Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Issue 4, October 2025
DOI : https://doi.org/10.33395/sinkron.v9i4.15415
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Comparative Performance Evaluation of MobileNetV3 and ResNet50 for Forest Fire Image Classification

Muhammad Rizky Amirullah Hidayat[1)], Djarot Hindarto[2)*], Asrul Sani[3)]
[1,2)]Informatika, Fakultas Teknologi Komunikasi dan Informatika, Universitas Nasional, Indonesia
[3)]Magister Teknologi Informasi, Fakultas Teknologi Komunikasi dan informatika, Universitas Nasional
[1)]muhammadrizkyamirullahhidayat2023@student.unas.ac.id, [2)]djarot.hindarto@civitas.unas.ac.id,
[3)]asrul.sani@civitas.unas.ac.id

Submitted :Oct 11, 2025 | **Accepted** : Oct 23, 2025 | **Published** : Oct 3, 2025

Abstract: Indonesia is one of the countries with a high incidence of forest and land fires (karhutla), especially during the dry season, thus requiring a fast and efficient early detection system. This study aims to compare the performance of two popular deep learning architectures, namely MobileNetV3 (Large and Small variants) and ResNet50, in forest fire image classification tasks using a transfer learning-based approach. This study emphasizes the comparison between accuracy and computational efficiency in a CPU-only environment, which represents real-world conditions of use in the field without GPU support. The dataset used is a combination of local field images from the Puncak area, Bogor, and a curated public forest fire dataset to ensure the model's generalization ability to diverse geographical conditions. The results of the experiment show that ResNet50 provides the highest accuracy with a training accuracy value of 0.677 and a validation accuracy of 0.647, but requires longer training and inference times. Meanwhile, MobileNetV3-Large and MobileNetV3-Small showed better computational efficiency, with only slightly lower accuracy (0.635 and 0.61) and high training stability. These findings confirm that lightweight models such as MobileNetV3 strike an optimal balance between accuracy, speed, and resource consumption, making them an ideal solution for implementing edge computing-based early detection systems. Overall, this research contributes by providing an empirical comparative analysis that can serve as a reference for selecting deep learning architectures for efficient and adaptive forest fire detection systems that are constrained by hardware limitations.

Keywords: Deep Learning, Forest Fires, Image Classification, MobileNetV3, ResNet50.

INTRODUCTION

Indonesia is a country characterized by vast tropical forests that serve as critical ecosystems supporting biodiversity and contributing significantly to global carbon absorption. However, these natural resources are increasingly threatened by recurring forest and land fires, especially during prolonged dry seasons influenced by climate anomalies such as El Niño. These fires not only destroy ecosystems and reduce biodiversity but also produce dense smoke that severely impacts human health and causes transboundary air pollution(Barmpoutis et al., 2020; Guede-Fernández et al., 2021). The economic losses and environmental degradation resulting from such fires are immense, making early detection and rapid response essential components of effective mitigation efforts (Muhammad et al., 2018). Detecting fires at an early stage allows authorities to respond before they spread uncontrollably, reducing the potential damage to both the environment and local communities.

Traditional detection methods, such as manual ground patrols or visual observation from watchtowers (Barmpoutis et al., 2020), have long been used in Indonesia. However, these conventional approaches are constrained by several factors including limited coverage, slow response times, and heavy dependence on weather and geographic conditions (Barmpoutis et al., 2020). Satellite-based observation systems can cover larger areas but are limited by their temporal resolution and cloud cover interference, which often delay fire detection (El-Madafri et al., 2023). As a result, there is an increasing need for automated systems capable of performing real-time analysis of image data obtained from drones, surveillance cameras, or satellites (Zhao et al., 2018). Advances

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Issue 4, October 2025
DOI : https://doi.org/10.33395/sinkron.v9i4.15415
e-ISSN : 2541-2019
p-ISSN : 2541-044X

in computer vision and deep learning provide a promising solution by enabling machines to identify fire and smoke patterns automatically, efficiently, and accurately .

Deep learning, particularly Convolutional Neural Networks (CNNs), has become the most widely adopted technique in visual recognition tasks because of its ability to automatically learn hierarchical features from raw images (Krizhevsky et al., 2012). Various studies have successfully applied CNNs for forest fire detection, outperforming traditional rule-based or color-segmentation methods (Shi et al., 2024). However, several technical challenges remain unresolved, especially regarding model selection and optimization for real-world deployment. Each CNN architecture exhibits different characteristics that result in trade-offs between accuracy, computational efficiency, and inference speed (Vdovjak et al., 2022). Deep and complex models generally provide higher accuracy but demand substantial computational resources, while lightweight models are more efficient but may suffer from reduced precision in complex scenes(A. G. Howard et al., 2017). Therefore, finding the right balance between accuracy and efficiency is crucial for developing an effective forest fire detection system that can operate in resource-limited environments.

MobileNetV3 and ResNet50 are two well-established CNN architectures that represent contrasting approaches to this trade-off. MobileNetV3 is designed for devices with limited computational capacity, such as mobile phones, drones, or IoT-based environmental sensors. It employs Depthwise Separable Convolutions (A. G. Howard et al., 2017) combined with Neural Architecture Search (NAS) and Squeeze-and-Excitation (SE) modules to achieve high efficiency and reduced computational cost without severely compromising accuracy(A. Howard et al., 2019; Hu et al., 2018). This makes MobileNetV3 particularly suitable for real-time edge computing applications. In contrast, ResNet50, introduced by He et al. (2016), is a deeper and more complex network that utilizes residual learning through skip connections to overcome the vanishing gradient problem. This design allows ResNet50 to extract rich feature representations from images, achieving superior accuracy across various computer vision tasks (Hindarto, 2023), but at the cost of longer inference times and higher energy consumption (Vdovjak et al., 2022).

