Journal of Applied Informatics and Computing (JAIC)
Vol.8, No.2, December 2024, pp. 589~597
e-ISSN: 2548-6861

Implementation of MobileNet Architecture for Skin Cancer Disease Classification

Haniifa Aliila Faudyta[1*], Jesica Trivena Sinaga[2*], Egia Rosi Subhiyakto[3]**
[*]Faculty of Computer Science, Universitas Dian Nuswantoro, Semarang, 50131, Indonesia
[**]Research Center for Intelligent Distributed Surveillance and Security (IDSS), Universitas Dian Nuswantoro, Semarang, 50131, Indonesia
111202113820@mhs.dinus.ac.id [1], 111202113808@mhs.dinus.ac.id [2], egia@dsn.dinus.ac.id [3]

Article Info

Article history:

Received 2024-10-28
Revised 2024-11-04
Accepted 2024-11-05

Keyword:

MobileNet,
Architecture,
CNN,
Skin Cancer,
Classification.

ABSTRACT

As the number of occurrences of skin cancer increases year, it becomes more and more crucial to identify the disease accurately and effectively. This study aims to implement and evaluate the MobileNet architecture for classifying nine types of skin lesions using the ISIC 2020 dataset and to compare MobileNet's performance with other CNN architectures, such as VGG-16 and LeNet, in terms of accuracy and computational efficiency. The study also investigates the impact of image preprocessing on model accuracy. The methodology comprises data collection, preprocessing, and model development, leveraging transfer learning from MobileNet pre-trained on ImageNet. Data preprocessing involves resizing images to 224 x 224 pixels and normalizing pixel values. To augment the dataset, techniques such as rotation, zooming, horizontal flipping, and brightness and contrast adjustment are applied. To address class imbalance, oversampling is used to obtain 500 images per class. The model architecture includes Global Average Pooling, a Dense layer with 1024 units and ReLU activation, and a Dropout layer with a 0.2 probability. Various training scenarios with batch sizes (8, 16, 32, 64) and learning rates (0.001, 0.0001) are evaluated, incorporating dropout and ReLU activations. The optimal performance was achieved with oversampling, dropout, and a learning rate of 0.0001, yielding a training accuracy of 99.64% and a validation accuracy of 86.89% after oversampling, resulting in 3,600 training and 900 validation images with an 80:20 data split. The results suggest overfitting due to dataset limitations. Future work should focus on fine-tuning and ensemble methods to improve validation performance.

I. INTRODUCTION

The number of skin cancer cases continues to rise each year, making it one of the most common types of cancer worldwide. The World Health Organization (WHO) says that skin cancer can be successfully treated if diagnosed early [1]. However, issues such as variability and subjectivity in assessment often become problematic with traditional detection approaches that rely on visual examinations by dermatologists [2]. The level of hospital services in Indonesia is relatively low, as indicated by the difficulties faced by the community in accessing medical services at several hospitals [3]. This condition indicates the need for an automated system that can assist in more efficiently identifying and categorizing skin cancer. Additionally, to enhance patient experience and optimize healthcare services, AI has begun to be integrated into digital healthcare services [4].

Advancements in machine learning and AI have improved the accuracy of medical diagnoses. To address image classification problems such as detecting skin cancer, various Deep Neural Network architectures have been used. In this case, MobileNet is a promising model, intended to improve efficiency and performance, especially on devices with limited resources [5]. It is hoped that by utilizing the capabilities of MobileNet, the burden on medical workers will be reduced and diagnosis will be more accessible.

Deep learning methods have been used in several previous studies to investigate skin cancer classification. The first study used a Convolutional Neural Network (CNN) architecture consisting of eight 2D Convolutional layers and successfully classified skin cancer into two categories: malignant and benign. This CNN achieves an accuracy of 75%. This model is more complex than the MobileNet architecture, which is lighter and suitable for devices with limited resources [6].

The second study used a larger dataset to classify eight types of skin cancer. However, the results only achieved an accuracy of 73.22% for VGG-16 and 68.11% for LeNet, which are still considered low accuracy levels. Therefore, this study proposes the use of MobileNet, which is expected to improve accuracy while also providing advantages in terms of model efficiency in classifying nine types of skin cancer [7].

The study objectives included implementing and evaluating MobileNet for skin cancer classification using the ISIC 2020 dataset, as well as comparing MobileNet's performance with other CNN architectures like VGG-16 and LeNet in terms of accuracy and computational efficiency. This research will also investigate the impact of image preprocessing on the model's accuracy.

