Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer
Vol. 23, No. 1, November 2023, pp. 1∼16
ISSN: 2476-9843, accredited by Kemenristekdikti, Decree No: 200/M/KPT/2020
DOI: 10.30812/matrik.v23i1.3185

Investigating The Effectiveness of Various Convolutional Neural Network Model Architectures for Skin Cancer Melanoma Classification

Rizky Hafizh Jatmiko , Yoga Pristyanto
Universitas AMIKOM, Yogyakarta, Indonesia

Article Info

Article history:

Received July 22, 2023
Revised September 22, 2023
Accepted October 04, 2023

Keywords:

Convolutional Neural Network
Deep Learning
Image Classification
Melanoma
Skin Cancer

ABSTRACT

Melanoma is one of the most dangerous types of skin cancer. Since 2018, the number of skin cancer cases in the US has increased and exceeded 100,000. Melanoma is the third most common cancer in Indonesia, following womb cancer and breast cancer. Standard detection of melanoma skin cancer biopsy is costly and time-consuming. This research aimed to build and compare melanoma skin cancer detection using various Convolutional Neural Network methods. This research used four CNN model architecture methods: VGG-16, LeNet, Xception, and MobileNet. The dataset for this research was image data that consisted of 9605 data divided into benign and malignant classes. The data was augmented to increase its quantity. After that, the data was trained using four CNN architecture models and evaluated using the confusion matrix. The result of this study was that the Xception model has the best accuracy and the lowest loss, with 93% accuracy and 19% loss, with precision 93%, recall 93.5%, and f1-score 93%. Meanwhile, the other model, VGG-16, gave 90% accuracy and 27% loss, LeNet had 89.7% accuracy and 28% loss, and mobileNet had 90.8% accuracy and 22.5% loss.

Corresponding Author:

Yoga Pristyanto, +6285156727484,
Faculty of Computer Science,
Universitas AMIKOM, Yogyakarta, Indonesia,
Email: yoga.pristyanto@amikom.ac.id.

How to Cite:
R. H. Jatmiko and Y. Pristyanto, ”Investigating The Effectiveness of Various Convolutional Neural Network Model Architectures for Skin Cancer Melanoma Classification”, *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 23, no. 1, pp. 1-16, Oct. 2023.

1. INTRODUCTION

Skin cancer is an overgrowth of skin cells, starting by damaging the DNA and resulting in growth, splitting uncontrollably, and having an irregular structure [1]. Ultraviolet rays can damage DNA and are a leading cause of skin cancer. Applying sunscreen can reduce UV exposure and prevent skin cancer [1]. Older people can be susceptible to skin cancer because they have low immunity, and other factors are because skin cancer has a long incubation period of about ten years. Therefore, the symptoms can only be seen in old age [1]. Skin cancer is one of the most common cancers worldwide, especially for white people [2]. Incidence and death continue to increase due to skin cancer. In 2018, there were 100,590 cases of skin cancer, including melanoma, recorded in America, resulting in incidents and deaths [3]. In Indonesia, melanoma skin cancer is the third most common cancer, with a prevalence rate of 5.9 - 7.8% annually, following womb and breast cancer [4]. Cases of malignant skin cancer are more common in Indonesia, accounting for 7.9%, compared to benign skin cancer [5]. In diagnosing skin cancer, doctors generally perform a biopsy, which takes skin samples for a detailed examination [1]. However, detecting skin cancer through a biopsy can be expensive and time-consuming [6]. By taking advantage of the technology, everyone can perform early diagnosis for skin cancer from everywhere using technology for classifying skin cancer melanoma. One of the most popular methods nowadays is deep learning. The implementation uses deep learning to classify data using various Convolutional Neural Network architectures. A Convolutional Neural Network (CNN) is a deep learning algorithm that uses large datasets to classify data [3]. CNN is a feedforward-type neural network that extracts features using a convolution structure without extracting them manually [7]. With its high network depth, CNN is often applied to image data [8]. CNN can adaptively optimize its performance [9]. The purpose of using CNN is to extract and analyze meaningful information from images effectively and perform classification from the information obtained [10]. Recently, the application of Convolutional Neural Networks (CNN) has been widely used in image detection, including melanoma skin cancer detection. Deep learning is a subset of machine learning with good computing performance, and deep learning is also a popular method used recently [10].

Several previous studies were used as references for this research. Research using the ResNet50 method has also been brought in research [4]. The existing research scenario is to compare several types of optimizers, as well as comparisons of several batch sizes, to determine which is the best. The research results [4] are that Adamax is the best optimizer with a batch size of 64, the accuracy is 99%, the precision is 99%, the recall is 99%, and the f1-score is 99%. Research [6] discussed designing a skin cancer detection system using the VGG-16 method. The data was divided into 75% for train data and 25% for test data. Research that talks about comparing two methods, ResNet50 and ResNet101 [9], shows a comparison of the results of ResNet50, which has an accuracy of 91%, and ResNet101, which results in an accuracy of 89%, explained in research [9] the reason why ResNet101 has lower accuracy than ResNet50 is that ResNet101 is more complex than ResNet50. The training data needs to be more significant for ResNet101. In research [11] discussing the comparison of the application of CNN and K-NN methods to detect skin cancer, it gave an accuracy of 76.56% for CNN and 75% for K-NN. In research [12], this research uses 2 CNN architectures, ResNet50 and VGG-16, to perform feature extraction, whereas the classification method uses a support vector machine; the accuracy results given are 65.33. Research [13] discusses performance comparisons between machine learning and deep learning. Research [13] shows that deep learning has higher accuracy than machine learning. The highest accuracy of machine learning is 75% using Ensemble, whereas the best result from deep learning is 87% using the ResNet50 Transfer Learning method, which has been fine-tuned. Research [14] conducted an experiment using the LeNet model with 44 test data and various train data. The results of this study explain that the amount of training data and the number of epochs during the training process affect the accuracy of the model, with the highest result using 176 data images and 100 epochs to produce an accuracy of 100%. Research using the LeNet model has also been conducted [14]. The technique proposed by research [14], namely the ANU-Net technique, was used to categorize skin cancer as melanoma or not melanoma based on the segmentation generated by the LeNet model. The results of the proposed method give an accuracy of 98.78%. In research [15] using the MobileNet model, which is used to detect melanoma, in this study, the model is also implemented on a smartphone camera, giving results of 95% accuracy in object detection and 70% for classification. The method proposed in Research [16] uses the Xception model as the base model by adding three layers and fine-tuning the Xception model using the dataset from the ISIC archive several times. After that, the model was optimized using six different optimizers. The highest result given is 99.78% accuracy using Adam and lazyAdam optimizers.