Previous studies have employed ResNet architectures for smoke and fire detection with strong classification accuracy but relatively high computational overhead (Vdovjak et al., 2022). Conversely, MobileNetV3 has been effectively used for image recognition tasks on mobile and embedded systems due to its lightweight structure and faster inference capabilities(Shi et al., 2024; Zheng et al., 2023). Despite these findings, comprehensive comparative studies (Hindarto, 2024) that analyze both models within the same experimental framework remain limited. In particular, very few studies have evaluated their performance in CPU-only computing environments (Vdovjak et al., 2022), which are more representative of real-world edge deployment conditions. Furthermore, most prior works rely solely on public datasets that may not fully capture the visual characteristics of tropical forest fires in Indonesia, such as dense vegetation, heavy smoke, and varying illumination (Akagic & Buza, 2022; El-Madafri et al., 2023).

This study conducts a comparative experimental analysis to evaluate the performance of MobileNetV3 (Large and Small variants) and ResNet50 models in forest fire image classification. The dataset combines public forest fire datasets with field images collected in the Puncak region of Bogor, Indonesia, to enhance model generalization through diverse visual and environmental conditions. Using transfer learning, the models were trained to accelerate convergence and improve accuracy despite limited data. Model performance was evaluated based on accuracy, and inference time under CPU-only conditions, representing realistic scenarios for low-cost early fire detection systems without advanced hardware. The findings are expected to identify the most efficient and accurate CNN architecture for adaptive and sustainable environmental monitoring and early warning systems in Indonesia.

LITERATURE REVIEW

The use of deep learning techniques, particularly Convolutional Neural Networks (CNN), has become the dominant approach for forest fire detection and classification due to its ability to automatically extract complex visual features (Muhammad et al., 2018). Various studies show that CNN-based methods outperform traditional techniques, especially in consistently detecting fire and smoke patterns under various lighting conditions (Frizzi et al., 2016). However, challenges such as weather variations, fog, and visual similarities between clouds and smoke remain significant obstacles (Barmpoutis et al., 2020). Therefore, CNN models that are not only accurate but also computationally efficient are needed for application in real-time detection systems in the field.

The MobileNetV3 architecture was developed to address resource limitations on mobile and edge devices. This model uses a combination of Depthwise Separable Convolution, Squeeze-and-Excitation (SE) (Hu et al., 2018), and Neural Architecture Search (NAS) (A. Howard et al., 2019) to achieve a balance between accuracy and computational efficiency. The MobileNetV3-Large variant is optimized for high accuracy, while MobileNetV3-Small is designed for speed and power efficiency. In various studies, MobileNetV3 has demonstrated good performance for classification and detection tasks with significantly lower memory consumption and inference time compared to conventional CNN models (Zheng et al., 2023). This makes it an ideal candidate for drone-based (UAV) or embedded ground sensor forest fire monitoring systems operating in remote areas (Shi et al., 2024; Zhao et al., 2018)

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Issue 4, October 2025
DOI : https://doi.org/10.33395/sinkron.v9i4.15415
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Meanwhile, ResNet50 is a deep network architecture introduced by (He et al., 2016) through the concept of residual learning. ResNet uses skip-connections or shortcut connections to overcome the vanishing gradient problem, allowing deeper networks to be trained more stably. ResNet50 has a high ability to extract complex and deep features, making it a popular model for image classification in various domains, including smoke and fire detection (Vdovjak et al., 2022). However, due to its large number of parameters, ResNet50 requires higher computational power and memory than MobileNet, making it less ideal for devices with limited resources (Vdovjak et al., 2022).

Previous studies have compared the performance of MobileNet and ResNet for fire classification and found that MobileNet has a significant computational advantage (Vdovjak et al., 2022). In the context of forest fires, studies directly comparing these two architectures are still ongoing. In addition, many previous studies have only used existing public datasets, which often lack diverse data representation, especially in terms of geographical conditions, vegetation types, and visual confounding elements such as fog or sunlight reflection (El-Madafri et al., 2023). Therefore, this study attempts to fill this gap by presenting an empirical analysis comparing MobileNetV3 and ResNet50 in forest fire image classification using a combination of local and public datasets.

METHOD

MobileNet

MobileNet is a Convolutional Neural Network (CNN) architecture designed to deliver high performance with optimal computational efficiency. This model was introduced by (A. G. Howard et al., 2017) and further developed into MobileNetV2 and MobileNetV3, with a focus on use in power-limited devices such as mobile phones, IoT, and drones. The main advantage of MobileNet lies in its use of the Depthwise Separable Convolution technique, which separates the convolution process into two stages: depthwise convolution to extract spatial features, and pointwise convolution to combine features between channels (A. G. Howard et al., 2017). This strategy significantly reduces the number of parameters and computational operations compared to standard convolution, without a significant decrease in accuracy. Thus, MobileNet has become a popular choice for real-time vision applications that require high efficiency, such as object detection and image classification on resource-constrained devices.