The study primary goal is to deploy and assess the MobileNet architecture for skin cancer classification using image datasets. Additionally, this study compares the performance of MobileNet by providing several experimental scenarios to assess its effectiveness and efficiency in detecting skin cancer. This research will also investigate the impact of image preprocessing on the model's accuracy.

It is hoped that this study will open our eyes to the potential use of MobileNet in skin cancer diagnosis. The result of this research can help medical professionals improve the speed and accuracy of detection as well as develop user-friendly mobile applications. As a result, this research can help raise awareness of the importance of early detection of skin cancer [8].

Finally, this research is expected to encourage collaboration between technology experts and medical professionals to create innovative solutions in the field of health. By combining expertise in medicine and information technology, we can create diagnostic tools that are not only effective but also sustainable and accessible to everyone [9].

This, this research not only examines the use of MobileNet in skin cancer classification but also aids in the development of automated diagnosis system that can enhance the overall quality of healthcare services.

II. METHOD

In this chapter, the method used in the research to detect and classify skin cancer employs the MobileNet architecture. This approach was chosen because it uses computational resources efficiently and can operate on Mobile devices while still achieving a high degree of image categorization accuracy. The outcomes of validation accuracy, training accuracy, validation loss, and training loss will be used to assess the model [10].

A. Data Collecting

The International Skin Imaging Collaboration (ISIC) 2020 provided the dataset for this study. This is a most recent iteration of the ISIC archive and was chosen because it is a comprehensive and validated source of Dermatoscopic images that can be used for skin cancer research, with 2242 images for the training data from various types of skin lesions and 107 images for the test data.

In this study, the dataset consists of 9 main classes of skin diseases, namely:

TABLE I
SAMPLE CLASSES OF SKIN CANCER

Name	Total Sample
Actinic Keratosis	114
Basal Cell Carcinoma	376
Dermatofibroma	95
Melanoma	438
Nevus	357
Pigmented Benign Keratosis	462
Seborrheic Keratosis	77
Squamous Cell Carcinoma	181
Vascular Lesion	139

Each class has a varying number of samples, with some classes, such as Seborrheic Keratosis and Dermatofibroma, having a relatively small number of samples compared to other classes, which can lead to data imbalance issues. Here are sample images from each class, as shown in Figure 1.

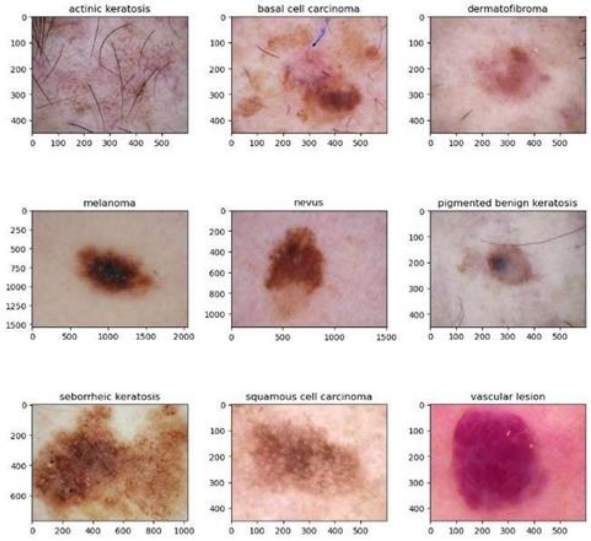

Figure 1. Sample Class Skin Cancer Disease

Apply the oversampling technique to the class with a few samples in order to rectify this imbalance. The training data is supplemented with fake samples using the random oversampling technique. The objective is to prevent prejudice against the majority class and enhance the representation of minority classes in the training model [11]. With oversampling, it is expected that the model can be more accurate in predicting classes with limited data.

The dataset is divided into three subsets [12]:

a) Training Data: The model is trained using this data.
b) Validation Data: To confirm how well the model performed throughout training.
c) Test Data: assess the model's ultimate performance.

TensorFlow is used to load the dataset, which is then randomly divided into training and validation data in an 80:20 ratio. The test data is prepared independently for assessment of the model's generalization. All images are resized to 224 x 224 pixels to be consistent with MobileNet input.

B. Preprocessing Data

An important stage in this research is data preprocessing to ensure that the images have the appropriate size and format for the MobileNet model. The preprocessing stage includes image resizing, pixel normalization, and data batching [13]. Proper preprocessing ensures correct training of the model, free from bias resulting from differences in image size or scale.

