Accredited SINTA 2 Ranking
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online at: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 8 No. 6 (2024) 701 - 709 e-ISSN: 2580-0760

Performance Analysis of MobileNetV3-based Convolutional Neural Network for Facial Skin Disorder Classification

Herimanto[1]*, Arie Satia Dharma[2], Junita Amalia[3], David Largo[4], Christin Adelia Pratiwi Sihite[5]
[1,2,3,4,5]Informatics, Faculty of Informatics and Electrical Engineering, Institut Teknologi Del, Toba
pardzheri@gmail.com[1], ariesatia@del.ac.id[2], junita.amalia@del.ac.id[3], dlargo02032002@gmail.com[4],
christinsihite2000@gmail.com[5]

Abstract

Accurately identifying facial skin types is essential for recommending the right skincare treatments and products. Misidentifying skin types can lead to negative consequences, such as irritation or worsening of skin conditions. This study investigated methods for classifying facial skin types into five categories: oily, acne-prone, dry, normal, and combination. A dataset of 1725 augmented facial images was used. Data augmentation techniques likely increased the dataset's diversity, which helps improve the model's generalization ability. The data underwent preprocessing, including rescaling, before being applied to two deep learning models, CNN and MobileNetV3. The models were evaluated based on accuracy and execution time to determine the most effective approach for classifying facial skin types. The CNN model achieved an accuracy of 64%, demonstrating its potential for image classification tasks. However, the MobileNetV3 model significantly outperformed CNN with an accuracy of 84%. This superior performance is attributed to MobileNetV3's advanced architecture, which is optimized for efficient feature extraction, and particularly relevant for capturing the subtle variations in facial skin types. Therefore, MobileNetV3 emerged as the more effective method for classifying facial skin types with higher accuracy.

Keywords: *CNN; MobileNetV3; Preprocessing; Deep Learning*

How to Cite: Herimanto, Arie Satia Dharma, Junita Amalia, David Largo, and Christin Adelia Pratiwi Sihite, "Performance Analysis of MobileNetV3-based Convolutional Neural Network for Facial Skin Disorder Classification", *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 8, no. 6, pp. 701 - 709, Dec. 2024.
DOI: https://doi.org/10.29207/resti.v8i6.5982

1. Introduction

The skin is the outermost organ of the human body, functioning as a protective barrier for other body parts. Every individual has different skin conditions influenced by factors such as the environment, immune system, chemicals, and physical attributes [1], [2], [3]. Facial skin is the most sensitive compared to other body areas, making it more prone to problems. Inappropriate skincare can lead to facial skin issues [2], [4]. In the context of skincare, understanding facial skin types is crucial for addressing skin problems and selecting suitable products [3]. [4]. Therefore, a classification system that can help in identifying facial skin types through image recognition is necessary. This classification approach can assist people in recognizing their skin type, saving costs, time, and effort, and eliminating the need for direct consultations with dermatologists [4], [5].

One successful classification approach is using Artificial Neural Networks (ANN), further developed into Deep Learning [6]. This deep learning application is ideal for image classification in computer vision tasks [7], which involves identifying and categorizing objects in images/videos into predefined categories. A frequently used method for image data is Convolutional Neural Network (CNN) [8]. CNNs have deep network layers and are widely used in image data, making them popular for object detection. Generally, CNN algorithms consist of three layers: convolutional, pooling, and fully connected layers. Various related studies have shown that CNNs have high accuracy in image processing.

Previous research has utilized CNN algorithms to classify facial skin types into four labels: normal, oily, combination, and dry, based on microscopic images. In this research, CNN was used with convolutional layers to extract features from images. The extracted features

Received: 10-08-2024 | Accepted: 23-10-2024 | Published Online: 24-12-2024

were then sent to pooling layers to reduce data dimensions. Dropout techniques were also used to reduce overfitting, and in the classification stage, a fully connected layer was implemented. Each class was computed using non-linearity with the ReLU activation function, achieving an accuracy of 99.5% [9]. Another study classified facial skin types using the CNN architecture ResNet-50 into three labels: normal, dry, and oily, with a dataset of 1,119 images. The modeling involved preprocessing with resizing and normalization, yielding an accuracy of 99.86% [5].

Besides CNN, MobileNetV3 offers a potential solution for detecting complex images, designed for high computational efficiency [10]. MobileNet is a CNN architecture that uses depthwise separable convolutions, reducing computation compared to standard convolutions. Depthwise separable convolutions work by separating the filtering operation (depthwise convolution) from the combining operation (pointwise convolution) into separate operations, thus performing fewer operations [11]. The main difference between MobileNet and other CNN architectures is the use of convolutional layers with filter thickness appropriate for the input image. MobileNetV3 also includes shortcuts between bottlenecks, enabling faster training with good accuracy [12]. Research conducted on skin segmentation and aging signs classification in 2022 [13], demonstrated that MobileNetV3 outperformed MobileNetV2, NASNet-Mobile, and EfficientNet-B0 with an accuracy of 94%.