The MobileNetV3 version is an improvement on the two previous generations, combining Neural Architecture Search (NAS) (A. Howard et al., 2019) and Squeeze-and-Excitation (SE) blocks (Hu et al., 2018) to enhance the representation of important features. This model comes in two variants, MobileNetV3-Large for higher accuracy and MobileNetV3-Small for maximum computational efficiency. In addition, the use of Hard-Swish (h-swish) activation replaces the traditional ReLU function and has been proven to improve non-linearity without adding significant computational load. The combination of network structure optimization, new activation functions, and SE blocks enables MobileNetV3 to achieve an ideal balance between accuracy and speed. Therefore, this model is widely used in various research and practical implementations, including in deep learning-based forest fire early detection systems.

1. Standard Convolution (As a Comparison)

In standard convolution, the number of operations (FLOPs) required is calculated as:

$$C_{standart} = D_k \times D_k \times M \times N \times D_f \times D_f \quad (1)$$

Where:
D_K = kernel size (e.g., 3 for a 3×3 kernel),
M = number of input channels,
N = number of output channels,
D_F = size of the output feature map (image height/width).

This formula shows that standard convolution has high complexity because each output channel is connected to all input channels.

2. Depthwise Separable Convolution (Key Concepts of MobileNet)

MobileNet replaces standard convolutions with two separate operations:
(a) Depthwise Convolution
Performed one kernel per input channel:

$$C_{depthwise} = D_K \times D_k \times M \times D_F \times D_F \quad (2)$$

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Issue 4, October 2025
DOI : https://doi.org/10.33395/sinkron.v9i4.15415
e-ISSN : 2541-2019
p-ISSN : 2541-044X

(b) Pointwise Convolution (1×1)
Combining depthwise results using convolution 1×1:

$$C_{pointwise} = M \times N \times D_F \times D_F \quad (3)$$

The total number of MobileNet operations is:

$$C_{MobileNet} = D_K^2 \times M \times D_F^2 + M \times N \times D_F^2 \quad (4)$$

3. Computational Efficiency Ratio
The efficiency comparison between MobileNet and standard convolutions is calculated as:

$$R = \frac{C_{MobileNet}}{C_{standart}} = \frac{1}{N} + \frac{1}{D_K^2} \quad (5)$$

For the kernel 3×3 ($D_K = 3$), value $R \approx 0.11$, This means that MobileNet is 8–9 times more efficient than conventional convolutions — with very little loss in accuracy.

4. Depth Multiplier (α)
MobileNet adds the α (alpha) to adjust the number of input and output channels, allowing the model to be tailored to the device's requirements.

$$M' = \alpha \times M, N' = \alpha \times N \quad (6)$$

Then the number of convolution operations becomes:

$$C_{MobileNet-\alpha} = D_K^2 \times \alpha M \times D_F^2 + \alpha^2 M \times N \times D_F^2 \quad (7)$$

The smaller the value of $\alpha (0 < \alpha \leq 1)$, the lighter the model, but its accuracy may decrease.

5. Resolution Multiplier (ρ)
In addition to α, MobileNet also introduces ρ (rho) to adjust the image input resolution:

$$D_F{}' = \rho \times D_F \quad (8)$$

So the total number of MobileNet convolution operations becomes:

$$C_{total} = D_K^2 \times \alpha M \times (\rho D_F)^2 + \alpha^2 M \times N \times (\rho D_F)^2 \quad (9)$$

The parameter ρ typically has a value of 1, 0.75, 0.5, or 0.25, where the smaller the ρ, the faster the inference process, but the less visual detail in the image.
Activation and Normalization Functions
Each convolution block is followed by:

$$y = ReLU6(BN(C_{MobileNet}(x))) \quad (10)$$

Where:
BN = Batch Normalization untuk training stabilization,
$ReLU6$ = The activation function is limited to the range [0, 6] to be more efficient for mobile/edge devices.

Mathematical Conclusions of MobileNet
In general, MobileNet operations can be summarized as follows:

$$y = ReLU6(BN(PointwiseConv(DepthwiseConv(x)))) \quad (11)$$

This approach enables MobileNet to achieve high computational efficiency while maintaining competitive accuracy, making it ideal for real-time vision systems such as early forest fire detection.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Issue 4, October 2025
DOI : https://doi.org/10.33395/sinkron.v9i4.15415
e-ISSN : 2541-2019
p-ISSN : 2541-044X

ResNet

ResNet50 (Residual Network 50 layers) is one of the Convolutional Neural Network (CNN) architectures developed by (He et al., 2016) and has become an important milestone in the evolution of deep learning. Before the emergence of ResNet, training very deep networks (more than 20–30 layers) often encountered vanishing gradient and accuracy degradation problems, where adding layers actually caused the model's performance to decline (He et al., 2016). ResNet overcomes this problem through the concept of residual learning, which involves adding shortcut connections that allow gradients to flow directly to the previous layer without having to pass through all the convolutional layers in between. This approach enables very deep networks—such as ResNet50 with 50 layers—to be trained stably, converge quickly, and produce high accuracy in image classification tasks.

The ResNet50 architecture consists of one initial convolutional layer followed by four main residual blocks, with a total of 50 layers including Conv layers, Batch Normalization, ReLU activation, and a fully connected layer at the end. Each residual block has a basic structure called an identity block and a convolutional block, both of which use a skip connection in the form of the formula $(x)+x$, where $F(x)$ is a non-linear transformation of the input x (He et al., 2016). In this way, the network only needs to learn the residual function, not the direct function from input to output, making it more efficient in feature learning. ResNet50 has proven to be superior in various image recognition tasks such as ImageNet classification (Krizhevsky et al., 2012), object detection, and medical imaging, due to its ability to extract deep features without losing important information from the early layers. This model has become a widely used baseline backbone in various modern deep learning architectures, including YOLO and Faster R-CNN.