With many algorithms that can be used, it provides many options, such as which algorithm gives the most accurate results. However, not all algorithms can be optimally used to classify skin cancer. This research will discuss and compare several popular models, especially VGG-16, LeNet, Xception, and MobileNet. The difference between this research and previous research is the model for the comparison. None of the previous research compares VGG-16, LeNet, Xception, and MobileNet. The novelty of this research is to compare the results of the accuracy, loss, and evaluation of the models for melanoma skin cancer classification using the same dataset and process. This study aims to compare the performance of the VGG-16, LeNet, Xception, and MobileNet models in classifying melanoma skin cancer using the same dataset and preprocessing process. Adjustments were used to compare models using the Adam optimizer with a loss function sparse categorical cross entropy with a batch size of 32 and epoch 50. There are two contributions to this research. First, it can be used for further research in the same field. Second, this research can be a reference for choosing models for classifying skin cancer.

2. RESEARCH METHOD

This research begins with data acquisition used to create deep learning models. After getting the data, data will be processed in the Data Preprocessing stage, which will be split into two: training and testing data. Then, after splitting data, the data will be trained using the CNN architecture model. The models used are VGG-16, LeNet, Xception, and MobileNet. The last step is to evaluate the model using the confusion matrix. The stages of the research carried out in this research can be seen in a research methodology flow in Figure 1.

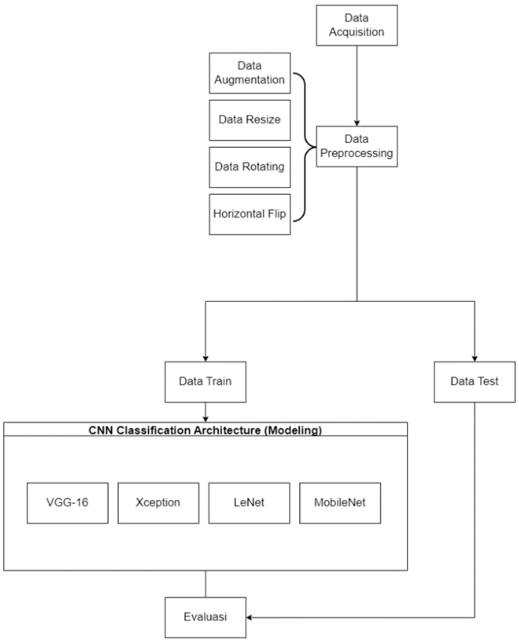
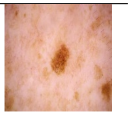
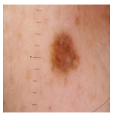


Figure 1. Research Flow

2.1. Data Acquisition

The data was obtained from the public dataset Kaggle Melanoma Skin Cancer Dataset of 10000 Images. The dataset consists of 9605 image data. Divide it into two classes of data: benign and malignant. In this study, the number of data that will be used was 9605 and will be split into two: 75% of the data will be used as training data, and 25% will be used as testing data. Table 1 shows an overview of the dataset before entering the preprocessing process.

Table 1. Dataset Overview

Class Name	Amount of Data	Image
Benign	5000	
Malignant	4605	

2.2. Data processing

This preprocessing stage aims to reproduce the data, which is expected to improve the model's performance. Data preprocessing consists of four steps: data augmentation, data resizing, data rotation, and horizontal flip. These four stages will provide more variety to the data, making it easier for the model to get optimal results. The size of the image will be generalized to 128 x 128 pixels. The process of rotating the images randomly and also flipping each image horizontally.

a. Data Augmentation
Data augmentation is a process of increasing image data that aims to increase the model's accuracy. The process of increasing image data uses the resizing, rotating method. Data augmentation is performed every epoch during model fitting.

b. Data Rescale
The data resizing process is one of the methods used to reproduce image data by changing the size of the image. Rescaling data is necessary to normalize the pixel value to a similar range to optimize the training process. The pixel value rescales from 0 to 255 to a new range of 0 to 1.

c. Data Rotating
Data rotating is also used to perform data augmentation or reproduce image data. The method is to rotate the image with a specific range. For this research, images rotate randomly between 0 and 20 degrees.

d. Horizontal Flip
This process is carried out to reverse the image data horizontally. Similar to data rotating, which rotates data for a predetermined range, horizontal flip only flips data horizontally. This technique provides additional variation in the augmentation data.

Data rescale, data rotating, and horizontal flip are single processes used to increase the variation of image data in the data augmentation process. This preprocessing process divides the data into training and test data. The distribution is 75% training data and 25% test data.

2.3. Modeling

This study uses four models: VGG-16, Xception, and LeNet. In every model, ReLU activation is used, as well as sigmoid activation, which is an activation that is often used. Sigmoid activation is used because the data only has malignant and benign classes. Each model also uses the same loss function, namely sparse categorical cross-entropy. These four models are expected to provide good results and evaluation values.