Various studies on facial skin classification show the efficacy of CNN and MobileNetV3 in classification. However, a comparison between CNN and MobileNetV3 methods is necessary as they offer different approaches in terms of computational efficiency and accuracy. CNN has shown superior accuracy through various studies but often requires significant computational resources and long training times [14], whereas MobileNetV3 is designed for high computational efficiency with depthwise separable convolutions, allowing for lower computational power usage. In this thesis, the researcher will classify facial skin types into five categories: normal, oily, acne-prone, combination, and dry, comparing the CNN and MobileNetV3 methods. Each classification step will be performed identically for both methods and then compared to determine the best model for facial skin type classification [15]. The comparison will evaluate precision, recall, F1-score, accuracy, and execution time. The best-performing model will be implemented into a mobile application using deep learning technology to classify facial skin types based on directly inputted facial images. This implementation aims to facilitate the determination of skin types, avoiding difficulties and errors in identifying facial skin types.

2. Research Methods

The method for carrying out processing in this study is explained in this session. From dataset preparation, a series of preprocessing stages to clean data from things that interfere with the data processing process, transforming data into the numerical form using the rescaling method to data mining and evaluation. The stages of this study are visualized in Figure 1.

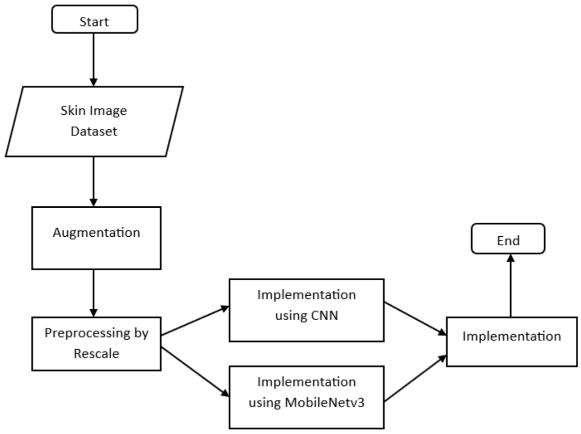

Figure 1. Proposed Method's Step

The method for carrying out the processing in this study is detailed in the following steps. The process begins with the collection of a skin image dataset from various sources, including Kaggle and web scraping from the Bing search engine, utilizing Python libraries. The collected dataset is then augmented to increase its diversity and size, which is crucial for enhancing the model's generalization and performance on unseen data. Following augmentation, the dataset undergoes preprocessing through rescaling to ensure uniformity and optimal input for the models. Subsequently, the preprocessed data is implemented using two different methods: Convolutional Neural Network (CNN) and MobileNetV3. Each method is applied separately to the dataset, and their respective implementations are compared. Finally, the results of both methods are evaluated to determine the best-performing model for facial skin type classification, concluding the process.

2.1. Research Data

The dataset will be collected through two methods: first, using the datasets available on Kaggle, a platform that provides various datasets for data analysis and machine learning. Through Kaggle, the researcher will search for studies whose datasets are related to this final project by entering appropriate search keywords. Second, additional data will be collected through the scraping process from the Bing search engine. This scraping process is conducted to gather relevant and up-to-date data that is not available in the Kaggle datasets. The data collection process using this scraping technique will be done using libraries in the Python programming language.

The facial skin characteristics data used were obtained based on analysis conducted through literature review. Each identified facial characteristic was categorized into the five predetermined skin type labels. These characteristics for each skin type will be used to classify each image dataset obtained through scraping and Kaggle data according to their respective labels. To enhance the validity of the data used, the dataset and label determination were verified directly by dermatologists to improve the accuracy of the data and information used in this study.

The dataset used in this research was collected from facial images of individuals residing in North Sumatra, specifically the Toba Regency, Medan, and the surrounding areas. This dataset will be divided into five folders according to their labels, and each label will contain images as illustrated in Figure 2.

