023.

[2] S. Shah dan A. Kumar, “Coffee: constituents and health benefits,” *Biotechnol. J. Int*, vol. 24, hlm. 22–38, 2020.

[3] M. Saeed, M. Naveed, J. BiBi, A. Ali Kamboh, L. Phil, dan S. Chao, “Potential nutraceutical and food additive properties and risks of coffee: A comprehensive overview,” *Critical reviews in food science and nutrition*, vol. 59, no. 20, hlm. 3293–3319, 2019.

[4] A. Wijaya, P. Glasbergen, dan S. Mawardi, “The mediated partnership model for sustainable coffee production: experiences from Indonesia,” *IFAM*, vol. 20, no. 5, hlm. 689–708, Okt 2017, doi: 10.22434/IFAMR2017.0021.

[5] M. Ridder, “Coffee consumption worldwide from 2012/13 to 2021/22 with a forecast to 2022/23,” statista. [Daring]. Tersedia pada: https://www.statista.com/statistics/292595/global-coffee-consumption/?__sso_cookie_checker=failed

[6] Doddy Rosadi, “Tiap Hari, 2 Miliar Cangkir Kopi Dikonsumsi di Seluruh Dunia,” Katadata.co.id. [Daring]. Tersedia pada: https://sisiplus.katadata.co.id/berita/gaya-hidup/363/tiap-hari-2-miliar-cangkir-kopi-dikonsumsi-di-seluruh-dunia#:~:text=Sekitar%20650%20juta%20orang%20di,(9%20kg/tahun).

[7] D. L. Brien dan J. Adams, “Coffee: a cultural and media focussed approach,” *M/C Journal*, vol. 15, no. 2, 2012.

[8] Sekertaris Jenderal, “Analisis Kinerja Perdagangan Kopi Semester II Tahun 2022,” Portal Satu Data Pertanian. [Daring]. Tersedia pada: https://satudata.pertanian.go.id/details/publikasi/450#:~:text=Portal%20Satu%20Data%20Pertanian&text=Kopi%20merupakan%20salah%20satu%20komoditas,angka%20estimasi%2C%20Ditjen%20Perkebunan).

[9] Kuntoro Boga Andri, “Tren 2025: Peluang dan Daya Saing Kopi Indonesia,” Balai Pengujian Standar Instrumen Tanaman Industri dan Penyegar. Diakses: 14 Maret 2025. [Daring]. Tersedia pada: https://tanamanindustri.bsip.pertanian.go.id/berita/tren-2025-peluang-dan-daya-saing-kopi-indonesia#:~:text=Sementara%20konsumsi%20domestik%20turun%20dari,Mesir%2C%20Jer