1. Basic Concepts of Residual Learning

ResNet introduces the idea that layers in a network do not need to directly learn the target function $H(x)$, but rather only need to learn the residual function defined as:

$$F(x) = H(x) - x \quad (1)$$

Thus, its original function can be rewritten as:

$$H(x) = F(x) + x \quad (2)$$

Description:

x : input from residual block.

$F(x)$: non-linear transformation resulting from multiple layers of convolution and activation

$H(x)$: target output of that block

2. Forward Process on Residual Block

The forward propagation process in a single residual block can be expressed as:

$$y = F(x, \{W_i\}) + x \quad (3)$$

where :

y = output from the residual block

x = input to that block

$F(x, \{W_i\})$ = the result of a series of convolution operations ($Conv \rightarrow BN \rightarrow ReLU$) with weight parameters W_i

If the input and output dimensions are different, a projection shortcut with 1×1 convolution is used, so that the formula becomes:

$$y = F(x, \{W_i\}) + W_{sx} \quad (4)$$

where Ws is the weight of the 1×1 shortcut convolution that equalizes the channel dimensions.

3. Backpropagation Process in the Residual Block.

The advantage of ResNet lies in the ease of gradient flow. The derivative of the loss function L with respect to the input x in the residual block is:

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial y} \cdot \frac{(1+\partial F(x)}{\partial (x)} \quad (5)$$

Because there is a “1” component of the shortcut connection, the gradient can flow directly to the previous layer without experiencing a vanishing gradient.

This is why ResNet is capable of training very deep networks (up to hundreds of layers).

4. ResNet50 structure (bottleneck block)

ResNet50 uses a bottleneck block structure for efficiency, consisting of three convolutional layers.:

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Issue 4, October 2025
DOI : https://doi.org/10.33395/sinkron.v9i4.15415
e-ISSN : 2541-2019
p-ISSN : 2541-044X

$$F(x) = W_3\sigma(BN(W_2\sigma(BN(W_1x)))) \quad (6)$$

where:
W_1: convolution 1×1 (channel dimension reduction)
W_2: convolution 3×3 (main feature extraction)
W_3: convolution 1×1 (restoring the number of channels)
σ: ReLU activation function
BN: Batch Normalization

So that the final output becomes:

$$y = F(x) + x \quad (7)$$

5. Final Activation Function

After each residual block, the results are passed through the ReLU activation function to introduce non-linearity:

$$Output = ReLU(F(x) + x) \quad (8)$$

Mathematical Conclusions

The ResNet50 function series can generally be described as follows:

$$yL = x + \sum_{i=1}^{L} F_i(x_i, W_i) \quad (9)$$

where L is the total number of residual blocks. With this approach, ResNet maintains gradient stability while improving the network's ability to learn complex features efficiently.

C. Data Collection and Dataset Description

Fig. 1 Image Dataset

Figure 1 presents a sample of augmented images used in the forest fire classification dataset, divided into two main categories: fire and non-fire. The fire class contains images showing visible flames, smoke, and orange-red color dominance representing active wildfire scenes. Meanwhile, the non-fire class includes images of forests, rivers, and landscapes without any visible fire or smoke, serving as negative samples for classification. Various augmentation techniques—such as rotation, brightness adjustment, zooming, and flipping—were applied to both classes to enhance data diversity and improve model robustness in recognizing fires under different environmental and lighting conditions.

The dataset used in this study is a combination of local field images and a public forest fire dataset, aiming to obtain data that represent real field conditions while enhancing the model's generalization capability. The local

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Issue 4, October 2025
DOI : https://doi.org/10.33395/sinkron.v9i4.15415
e-ISSN : 2541-2019
p-ISSN : 2541-044X

dataset was collected from the Puncak region, Bogor, Indonesia, using drone cameras and smartphone cameras under various lighting conditions, levels of smoke density, and vegetation types. A total of 500 local images were collected, consisting of 250 forest fire images and 250 non-fire images (normal forest areas without fire or smoke). These local images represent the visual characteristics of tropical forest fires in Indonesia, providing realistic context for model training and evaluation.

In addition to the local data, this study also utilized a public dataset sourced from the Wildfire Smoke Dataset (El-Madafri et al., 2023), which contains approximately 2,700 annotated images. The public dataset includes various wildfire conditions from different parts of the world, covering variations in fire intensity, smoke density, visibility range, and environmental background. To ensure data quality, a data cleaning process was performed by removing duplicate, corrupted, and irrelevant images. Furthermore, class balancing was applied to maintain a proportional distribution between fire and non-fire images across the entire dataset.

All images underwent a preprocessing stage, which included resizing to 224×224 pixels, normalizing pixel values to the [0,1] range, and applying data augmentation techniques to improve model robustness. The augmentation techniques used included random rotation, horizontal flipping, brightness adjustment, and moderate zooming to simulate diverse visual conditions without increasing the actual dataset size. This process aimed to strengthen the model's resilience against variations in lighting, camera angles, and fire shapes encountered in real-world scenarios.

The combined dataset of 500 local images and 2,700 public images, resulting in a total of 3,200 images, was split using a stratified sampling method to maintain balanced class distribution. The data were divided into 70% for training, 15% for validation, and 15% for testing. The training set was used to train the model, the validation set was employed to monitor model performance and prevent overfitting, while the testing set was used to evaluate the model's generalization ability on unseen data.

This dataset integration approach ensures that the MobileNetV3 (Large and Small) and ResNet50 models are fairly tested in scenarios that reflect both real-world field conditions and global variations. Consequently, the comparative performance analysis of these architectures provides comprehensive insights into the balance between accuracy, computational efficiency, and generalization capability in forest fire image classification tasks.