MobileNet requires images of 224 x 224 pixels with three color channels (RGB) [14]. Therefore, each image in the dataset is resized. This ensures that the input dimensions are consistent and align with the model. To improve training efficiency and stability, each image's pixel values are normalized to fall between 0 and 1. To do this, divide the value of each pixel by 255, which is the maximum value of an RGB pixel. This way, all features are on the same scale, which speeds up the convergence process during training.

To enable training, the dataset is divided into several batches. Each batch contains a number of images, and the batch sizes in this study are 8, 16, 32, and 64. The dataset is cached and prefetched using TensorFlow to speed up the process. This ensures that the data is ready before it is needed by the model, thus reducing idle time during training. The model can train more efficiently because the caching and prefetching processes help reduce latency when reading data from memory or disk [15].

During the dataset preprocessing stage, TensorFlow is automatically used to perform all preprocessing processes, including measurement, normalization, and batching.

C. Augmentation

Augmentation increases image variation and the model's generalization ability [16]. Increasing image variation allows the model to recognize objects even when they are moving, enlarging, or experiencing changes in brightness.

Some of the development methods used in this study are:

a) Rotate: the image is randomly rotated by up to 10 degrees more or less to depict angle variation.
b) Zoom_random: the image is enlarged up to 80% to illustrate different viewing distances.
c) Flip_left_right: To increase orientation variation, rotate the image horizontally.
d) Random_contrast: To simulate various lighting conditions, the image contrast is randomly altered.
e) Random_brightness: The brightness of the image is changed randomly.

By using this augmentation, each class of skin lesions generates 500 new images to improve data distribution and avoid bias during training.

D. Building a Model with MobileNet

This study's goal is to employ Mobilenet (feature extractor) as a foundational model along with weights that have been pre-trained on the ImageNet dataset. The goal is to leverage the model's initial knowledge of common image features, such as patterns and textures so that training can start from a better point compared to training on initial data. The initial classification layer of MobileNet was removed to be replaced with a new layer specifically designed to detect 9 classes of skin lesions.

MobileNet reduces the number of parameters and computational operations compared to conventional convolutional models, making it faster and lighter [17]. This is very helpful when applied to devices with computational limitations, such as smartphones or real-time systems. Here is Figure 2, which contains a summary of the model that will be crated in this study.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 mobilenet_1.00_224 (Functi  (None, 7, 7, 1024)        3228864
 onal)

 global_average_pooling2d (  (None, 1024)              0
 GlobalAveragePooling2D)

 dropout (Dropout)           (None, 1024)              0

 dense (Dense)               (None, 1024)              1049600

 dense_1 (Dense)             (None, 9)                 9225

=================================================================
Total params: 4287689 (16.36 MB)
Trainable params: 4265801 (16.27 MB)
Non-trainable params: 21888 (85.50 KB)
_________________________________________________________________
```

Figure 2. Summary Model MobileNet

In this research, a new layer was added to adjust the model's output to this multi-class classification task. The additional layers include:

a) Global Average Pooling: Figuring out the feature map's overall average to reduce the feature dimensions of MobileNet. By reducing the number of parameters that need to be trained, this method successfully prevents overfitting [18].

b) Dense Layer: Added to enable the model to learn intricate and particular skin lesion patterns, this layer has 1024 units and a ReLU activation function.
c) Dropout layer with a value of 0.2: To avoid overfitting, dropout is applied after the dense layer [19]. 20% of the neurons in the thick layer will be randomly deactivated during training by dropout. This makes the model less reliant on particular attributes and more universal.

To provide the model with initial knowledge about common visual features, transfer learning uses pretrained weights from ImageNet. The initial layers of MobileNet are frozen during training, so the pretrained weights do not change, and only the additional layers are trained. This speeds up the training process and prevents the model from losing the initial knowledge from ImageNet.

After several epochs, the model can be fine-tuned by unfreezing the last layer of MobileNet for retraining. At this point, a lower learning rate is used to maintain training stability and ensure that the model does not lose previous data.

Following the addition of each layer, the Adam Optimizer was used to assemble the model with Learning Rates of 0.001 and 0.0001. The Adam optimizer was used because it can accelerate convergence by modifying the learning rate during training [20]. Accuracy metrics were employed to assess the model's performance during training.

The model is trained using data training and validation, and its performance is assessed at each epoch to prevent overfitting. In this process, dropout is essential since it improves the model generalization to fresh data and motivates it to learn more broad characteristics.

To make sure the model can correctly predict skin lesion classifications, test data is used after training is finished. The fine-tuning techniques aid in the model's adaption to the unique features of the skin lesion dataset, while the application of Dropout and Global Average Pooling guarantees that the model stays effective and does not overfit.

E. Result Visualization

Accuracy and loss graphs are used to display the model's performance during training and validation. This graph facilitates the analysis of the model evolution throughout time. Additionally, the graph aids in identifying indications of either overfitting or underfitting.

a) Accuracy: The proportion of all test data that were correctly predicted.
b) Loss: Assessing the discrepancy between the actual labels and the model prediction.
c) Confusion Matrix: Used to examine how well the model performs for every class and pinpoint those that are hard to forecast.
d) Classification Report: Shows the precisions, recall, f1-score, and support for each class as well as the classification outcomes.

The accuracy and loss curves for training and validation data. If the validation accuracy curve stagnates or decreases while the training accuracy curve increases, it indicates that the model is overfitting.

III. RESULT AND DISCUSSION

This chapter will discuss the result of implementing the MobileNet model for skin cancer image classification, as well as analyze the model's performance based on several established evaluation metrics. This model was trained using a dataset of images and skin lesions and optimized with data augmentation techniques to improve its performance in detecting 9 classes of skin cancer. In this evaluation, the results of the MobileNet model will be compared by batch size to determine which one has the best accuracy.

A. Result of Preprocessing and Augmentation

At the preprocessing and augmentation stage, the training data is prepared to improve the model's performance in classifying skin lesions.

