Performance Evaluation of Pre-Trained Convolutional Neural Network Model for Skin Disease Classification

Afandi Nur Aziz Thohari[1], Liliek Triyono[2], Idhawati Hestiningsih[3], Budi Suyanto[4] , Amran Yobioktobera[5]

[1,2,3,4,5]*Department Electrical Engineering, Politeknik Negeri Semarang, Indonesia*

[1]afandi@polines.ac.id, [2]liliek.triyono@polines.ac.id,[3]hestidha@gmail.com, [4]hestidha@gmail.com, [5]amranyobi@polines.ac.id

Abstract - Indonesia is a tropical country that has various skin diseases. Tinea versicolor, ringworm, and scabies are the most common types of skin diseases suffered by the people of Indonesia. The classification of the three skin diseases can be automatically completed by artificial intelligence and deep learning technology because the classification process using an expert will require a lot of money and time. The challenge in classifying skin diseases is in the process of collecting data. Because health data cannot be obtained freely, there must be approval from the patient or hospital. Therefore, to overcome the limited amount of data, Pre-Trained CNN is used. The Pre-Trained CNN model has many patterns from thousands of images, so we do not need many images to train the model. In this study, a comparison of five pre-trained CNN models was conducted, namely VGGNet16, MobileNetV2, InceptionResNetV2, ResNet152V2, and DenseNet201. The aim is to find out which CNN model can produce the best performance in classifying skin diseases with a limited amount of image data. The test results show that the ResNet152V2 model has the best classification ability with the highest accuracy, precision, recall, and F1 score values, namely 95.84%, 0.963, 0.96, and 0.956. As for the training execution time, the ResNet152V2 model has the fastest time to achieve 95% accuracy. That's happened because the addition of the dropout parameter is 20%.

Keywords: Classification, Deep Learning, Pre-Trained Model, ResNet152V2, Skin Disease

I. INTRODUCTION

The development of artificial intelligence in the current era is increasingly massive. The technology that makes machines do human work is applied in almost all fields, including the health sector. One of the applications of artificial intelligence in the health sector is the prediction of diabetes using machine learning [1-2]. Artificial intelligence is divided into several branches of science, namely machine learning and deep learning. In the case of processing large amounts of data and consisting of many features, deep learning can be used because the technology that requires a lot of data has many layers of neural networks. Deep learning is widely used to process image data, and this technology is capable of performing image classification, object detection, and image segmentation [3]. One application of deep learning in the health sector is the detection of COVID-19 through X-ray images [4-5], malaria parasite detection [6], and predicting the 1p/19q co-deletion status [7]. The use of deep learning for health must be very careful because if the system detects a false negative, it will endanger someone's life. Therefore, in building an artificial intelligence system for health, it is necessary to be accompanied by experts and carry out continuous testing.

The development of machine learning models for health has several challenges, such as limited data. The data used to develop the model is personal and cannot be provided without the patient's permission or the hospital. Therefore, the Pre-Trained Model is used to overcome the data limitations. Pre-Trained model keeps the model have high accuracy even though the data used is limited. In this study, the use of deep learning will be studied for the classification of skin diseases in images. Skin diseases used as research objects are tinea versicolor, ringworm, and scabies. Using this disease is because it is most often found in Indonesia [8]. The skin disease classification process uses the Convolution Neural Network (CNN) algorithm. Previously, there have been several studies using skin diseases as research objects.

One of the studies that use the CNN algorithm for skin disease classification is a study conducted by [9]. The skin diseases used as research objects are acne, keratosis, eczema herpeticum, and urticaria. The CNN architecture built in this study has 11 layers consisting of a pooling layer, a fully connected layer, and an activation layer. The accuracy of the identification of skin disease is 91.025%. Then there is also research using the classification of skin diseases with five classes, namely healthy, acne, eczema, benign, and malignant. The CNN

architecture development uses AlexNet as a pre-trained model, then the SVM algorithm as a classifier [10]. The result is that the overall value of accuracy is 86.21%. The last research is the classification of skin diseases using the MobileNet architecture [11]. The accuracy of the built model is 94.4%. The training model is then deployed into an android application.

This study aimed to compare 5 pre-trained CNN models for skin disease classification with a limited number of datasets. The pre-trained CNN model used has many layers and proven to be accurate in ImageNet data classification [12]. The comparison is made by looking at the confusion matrix results and the execution time during training.

II. METHOD

The method used in this study is shown in the flowchart Fig 1. Just like when building a model using the CNN method, the first thing to add is a dataset. Then divide the dataset into train data and test data. After that, the pre-processing stage builds a network model, trains the model, and finally evaluates the model. However, there is one other process in the flowchart of Fig. 1, which is to compare the evaluation results of each CNN architecture that has been tested.