man%2C%20dan%20Malaysia.
[10] H. Deribe, “Review on Factors which Affect Coffee (Coffea Arabica L.) Quality in South Western, Ethiopia,” *International Journal of Forestry and Horticulture*, vol. 5, no. 1, hlm. 12–19, 2019.
[11] V. V. Freitas, L. L. R. Borges, M. C. T. R. Vidigal, M. H. dos Santos, dan P. C. Stringheta, “Coffee: A comprehensive overview of origin, market, and the quality process,” *Trends in Food Science & Technology*, hlm. 104411, 2024.
[12] M. A. Markom *dkk.*, “Intelligent electronic nose system for basal stem rot disease detection,” *Computers and Electronics in Agriculture*, vol. 66, no. 2, hlm. 140–146, 2009.
[13] E. E. Citra, S. Mutmainah, dan B. Hermanto, “Breast Cancer Detection Using EfficientNetV2 Variants and Data Augmentation: A Comparative Study,” *Jurnal Komputasi*, vol. 13, no. 1, hlm. 13–24, 2025.
[14] D. G. Thippanna, M. D. Priya, dan T. A. S. Srinivas, “An Effective Analysis of Image Processing with Deep Learning Algorithms,” *Int. J. Comput. Appl*, vol. 975, hlm. 8887, 2023.
[15] B. Meena, K. V. Rao, dan S. Chittineni, “A survey on deep learning methods and tools in image processing,” *Int. J. Sci. Technol. Res.*, vol. 9, no. 2, hlm. 1057–1062, 2020.
[16] E. L. Lydia, G. J. Moses, N. Sharmili, K. Shankar, dan A. Maseleno, “Image classification using deep neural networks for malaria disease detection,” *International Journal on Emerging Technologies*, vol. 10, no. 4, hlm. 66–70, 2019.
[17] R. H. Paradisa, A. Bustamam, W. Mangunwardoyo, A. A. Victor, A. R. Yudantha, dan P. Anki, “Deep feature vectors concatenation for eye disease detection using fundus image,” *Electronics*, vol. 11, no. 1, hlm. 23, 2021.
[18] M. Iqbal, M. N. Setiawan, M. I. Irawan, K. M. N. K. Khalif, N. Muhammad, dan M. K. B. M. Aziz, “Cardiovascular disease detection from high utility rare rule mining,” *Artificial Intelligence in Medicine*, vol. 131, hlm. 102347, 2022.
[19] A. H. Basori, A. B. F. Mansur, dan H. Y. Riskiawan, “SMARF: Smart farming framework based on big data, iot and deep learning model for plant disease detection and prevention,” dipresentasikan pada International Conference on Applied Computing to Support Industry: Innovation and Technology, Springer, 2019, hlm. 44–56.
[20] R. Widyatama dan S. Hansun, “Expert System for Chili Plants Disease Detection Using Certainty Factor Method,” *International Journal of Innovative Technology and Exploring Engineering (IJITEE) Volume*, vol. 9, hlm. 1145–1151, 2019.
[21] I. N. Yulita, N. A. Amri, dan A. Hidayat, “Mobile application for tomato plant leaf disease detection using a dense convolutional network architecture,” *Computation*, vol. 11, no. 2, hlm. 20, 2023.
[22] H.-C. Chen *dkk.*, “AlexNet convolutional neural network for disease detection and classification of tomato leaf,” *Electronics*, vol. 11, no. 6, hlm. 951, 2022.
[23] Y. Aufar dan T. P. Kaloka, “Robusta coffee leaf diseases detection based on MobileNetV2 model,” *International Journal of Electrical and Computer Engineering*, vol. 12, no. 6, hlm. 6675, 2022.
[24] D. Novtahaning, H. A. Shah, dan J.-M. Kang, “Deep learning ensemble-based automated and high-performing recognition of coffee leaf disease,” *Agriculture*, vol. 12, no. 11, hlm. 1909, 2022.
[25] M. Muharromah, A. Kristiana, S. Slamin, D. Dafik, I. Agustin, dan R. Baihaki, “The analysis of the implementation of convolutional neural network architectures for coffee leaf disease image classification,” dipresentasikan pada AIP Conference Proceedings, AIP Publishing, 2024.
[26] K. J. Ayikpa, D. Mamadou, P. Gouton, dan K. J. Adou, “Experimental Evaluation of Coffee Leaf Disease Classification and Recognition Based on Machine Learning and Deep Learning Algorithms,” *Journal of Computer Science*, vol. 18, no. 12, hlm. 1201–1212, 2022.
[27] M. Nawaz, T. Nazir, A. Javed, S. T. Amin, F. Jeribi, dan A. Tahir, “CoffeeNet: A deep learning approach for coffee plant leaves diseases recognition,” *Expert Systems with Applications*, vol. 237, hlm. 121481, 2024.
[28] E. Hitimana *dkk.*, “An intelligent system-based coffee plant leaf disease recognition using deep learning techniques on Rwandan Arabica dataset,” *Technologies*, vol. 11, no. 5, hlm. 116, 2023.

[29] R. Karthik, J. J. Alfred, dan J. J. Kennedy, “Inception-based global context attention network for the classification of coffee leaf diseases,” *Ecological Informatics*, vol. 77, hlm. 102213, 2023.

[30] S. Signo, C. Tuquero, E. Arboleda, S. Signo, C. Tuquero, dan E. Arboleda, “Coffee disease detection and classification using image processing: A Literature review,” *Int. J. Sci. Res. Arch*, vol. 11, hlm. 1614–1621, 2024.

[31] A. H. Hidayah, S. Zuhriyah, B. E. W. Asrul, Y. Yuyun, dan E. Prakasa, “Integration of yolov5 algorithm and opencv in innovative smart parking management approach,” *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 8, no. 3, hlm. 413–422, 2024.

[32] E. Syahrudin, E. Utami, dan A. D. Hartanto, “Object Detection with YOLOv8 and Enhanced Distance Estimation Using OpenCV for Visually Impaired Accessibility,” *JOIV: International Journal on Informatics Visualization*, vol. 9, no. 2, hlm. 575–584, 2025.