1. VGG-16
The Visual Geometric Group (VGG) was introduced at ILSVRC 2014, where VGG was the runner-up [17], and Simonyan and Zisserman proposed the VGG-16 model at the University of Oxford [18]. Table 2 shows VGG architecture comprising five building blocks [17]. Each block has a convolutional layer and a max pooling layer. The first block consists of two convolutional layers and one max pooling layer, each with a filter of 64. Block two is the same as block 1, consisting of 2 convolutional layers and one max pooling layer with 128 filters. In blocks 3 to 4, there are three convolution layers and one max pooling with each filter 256, 512, and 512. For a fully connected layer, use a global max pooling layer, a dense layer with ReLU activation, a hyperparameter dropout layer to reduce overfitting, and a dense layer using sigmoid activation.

Table 2. VGG-16 Model Architecture

Block	Layer Name	Output Shape
	Input Layer	None, 128, 128, 3
	Convolutional Layer	None, 128, 128, 64
Block 1	Convolutional Layer	None, 128, 128, 64
	Max Pooling Layer	None, 64, 64, 64
Block 2	Convolutional Layer	None, 64, 64, 128
	Convolutional Layer	None, 64, 64, 128
	Max Pooling Layer	None, 32, 32, 128
Block 3	Convolutional Layer	None, 32, 32, 256
	Convolutional Layer	None, 32, 32, 256
	Convolutional Layer	None, 32, 32, 256
	Max Pooling Layer	None, 16, 16, 256

(continued on next page)

Table 2 (continued)

Block	Layer Name	Output Shape
Block 4	Convolutional Layer	None, 16, 16, 512
	Convolutional Layer	None, 16, 16, 512
	Convolutional Layer	None, 16, 16, 512
	Max Pooling Layer	None, 8, 8, 512
Block 5	Convolutional Layer	None, 8, 8, 512
	Convolutional Layer	None, 8, 8, 512
	Convolutional Layer	None, 8, 8, 512
	Max Pooling Layer	None, 4, 4, 512
	Global Max Pooling Layer	None, 512
	Dense	None, 512
	Dropout	None, 512
	Dense	None, 2

2. LeNet

LeNet was published in 1998 [17] and is one of the earliest models of CNN architecture, invented by Huben and Wiesel [19]. LeNet is comprised of two convolutional layers and two average pooling layers [17] with filters 6 and 16. Table 3 shows the layer used. There is a hyperparameter dropout layer to avoid overfitting models and several fully connected layers such as flatten, Dense with ReLU activation, and Dense with sigmoid activation.

Table 3. LeNet Architecture Model

Layer Name	Output Shape
Convolutional Layer	(None, 124, 124, 6)
Dropout Layer	(None, 124, 124, 6)
Average Pooling Layer	(None, 62, 62, 6)
Dropout Layer	(None, 62, 62, 6)
Convolutional Layer	(None, 58, 58, 16)
Average Pooling Layer	(None, 29, 29, 16)
Flatten	(None, 13456)
Dense Layer	(None, 128)
Dense Layer	(None, 2)

3. Xception

Extreme Inception is a CNN architectural model invented by F. Chollet in 2017 [20]. As the name implies extreme Inception, Xception introduces a new inception layer. This layer is composed of a depthwise convolutional layer [21]. The arrangement of layers on Xception is quite different from the other three models. Xception is divided into three main parts: entry, middle, and exit flow [20]. Tables 4, 5, and 6 show three main parts of the Xception model.

Table 4. Xception Architecture Model Entry Flow

Layer Name	Output Shape
Input Layer	None, 128, 128, 3
Convolutional Layer	None, 64, 64, 32
Batch Normalization Layer	None, 64, 64, 32
Activation Layer	None, 64, 64, 32
Convolutional Layer	None, 64, 64, 64
Batch Normalization Layer	None, 64, 64, 64
Activation Layer	None, 64, 64, 64
Activation Layer	None, 64, 64, 64
Separable Convolutional Layer	None, 64, 64, 128
Batch Normalization Layer	None, 64, 64, 128
Activation Layer	None, 64, 64, 128
Separable Convolutional Layer	None, 64, 64, 128
Batch Normalization Layer	None, 64, 64, 128
Max Pooling Layer	None, 32, 32, 128
Convolutional Layer	None, 32, 32, 128

(continued on next page)

Tabel 4 (continued)

Layer Name	Output Shape
Add Layer	None, 32, 32, 128
Activation Layer	None, 32, 32, 128
Separable Convolutional Layer	None, 32, 32, 256
Batch Normalization Layer	None, 32, 32, 256
Activation Layer	None, 32, 32, 256
Separable Convolutional Layer	None, 32, 32, 256
Batch Normalization Layer	None, 32, 32, 256
Max Pooling	None, 16, 16, 256
Convolutional Layer	None, 16, 16, 256
Add Layer	None, 16, 16, 256
Activation Layer	None, 16, 16, 256
Separable Convolutional Layer	None, 16, 16, 728
Batch Normalization Layer	None, 16, 16, 728
Activation Layer	None, 16, 16, 728
Separable Convolutional Layer	None, 16, 16, 728
Batch Normalization Layer	None, 16, 16, 728
Max Pooling	None, 8, 8, 728
Convolutional Layer	None, 8, 8, 728
Add Layer	None, 8, 8, 728

The process in the middle flow is repeated eight times. The layers in Table 5 are the first and second iterations of the middle flow. The middle flow in its first iteration starts with the input layer and then comprises the activation layer, separable convolutional layer, batch normalization layer, and add layer.