A. Dataset

The dataset used for the training process is the image of skin diseases. The image has been grouped into three classes: ringworm, scabies, and tinea versicolor—grouping based on each type of disease pattern. The ringworm has a circular pattern and is red. In contrast, scabies has a spreading pattern and is in the form of red spots. Then tinea versicolor has a spreading pattern and is white. The total number of images is 144. The limited number of datasets is due to the difficulty of obtaining an image dataset of ringworm, scabies, and tinea versicolor. The source of the dataset search is from google image and the dermet.com website. The representation of the input image used as the dataset is shown in Fig. 2. After getting the dataset, then split the dataset to be divided into training data and test data. The comparison of datasets is 80% for training data and 20% for test data.

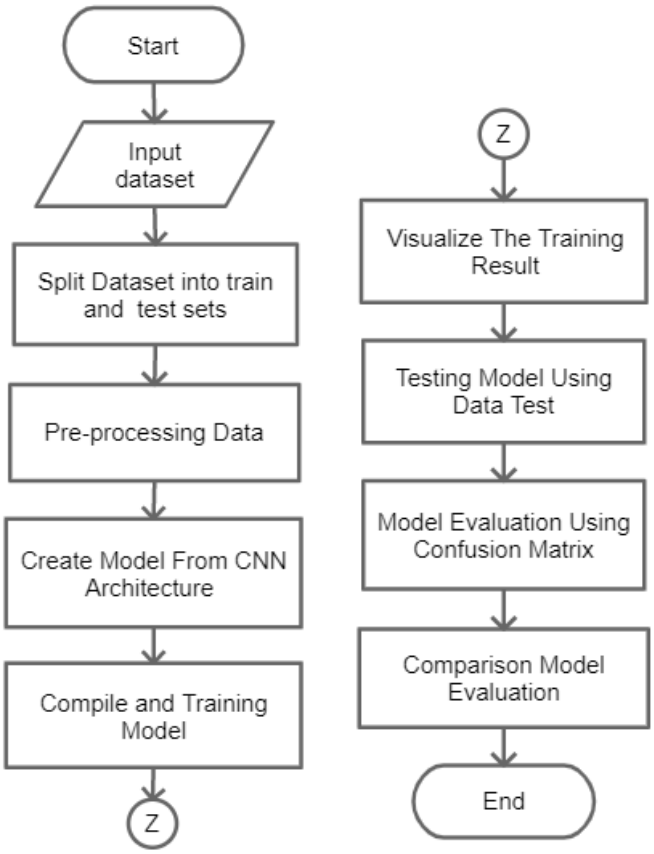

Fig. 1 Research flowchart

Fig. 2 Input image

B. Pre-Processing

At this stage, the image resizing process is carried out to uniform the image's resolution. All images are resized to a resolution of 224x224 pixels. Then at this stage, image labeling is also carried out to facilitate the learning process during training. The pre-processing stage is essential because, at this stage, the data augmentation and data pipeline processes are also carried out. The process of data augmentation carried out is rescaling and validation. The rescaling value of the data in this study is 1/255, while the validation value is 20%. Then the data pipeline is carried out by converting the image data into an array that TensorFlow can read.

C. Create Model

After pre-processing the data, the next step is to create a CNN model. In this study, five pre-trained CNN models were used for image classification. The five pre-trained models are VGGNet16, MobileNetV2, DenseNet201, InceptionResNetV2, and ResNet152V2. The reason for choosing the five pre-trained models is that they have an architecture that can run on devices with limited resources [13].

1) VGGNet16: It is a Pre-Train model which consists of 16 layers. The VGGNet16 architecture is divided into two parts: the feature extraction layer and the fully connected layer [14]. The feature extraction layer is a layer that functions to recognize the pattern from the image then convert it into a one-dimensional matrix format. Then the fully connected layer functions to study patterns that have been extracted previously. So in the fully connected layer, the machine will learn to recognize objects contained in the image.

2) MobileNetV2: MobileNetV2 is a development of the MobileNetV1 architecture. There are two new features in the MobileNetV2 architecture, namely linear bottlenecks and shortcut connections between bottlenecks [15]. As the name suggests, MobileNet is used on devices with limited resources, such as cell phones. So that the training model with MobileNet can be deployed to mobile devices. MobileNetV2 has accuracy and a faster execution time than MobileNetV1. The architecture of MobileNetV2 has got more layers, as shown in Table I.

