International Journal on Advanced Science Engineering Information Technology

Vol.13 (2023) No. 4
ISSN: 2088-5334

Mobile Skin Disease Classification using MobileNetV2 and NASNetMobile

Idhawati Hestiningsih [a], Afandi Nur Aziz Thohari [a,*], Kurnianingsih [a], Nur Diyana Kamarudin [b]

[a] Department of Electrical Engineering, Politeknik Negeri Semarang, Prof. Soedarto SH, Semarang, Indonesia
[b] Faculty of Science and Defence Technology, National Defence University of Malaysia, Kem Perdana Sungai Besi, Malaysia
Corresponding author: *afandi@polines.ac.id

Abstract— **The people of Indonesia often suffer from three diseases: tinea versicolor, ringworm, and scabies. Most Indonesians cannot distinguish the type of skin disease they suffer from because some have the same characteristics and patterns. Therefore, this study built an M-Health application to predict skin diseases. The prediction process uses a deep learning model deployed in the smartphone. The challenge of this study is the limited number of datasets because the data used is personal and requires permission from the patient or hospital. Therefore, transfer learning is used to overcome these data limitations. The transfer learning process in this study uses two pre-trained models, MobileNetV2 and NASNetMobile. To obtain a model with high accuracy, performed modifications to the architecture MobileNetV2 and NASNetMobile. The test results showed that MobileNetV2 performs best using a learning rate of 0.0005 and activation function ELU. While NASNetMobile produces the best performance using a learning rate of 0.0001 and activation function ReLU6. The test results using data from gallery smartphones show that NASNetMobile has an accuracy of 91.6%, while MobileNetV2 has an accuracy of 88.9%. They were testing using the camera in real-time, which resulted in accuracy that was not as accurate as if using data from the gallery. Accuracy using a smartphone camera shows an accuracy of 75% when using NASNetMobile and 72.2% when using MobileNetV2. Accuracy will increase when using a flashlight while capturing objects. Based on the testing results using a flashlight, the NASNetMobile accuracy value increased to 80.5%, while MobileNetV2 accuracy changed to 77.8%.**

Keywords— **M-Health; skin disease; transfer learning; MobileNetV2; NASNetMobile.**

Manuscript received 3 Sep. 2022; revised 3 Oct. 2022; accepted 7 Nov. 2022. Date of publication 31 Aug. 2023.

I. INTRODUCTION

Indonesia is a tropical country with high humidity, which allows the emergence of various skin diseases caused by fungi. [1]. In addition to fungi, Indonesian people often suffer skin diseases from a dirty or unhealthy environment [2]. The people of Indonesia often sustain three skin diseases: tinea versicolor, scabies, and ringworm [3]. These three diseases are straightforward to spread if not treated immediately. Although it is not classified as a dangerous disease, it worsens and can interfere with activities if it is not treated immediately. Most Indonesians cannot distinguish the type of skin disease they suffer from because several skin diseases have the same characteristics or patterns. For example, scabies and ringworms have almost the same pattern and color.

Therefore, a mobile health application (M-Health) was built to classify skin diseases in this study. The purpose of making the M-Health application is to make it easier for users to identify skin diseases anywhere and anytime. Because according to a survey from GSMA Intelligence in January 2022, there were 370.1 million smartphone connections out of 277.7 million Indonesians [4].

This M-Health application can be used offline or does not require an internet connection. Users only need to point the smartphone camera at the skin disease object they want to identify. The M-Health application uses deep learning technology to classify skin diseases. The classification process is carried out by matching the characteristics and patterns of the disease with the model that has been built. Several smartphone applications for diagnosing skin diseases are available on the Google Play Store. When typing the keyword "Skin Disease Apps" in the search field of the play store, several applications such as Medgic [5], Model Dermatology-Skin Disease [6], All Skin Diseases [7], and Treatment AI Skin Disease Detection[8] will appear.

These applications use artificial intelligence technology to predict skin diseases using smartphone cameras. However, the application (Medgic, Model Dermatology-Skin Disease, All Skin Diseases, and Treatment AI Skin Disease Detection) cannot be justified for its validity because it has not been through testing and comprehensive study. Research on developing skin disease classification applications using deep learning technology has been carried out several times. One of them is a study conducted by Wibowo et al. [9] regarding the classification of skin cancer using an Android smartphone. The study modified the parameters of MobileNetV2 until the optimal accuracy value was obtained.

Then research from Srinivasu et al. [10] performed skin cancer classification using MobileNetV2 and LSTM. The test results show that the accuracy of the resulting model is better than the model on the MobileNetV1 and VGGNet architectures. Then there is research on developing mobile applications for skin disease detection using MobileNet [11]. As a result, the application can predict the type of skin disease through images taken from the smartphone gallery with an accuracy of 91%.