Table 5. Table Architecture Model Xception Middle Flow

Layer Name	Output Shape
Input Layer	None, 8, 8, 728
Activation Layer	None, 8, 8, 728
Separable Convolutional Layer	None, 8, 8, 728
Batch Normalization Layer	None, 8, 8, 728
Activation Layer	None, 8, 8, 728
Separable Convolutional Layer	None, 8, 8, 728
Batch Normalization Layer	None, 8, 8, 728
Activation Layer	None, 8, 8, 728
Separable Convolutional Layer	None, 8, 8, 728
Batch Normalization Layer	None, 8, 8, 728
Add Layer	None, 8, 8, 728
Activation Layer	None, 8, 8, 728
Separable Convolutional Layer	None, 8, 8, 728
Batch Normalization Layer	None, 8, 8, 728
Activation Layer	None, 8, 8, 728
Separable Convolutional Layer	None, 8, 8, 728
Batch Normalization Layer	None, 8, 8, 728
Activation Layer	None, 8, 8, 728
Separable Convolutional Layer	None, 8, 8, 728
Batch Normalization Layer	None, 8, 8, 728
Add Layer	None, 8, 8, 728

The exit flow shown in Table 6 also comprises the activation, separable convolutional, batch normalization to make the network more robust, and max pooling layer. The fully connected layer uses the global average pooling layer, and the dense layer uses sigmoid activation.

Table 6. Architecture Model Xception Exit Flow

Layer Name	Output Shape
Input Layer	None, 128, 128, 3
Activation Layer	None, 128, 128, 3
Separable Convolutional Layer	None, 128, 128, 728
Batch Normalization Layer	None, 128, 128, 728
Activation Layer	None, 128, 128, 728
Separable Convolutional Layer	None, 128, 128, 1024
Batch Normalization Layer	None, 128, 128, 1024
Max Pooling Layer	None, 64, 64, 1024
Convolutional Layers	None, 64, 64, 1024
Add Layer	None, 64, 64, 1024
Separable Convolutional Layer	None, 64, 64, 1536
Batch Normalization Layer	None, 64, 64, 1536
Activation Layer	None, 64, 64, 1536
Separable Convolutional Layer	None, 64, 64, 2048
Batch Normalization Layer	None, 64, 64, 2048
Activation Layer	None, 64, 64, 2048
Global Average Pooling Layer	None, 2048
Dense	None, 2

4. MobileNet
MobileNet, proposed by Google in 2017 [22], is a model often used for mobile applications. MobileNet uses depthwise separable convolutions composed of depthwise and pointwise convolutions [20], [23]. New features added to mobileNetV2 are Linear Bottleneck Structure, Residual Structure, and Inverse Residual Structure [24]. Table 7 shows MobileNet architecture composed of the input layer and convolutional layer. It used a batch normalization activation layer using ReLU, a Depthwise convolutional layer, a global average pooling layer, and a dense layer using a sigmoid activation layer to make the network robust.

Table 7. Table Architecture Model MobileNet

Layer Name	Output Shape
Input Layer	None, 128, 128, 3
Convolutional Layers	None, 64, 64, 32
Batch Normalization	None, 64, 64, 32
Activation Layer	None, 64, 64, 32
Depthwise convolution layer	None, 64, 64, 32
Batch Normalization	None, 64, 64, 32
Activation Layer	None, 64, 64, 32
Convolutional Layers	None, 64, 64, 64
Batch Normalization	None, 64, 64, 64
Depthwise convolution layer	None, 64, 64, 64
Activation Layer	None, 64, 64, 64
Depthwise convolution layer	None, 32, 32, 64
Batch Normalization	None, 32, 32, 64
Activation Layer	None, 32, 32, 64
. . .	. . .
Global Average Pooling Layer	None, 1024
Dense Layer	None, 2

2.4. Evaluation

The confusion matrix is used to see the model's performance in this study. The confusion matrix shown in Table 8 describes the prediction results made by the model.

Table 8. Confusion Matrix

	Prediction	
	Positive	Negative
Actual	TP (True Positive)	FN (False Negative)
	FP (False Positive)	TN (True Negative)

In evaluating the confusion matrix, the values of TP and TN indicate that the model can classify data based on its class. In contrast, FP and FN values indicate that the model incorrectly classifies the data [6]. To support the confusion matrix results, an analysis of the classification evaluation parameters is carried out by calculating the values of precision, recall, and f1-score, which can be calculated using Equation (1), (2), (3), and (4).

$$\begin{aligned}
Accuration &= \frac{TP + TN}{TP + TN + FP + FN} && (1)\\
Accuration &= \frac{TP}{TP + FP} && (2)\\
Accuration &= \frac{TP}{TP + FN} && (3)\\
Accuration &= F1 - Score = 2 \times \frac{PrecisiCon \times Recall}{PrecisiCon + Recall} && (4)
\end{aligned}$$

3. RESULT AND ANALYSIS

At the model training stage, the image used is an image obtained from a dataset platform that is generally accessible to anyone, namely Kaggle. The image data used was 9605 consisting of 2 classes, namely benign, which consisted of 5000 image data, and 4605 malignant image data. Image data will be divided into training data, as much as 75% of the overall data, and testing data, as much as 25%. The total data used in the training data is 7204, and the test data is 2401. The data splitting scheme is shown in Table 9. The size of the image will be generalized to 128x128 pixels. To increase the amount of data, data augmentation is carried out to improve the results of the model's accuracy. The augmentation process is done by changing the image size, rotating the image, and flipping the image horizontally. The data will be augmented at each epoch. The configuration for model comparison uses the Adam optimizer, using a loss function sparse categorical cross entropy with an epoch of 50 and a batch size of 32.

Table 9. Data Splitting Scheme

Class Name	Training Data	Testing Data
Benign	3750	1250
Malignant	3454	1151
Total Data	7204	2401

3.1. VGG-16

The results of the VGG-16 architectural model shown in Figure 3 produce an accuracy of 90% with a loss of 27%. The confusion matrix shown in Figure 2 shows that the VGG-16 architectural model can detect 1205 benign skin cancers, and only about 45 are detected incorrectly. As for the malignant class, the errors made were high, that is, 189, but for accurate detection, it was high 962. The precision, recall, and f1-scores shown in Table 10 results were quite good. All three were above 90%.