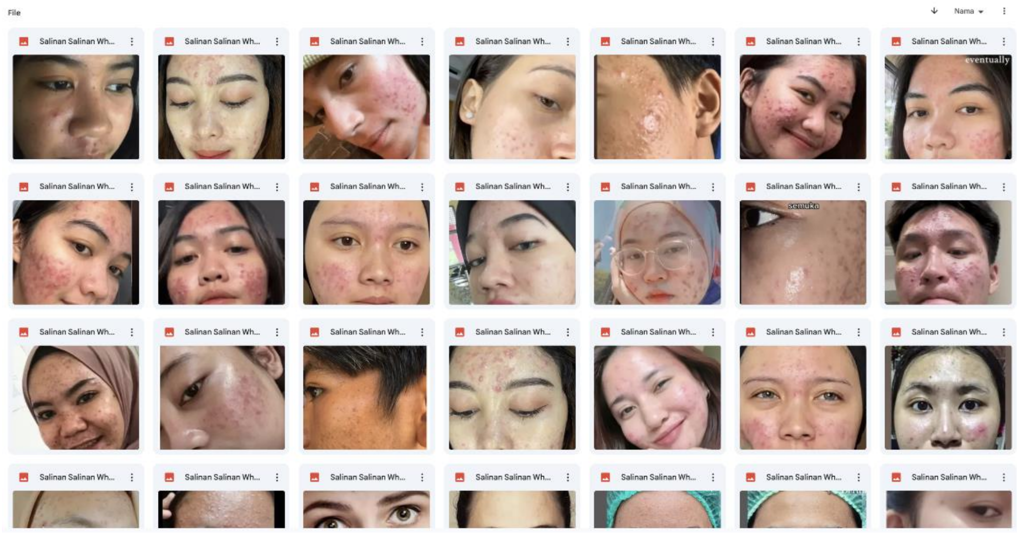

Figure 2. The Dataset

2.2. Convolutional Neural Network (CNN)

The first step is the initialization and initial preparation for the CNN model. This includes defining the model architecture, setting parameters, and loading the dataset. Next is the convolution stage, where the input image matrix is convolved with kernels or filters to extract features from the images. Each kernel will produce a feature map that represents specific information in the image. This convolution process can be repeated to allow the network to understand more complex features as the depth increases, depending on the model being built.

After the convolution process, the result is processed by the ReLU activation function to introduce nonlinearity into the network, allowing it to learn more complex features from the data. ReLU also helps reduce the vanishing gradient problem that often occurs in deep neural networks. The next stage is Max Pooling, which is used to reduce the spatial dimensions of the feature maps produced by convolution. Max pooling selects the maximum value from a specific area to represent the entire area, making the resulting representation more resistant to shifts and helping to reduce overfitting [16].

The repetition of these layers processes lower-level information in the initial layers and higher-level information in the deeper layers. After several convolution and pooling layers, the dimension-reduced data is flattened into a one-dimensional vector to be processed by the fully connected layer, which then takes the learned features to perform classification. With repeated convolution and pooling, CNNs can enhance their capacity to learn increasingly complex features, improving performance in tasks such as image recognition.

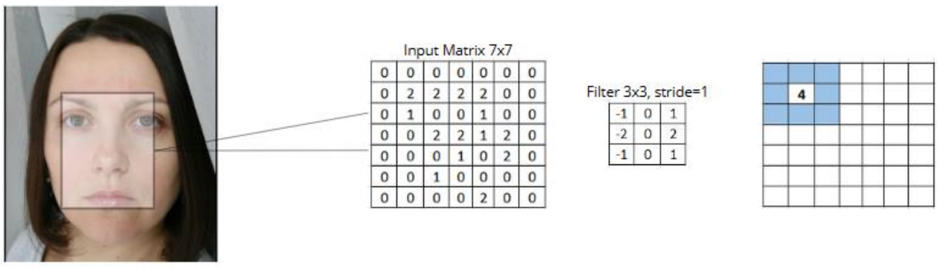

Figure 3. Example of Feature Map

The input is a 7×7 matrix, with a 3×3 kernel and a stride of 1. The calculation performed to obtain the feature map is as follows:

$(0*(-1)) + (0*0) + (0*1) + (0*(-2)) + (2*0) + (2*2) + (0*(-1)) + (1*0) + (0*1) = 4$

The value obtained from this calculation represents the process of extracting a feature map. This value is then used in the activation process with ReLU as shown in Figure 3. ReLU activation involves converting negative values to 0 in the feature map using the ReLU function as shown in Figure 4.

2	2	0	0	0	0	0
5	4	0	1	0	0	0
1	4	0	1	0	0	0
0	4	4	0	1	0	0
0	3	4	0	2	0	0
0	2	1	2	1	4	2
0	1	0	1	0	2	0

Figure 4. Feature Map Result by ReLU

The next step is max pooling, which is performed to extract the maximum value from each region of the feature map obtained after the ReLU convolution as shown in Figure 5.

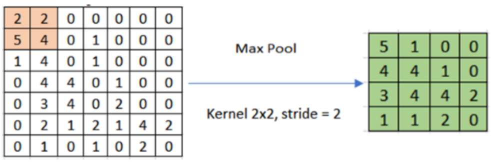

Figure 5. Max Pooling

The next step is to combine each channel from the pooling process and convert the matrix into a one-dimensional vector. This vector will then be mapped to the features of each class. Max pooling is a technique used in CNNs to reduce the dimensionality of the feature map resulting from convolution. This process involves taking the maximum value from each region defined by the pooling operation, which follows the ReLU convolution.

The resulting feature map will then be transformed into a one-dimensional array, a process known as flattening. This flattened vector is then connected to the neurons in the fully connected (dense) layer. During the pooling stage, the feature map is still in 2D form, so it must be converted into a one-dimensional vector or row array to be used in the fully connected layer as shown in Figure 6.