3) ResNet152V2: Residual Neural Network abbreviated as ResNet is a Pre-Train CNN model that can not only be used for image classification but can also be used for object detection and semantic segmentation. ResNet has the advantage of training networks with a vast number of layers. In general, CNN has a limited number of layers and cannot reach the deepest layer. Because the more profound the layer, the greater the error in the accuracy of the test data, often called overfitting [16].

Therefore, ResNet offers the concept of the residual block to overcome the occurrence of overfitting and allows the network to reach the deepest layer. ResNet has various types of layer depth, ranging from 18, 34, 50, 101, to 152 [17]. In this study, 152 layers will be used to classify images. The reason for choosing 152 layers is because it has the best accuracy. An illustration of the use of the ResNet152V2 architecture can be seen in Fig. 3.

4) DenseNet201: DenseNet architecture differs from traditional CCN architecture in that all layers are connected incrementally, as in Fig. DenseNet offers different techniques in processing image data. DenseNet architecture consists of Dense Block in which there are several layers [18]. Each layer has a feature map connected, starting from the first layer until the new layer is created. The structure of the Dense Block can be seen in Fig. 4.

Fig. 4 shows that the first layer has a k_0 feature map, the second layer has a $k_0 + k$ feature map, and the last layer has a $k_0 +$ 4k feature map. DenseNet consists of several Dense Blocks for processing data. Among the Dense Blocks, there is a Transition Layer in which there are operations such as batch normalization, convolution, and pooling. After arriving at the last Dense Block, the prediction process is carried out using global average pooling, fully connected layer, and activation using softmax. In DenseNet201, there are four Dense Blocks and three transfer layers whose architecture is shown in Fig. 5.

TABLE I
MOBILENETV2 ARCHITECTURE

Input	Operator	*t*	*c*	*n*	*s*
$224^2 \; x \; 3$	conv2d	-	32	1	2
$112^2 \; x \; 32$	bottleneck	1	16	1	1
$112^2 \; x \; 16$	bottleneck	6	24	2	2
$56^2 \; x \; 24$	bottleneck	6	32	3	2
$28^2 \; x \; 32$	bottleneck	6	64	4	2
$14^2 \; x \; 64$	bottleneck	6	96	3	1
$14^2 \; x \; 96$	bottleneck	6	160	3	2
$7^2 \; x \; 160$	bottleneck	6	320	1	1
$7^2 \; x \; 320$	conv2d 1x1	-	1280	1	1
$7^2 \; x \; 1280$	avgpool 7x7	-	-	1	-
1 x 1 x 1280	conv2d 1x1	-	k	-	