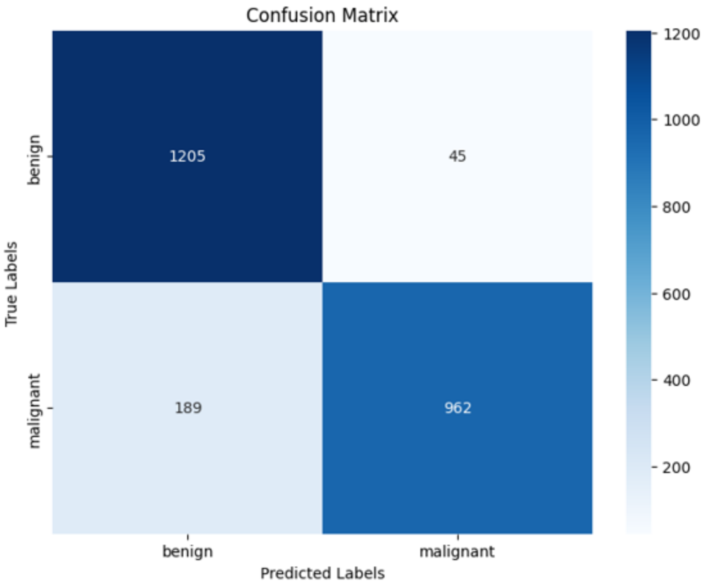


Figure 2. Confusion Matrix VGG-16

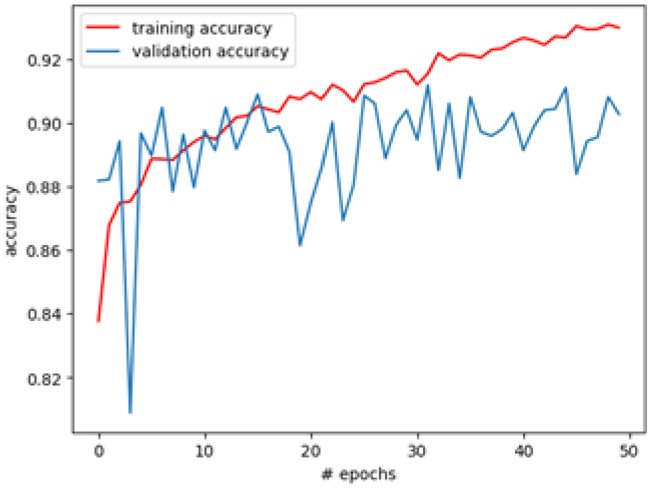


Figure 3. VGG-16 Accuracy

Table 10. VGG-16 Evaluation

Class	Precision	Recall	F1-Score
Benign	86%	96%	91%
Malignant	96%	84%	89%
Average	91%	90%	90%

3.2. LeNet

The training was carried out on the LeNet architecture model, which has an accuracy of 89.7% and a loss of 28%, as shown in Figure 5. The evaluation of the LeNet architectural model shown in the Figure 4 confusion matrix shows that from the 7204 data tested, the data can be classified correctly according to their respective classes. Besides that, Table 11 shows the results of other parameters used to measure model performance, measurements using precision, recall, and F1-Score.

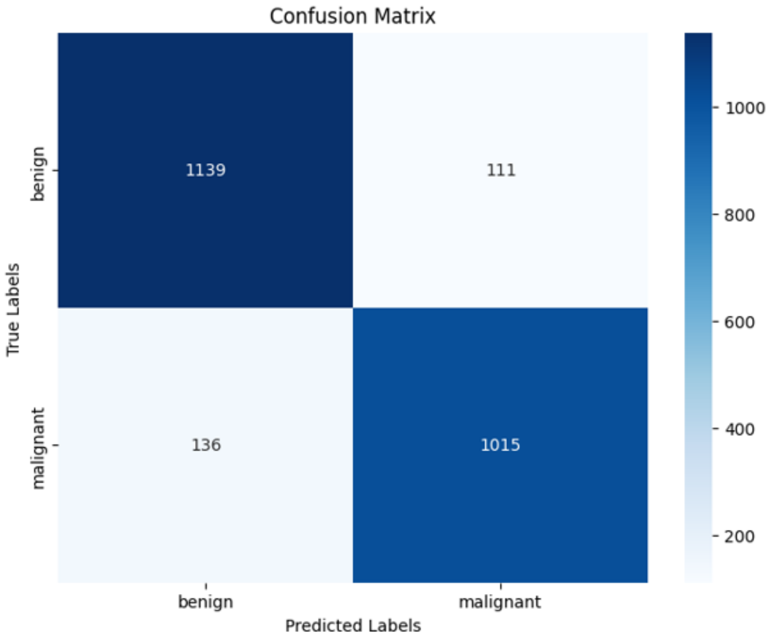


Figure 4. Confusion Matrix LeNet

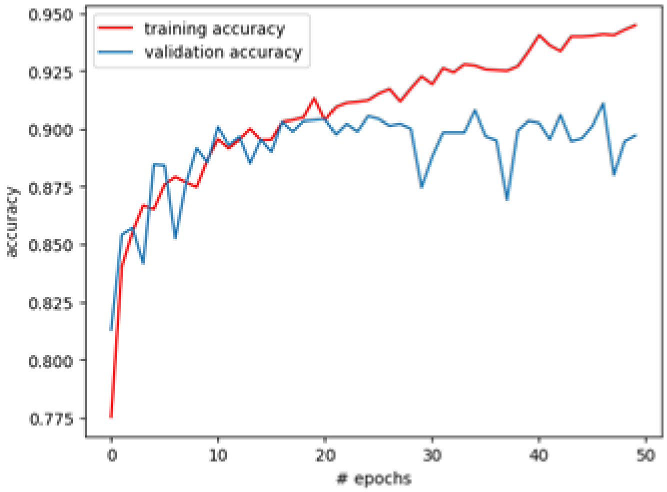


Figure 5. LeNet Accuracy

Table 11. LeNet Evaluation

Class	Precision	Recall	F1-Score
Benign	89%	91%	90%
Malignant	90%	88%	89%
Average	89.5%	89.5%	89.5%

3.3. Xception

Furthermore, the Xception architecture model was tested; it is shown in Figure 7 that the accuracy is 93% with a loss of 19%. The evaluation results shown in Figure 6, the confusion matrix from the Xception model, showed that the model could classify benign and malignant image data based on each class correctly, and image data for each class was identified incorrectly as less than 100. Table 12 contains the evaluation parameters precision, recall, and f1-score. All three give values above 90%, precision produces 93%, recall is 93.5%, and f1-score is 93%. With the evaluation value as described, the exception architecture model provides outstanding model performance.