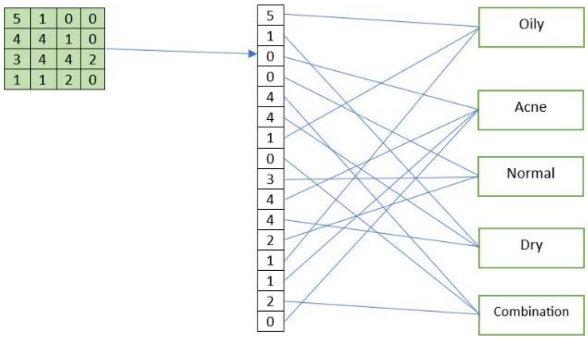

Figure 6. Fully-Connected Layer

The choice of using a row vector simplifies the process for the fully connected layer, where each element of the vector becomes an input to the neurons in the dense layer. Converting to a 1D vector allows the model to learn and connect the local features that have been detected to the desired output classes. The fully connected layer is connected to output neurons representing classes such as oily, acne-prone, normal, dry, and combination. Each neuron in the output layer provides a score indicating how confident the model is that the input belongs to that class. The class with the highest score will be the final prediction of the model.

$$\text{Oily Class} = \frac{e(5+1+1)}{e((5+1+0+0+4+4+1+0+3+4+4+2+1+1+2+0)} = 0.21$$

$$\text{Acne Class} = \frac{e(0+0+4+1)}{e((5+1+0+0+4+4+1+0+3+4+4+2+1+1+2+0)} = 0.14$$

$$\text{Normal Class} = \frac{e(4+0+2)}{e((5+1+0+0+4+4+1+0+3+4+4+2+1+1+2+0)} = 0.18$$

$$\text{Dry Class} = \frac{e(1+4+4)}{e((5+1+0+0+4+4+1+0+3+4+4+2+1+1+2+0)} = 0.28$$

$$\text{Combination Class} = \frac{e(5+1+1)}{e((5+1+0+0+4+4+1+0+3+4+4+2+1+1+2+0)} = 0.21$$

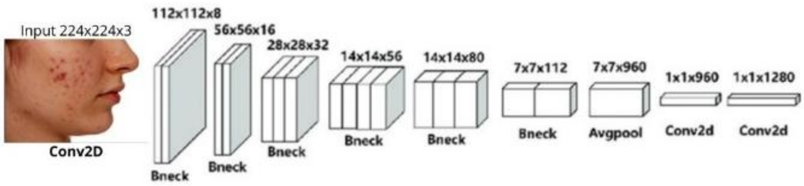

Figure 7. MobileNetV3 architecture

From the calculation results, it is concluded that the class with the highest probability is the Dry class.

An image with dimensions 224×224×3224 \times 224 \times 3224×224×3 (height ×\times× width ×\times× RGB channels) is processed using 2D convolution. Using a 3×33 \times 33×3 filter with a stride of 2 and padding "same" results in an output dimension of 112. Padding "same" means adding extra pixels to the edges of the image to ensure the filter can be applied to the entire image.

Equation 1 is used to calculate the output size,

$$Output = \left(\frac{Input\ size - \text{filter size} + \text{Stride x Padding}}{\text{Stride}}\right) + 1 \quad (1)$$

The output is then processed using 8 filters in the bottleneck (Bneck) layer. Bneck is short for bottleneck layers, which are a core component of the MobileNetV3 architecture. Each bottleneck block typically includes a separate depthwise convolution followed by a pointwise convolution. These bottleneck layers are designed for high computational efficiency and reduced parameter count.

2.3. Optimizer: Adam

In this final research, the modeling process will include adding an optimizer. An optimizer is a method used to reduce the loss value during model training by adjusting the weights [17]. The optimization method used in this study is Adam. Based on tests conducted in CNN model research [18], [19]. To assess the impact of different optimizers on CNN models, it was found that Adam achieved a more optimal accuracy (98.89%) compared to Adamax (88.89%) and RMSprop (96.67%). This testing forms the basis for selecting Adam as the optimizer for this facial skin type classification research.

Adam was chosen because it offers faster and more stable model convergence. Adam combines the advantages of RMSprop (which handles adaptive learning rates) and momentum (which helps speed up convergence). This method is also more stable and reliable in handling varied data and gradient changes during training [19], [20], [21]. Using the Adam optimizer will involve evaluating metrics by measuring accuracy, precision, and recall to provide a comprehensive view of the model's performance for classification [11], [19].

In the CNN and MobileNetV3 model for skin type classification, Adam will be invoked from the Keras library, which is part of TensorFlow. When Adam is called, Keras will use its internal implementation of Adam, containing all necessary logic and algorithms for running the optimizer. In Keras, Adam starts by initializing model parameters and two momentum variables—mean gradient and mean squared gradient—which are initially set to zero. During each iteration, Adam calculates the gradient of the loss function concerning each model parameter, indicating the direction and magnitude by which the model parameters need to be adjusted to reduce loss.

Next, momentum updates are performed. Adam updates the first moment by computing the exponential moving average of the current gradient and the previous first moment, giving insight into the average gradient direction. Adam also updates the second moment by calculating the exponential moving average of the squared current gradient and the previous second moment, providing insight into the average gradient magnitude. Since the first and second moments are initialized to zero, they are biased towards zero in the early iterations. Therefore, Adam performs bias correction for both moments to more accurately represent the true averages. After bias correction, Adam updates each model parameter by decreasing it with a dynamically computed learning rate based on the corrected moments. This means model parameters are updated faster or slower depending on the average gradient magnitude and the average squared gradient magnitude.