```
total training image count = 4500

--------------------------------------
Class name =  actinic keratosis
count      =  500
proportion =  0.1111111111111111
--------------------------------------
Class name =  basal cell carcinoma
count      =  500
proportion =  0.1111111111111111
--------------------------------------
Class name =  dermatofibroma
count      =  500
proportion =  0.1111111111111111
--------------------------------------
Class name =  melanoma
count      =  500
proportion =  0.1111111111111111
--------------------------------------
Class name =  nevus
count      =  500
proportion =  0.1111111111111111
--------------------------------------
Class name =  pigmented benign keratosis
count      =  500
...
Class name =  vascular lesion
count      =  500
proportion =  0.1111111111111111
--------------------------------------
```

Figure 3. Number of Data per Class After Performing Oversampling

Figure 3 shows the results of oversampling and the distribution of the training data used in this study. The total number of images used is 4500 images evenly divided into 9 classes of skin lesions. Each class has 500 images, making up 11.11% of the entire training dataset. Oversampling with augmentation was chosen to address class imbalance within the dataset. This technique is effective for avoiding bias toward the majority class, allowing the model to learn adequately from each class during training. Compared to other data balancing methods like SMOTE or undersampling, physical augmentation is more suitable in this case, as visual models often require more variation in visual features.

Given that the dataset consists of skin images, this augmentation technique enhances the diversity of patterns seen by the model without reducing data from majority classes. The selection of 500 samples per class is intended to balance the data without excessive inflation, which could increase the risk of overfitting. This ensures each class is

sufficiently represented in training while maintaining an efficient dataset size.

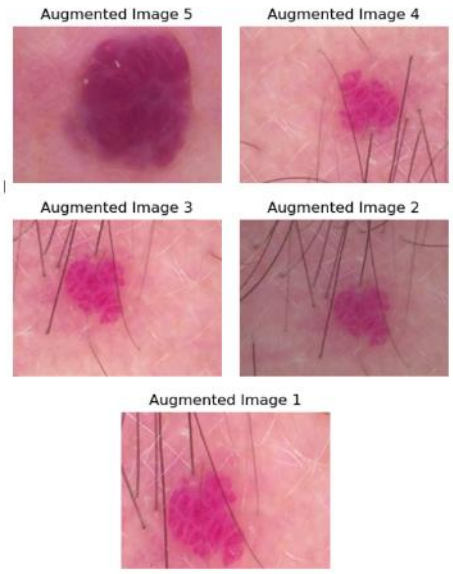

Figure 4. Result of the Dataset After the Augmentation Process

In Figure 4, an example of the dataset that has undergone the augmentation process is shown. Augmentation is done to increase image variation without manually adding more images. The augmentation techniques used include rotation, zoom, flip, contrast, and brightness. The generated image retains the main characteristics of the skin lesions, but with a different appearance. This helps the model train its generalization ability so it can accurately classify images of skin lesions under various conditions.

B. Model Results

The result of the model training with an 80:20 data split ratio, where 80% of the total dataset is used as training data, while the remaining 20% is used as validation data. With a Dense layer of 1024.

TABLE II
SCENARIO WITHOUT OVERSAMPLING AUGMENTATION, DROPOUT, AND RELU

Batch Size	Epoch	Learning Rate	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
8	25	0.001	99.17%	0.0249	58.68%	2.4766
16	25	0.001	99.45%	0.0330	57.14%	2.1709
32	25	0.001	99.53%	0.0337	59.56%	1.6665
64	25	0.001	80.28%	0,6129	49.67%	1.5548

The outcomes of the model testing with different batch sizes across 25 epochs at a learning rate of 0.001 are displayed in Table II. The findings of this investigation are without augmentation, dropout, and ReLU. Overall, the increase in batch size does not yield consistent results in validation. For example, small batch sizes like 8 and 16 show very high training accuracy, namely 99.17% and 99.45%, but their validation accuracy is relatively low, 58.68% and 57.14%. Meanwhile, although larger batch sizes like 32 and 64 show significant improvements in training accuracy (99.53% and

80.28%), the validation accuracy does not follow a clear pattern, with validation loss decreasing at batch size 64 but validation accuracy remaining low. This indicates that there may be overfitting or a mismatch between the batch size and the dataset used.