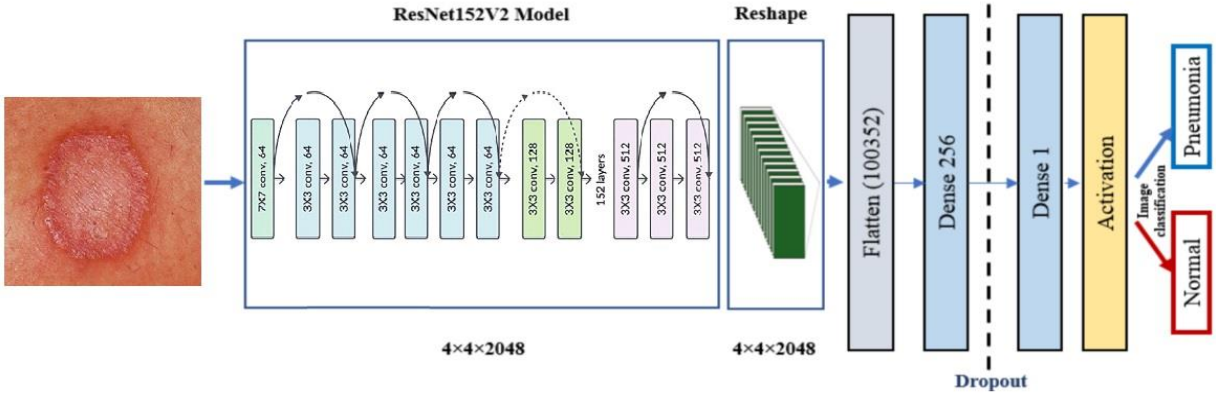

Fig. 3 ResNet152V2 architecture

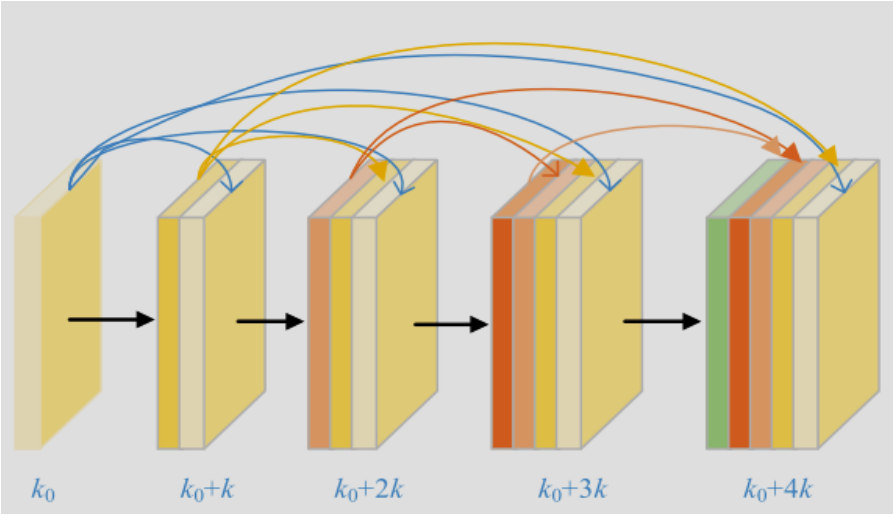

Fig. 4 Dense block structure

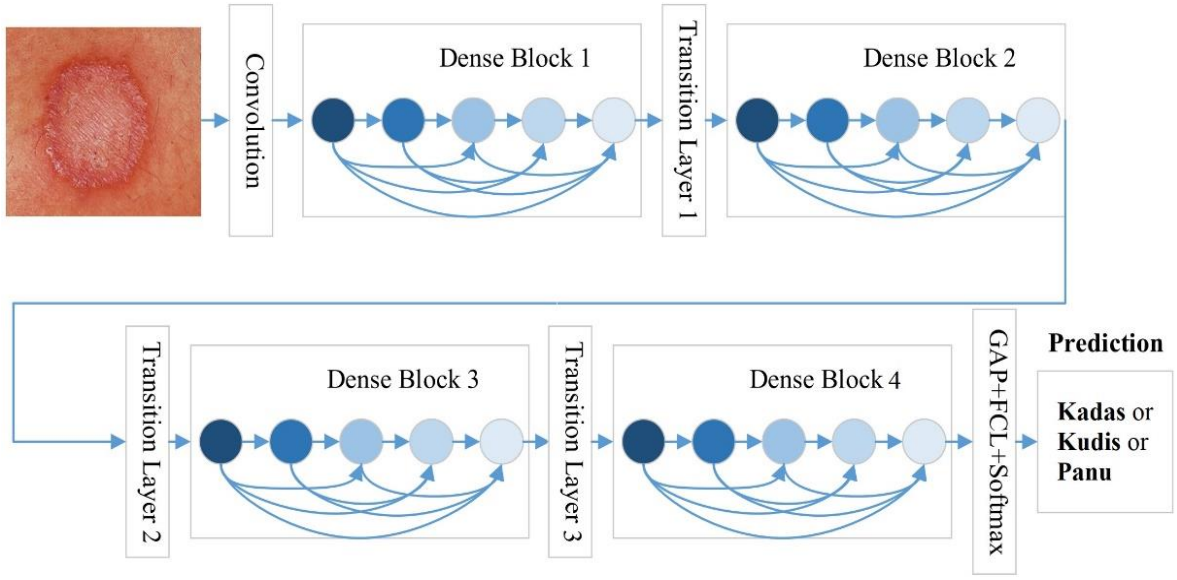

Fig. 5 DenseNet201 architecture

5) InceptionResNetV2: InceptionResNetV2 is the result of improvements from previous versions of Inception. Overall this architecture consists of a stem and three modules [19]. The stem is an initial set operation performed before introducing the inception blocks. At the same time, the modules contained in this architecture are Inception-A, Inception-B, and Inception-C. The process in InceptionResNetV2 is that after the pre-processing stage, the image enters the model training process. Then it will be continued to the average pooling process and ended by the fully connected layer process as the classification layer.

D. Training Process

The training process is carried out using Google Colab. The Graphical Processing Unit (GPU) is used as a place to speed up the training process. The number of epochs set during training is 100. It means that the machine will learn 100 times. After the training is complete, the training results are visualized using a line chart. Visualization of the results of this training is essential to know the value of accuracy and loss of each epoch. Besides this, it is also used as an analysis material to determine the quality of the resulting model (Overfitting or Underfitting).