3. Results and Discussions

The data collection process resulted in a total of 521 facial skin images, sourced from Kaggle and web scraping. These images were categorized into different skin types: normal, acne, oily, dry, and combination. This dataset was crucial for training the Convolutional Neural Network (CNN) and MobileNetV3 models. Table 1 is the detailed count of collected facial skin images based on skin types.

Table 1. Source Datasets

Skin Type	Count
Normal	129
Oily	111
Acne	127
Dry	88
Combination	66
TOTAL	521

The collected dataset includes images in jpg, jpeg, and png formats. In the initial dataset splitting, the 521 images were divided into training and testing sets with an 80-20 split ratio.

In this research, we conducted several significant experiments to evaluate the performance of MobileNetV3 from multiple perspectives as shown in Figure 7. These experiments included comparing various data augmentation techniques to determine the most effective method for improving accuracy and reducing loss, as well as comparing the performance of CNNs without using the MobileNetV3 architecture based on the confusion matrix. The results of the augmentation techniques experiments are presented in Table 2. Based on Table 2, the results show that the Rescale preprocessing method outperformed other techniques in classifying facial skin types. The Rescale model achieved the lowest Loss (0.2594), highest Accuracy (90.40%), Precision (0.9228), Recall (0.885), and F1-Score (0.9054), indicating superior performance in minimizing prediction errors and achieving a balance between precision and recall. Consequently, the rescale augmentation technique will be adopted for model development.

Table 2. Comparison of Augmentation Techniques

Augmentation Techniques	Model Evaluation				
	Loss	Accuracy	Precision	Recall	F1 Score
Rescale	0.2594	0.9040	0.9228	0.885	0.9054
Rescale & Noise Removal	0.2833	0.8854	0.9013	0.8762	0.8885
Rescale & Greyscale	0.3272	0.9009	0.9172	0.8576	0.8864

The subsequent experiment involves evaluating the model's performance during training, based on accuracy and loss graphs. In the first trial, we tested a CNN algorithm without a specific architecture. The results are depicted in Figure 8.

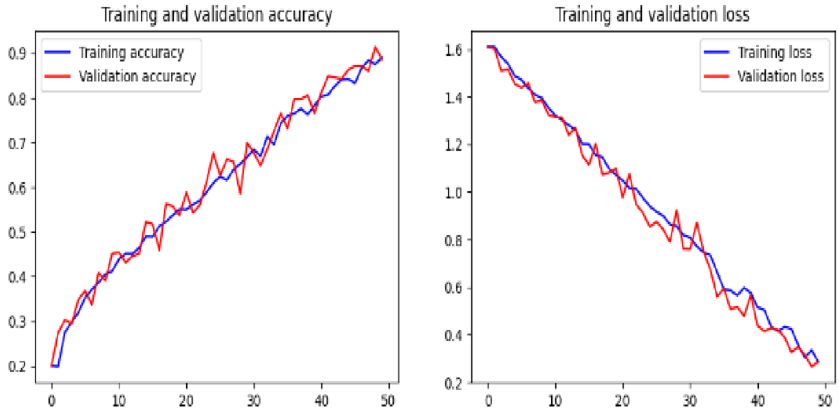

Figure 8. Accuracy and Loss Grafik CNN Model

Figure 8 illustrates the changes in accuracy and loss during the training and validation of the CNN model without a specific architecture over 50 epochs. The left graph shows the training accuracy (blue line) and validation accuracy (red line). Training accuracy gradually increased from approximately 0.2 to nearly 0.9, while validation accuracy also increased in a similar pattern, indicating that the model was able to learn patterns effectively during training. The right graph shows the training loss (blue line) and validation loss (red line). Training loss decreased gradually from around 1.6 to approximately 0.2, while validation loss also showed a similar decrease, indicating that the model became better at minimizing errors.

To further evaluate the performance of the CNN model, the next step involved assessing the model's performance based on evaluation metrics. The results are presented in Table 3.

Table 3. Evaluation Result CNN Model

	Precision	Recall	FI-Score	Support
Acne	0.53	0.90	0.67	10
Oily	0.50	0.50	0.50	10
Dry	0.67	0.80	0.73	10
Combination	1.00	0.50	0.67	10
Normal	1.00	0.90	0.75	10
Accuracy	0.66			50

Based on the evaluation results presented in Table 3, the CNN model without a specific architecture exhibited relatively poor performance, achieving an accuracy of only 0.66 or 66%. This underperformance is further illustrated by the confusion matrix in Figure 9.