TABLE III
SCENARIO WITHOUT OVERSAMPLING AUGMENTATION

Batch Size	Epoch	Learning Rate	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
8	25	0.0001	100.00%	0.0019	63.50%	1.6089
16	25	0.0001	100.00%	2.0149	66.37%	2.6671
32	25	0.0001	99.94%	0.0093	62.83%	1.4464
64	25	0.0001	99.61%	0.0151	63.27%	1.4517

Table III shows the test results without any augmentation. This model adds dropout, ReLU, and a learning rate of 0.0001. Because, the first experiment without oversampling augmentation, dropout, ReLU, and learning rate 0.001 resulted in very low validation accuracy and severe overfitting, so the learning rate was lowered to reduce the model's aggressive updates during training and allow for more stable convergence. This model achieved a very high training accuracy of 99.62% to 100%. However, the validation accuracy ranged from 62.83% to 66.37%, which is much lower than the training accuracy. In addition, the validation loss remains high, even though the batch size changes, with validation loss values ranging from 1.4464 to 2.6671. this allows for the occurrence of overfitting. However, the validation accuracy after adding dropout and ReLU is slightly higher compared to the previous scenario.

TABLE IV
SCENARIO WITH OVERSAMPLING AUGMENTATION, DROPOUT, AND RELU

Batch Size	Epoch	Learning Rate	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
8	25	0.0001	99.64%	0.0115	86.89%	0.6128
16	25	0.0001	99.81%	0.0058	85.49%	0.5854
32	25	0.0001	99.94%	0.0054	85.22%	0.6156
64	25	0.0001	100.00%	0.0033	84.00%	0.5762

Table IV during the training process, tests were conducted with various combinations of batch size, this is the quantity of samples processed across 25 epochs prior to updating the model weights, using augmentation, and an Adam learning rate of 0.0001. Because in the previous experiment when adding a learning rate of 0.0001 the validation accuracy had increased sufficiently, the learning rate of 0.0001 was maintained, as lowering it further would result in slower convergence and longer training times. The results obtained show variations in training and validation accuracy, as well as loss for both categories.

In earlier experiments using basic augmentation without oversampling, the model's performance showed only minimal improvement compared to previous trials. Therefore, oversampling with augmentation was incorporated to increase validation accuracy and reduce the potential for high overfitting. This approach allows the model to better generalize by balancing the data effectively across classes, ensuring each class has sufficient representation, and enhancing the diversity of features in the dataset.

The model obtained a training accuracy of 99.64% with a loss of 0.0115 at a batch size of 8. Nevertheless, the model only obtained an accuracy of 86.89% with a loss of 0.6218 for the validation data. A decrease in accuracy on the validation data, suggesting possible overfitting, was seen despite the excellent training accuracy.

The model increased training accuracy to 99.81% with a batch size of 16 and with a loss of 0.0058, but validation accuracy slightly decreased to 85.49% with a loss of 0.5854. This indicates a slight improvement in training, but not significant in validation.

At a batch size of 32, the training accuracy of the model reached 99.94% with a loss of 0.0054, but the validation accuracy dropped to 85.22% with a loss of 0.6156. Although the training is getting closer perfect, the decline in performance during validation remains a concern.

At a batch size of 64, the model achieved a perfect training accuracy of 100% with a very low loss of 0.0033. However, the validation accuracy actually decreased to 94.00% with a loss of 0.5762. this indicates that increasing the batch size does improve training, but it causes the model to overfit more, resulting in a continuous decline in validation accuracy.

C. Result Visualization