Another study used ResNet152 and InceptionResNetV2 pre-trained models to detect five skin diseases, including acne, blackheads, dark-circle and spots [12]. The accuracy value of additional triplet loss function parameters is 87.42%, and the sensitivity is 97.04%. Another multi-class study carried out by Goceri [13] reported MobileNetV2 to develop skin disease diagnosis applications. The application can diagnose five types of skin diseases caused by cancer and tumors with an accuracy above 90%.

Based on the results of a review of studies [9]-[13], it can be concluded that specific testing regarding the model's accuracy on smartphones has not been carried out. Testing is only done using the confusion matrix on google collab or Jupiter notebook. The research object used in previous studies is skin diseases caused by cancer and tumors. This study discusses using deep learning to classify skin diseases caused by fungi and mites.

In this study, the dataset was taken manually through a search on google images and the dermnetnz.org website. The number of datasets used in this study is limited because it requires a permit to retrieve data from the hospital. Therefore, in the classification process, transfer learning techniques are used. Transfer learning is one solution to overcome the limited data when classifying images using deep learning [14]. One of the transfer learning processes is using pre-trained models to build a new CNN architecture. In this study, two pre-trained models were used, namely MobileNetV2 and NASNetMobile. The two pre-trained models were chosen because they can build CNN models on devices with limited resources, such as smartphones, single-board computers, etc [15].

This study compares the performance of the models produced by MobileNetV2 and NASNetMobile. To increase the accuracy of the model, modifications were made to the architecture of the pre-trained model. Modifications are made using hyperparameter tuning on the neural network layer. The challenge of this research is to choose the appropriate parameter values to produce an accurate model for image classification on mobile devices. The program code in this study is available and publicly accessible at the link address https://github.com/afandi354/Skin-Disease-Classification-Deep-Learning.

II. Materials and Method

The research method in this study is shown in Figure 1. There are five main stages to making this M-Health application: collecting datasets, pre-processing, feature extraction, evaluation, and exporting models into a format that mobile devices can read. More details about the proposed method can be seen in Figure 1.

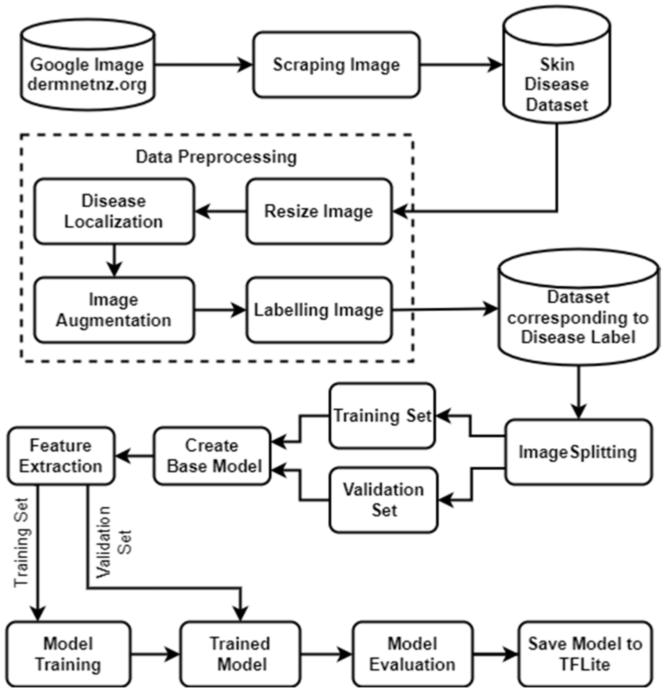

Fig. 1 Our Purpose Method

A. Collecting Dataset

The dataset comes from Google Images and the dermnetnz.org website. The total number of datasets used is 180 images. The percentage of the dataset used for training is 80%, while for testing is 20%. Based on this percentage, the number of training data is 144 images, while the number of testing data is 36. The dataset is grouped into scabies, ringworm, and tinea versicolor. The display of the skin disease dataset can be seen in Figure 2.

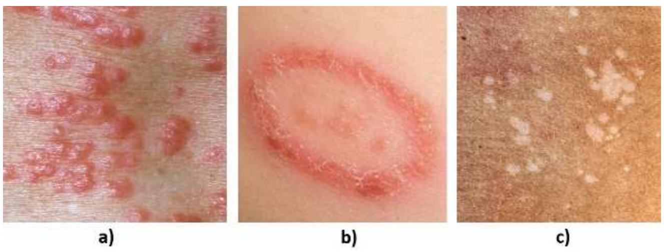

Fig. 2 Skin Disease Datase: a) Scabies, b) Ringworm, c) Tinea Versicolor

Ringworm has a characteristic red color, ring-shaped that extends, and there are scales from the skin. While scabies is in the form of lumps, there are red spots. Tinea versicolor is white, wide in shape, and looks like white spots at first glance. The dataset collection process is problematic because it is not found on dataset provider sites such as Kaggle, world data bank, UC Irvine machine learning repository, Etc. Therefore, the authors searched themselves from Google Images and the dermnetnz.org website by downloading them one by one. The number of datasets that have been successfully compiled is minimal. Therefore, an image augmentation process is used to increase the number of datasets [16].