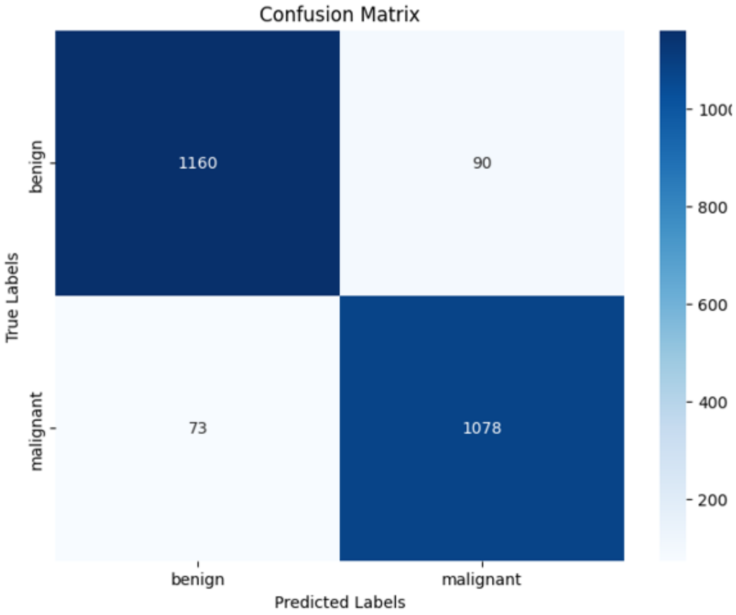


Figure 6. Confusion Matrix Xception

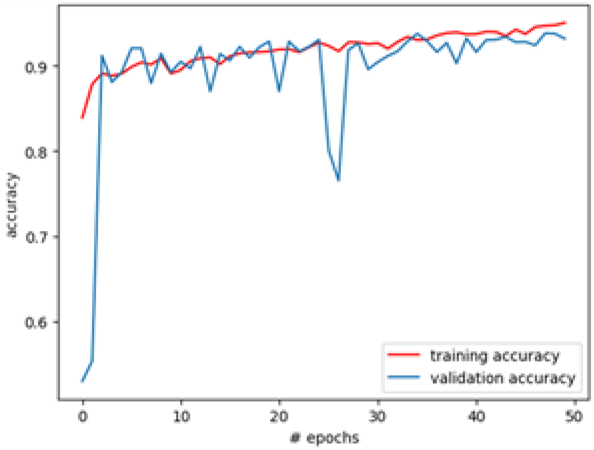


Figure 7. Xception Accuracy

Table 12. Xception Evaluation

Class	Precision	Recall	F1-Score
Benign	94%	93%	93%
Malignant	92%	94%	93%
Average	93%	93.5%	93%

3.4. MobileNet

Training conducted on the MobileNet architectural model shown in Figure 9 has an accuracy of 90.8% and a loss of 22.5%. The Confusion matrix's value shown in Figure 8 can be seen as true positive and negative, indicating that the model can properly classify data based on each class. Meanwhile, based on Table 13 showing precision, recall, and f1-score, all have an average value above 90%.