The results of the model training will be visualized to give a more comprehensive view of the model's functionality in classifying data. Several methods used include the confusion matrix, accuracy and loss graphs, and the classification report. The results below represent the model, which achieved an accuracy of 99.64% and a validation accuracy of 86.89%.

1) *Accuracy and Loss Graph*

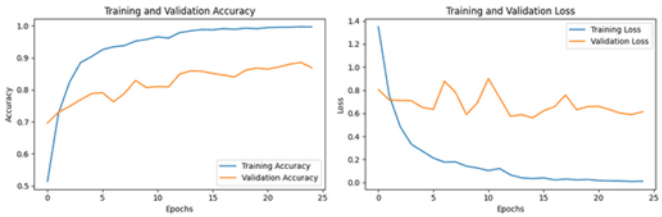

Figure 5. Accuracy and Loss Graph

The model's performance during training and validation is displayed in Figure 5. On the left graph, the accuracy of training and validation is displayed in relation to the number of epochs. The training accuracy (blue line) consistently increases with the addition of epochs and approaches a value close to 1 (100%). However, the validation accuracy (orange line), although initially increasing, starts to stabilize and slightly decrease after a certain number of epochs. In the second graph (right), it is evident that as the number of epochs rises, the training loss keeps declining, indicating that the model is learning better from the training data. However, the validation loss shows a fluctuating pattern, initially decreasing but then starting to increase.

2) *Confusion Matrix*

The validation accuracy after adding augmentation, dropout, and ReLU improved and was better than the previous scenarios. Therefore, the best result was an accuracy of 99.64% with a validation accuracy of 86.89%.

The confusion matrix showing the classification is shown in Figure 6, results of the model on the skin cancer detection dataset. Rows in the matrix represent the true labels, while columns represent the model predictions (predict label). Each cell contains the number of cases where the actual label and the prediction match or do not match.

Overall, the model works quite well in classifying several categories, such as "Vascular Lesion" with 108 correct predictions and "Dermatofibroma" with 100 correct predictions. However, there are some prediction errors, such as in the case of "Melanoma", which is often misclassified as "Venus" or "Seborrheic Keratosis". Darker colors indicate larger numbers, with errors spread across several categories, highlighting the challenge of distinguishing between different types of skin cancer.

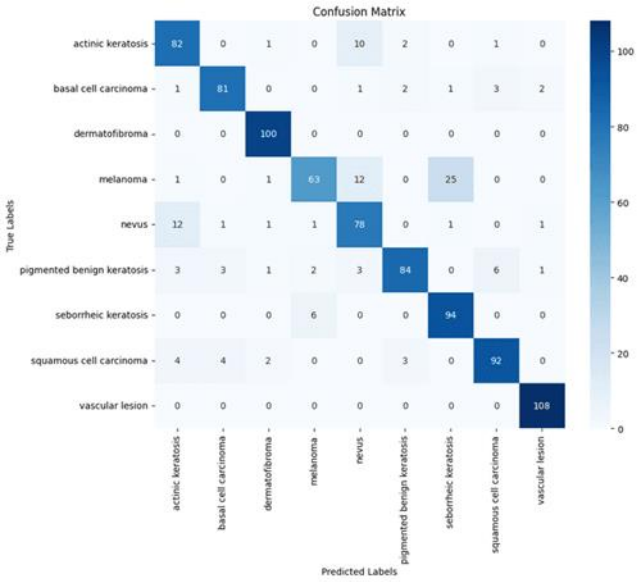

Figure 6. Confusion Matrix

3) *Classification Report*

	precision	recall	f1-score	support
actinic keratosis	0.80	0.85	0.82	96
basal cell carcinoma	0.91	0.89	0.90	91
dermatofibroma	0.94	1.00	0.97	100
melanoma	0.88	0.62	0.72	102
nevus	0.75	0.82	0.78	95
pigmented benign keratosis	0.92	0.82	0.87	103
seborrheic keratosis	0.78	0.94	0.85	100
squamous cell carcinoma	0.90	0.88	0.89	105
vascular lesion	0.96	1.00	0.98	108
accuracy			0.87	900
macro avg	0.87	0.87	0.87	900
weighted avg	0.87	0.87	0.87	900

Figure 7. Classification Report

Figure 7 displays the classification report from the skin cancer detection model results, which includes important metrics such as precision, recall, f1-score, and support for each class.

Precision shows the accuracy of the model predictions for each class, with the highest value for "Vascular Lesion" (0.96) and the lowest value for "Nevus" (0.75). Recall measures the model's ability to find all instances of a particular class, with "Vascular Lesion" having a perfect recall value (1.00), while "Melanoma" has the lowest recall value (0.62). The f1-score is a combination of precision and recall, with the highest value for "Vascular Lesion" (0.98) and the lowest for "Melanoma" (0.72). Support shows the number of samples tested for each class.

Overall, the MobileNet model used in this study achieved an accuracy of 0.87, with the macro and weighted averages

for precision, recall, and f1-score also being 0.87. This result shows that the model has consistent performance across all classes. This model works very well in categories such as “Vascular Lesion” and “Dermatofibroma”, but still faces challenges in more difficult categories like “Melanoma” and “Nevus”.