RESULT

MobilenetV3Large

Fig. 2 Training MobileNetV3Large

Figure 2 shows the training process of the MobileNetV3-Large model until it reaches 100 epochs with relatively stable results in each iteration. The training accuracy value ranges from 0.62 to 0.64, while the validation accuracy is around 0.61, indicating that the model is able to maintain consistent performance without significant overfitting. The loss and validation loss values also show good convergence in the range of 0.63, indicating that the optimization process is running effectively and the model has reached a stable point. Overall, these results show that MobileNetV3-Large successfully achieves a balance between accuracy and computational efficiency during the training process for forest fire image classification.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Issue 4, October 2025
DOI : https://doi.org/10.33395/sinkron.v9i4.15415
e-ISSN : 2541-2019
p-ISSN : 2541-044X

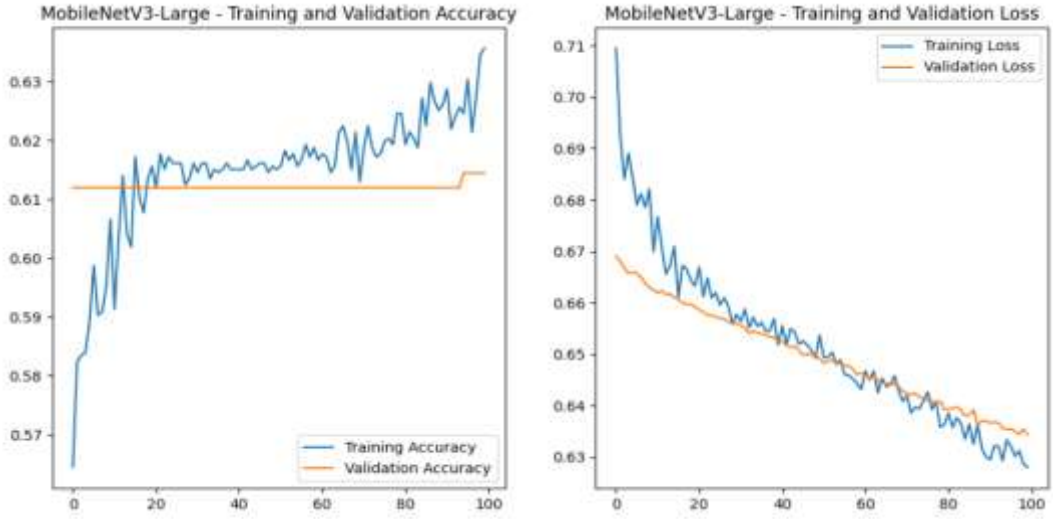

Fig. 3 Training Graft MobileNetV3Large

Figure 3 shows a graph of the MobileNetV3-Large model training results, illustrating the development of accuracy and loss in the training and validation data over 100 epochs. The graph on the left shows a gradual increase in training accuracy, reaching a value of around 0.63, while validation accuracy remains stable at around 0.61, indicating that the model has fairly good generalization capabilities. On the other hand, the graph on the right shows a consistent decrease in loss values for both training and validation data, indicating that the learning process is effective and convergent. Overall, these results indicate that MobileNetV3-Large is able to achieve a balance between accuracy and efficiency with stable performance without significant signs of overfitting.

MobilenetV3Small

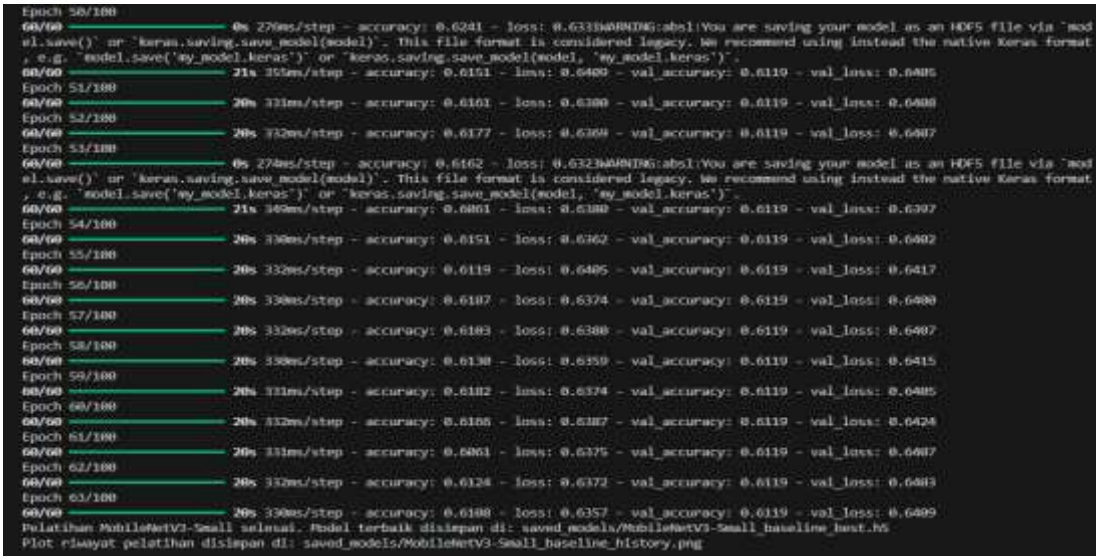

Fig. 4 Training MobileNetV3Small

Figure 4 shows the training process of the MobileNetV3-Small model until it reaches 100 epochs with relatively stable results throughout the training. The training accuracy value ranges from 0.61 to 0.62, while the validation accuracy is constant at around 0.61, indicating that the model is at a consistent convergence point. The loss and validation loss values also gradually decrease until they stabilize at around 0.64, indicating that the learning process has achieved a balance between training and validation. Overall, these results show that MobileNetV3-Small has efficient and stable performance, even though its accuracy is lower than larger variants or models with deeper architectures.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Issue 4, October 2025
DOI : https://doi.org/10.33395/sinkron.v9i4.15415
e-ISSN : 2541-2019
p-ISSN : 2541-044X