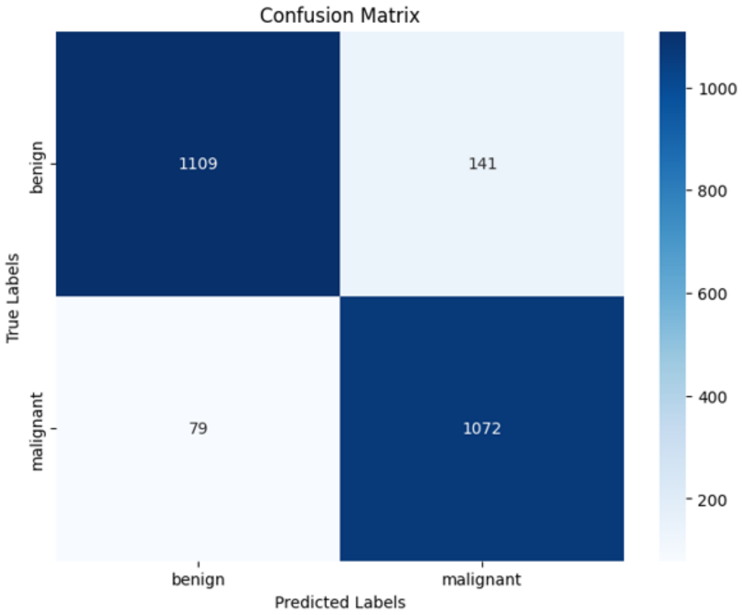


Figure 8. Confusion Matrix MobileNet

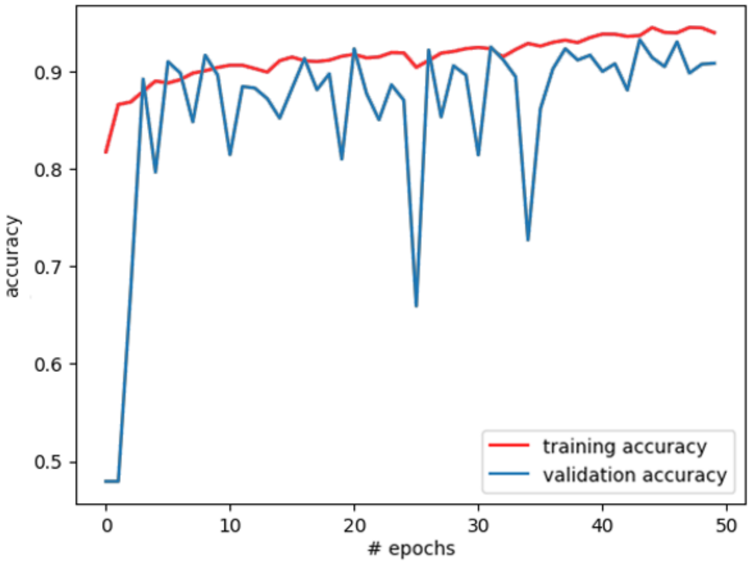


Figure 9. MobileNet Accuracy

Table 13. MobileNet Evaluation

Class	Precision	Recall	F1-Score
Benign	93%	89%	91%
Malignant	88%	93%	91%
Average	90.5%	91%	91%

This section analyzes and compares the results obtained from four models: VGG-16, LeNet, Xception, and MobileNet. The VGG-16 architectural model produces an accuracy value of 90% with a relatively high loss of 27%. Based on Figure 3, it can be seen that the accuracy validation has fluctuated. The evaluation results were quite good, with precision, recall, and f1-score above 90%. The LeNet architectural model yielded the slightest accuracy among the other models, with just 89.7% and a relatively high loss of 28%. Figure 5 shows that the accuracy validation has decreased starting from around the 20th epoch. The Xception architecture model produces the highest accuracy among the other models, 93%, and loss, which is relatively low at 19%. Based on Figure 7, accuracy and validation do not fluctuate much and tend to be stable. Even though one point has decreased, the rest is stable. The evaluation results are also the best, with 93% precision, 93.5% recall, and 93% f1-score. The MobileNet architectural model gives a yield of 90.8% and a relatively small loss of 22.5%. Even though Figure 9 shows fluctuations in accuracy validation, the MobileNet model gives quite good results. The evaluation is also good, with a precision of 90.5%, recall of 91%, and f1-score of 91%.

Therefore, using the Xception model for classifying melanoma skin cancer gives the best and the most stable performance during the process than the other model used in this research supported by research [13], [16]; even though the Xception model results in research [13] is lower than result from research [16].

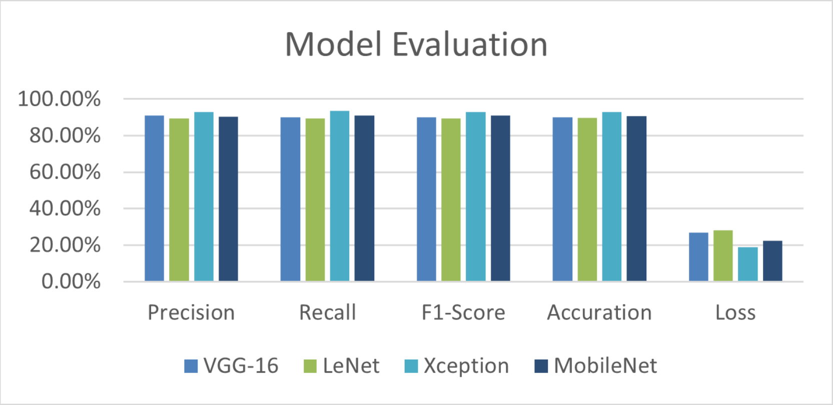


Figure 10. Chart Model Evaluation Comparison

Table 14. Table Model Evaluation Comparison

Model	Precision	Recall	F1-Score	Accuracy	Loss
VGG-16	91%	90%	90%	90%	27%
LeNet	89.50%	89.50%	89.50%	89.70%	28%
Xception	93%	93.5%	93%	93%	19%
MobileNet	90.5%	91%	91%	90.8%	22.5%

4. CONCLUSION

The study used four architectural models of the convolutional neural network VGG-16, LeNet, Xception, and MobileNet to classify melanoma skin cancer. Using image data from the Kaggle platform, the total data is 9605, with a 75% division of training data totaling 7204 and 25% testing data of 2401. Data preprocessing is done by augmenting the data by changing the image size to 128 x 128 pixels, rotating and flipping the image horizontally. The augmentation process occurs in each running epoch. The evaluation process uses a confusion matrix by testing precision, recall, and f1-score. Of the four models that have been tested, the results obtained are that the Xception architecture model gives the highest accuracy, which is at 93%, with a reasonably low loss at 19%. The Xception architectural model can be used to assist in diagnosing melanoma skin cancer. Developments that can be carried out in subsequent studies include adding image data for the malignant class so that it has a number that is not far from the benign class. In the augmentation process, transformations can be added to image data other than those used in this study, and also, when training the model, the number of epochs can be increased. Also, batch size values can be changed. Besides that, it can add research scenarios, such as changing the optimizer and giving other hyperparameters, such as the learning rate or any other hyperparameters.

5. ACKNOWLEDGEMENTS

Thanks to the Department of Research and Community Service, Universitas Amikom Yogyakarta, for supporting and striving to finish this research.

6. DECLARATIONS

AUTHOR CONTRIBUTION

Rizky Hafizh Jatmiko, the first author, is a data collector and model maker. Yoga Pristyanto, the second author, analyzes and helps to give a correction of each model's results.

FUNDING STATEMENT

The researcher would like to sincerely thank the Department of Research and Community Service, Universitas Amikom Yogyakarta, for supporting and funding this research.

COMPETING INTEREST

Further research can be directed to make the best model either in the same field as this research or in different fields for image detection by considering the amount of data sharing and other things.

REFERENCES

[1] J. Setiabudi, M. Wardhana, I. G. A. A. E. Indira, and N. M. D. Puspawati, "Profil Pra Kanker dan Kanker Kulit RSUP Sanglah Periode 2015 - 2018," *Jurnal Medika Udayana*, vol. 10, no. 3, pp. 83–89, mar 2021.