Compared to two previous studies, this study shows better results. In the first study, which used a Convolutional Neural Network (CNN) with eight 2D convolutional layers to classify skin cancer into two categories, namely “Malignant” and “Benign”, an accuracy of 75% was achieved.

The second study, which used VGG-16 and LeNet for the classification of 8 types of skin cancer, recorded accuracies of 73.22% and 68.11%, respectively. Both showed lower results compared to MobileNet in this study, which was able to achieve a higher accuracy (99.64%) across 9 classes. Thus, MobileNet shows a significant improvement in both accuracy and efficiency compared to the more complex CNN and VGG-16, making it a superior alternative for multi-class classification in 9 categories of skin cancer.

IV. CONCLUSION

Classification of skin cancer images using the MobileNet architecture has been successfully created, leveraging the ISIC 2020 dataset as the sole data source. The ISIC dataset was selected for its high-quality, well-organized images with clear classifications, which facilitated a structured training process. Alternative datasets explored were found lacking in terms of image quality, classification clarity, and organization, which would have complicated the identification of classes. The study objectives is implementing and evaluating MobileNet for skin cancer classification using the ISIC 2020 dataset, and comparing MobileNet's performance with other CNN architectures like VGG-16 and LeNet in terms of accuracy and computational efficiency. The architecture applied to this categorization uses a 224 x 224 input shape, a Learning Rate of 0.0001, 25 epochs, and a Dense Layer of 1024, with an oversampling of 500. Oversampling was especially beneficial in balancing data distribution across classes without sacrificing information quality, while the dropout and ReLU combination enhanced model generalization and helped prevent overfitting. Initial experiments using only basic augmentation showed only marginal performance improvements, prompting the addition of oversampling to increase validation accuracy and further reduce overfitting risks. This choice ensured balance across classes and introduced meaningful variability in visual features, a crucial factor for skin cancer image analysis. After the oversampling process, the training data amounted to 3600, and the validation data amounted to 900. The result of the model training with an 80:20 data split ratio, where 80% of the total dataset is used as training data, while the remaining 20% is used as validation data. With a Dense layer of 1024.

As a result, the model’s validation accuracy was only 86.89%, but its training accuracy was 99.64%. This indicates that although the MobileNet model works incredibly well with the training data, its performance for skin cancer classification on the validation data is less than optimal. The model tends to overfit, where it memorizes inadequate capacity to generalize to new data from the training data. Analysis shows that this limitation is caused by a lack of quantity and diversity of the datasets used.

Future work could involve advanced transfer learning techniques, such as employing ensemble methods that combine multiple architectures like ResNet-50 and EfficientNet using weighted voting based on individual validation performances. Following this, fine-tuning specific layers of pre-trained models at a lower learning rate (1e-4 to 1e-5) could be applied to capture task-specific features. With larger and more diverse datasets, these approaches have the potential to further improve validation performance, leading to more accurate and robust classifications.

REFERENCES

[1] N. M. Mahmoud and A. M. Soliman, “Early automated detection system for skin cancer diagnosis using artificial intelligent techniques,” *Sci Rep*, vol. 14, no. 1, Dec. 2024, doi: 10.1038/s41598-024-59783-0.