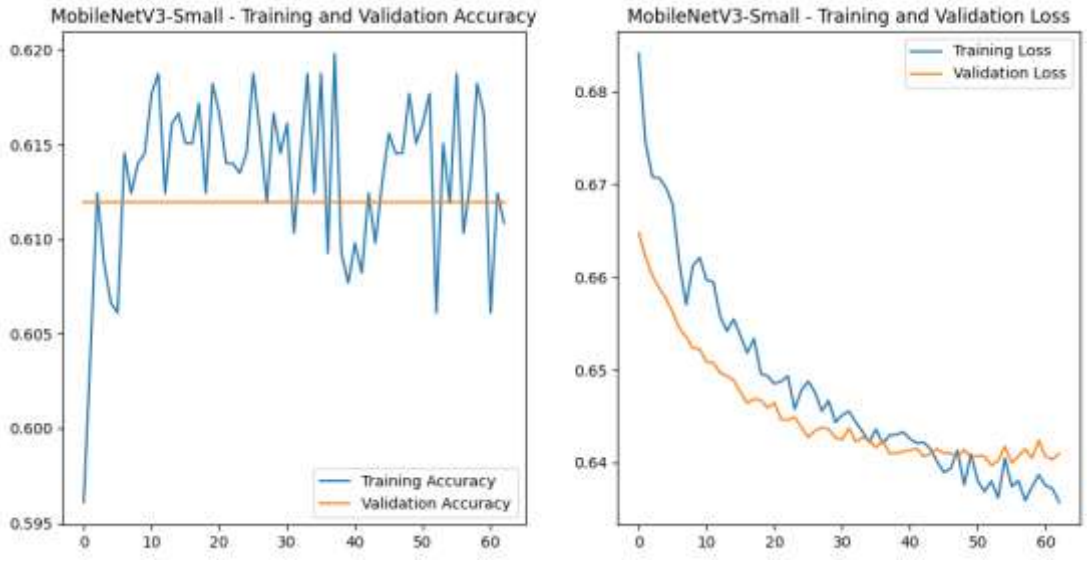

Fig. 5 Training Graft MobileNetV3Small

Figure 5 shows a graph of the MobileNetV3-Small model training results, illustrating the development of accuracy and loss in the training and validation data during the training process. The graph on the left shows that the training accuracy fluctuates in the range of 0.61 to 0.62, while the validation accuracy is relatively stable at around 0.61, indicating that the model has reached a point of convergence without significant improvement. On the other hand, the graph on the right shows a consistent decrease in loss values for both training and validation data until both reach stability at around 0.64. Overall, these results show that MobileNetV3-Small is able to learn efficiently and stably despite its lower architectural complexity, making it suitable for implementation on devices with computational limitations.

Table 1 shows the training performance results of the three deep learning models used in this study, namely ResNet50, MobileNetV3-Large, and MobileNetV3-Small. Based on accuracy and loss values, the ResNet50 model obtained the best results with a training accuracy of 0.677 and a validation accuracy of 0.647, accompanied by the lowest loss value among the three. Meanwhile, MobileNetV3-Large showed moderate performance with a training accuracy of 0.635 and a validation accuracy of 0.614, while MobileNetV3-Small produced the lowest accuracy but remained stable with high efficiency. Overall, this table indicates a trade-off between accuracy and computational efficiency, where ResNet50 excels in accuracy while MobileNetV3 offers better efficiency for implementation on devices with limited resources.

ResNet50

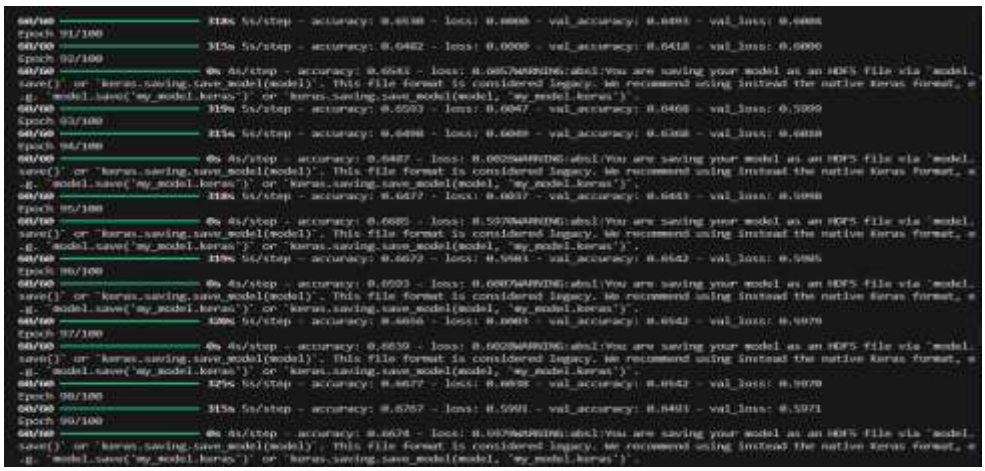

Fig. 6 Training ResNet50.