[33] I. D. Mienye, T. G. Swart, G. Obaido, M. Jordan, dan P. Ilono, “Deep Convolutional Neural Networks in Medical Image Analysis: A Review,” *Information*, vol. 16, no. 3, 2025, doi: 10.3390/info16030195.

[34] S. Prabu, B. Natarajan, R. George, S. Sujitha, P. D. Bharathi, dan S. Rajeswari, “Capturing the Essence: Exploring Cutting-Edge Feature Extraction Methods in Digital Image Processing,” dipresentasikan pada 2023 3rd International Conference on Mobile Networks and Wireless Communications (ICMNWC), IEEE, 2023, hlm. 1–5.

[35] K. K. Kumar, K. Chaduvula, dan B. Markapudi, “A detailed survey on feature extraction techniques in image processing for medical image analysis,” *European Journal of Molecular & Clinical Medicine*, vol. 7, no. 10, hlm. 2020, 2020.

[36] A. O. Salau dan S. Jain, “Feature extraction: a survey of the types, techniques, applications,” dipresentasikan pada 2019 international conference on signal processing and communication (ICSC), IEEE, 2019, hlm. 158–164.

[37] A. K. Bharodiya, “Feature extraction methods for ct-scan images using image processing,” dalam *Computed-Tomography (CT) Scan*, IntechOpen, 2022.

[38] V. Chavan, J. Shah, M. Vora, M. Vora, dan S. Verma, “A Review on Feature Extraction Methods for Image Analysis,” dipresentasikan pada Proceedings of the 3rd International Conference on Advances in Science & Technology (ICAST), 2020.

[39] G. Guillen dan G. Guillen, “Digital image processing with Python and OpenCV,” *Sensor Projects with Raspberry Pi: Internet of Things and Digital Image Processing*, hlm. 97–140, 2019.

[40] M. Khan, S. Chakraborty, R. Astya, dan S. Khepra, “Face detection and recognition using OpenCV,” dipresentasikan pada 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), IEEE, 2019, hlm. 116–119.

[41] G. K. L. Alcantara, I. D. J. Evangelista, J. V. B. Malinao, O. B. Ong, R. S. D. Rivera, dan E. L. U. Ambata, “Head detection and tracking using OpenCV,” dipresentasikan pada 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), IEEE, 2018, hlm. 1–5.

[42] P. Castro, G. Pathinettampadian, S. Thanigainathan, V. Prabakar, R. A. Krishnan, dan M. K. Subramaniyan, “Measurement of additively manufactured part dimensions using OpenCV for process monitoring,” *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering*, hlm. 09544089241227894, 2024.

[43] E. Cervera, “GPU-accelerated vision for robots: Improving system throughput using OpenCV and CUDA,” *IEEE Robotics & Automation Magazine*, vol. 27, no. 2, hlm. 151–158, 2020.

[44] J. Sigut, M. Castro, R. Arnay, dan M. Sigut, “OpenCV basics: A mobile application to support the teaching of computer vision concepts,” *IEEE Transactions on Education*, vol. 63, no. 4, hlm. 328–335, 2020.

[45] A. B. Abadi dan S. Tahcfulloh, “Digital Image Processing for Height Measurement Application Based on Python OpenCV and Regression Analysis,” *JOIV : Int. J. Inform. Visualization*, vol. 6, no. 4, hlm. 763, Des 2022, doi: 10.30630/joiv.6.4.1013.

[46] K. Maharana, S. Mondal, dan B. Nemade, “A review: Data pre-processing and data augmentation techniques,” *Global Transitions Proceedings*, vol. 3, no. 1, hlm. 91–99, Jun 2022, doi:

10.1016/j.gltp.2022.04.020.

[47] S. Golazad, A. Mohammadi, A. Rashidi, dan M. Ilbeigi, “From raw to refined: Data preprocessing for construction machine learning (ML), deep learning (DL), and reinforcement learning (RL) models,” *Automation in Construction*, vol. 168, hlm. 105844, Des 2024, doi: 10.1016/j.autcon.2024.105844.

[48] T. Kurniawan, K. Khadijah, dan R. Kusumaningrum, “An Efficient Model for Waste Image Classification Using EfficientNet-B0,” *J. Tek. Inform. (JUTIF)*, vol. 6, no. 3, hlm. 1147–1158, Jun 2025, doi: 10.52436/1.jutif.2025.6.3.4417.

[49] K. F. Habie, M. Murinto, dan S. Sunardi, “Impact of Optimizer Selection on MobileNetV1 Performance for Skin Disease Detection Using Digital Images,” *J. Tek. Inform. (JUTIF)*, vol. 6, no. 3, hlm. 1589–1604, Jul 2025, doi: 10.52436/1.jutif.2025.6.3.4685.