E. Testing Model

After the deep learning model is obtained, the next step is to test the model using test data. The number of test data is eight images for each class. So the total test data is 24 images. In the model testing process, loading the image into memory is also carried out to see the predictions generated by the model.

F. Model Evaluation

After going through the training process and getting a deep learning model, the next step is to evaluate the model using a confusion matrix. In the case of classification or supervised learning, the confusion matrix is the most suitable technique to measure model performance. In measuring performance using a confusion matrix, there are 4 (four) terms .as a representation of the results of the classification process, as shown in Table II. The four terms are True Positive, True Negative, False Positive, and False Negative. Then based on the values of TN, FP, FN, TP can be obtained the value of accuracy, precision and recall, and F1 Score.

TABLE II
CONFUSION MATRIX FOR BINARY CLASSIFICATION [20]

Data Class	Classified as pos	Classified as neg
pos	True Positive (TP)	False Negative (FN)
neg	False Positive (FP)	True Negative (TN)

Accuracy describes how accurate the model is in making predictions correctly. Calculating the value of accuracy can be done using (1). The precision value describes the number of correctly classified positive category data divided by the total data classified as positive. Precision can be obtained by using (2). Meanwhile, recall shows how many the system correctly classifies percent of the positive category data. The recall value is obtained by using (3). Finally, the F1 Score is a weighted comparison of the average precision and recall. The recall value is obtained by (4).

$$Accuracy = \frac{(TP+TN)}{(TN+FP+FN+TN)} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F1\ Score = 2x\frac{Recall\ x\ Precision}{Recall + Precision} \quad (4)$$

III. RESULTS AND DISCUSSION

Each pre-trained model is trained using the same parameters, namely using 100 epochs. The image size used is 224x224 pixels, and the total batch size is 128. The history of the training process for each model is shown in Figures and Figures. The image shows the history of the accuracy of the train data during the training process. It can be concluded that each model has poor accuracy at the beginning of the epoch. However, starting from the 15th epoch, the model's accuracy began to rise except for VGGNet16, where the increase in accuracy tends to be slow. Through the graph in Fig. 7-8, it is also known that each model does not experience overfitting because the accuracy of each model tends to increase.