[2] Z. Khazaei, F. Ghorat, A. M. Jarrahi, H. A. Adineh, M. Sohrabivafa, and E. Goodarzi, "Global incidence and mortality of skin cancer by histological subtype and its relationship with the human development index (HDI); an ecology study in 2018," *World Cancer Research Journal*, vol. 6, pp. 1–14, 2019.

[3] 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.

[4] N. Alyyu, R. Fuadah, and N. Pratiwi, "Klasifikasi Kanker Kulit Ganas Dan Jinak Menggunakan Metode Convolutional Neural Network," in *e-Proceeding of Engineering*, vol. 8, dec 2022, pp. 3200–3206.

[5] S. Sa'idah, I. Putu, Y. Nugraha Suparta, and E. Suhartono, "Modifikasi Convolutional Neural Network Arsitektur GoogLeNet dengan Dull Razor Filtering untuk Klasifikasi Kanker Kulit," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, vol. 11, no. 2, pp. 148–153, may 2022.

[6] R. Agustina, R. Magdalena, and N. K. C. Pratiwi, "Klasifikasi Kanker Kulit menggunakan Metode Convolutional Neural Network dengan Arsitektur VGG-16," *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 10, no. 2, p. 446, apr 2022.

[7] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 6999–7019, dec 2022.

[8] P. Adi Nugroho, I. Fenriana, and R. Arijanto, "Implementasi Deep Learning Menggunakan Convolutional Neural Network (CNN) pada Ekspresi Manusia," *Jurnal ALGOR*, vol. 2, no. 1, pp. 12–21, 2020.

[9] Z. Niswati, R. Hardatin, M. N. Muslimah, and S. N. Hasanah, "Perbandingan Arsitektur ResNet50 dan ResNet101 dalam Klasifikasi Kanker Serviks pada Citra Pap Smear," *Faktor Exacta*, vol. 14, no. 3, pp. 160–167, oct 2021.

[10] E. Yilmaz and M. Trocan, “A modified version of GoogLeNet for melanoma diagnosis,” *Journal of Information and Telecommunication*, vol. 5, no. 3, pp. 395–405, 2021.

[11] T. R. Savera, W. H. Suryawan, and A. W. Setiawan, “Deteksi Dini Kanker Kulit menggunakan K-NN dan Convolutional Neural Network,” *Jurnal Teknologi Informasi Dan Ilmu Komputer*, vol. 7, no. 2, pp. 373–378, 2020.

[12] R. Yohannes and M. Rivan, “Klasifikasi Jenis Kanker Kulit Menggunakan CNN-SVM,” *Jurnal Algoritme*, vol. 2, no. 2, pp. 133–144, apr 2022.

[13] S. Bechelli and J. Delhommelle, “Machine Learning and Deep Learning Algorithms for Skin Cancer Classification from Dermoscopic Images,” *Bioengineering*, vol. 9, no. 3, pp. 1–18, mar 2022.

[14] V. Radhika and B. S. Chandana, “Skin Melanoma Classification from Dermoscopy Images using ANU-Net Technique,” *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 13, no. 10, pp. 928–938, 2022.

[15] A. Wibowo, C. A. Hartanto, and P. W. Wirawan, “Android skin cancer detection and classification based on mobilenet v2 model,” *International Journal of Advances in Intelligent Informatics*, vol. 6, no. 2, pp. 135–148, jul 2020.

[16] M. N. Qureshi, M. S. Umar, and S. Shahab, “A Transfer-Learning-Based Novel Convolution Neural Network for Melanoma Classification,” *Computers*, vol. 11, no. 5, pp. 1–18, apr 2022.

[17] A. Ajit, K. Acharya, and A. Samanta, “A Review of Convolutional Neural Networks,” in *International Conference on Emerging Trends in Information Technology and Engineering, ic-ETITE 2020*. Institute of Electrical and Electronics Engineers Inc., feb 2020, pp. 1–5.

[18] A. Victor Ikechukwu, S. Murali, R. Deepu, and R. Shivamurthy, “ResNet-50 vs VGG-19 vs training from scratch: A comparative analysis of the segmentation and classification of Pneumonia from chest X-ray images,” *Global Transitions Proceedings*, vol. 2, no. 2, pp. 375–381, nov 2021.

[19] M. Ezar Al Rivan and Orlando, “Klasifikasi Jenis Kanker Kulit Manusia Menggunakan Convolutional Neural Network,” in *2nd MDP Student Conference 2023*. 2nd MDP STUDENT CONFERENCE (MSC) 2023, 2023, pp. 144–150.

[20] O. Rochmawanti, F. Utaminingrum, and F. A. Bachtiar, “Analisis Performa Pre-Trained Model Convolutional Neural Network dalam Mendeteksi Penyakit Tuberkulosis,” *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 8, no. 4, pp. 805–8013, 2021.

[21] M. Rahimzadeh and A. Attar, “A modified deep convolutional neural network for detecting COVID-19 and pneumonia from chest X-ray images based on the concatenation of Xception and ResNet50V2,” *Informatics in Medicine Unlocked*, vol. 19, p. 100360, jan 2020.

[22] M. Sandler, A. Howard, M. Zhu, and A. Zhmoginov, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),*, 2018, pp. 4510–4520.

[23] W. Sae-Lim, W. Wettayaprasit, and P. Aiyarak, “Convolutional Neural Networks Using MobileNet for Skin Lesion Classification,” in *Artificial Intelligence Research Lab, Department of Computer Sciencence of Songkla University*. 2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE), jul 2019, pp. 242–247.

[24] Y. Zhou, S. Chen, Y. Wang, and W. Huan, “Review of research on lightweight convolutional neural networks,” in *IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC 2020)*, 2020, pp. 1713–1720.

[This page intentionally left blank.]