Figure 6 shows the deep learning model training process running until it reaches the 100th epoch. The accuracy and validation accuracy values consistently range from 0.64 to 0.67, indicating that the model has reached stability

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Issue 4, October 2025
DOI : https://doi.org/10.33395/sinkron.v9i4.15415
e-ISSN : 2541-2019
p-ISSN : 2541-044X

without significant improvement at the end of training. The loss and validation loss values are also relatively constant at around 0.59 to 0.60, indicating that the model does not experience severe overfitting. In addition, there is a warning from Keras suggesting the use of the new .keras model storage format instead of the old .h5 format to be more in line with current model storage standards.

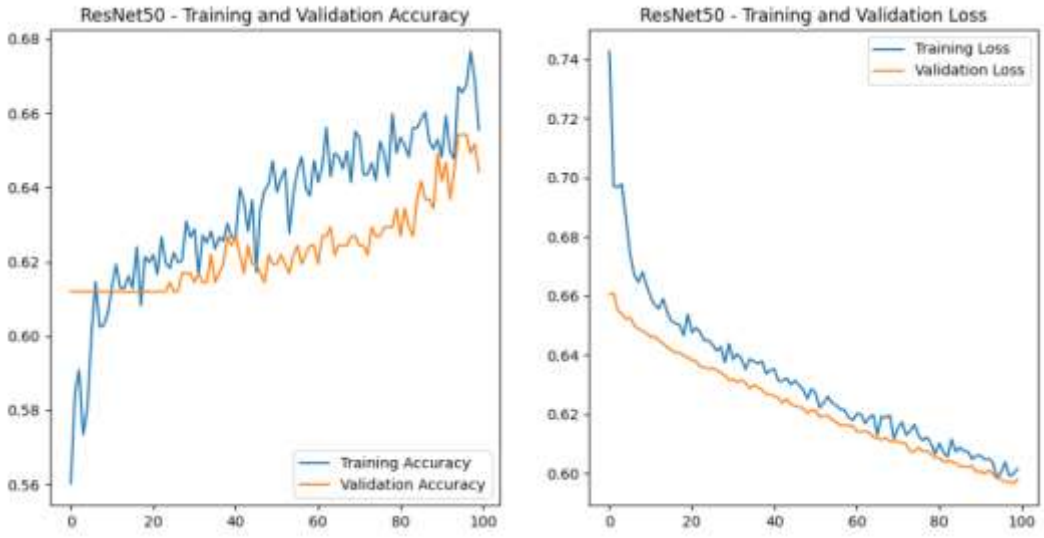

Fig. 7 Training Graft ResNet50

Figure 7 shows the training results of the ResNet50 model, illustrating the relationship between accuracy metrics and loss in the training and validation data over 100 epochs. The graph on the left shows a gradual increase in accuracy in both the training and validation data, with the accuracy value reaching around 0.67 at the end of training, indicating that the model was able to learn steadily without significant signs of overfitting. Meanwhile, the graph on the right shows a consistent decrease in loss values for both training and validation data, which means that the optimization process has successfully reduced the model's prediction errors gradually. The balanced movement pattern between the training and validation curves confirms that the ResNet50 model has good generalization capabilities for new data in forest fire image classification tasks.

Table 1 Performance Training Model

Method	Accuracy	loss	Val_accuracy	Val_loss
ResNet50	0.677	0.59	0.647	0.697
MobilenetV3Large	0.635	0.628	0.614	0.635
MobilenetV3Small	0.61	0.63	0.61	0.64

Table 1 shows the training performance results of the three deep learning models used in this study, namely ResNet50, MobileNetV3-Large, and MobileNetV3-Small. Based on accuracy and loss values, the ResNet50 model obtained the best results with a training accuracy of 0.677 and a validation accuracy of 0.647, accompanied by the lowest loss value among the three. Meanwhile, MobileNetV3-Large showed moderate performance with a training accuracy of 0.635 and a validation accuracy of 0.614, while MobileNetV3-Small produced the lowest accuracy but remained stable with high efficiency. Overall, this table indicates a trade-off between accuracy and computational efficiency, where ResNet50 excels in accuracy while MobileNetV3 offers better efficiency for implementation on devices with limited resources.

DISCUSSIONS

The training results show that the ResNet50, MobileNetV3-Large, and MobileNetV3-Small models have different performance characteristics, reflecting a trade-off between accuracy and computational efficiency. ResNet50 provides relatively high and stable accuracy at each epoch, demonstrating the architecture's ability to extract complex visual features through residual learning mechanisms. However, this model requires longer training time and greater computational resources, making it less ideal for application on devices with limited power. On the other hand, MobileNetV3-Large is able to achieve accuracy close to ResNet50 with a much lighter computational load, proving the effectiveness of its architectural optimization through depthwise separable

*name of corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

convolution and squeeze-and-excitation block techniques. Meanwhile, MobileNetV3-Small, despite having slightly lower accuracy, still shows stable and efficient performance in terms of memory usage and training time. This difference confirms that architectural efficiency does not always mean a significant decrease in accuracy, especially when the model is used with a transfer learning approach. Thus, MobileNetV3 can be considered the optimal solution for implementing fire detection systems on edge devices or in environments with limited computing power.