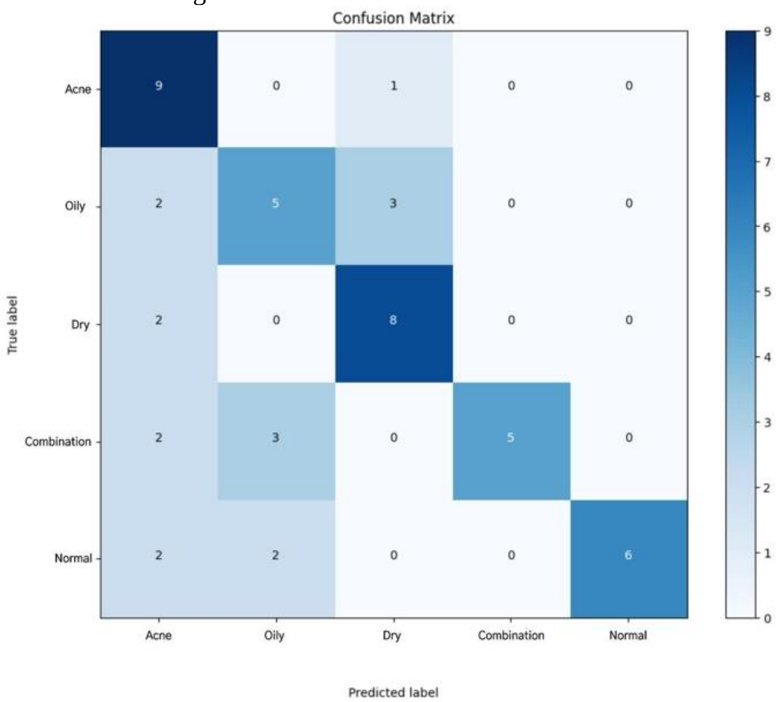

Figure 9. Confusion Matrix Model CNN

The confusion matrix depicted in Figure 9 highlights the CNN model's limited predictive capabilities. The model demonstrates the best performance for the class 'agne', achieving a true positive count of 9.

The subsequent experiment will demonstrate the performance of the MobileNetV3 model when trained using the same dataset and parameters as the CNN model. The loss and accuracy graphs of the MobileNetV3 model are presented in Figure 10.

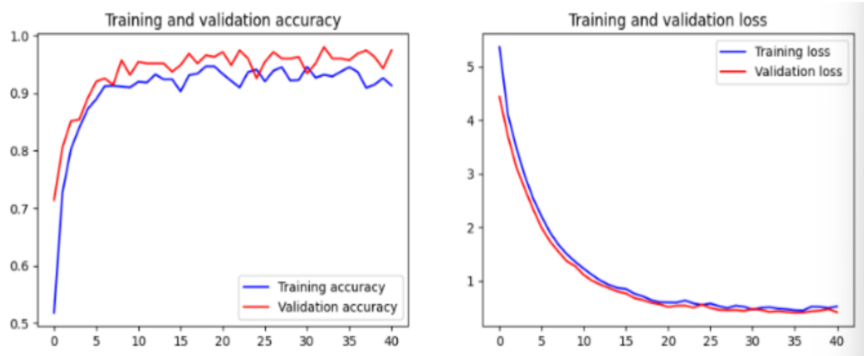

Figure 10. Accuracy and Loss Grafik MobileNetV3 Model

Based on the loss and accuracy graphs in Figure 10, the MobileNetV3 model exhibited rapid initial learning, as evidenced by the swift increase in accuracy and decrease in loss. After approximately 15 epochs, the model reached convergence, indicated by the stabilization of the accuracy and loss curves. This suggests that the model has ceased to make substantial performance improvements. Minor fluctuations in validation accuracy towards the end of training indicate that overfitting is not a significant concern. Overfitting occurs when a model becomes overly specialized to the training data, compromising its ability to generalize to new data. In this case, the MobileNetV3 model appears to maintain a healthy balance between training and validation.

This significant outcome is also evident in the model evaluation during training, as presented in Table 4.

Table 4. Evaluation Result MobileNetV3 Model

	Precision	Recall	FI-Score	Support
Acne	0.91	1.00	0.95	10
Oily	0.73	0.80	0.76	10
Dry	0.83	1.00	0.91	10
Combination	1.00	0.40	0.57	10
Normal	0.83	1.00	0.91	10
Accuracy	0.84			50

Table 4 indicates that the model achieved a satisfactory performance with an accuracy of 84%. This result was obtained using the same dataset and parameters as those used in the CNN model.

Meanwhile, the performance of the MobileNetV3 model in classifying skin disorders is presented in Figure 11.