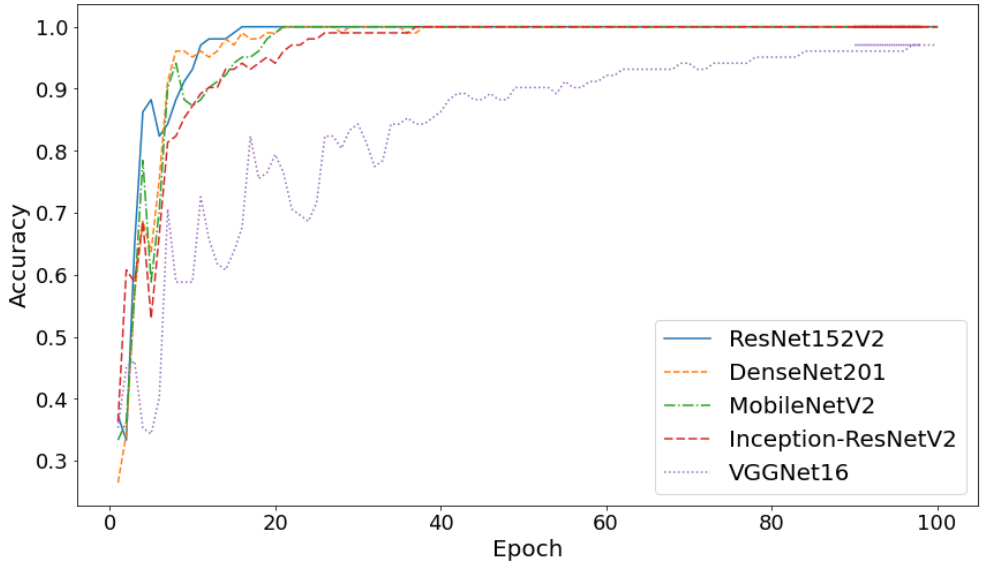

Fig. 7 Accuracy value during the training process

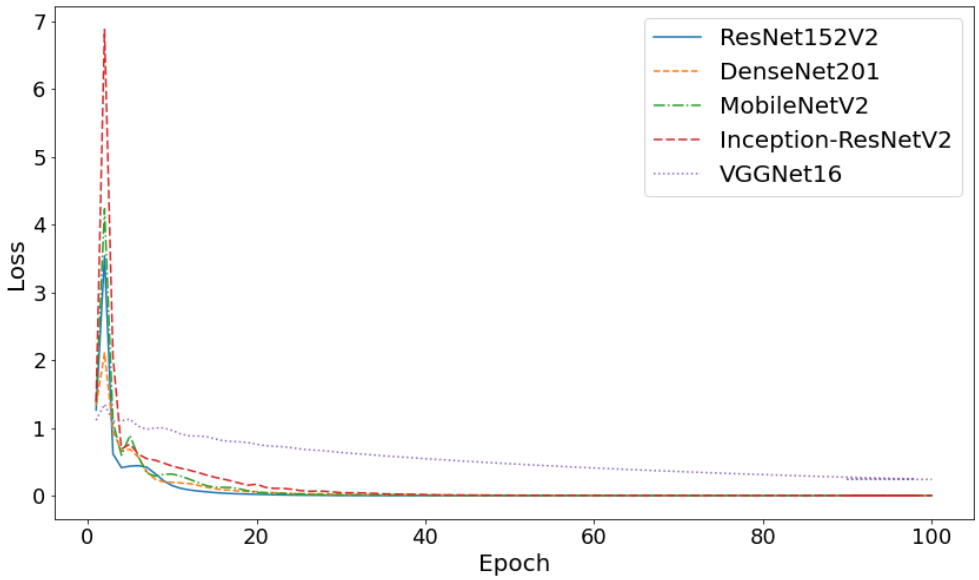

Fig. 8 Loss value during the training process

The result of the training process is a deep learning model for skin disease classification. The model that has been obtained is evaluated using a configuration matrix. The model evaluation results of each architecture are shown in Table III. Based on the data in Table III, it is known that the ResNet152V2 architecture has the highest precision, recall, and F1-score values. It shows that the ResNet152V2 architecture has the smallest error rate value compared to the other four CNN architectures.

Based on Table III, data visualization can be made to determine the best model accuracy. Visualization of the accuracy values of the five pre-trained models is shown in Figure 9. Based on the graph in Fig. 9, it is known that the accuracy value of the ResNet152V2 architecture is the highest at 95.83%. So for the classification of skin disease images, the model suitable for deployment is the model of the ResNet152V2 architecture.

The training process is carried out on Google Colab using the GPU as a hardware accelerator. The number of epochs used during the training is 100. The visualization results of the training time for each architecture are shown in Fig. 10. Based on the graph in Fig. 10, it is known that MobileNetV2 has the fastest training time. The number of layers in the MobileNetV2 architecture is not too many. In addition, the MobileNet architecture is indeed used for devices that have limited resources [21].

A way to speed up training without compromising model accuracy is to use dropout. Dropout works by reducing the complexity of the neural network model without changing the model's architecture [22]. The dropout parameter used to reduce complexity is 20%. Then a callback is also used to stop the training process when the model accuracy has reached 95%. As a result, after using dropouts and callbacks, the training time is almost 50% faster. The training time for MobileNetV2 and ResNet152V2 is the same, namely 98 seconds. Comparison of training time after using dropout and callback can be seen in Fig. 11.

TABLE III
COMPARISON PRECISION, RECALL DAN F1-SCORE

Architecture	Accuracy	Precision	Recall	F1-score
VGGNet16	0,541	0,683	0,583	0,58
MobileNetV2	0,791	0,86	0,833	0,826
InceptionResNetV2	0,875	0,89	0,876	0.873
ResNet152V2	**0,958**	**0,963**	**0,96**	**0,956**
DenseNet201	0,791	0,833	0,793	0,78