In addition, the experimental results show that models tested in a CPU-only environment have limitations in training speed compared to other research reports that use GPUs. This reinforces the importance of real-device testing, as emphasized in the introduction, to assess the suitability of models in actual field conditions. The use of a combined dataset of local and public images also shows that model performance can vary depending on lighting conditions, background, and smoke texture. ResNet50 proved to be more sensitive to environmental variations due to its high layer complexity, while MobileNetV3 showed better generalization capabilities on heterogeneous data. This shows that lightweight architecture designs such as MobileNetV3 have advantages not only in efficiency but also in adaptability to real-world conditions in the field. In other words, this study proves that model selection must consider the operational context and hardware limitations in addition to accuracy aspects. Overall, the results of this discussion address the main research problem by providing a comprehensive overview of the balance between accuracy, efficiency, and generalization capabilities in deep learning-based forest fire detection.

CONCLUSION

Indonesia faces recurring threats of forest and land fires every year, making early detection a crucial step in disaster mitigation and control. The main problem in developing a computer vision-based detection system is finding a balance between model accuracy and computational efficiency so that it can be implemented on devices with limited resources. This study proposes a comparative approach to three deep learning architectures—ResNet50, MobileNetV3-Large, and MobileNetV3-Small—using a combined dataset that includes local and public images to improve model generalization to real-world conditions in the field. Through transfer learning methodology and CPU-only testing, this study successfully describes model performance in a realistic context without GPU accelerator support. Experimental results show that ResNet50 achieves the highest accuracy but with longer training and inference times, while MobileNetV3-Large and Small provide better computational efficiency with minimal accuracy degradation. Further analysis shows that MobileNetV3 has an optimal balance between speed, resource consumption, and generalization ability to fire image variations. These findings confirm that lightweight models can be a viable alternative for implementing edge computing-based forest fire detection systems. Overall, this research contributes to the literature by providing empirical evidence that deep learning architecture efficiency does not have to sacrifice accuracy, and the results can form the basis for practical recommendations for developers and environmental monitoring agencies in selecting appropriate models for early detection of forest fires in Indonesia.

REFERENCES

Akagic, A., & Buza, E. (2022). LW-FIRE: A Lightweight Wildfire Image Classification with a Deep Convolutional Neural Network. *Applied Sciences (Switzerland)*, *12*(5). https://doi.org/10.3390/app12052646

Barmpoutis, P., Papaioannou, P., Dimitropoulos, K., & Grammalidis, N. (2020). A review on early forest fire detection systems using optical remote sensing. *Sensors*, *20*(22), 6442. https://doi.org/10.3390/s20226442

El-Madafri, I., Peña, M., & Olmedo-Torre, N. (2023). The Wildfire Dataset: Enhancing Deep Learning-Based Forest Fire Detection with a Diverse Evolving Open-Source Dataset Focused on Data Representativeness and a Novel Multi-Task Learning Approach. *Forests*, *14*(9). https://doi.org/10.3390/f14091697

Frizzi, S., Kaabi, R., Bouchouicha, M., Ginoux, J.-M., Moreau, E., & Fnaiech, F. (2016). Convolutional neural network for video fire and smoke detection. *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*, 877–882. https://doi.org/10.1109/IECON.2016.7793196

Guede-Fernández, F., Martins, L., de Almeida, R. V., Gamboa, H., & Vieira, P. (2021). A deep learning based object identification system for forest fire detection. *Fire*, *4*(4), 1–17. https://doi.org/10.3390/fire4040075

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778. https://doi.org/10.1109/CVPR.2016.90

Hindarto, D. (2023). Comparison of Detection With Transfer Learning Architecture Restnet18, Restnet50,

*name of corresponding author

Restnet101 on Corn Leaf Disease. *Jurnal Teknologi Informasi Universitas Lambung Mangkurat (JTIULM)*, *8*(2), 41–48. https://doi.org/10.20527/jtiulm.v8i2.174

Hindarto, D. (2024). *Journal of Computer Networks , Architecture and High Performance Computing Comparison Accuracy of CNN and VGG16 in Forest Fire Identification : A Case Study Journal of Computer Networks , Architecture and High Performance Computing*. *6*(1), 137–148. https://doi.org/10.47709/cnahpc.v6i1.3371

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. https://doi.org/10.48550/arXiv.1704.04861

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., & others. (2019). Searching for mobilenetv3. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1314–1324. https://doi.org/10.1109/ICCV.2019.00140

Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7132–7141. https://doi.org/10.1109/CVPR.2018.00745

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, *25*. https://doi.org/10.1145/3065386

Muhammad, K., Ahmad, J., & Baik, S. W. (2018). Early fire detection using convolutional neural networks during surveillance for effective disaster management. *Neurocomputing*, *288*, 30–42. https://doi.org/10.1016/j.neucom.2017.04.083

Shi, G. X., Wang, Y. N., Yang, Z. F., Guo, Y. Q., & Zhang, Z. W. (2024). Wildfire Identification Based on an Improved MobileNetV3-Small Model. *Forests*, *15*(11). https://doi.org/10.3390/f15111975

Vdovjak, K., Maric, P., Balen, J., Grbic, R., Damjanovic, D., & Arlovic, M. (2022). *Modern CNNs Comparison for Fire Detection in RGB Images*. 239–254.

Zhao, Y., Ma, J., Li, X., & Zhang, J. (2018). Saliency detection and deep learning-based wildfire identification in uav imagery. *Sensors (Switzerland)*, *18*(3). https://doi.org/10.3390/s18030712

Zheng, H., Duan, J., Dong, Y., & Liu, Y. (2023). Real-time fire detection algorithms running on small embedded devices based on MobileNetV3 and YOLOv4. *Fire Ecology*, *19*(1). https://doi.org/10.1186/s42408-023-00189-0

*name of corresponding author