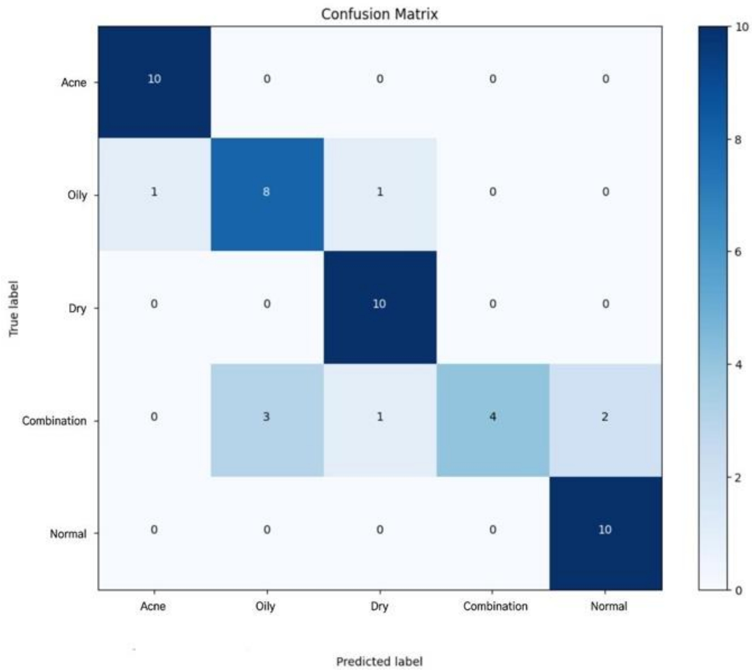

Figure 11. Confusion Matrix Model MobileNetV3

Based on Figure 11, the MobileNetV3 model demonstrates a strong ability to recognize patterns in the training data and apply them effectively to the validation data. The confusion matrix indicates that the MobileNetV3 model performs exceptionally well in classifying oily, dry, and normal skin types, with all samples being correctly classified. However, there are some misclassifications for the oily and combination skin types.

From these results, it can be concluded that MobileNetV3 far outperforms CNN with significant performance. The MobileNetV3 outperformed the CNN with an accuracy of 84%, precision of 0.83, recall of 0.84, and an F1-Score of 0.83.

To facilitate a clearer comparison between the CNN and MobileNetV3 models, the following table presents the evaluation results of both models during training.

Table 5. Comparison Result of CNN and MobileNetV3

Distinguishing Parameters	Model	
	Convolutional Neural Network	MobileNetV3
Precision (%)	0.74	0.86
Recall (%)	0.66	0.84
F1-Score (%)	0.66	0.84
Accuracy (%)	0.66	0.84
Execution Time (s)	9.361	2.071

Based on Table 5, a substantial discrepancy in model evaluation metrics is evident between the CNN and MobileNetV3 models. MobileNetV3 outperforms CNN by a significant margin of 18% in terms of accuracy. This can be attributed to the depthwise separable convolutions employed in MobileNetV3, which significantly reduce the computational burden, making it a more efficient model compared to CNN. Furthermore, the analysis revealed that MobileNetV3 is particularly adept at handling intricate facial images, a result attributed to its architectural design incorporating depthwise separable convolutions and shortcut connections. This combination accelerates the training process without compromising accuracy. Conversely, CNN, being a relatively larger model, may struggle with limited datasets, unlike MobileNetV3, which is specifically designed for resource-constrained environments.

4. Conclusions

The research findings unequivocally demonstrate that MobileNetV3 significantly outperforms the Convolutional Neural Network (CNN) in classifying facial skin types, achieving an accuracy rate of 84% compared to CNN's 66%. Not only is MobileNetV3 more accurate, but it also processes complex facial image data more efficiently. To enhance accessibility, the top-performing MobileNetV3 model has been integrated into an Android-based application, empowering users to capture images, access the camera, and independently classify their facial skin type. Although MobileNetV3 exhibits impressive performance, further research is warranted to optimize its accuracy and efficiency across diverse conditions. Exploring architectural modifications and expanding the dataset could yield promising results.

Acknowledgements

This research is an internal research grant program supported by the Research and Community Service Institution (LPPM) Institut Teknologi Del.

References

[1] M. Ath-Thariq and T. N. Suharsono, "Deteksi Penyakit Kulit Serupa Pada Wajah Berbasis Mobile dengan Metode Convolutional Neural Network," *Innov. J. Soc. Sci. Res.*, vol. 3, no. 5 SE-Articles, pp. 876–887, Oct. 2023, [Online]. Available: https://j-innovative.org/index.php/Innovative/article/view/4936

[2] J. Zdrada, W. Odrzywołek, A. Stolecka-Warzecha, S. Wilczyński, and B. Błońska-Fajfrowska, "The influence of cosmetics dedicated to oily and acne-prone skin on skin parameters.," *J. Cosmet. Dermatol.*, vol. 21, no. 11, pp. 6092–6099, Nov. 2022, doi: 10.1111/jocd.15143.

[3] M. A. Farage, "The Prevalence of Sensitive Skin," *Front. Med.*, vol. 6, 2019, [Online]. Available: https://www.frontiersin.org/journals/medicine/articles/10.3389/fmed.2019.00098

[4] S. D. Kusumaningrum and I. Muhimmah, "Analisis Faktor dan Metode untuk Menentukan Tipe Kulit Wajah: Tinjauan Literatur," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 10, no. 4 SE-Ilmu Komputer, pp. 753–762, Aug. 2023, doi: 10.25126/jtiik.20241046955.

[5] D. A. Agustina, "KLASIFIKASI CITRA JENIS KULIT WAJAH DENGAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN) RESNET-50," *Jurnal Riset Sistem Informasi*, vol. 1, no. 3 SE-Articles. pp. 1–7, Jul. 03, 2024. doi: 10.69714/13sbby24.

[6] R. Nuraini, N. Sadikin, and Y. Rahmanto, "Klasifikasi Citra Jenis Daun Berkhasiat Obat Menggunakan Algoritma Jaringan Syaraf Tiruan Extreme Learning Machine," *J. Comput. Syst. Informatics*, vol. 4, pp. 145–154, Dec. 2022, doi: 10.47065/josyc.v4i1.2586.