[50] A. R. Priambodo dan C. Fatichah, “Leveraging Convolutional Block Attention Module (Cbam) For Enhanced Performance In Mobilenetv3-Based Skin Cancer Classification,” *J. Tek. Inform. (JUTIF)*, vol. 6, no. 3, hlm. 1389–1404, Jun 2025, doi: 10.52436/1.jutif.2025.6.3.4546.

[51] E. E. Citra, S. Mutmainah, dan B. Hermanto, “Optimization of EfficientNetV2 with Data Augmentation for Early Breast Cancer Detection,” *Jurnal Komputasi*, vol. 13, no. 1, hlm. 13–24, 2025.

[52] R. T. Hasan dan A. B. Sallow, “Face detection and recognition using opencv,” *Journal of Soft Computing and Data Mining*, vol. 2, no. 2, hlm. 86–97, 2021.

[53] V. Goel, V. Kumar, A. S. Jaggi, dan P. Nagrath, “Text extraction from natural scene images using OpenCV and CNN,” *Int J Inf Technol Comput Sci (IJITCS)*, vol. 11, no. 9, hlm. 48–54, 2019.

[54] M. El Sakka, J. Mothe, dan M. Ivanovici, “Images and CNN applications in smart agriculture,” *European Journal of Remote Sensing*, vol. 57, no. 1, hlm. 2352386, 2024.

[55] T. Shahriar, “Comparative Analysis of Lightweight Deep Learning Models for Memory-Constrained Devices,” 9 September 2025, *arXiv*: arXiv:2505.03303. doi: 10.48550/arXiv.2505.03303.

[56] A. G. Howard *dkk.*, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” 17 April 2017, *arXiv*: arXiv:1704.04861. doi: 10.48550/arXiv.1704.04861.

[57] M. Tan dan Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” dipresentasikan pada International conference on machine learning, PMLR, 2019, hlm. 6105–6114.

[58] A. D. Febriani dan M. D. Kartikasari, “Detecting Avocado Freshness In Real-Time: A Yolo-Based Deep Learning Approach,” *J. Tek. Inform. (JUTIF)*, vol. 6, no. 3, hlm. 1485–1502, Jun 2025, doi: 10.52436/1.jutif.2025.6.3.4626.

[59] P. Setiaji, W. A. Triyanto, dan M. Nurhaliza, “Real-Time Traffic Density and Anomaly Monitoring Using YOLOv8, OpenCV and Pattern Recognition for Smart City Applications in Demak,” vol. 6, no. 4, 2025.

[60] I. P. Kamila, C. A. Sari, E. H. Rachmawanto, dan N. R. D. Cahyo, “A Good Evaluation Based on Confusion Matrix for Lung Diseases Classification using Convolutional Neural Networks,” *Advance Sustainable Science, Engineering and Technology*, vol. 6, no. 1, hlm. 0240102, 2024.

[61] P. M. Bidarakundi dan B. M. Kumar, “A Coffee Leaf Disease Detection via Deep Learning Algorithms,” dipresentasikan pada 2024 International Conference on Trends in Quantum Computing and Emerging Business Technologies, IEEE, 2024, hlm. 1–5.

[62] S. Kulkarni, H. Shravan, M. Veeresh, P. D. Shenoy, dan K. Venugopal, “Detection of coffee leaf diseases using densenet-264 and slic segmentation,” dipresentasikan pada 2023 10th IEEE Uttar Pradesh section international conference on electrical, electronics and computer engineering (UPCON), IEEE, 2023, hlm. 769–774.

[63] L. A. Pinto, L. Mary, dan S. Dass, “The real-time mobile application for identification of diseases in coffee leaves using the CNN model,” dipresentasikan pada 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), IEEE, 2021, hlm. 1694–1700.

[64] M. Kumar, P. Gupta, dan P. Madhav, “Disease detection in coffee plants using convolutional neural network,” dipresentasikan pada 2020 5th International Conference on Communication

and Electronics Systems (ICCES), IEEE, 2020, hlm. 755–760.

[65] E. B. Paulos dan M. M. Woldeyohannis, “Detection and classification of coffee leaf disease using deep learning,” dipresentasikan pada 2022 International Conference on Information and Communication Technology for Development for Africa (ICT4DA), IEEE, 2022, hlm. 1–6.