[2] R. Attar *et al.*, “Reliable detection of eczema areas for fully automated assessment of eczema severity from digital camera images,” Nov. 07, 2022. doi: 10.1101/2022.11.05.22281951.

[3] H. Yuniarti, R. Sigit, and M. A. Rofiq, “Penerapan Fuzzy Tsukamoto pada Alat Deteksi Penyakit Hipoksemia, Hipotermia, Hipertensi, dan Diabetes untuk Health Care Kiosk,” 2020. [Online]. Available: http://jurnal.polibatam.ac.id/index.php/JAIC

[4] A. A. Dzaky *et al.*, “Optimization Chatbot Services Based on DNN-Bert for Mental Health of University Students,” 2024. [Online]. Available: http://jurnal.polibatam.ac.id/index.php/JAIC

[5] C. Y. Kim, K. S. Um, and S. W. Heo, “A novel MobileNet with selective depth multiplier to compromise complexity and accuracy,” *ETRI Journal*, vol. 45, no. 4, pp. 666–677, Aug. 2023, doi: 10.4218/etrij.2022-0103.

[6] Luqman Hakim, Z. Sari, and H. Handhajani, “Klasifikasi Citra Pigmen Kanker Kulit Menggunakan Convolutional Neural Network,” *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 2, pp. 379–385, Apr. 2021, doi: 10.29207/resti.v5i2.3001.

[7] M. Ezar Al Rivan and K. Kunci, “2 Nd Mdp Student Conference (MSC) 2023 144| Universitas Multi Data Palembang Klasifikasi Jenis Kanker Kulit Manusia Menggunakan Convolutional Neural Network”.

[8] A. Wibowo, “Comparison of Naive Bayes Method with Support Vector Machine in Helpdesk Ticket Classification,” 2023. [Online]. Available: http://jurnal.polibatam.ac.id/index.php/JAIC

[9] A. Georgiou *et al.*, “Diagnostic Informatics—The Role of Digital Health in Diagnostic Stewardship and the Achievement of Excellence, Safety, and Value,” Jun. 10, 2021, *Frontiers Media SA*. doi: 10.3389/fdgth.2021.659652.

[10] S. Salsabila and A. Suharso, “Comparison of Deep Learning Architectures in Identifying Types of Medicinal Plant Leaf Images,” 2024. [Online]. Available: http://jurnal.polibatam.ac.id/index.php/JAIC

[11] S. Cusworth, G. V. Gkoutos, and A. Acharjee, “A novel generative adversarial networks modelling for the class imbalance problem in high dimensional omics data,” *BMC Med Inform Decis Mak*, vol. 24, no. 1, Dec. 2024, doi: 10.1186/s12911-024-02487-2.

[12] A. Fuadi *et al.*, “Perbandingan Arsitektur Mobilenet Dan Nasnetmobile Untuk Klasifikasi Penyakit Pada Citra Daun Kentang.”

[13] N. Saefulloh, J. Indra, and A. Ratna Juwita, "Implementasi Algoritma Convolutional Neural Network (CNN) Untuk Klasifikasi Kecacatan Pada Proses Welding di Perusahaan Manufacturing," *Technology and Science (BITS)*, vol. 6, no. 1, pp. 387–394, 2024, doi: 10.47065/bits.v6i1.5321.

[14] G. Ashari Rakhmat and M. Fikri Haekal, "MIND (Multimedia Artificial Intelligent Networking Database Peningkatan Performa MobilenetV3 dengan Squeeze-and-Excitation (Studi Kasus Klasifikasi Kesegaran Ikan Berdasarkan Mata Ikan)," *Journal MIND Journal | ISSN*, vol. 8, no. 1, pp. 27–41, 2023, doi: 10.26760/mindjournal.v8i1.27-41.

[15] E. R. Lucas Filho, L. Yang, K. Fu, and H. Herodotou, "Streaming Machine Learning for Supporting Data Prefetching in Modern Data Storage Systems," in *AI4Sys 2023 - Proceedings of the 1st Workshop on AI for Systems*, Association for Computing Machinery, Inc, Aug. 2023, pp. 7–12. doi: 10.1145/3588982.3603608.

[16] I. Topan Adib Amrulloh *et al.*, "Evaluasi Augmentasi Data Pada Deteksi Penyakit Daun Tebu Dengan Yolo v8," 2024.

[17] Y. Zou, "Research On Pruning Methods for Mobilenet Convolutional Neural Network," 2024.

[18] R. D. Djuardi and T. H. Rochadiani, "Klasifikasi Suara Anjing Menggunakan Pretrained Model Yet Another Mobile Network Berbasis Convolutional Neural Network," *Technology and Science (BITS)*, vol. 6, no. 1, 2024, doi: 10.47065/bits.v6i1.5165.

[19] J. Sistem Komputer dan Kecerdasan Buatan *et al.*, "Pengembangan Model Klasifikasi Produk Furnitur Sebagai Visual Search Menggunakan Algoritma Convolutional Neural Network," 2024.

[20] M. Reyad, A. M. Sarhan, and M. Arafa, "A modified Adam algorithm for deep neural network optimization," *Neural Comput Appl*, vol. 35, no. 23, pp. 17095–17112, Aug. 2023, doi: 10.1007/s00521-023-08568-z.