[7] A. Peryanto, A. Yudhana, and R. Umar, "Rancang Bangun Klasifikasi Citra Dengan Teknologi Deep Learning Berbasis Metode Convolutional Neural Network," *Format J. Ilm. Tek. Inform.*, vol. 8, p. 138, Feb. 2020, doi: 10.22441/format.2019.v8.i2.007.

[8] M. Efrian and U. Latifa, "IMAGE RECOGNITION BERBASIS CONVOLUTIONAL NEURAL NETWORK (CNN) UNTUK MENDETEKSI PENYAKIT KULIT PADA MANUSIA," *Power Elektron. J. Orang Elektro*, vol. 11, p. 276, Jul. 2022, doi: 10.30591/polektro.v12i1.3874.

[9] S. Saidah, N. Fuadah, F. Alia, N. Ibrahim, R. Magdalena, and S. Rizal, "Facial Skin Type Classification Based on Microscopic Images Using Convolutional Neural Network (CNN)," 2021, pp. 75–83. doi: 10.1007/978-981-33-6926-9_7.

[10] N. Afarini and D. Hindarto, "OPTIMIZATION OF CNN + MOBILENETV3 FOR INSECT IDENTIFICATION: TOWARD HIGH ACCURACY," *J. Teknol. Inf. Univ. Lambung Mangkurat*, vol. 9, pp. 21–28, May 2024, doi: 10.20527/jtiulm.v9i1.199.

[11] X. Chu, B. Zhang, and R. Xu, *MoGA: Searching Beyond MobileNetV3*. 2019. doi: 10.48550/arXiv.1908.01314.

[12] F. Hariz, I. Yulita, and I. Suryana, "Human Activity Recognition Berdasarkan Tangkapan Webcam Menggunakan Metode Convolutional Neural Network (CNN) Dengan Arsitektur MobileNet," *JITSI J. Ilm. Teknol. Sist. Inf.*, vol. 3, pp. 103–115, Dec. 2022, doi: 10.30630/jitsi.3.4.97.

[13] C.-I. Moon and O. Lee, "Skin Microstructure Segmentation and Aging Classification Using CNN-Based Models," *IEEE Access*, vol. PP, p. 1, Dec. 2021, doi: 10.1109/ACCESS.2021.3140031.

[14] E. Budi, A. Chan, P. Alda, and F. Idris, "Optimasi Model Machine Learning untuk Klasifikasi dan Prediksi Citra

Menggunakan Algoritma Convolutional Neural Network," *Resolusi Rekayasa Tek. Inform. dan Inf.*, vol. 4, pp. 502–509, Jun. 2024, doi: 10.30865/resolusi.v4i5.1892.

[15] P. M. B. G. Maia Campos, M. O. Melo, and D. G. Mercurio, "Use of Advanced Imaging Techniques for the Characterization of Oily Skin.," *Front. Physiol.*, vol. 10, p. 254, 2019, doi: 10.3389/fphys.2019.00254.

[16] Y. D. Rosita and L. S. Moonlight, "Perbandingan Metode Prediksi untuk Nilai Jual USD: Holt-Winters, Holt's, dan Single Exponential Smoothing," *JTIM J. Teknol. Inf. dan Multimed.*, vol. 5, no. 4 SE-Articles, Jan. 2024, doi: 10.35746/jtim.v5i4.473.

[17] R. Kurniawan, P. Wintoro, Y. Mulyani, and M. Komarudin, "IMPLEMENTASI ARSITEKTUR XCEPTION PADA MODEL MACHINE LEARNING KLASIFIKASI SAMPAH ANORGANIK," *J. Inform. dan Tek. Elektro Terap.*, vol. 11, Apr. 2023, doi: 10.23960/jitet.v11i2.3034.

[18] A. Rusdy Prasetyo, Sussi, and B. Aditya, "ANALISIS PERBANDINGAN ALGORITMA SUPPORT VECTOR MACHINE (SVM) DAN CONVOLUTIONAL NEURAL NETWORK (CNN) UNTUK SISTEM DETEKSI KATARAK," *J. Ilm. Tek. Mesin, Elektro dan Komput.*, vol. 3, no. 1 SE-Articles, pp. 1–10, Mar. 2023, doi: 10.51903/juritek.v3i1.604.

[19] R.-Y. Sun, "Optimization for Deep Learning: An Overview," *J. Oper. Res. Soc. China*, vol. 8, pp. 1–46, Jun. 2020, doi: 10.1007/s40305-020-00309-6.

[20] R. Cobilla *et al.*, "Classification of the Type of Brain Tumor in MRI Using Xception Model," in *2023 International Conference on Electronics, Information, and Communication (ICEIC)*, 2023, pp. 1–4. doi: 10.1109/ICEIC57457.2023.10049979.

[21] M. Alwanda, R. Ramadhan, and D. Alamsyah, "Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk Pengenalan Doodle," *J. Algoritm.*, vol. 1, pp. 45–56, Oct. 2020, doi: 10.35957/algoritme.v1i1.434.