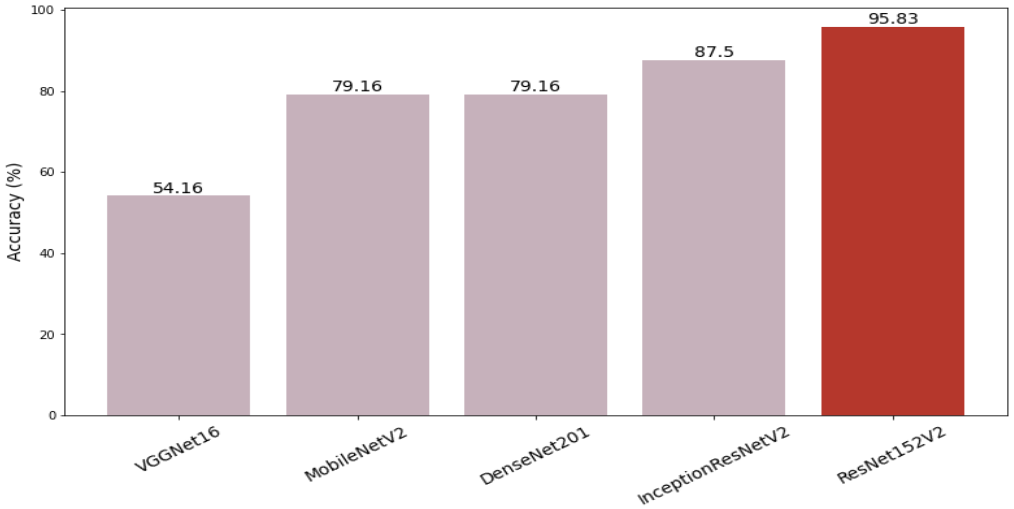

Fig. 9 Accuracy Comparison

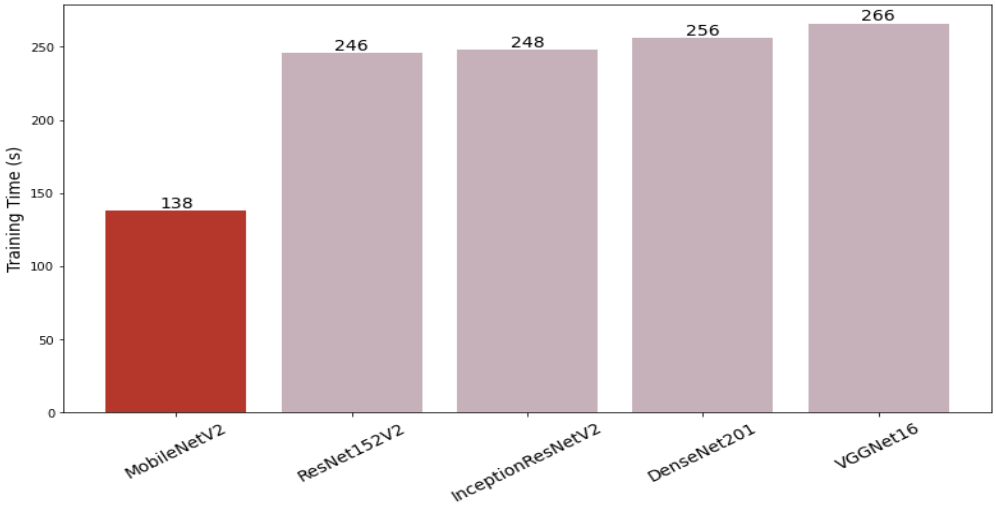

Fig. 10 Training Time Comparison

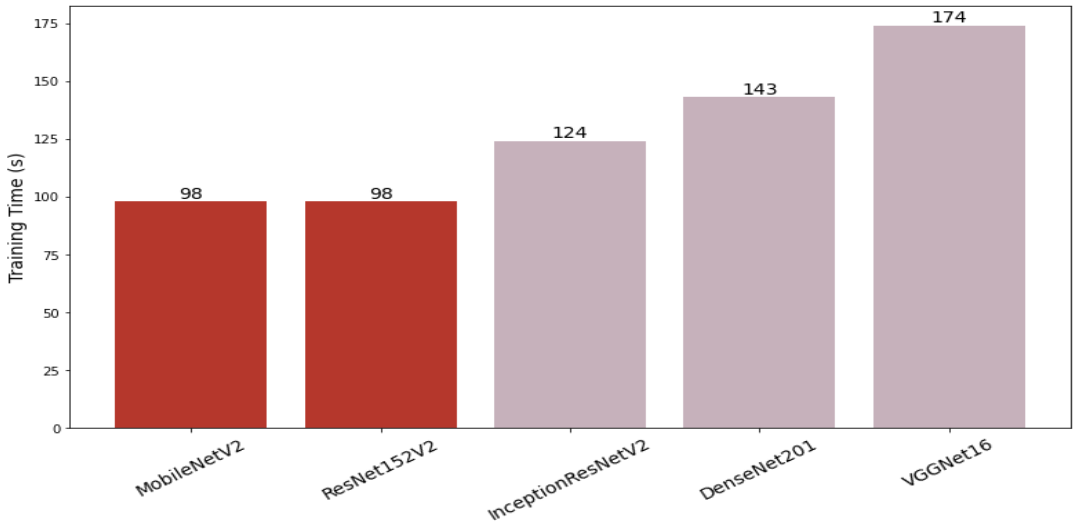

Fig. 11 Comparison of training times using dropouts and callbacks

ResNet152V2 model training time is much reduced because ResNet152V2 achieves 95% accuracy faster than other models. It is reasonable because ResNet152V2 is specifically for object identification in the image. Using dropout also speeds up ResNet152V2 training time because the number of hidden layers is reduced.

IV. CONCLUSION

One way to overcome the limited number of datasets is to use the Pre-Trained Model. Because the Pre-Trained Model already stores various patterns of training results from thousands of images. This study used five Pre-Trained CNN models, namely VGGNet16, MobileNetV2, InceptionResNetV2, ResNet152V2, and DenseNet201, to build a new CNN network architecture. The use of the Pre-Trained model was carried out due to the limited number of skin disease datasets. Determining the Pre-Trained model that has the best performance is to a comparison of the confusion matrix and training execution time. After testing, the results show that ResNet152V2 has the highest accuracy, precision, recall, and F1 scores, namely 95.84%, 0.963, 0.96, 0.956. Then the fastest training execution time is MobileNetV2. However, the use of dropouts and callbacks can also speed up the training time for ResNet152V2 to be the same as MobileNetV2, which is 98 seconds.

REFERENCES

[1] M. K. Hasan, M. A. Alam, D. Das, E. Hossain, and M. Hasan, "Diabetes prediction using ensembling of different machine learning classifiers," *IEEE Access*, vol. 8, pp. 76516–76531, 2020.

[2] D. Sisodia and D. S. Sisodia, "Prediction of Diabetes using Classification Algorithms," *Procedia Comput. Sci.*, vol. 132, no. Iccids, pp. 1578–1585, 2018.

[3] J. Brownlee, *Deep Learning for Computer Vision: Image Classification, Object Detection, and Face Recognition in Python*, 1.8. Machine Learning Mastery, 2019.

[4] D. Haritha, N. Swaroop, and M. Mounika, "Prediction of COVID-19 Cases Using CNN with X-rays," *Proc. 2020 Int. Conf. Comput. Commun. Secur. ICCCS 2020*, 2020.

[5] M. Heidari, S. Mirniaharikandehei, A. Z. Khuzani, G. Danala, Y. Qiu, and B. Zheng, "Improving the performance of CNN to predict the likelihood of COVID-19 using chest X-ray images with preprocessing algorithms," *Int. J. Med. Inform.*, vol. 144, no. September, p. 104284, 2020.

[6] S. Rajaraman *et al.*, "Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images," *PeerJ*, vol. 2018, no. 4, pp. 1–17, 2018.

[7] Z. Akkus *et al.*, "Predicting 1p19q Chromosomal Deletion of Low-Grade Gliomas from MR Images using Deep Learning," pp. 1–7, 2016, [Online]. Available: *http://arxiv.org/abs/1611.06939*. [Accessed 6 December 2021].

[8] E. S. S. Daili, Sri Linuwih Menaldi, and I Made Wisnu, *Penyakit Kulit yang Umum di Indonesia*, I. Jakarta: PT Medical Multimedia Jakarta, 2006.

[9] T. Shanthi, R. S. Sabeenian, and R. Anand, "Automatic Diagnosis of Skin Diseases Using Convolution Neural Network," *Microprocess. Microsyst.*, vol. 76, p. 103074, 2020.

[10] N. Hameed, A. M. Shabut, and M. A. Hossain, "Multi-Class Skin Diseases Classification Using Deep Convolutional Neural Network and Support Vector Machine," in *International Conference on Electronics, Communication and Aerospace Technology (ICECA 2018)*, 2019, pp. 1–7.

[11] J. Velasco *et al.*, "A Smartphone-Based Skin Disease Classification Using MobileNet CNN," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 8, no. October, pp. 2–8, 2019.

[12] O. Rochmawanti, F. Utaminingrum, and F. A. Bachtiar, "Analisis Performa Pre-Trained Model Convolutional Neural Network dalam Mendeteksi Penyakit Tuberkulosis," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 4, p. 805, 2021.

[13] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5455–5516, 2020.

[14] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *International Conference on Learning Representations 2015*, 2014, pp. 1–14.

[15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[16] Y. Hu, A. Huber, J. Anumula, and S.-C. Liu, "Overcoming the vanishing gradient problem in plain recurrent networks," 2018, no. Section 2, pp. 1–20.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, vol. 2016-Decem, pp. 770–778.

[18] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017, vol. 2017-Janua, pp. 2261–2269.

[19] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z.

Wojna, "Rethinking the Inception Architecture for Computer Vision," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, vol. 2016-Decem, pp. 2818–2826.

[20] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, 2009.

[21] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," in *International Conference on Learning Representations 2017*, 2017, pp. 1–9.

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 56, p. 1929−1958, 2014.