B. Pre-processing Process

Pre-processing is the most challenging process because it has to normalize the data. The normalization process is carried out by uniform image resolution. The stages carried out in pre-processing are image resizing, localization, image augmentation, and labeling. The result of pre-processing is an image ready to be processed into vector form.

1) Resize Image: The images used as datasets have different sizes. Therefore, it is necessary to equalize the image size to make it easier for machines to recognize objects. This study resizes the image so that all processed images have the same size of 224x224.

2) Localization: Detection of objects in the image requires a localization process. Localization aims to get the Region of Interest (ROI) of the object you want to detect [17]. Through the localization process, the machine will find it easier to pick up the pixels needed to detect objects and eliminate unnecessary ones.

3) Image Augmentation: Image augmentation is a technique to increase the amount of data by changing the image structure of the existing image. Changes in image structure are done by changing the pixel image values. There are six augmentation techniques implemented to increase the number of datasets. The augmentation techniques include rescaled, rotation, zoom, horizontal flipping, random shift, and edge enhancement. The augmentation process produces a new image with the same features as the original [18]. The number of images after the augmentation process is shown in Table 1.

TABLE I
DETAIL OF SKIN DISEASE DATASET

Skin Disease	**Original Image**	**Training (80%)**	
		Before Augmentation	**After Augmentation**
Tinea Versicolor	48	38	228
Scabies	48	38	228
Ringworm	48	38	228

4) Labeling: The last step of the pre-processing process is labeling. Each image is labeled in this process to facilitate training and disease classification if new data is available. Labeling a dataset is accomplished by organizing each image into a directory. The directory name has been changed to match the name of the skin diseases. Enter the image according to the name of the newly formed directory. The Python program will automatically detect that there are three classes of objects to recognize. The Python program is also used for labeling, and the labeling results are in the form of a txt file with the name of the skin diseases.

C. Image Splitting

After the pre-processing stage, the image has a uniform size and shape format to be further processed in the feature extraction process. However, images must be grouped into training and validation sets before entering feature extraction to facilitate the model validation process. A comparison of the training and validation sets is 80% versus 20%.

D. Create Base Model

At this stage, the process of making the base model of the pre-trained model architecture is carried out. This study uses two pre-trained models to detect skin diseases, namely MobileNetV2 and NASNetMobile.

1) MobileNetV2: MobileNetV2 is a pre-trained CNN model used to build machine-learning models on devices with limited resources [19]. The MobileNetV2 architecture has fewer layers and fewer processing parameters. The TensorFlow lite model produced by MobileNetV2 is small, so the size of the .apk file installed on Android also has a small size. MobileNetV2 has been used several times for object detection on Android smartphones. Examples are used to detect skin cancer [20] and face mask detectors [21]. However, some modifications to the parameters of MobileNetV2 are needed to produce higher accuracy—some modifications of MobileNetV2 in this study as shown in Figure 3.

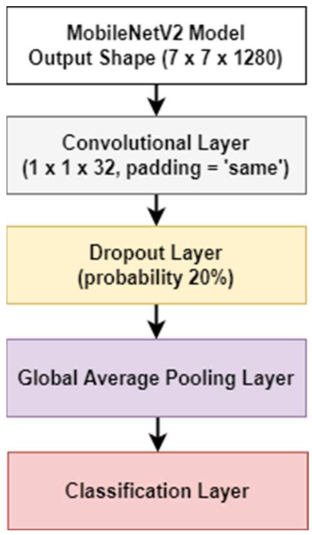

Fig. 3 Purpose CNN Architecture using MobileNetV2 Pre-Trained Model

The CNN architecture referred to in Figure 3 has several additional parameters. One of the added parameters is dropout and padding. The function of adding dropout parameters is to reduce the number of neurons in the hidden layer. The dropout value installed is 20%, meaning 20% of the neurons will be deleted. In comparison, the padding settings are the same so that the output dimensions remain the same as the input dimensions or do not decrease drastically.

2) NASNetMobile: NASNet is a pre-trained CNN model consisting of building blocks on a small dataset and then transferring the block to a larger dataset. The advantage of the NASNet model is that it has a smaller model size and lower complexity (FLOPs) [20]. The NASNet model results from training over one million images from the ImageNet dataset. The complexity of the NASNet architecture can be divided into two types, NASNetLarge and NASNetMobile. In this study, NASNetMobile was used for the image classification of skin diseases. NASNetMobile is a development of the NASNet architecture, consisting of two cells, the Normal Cell and the Reduction Cell [22]. Normal cells are Convolutional cells that return a highlight outline of the exact measurement. At the same time, the reduction cells are Convolutional cells that return an included outline where the included outline's tallness and width are diminished by a calculation of two. More details about the NASNetMobile architecture can be seen in Figure 4.

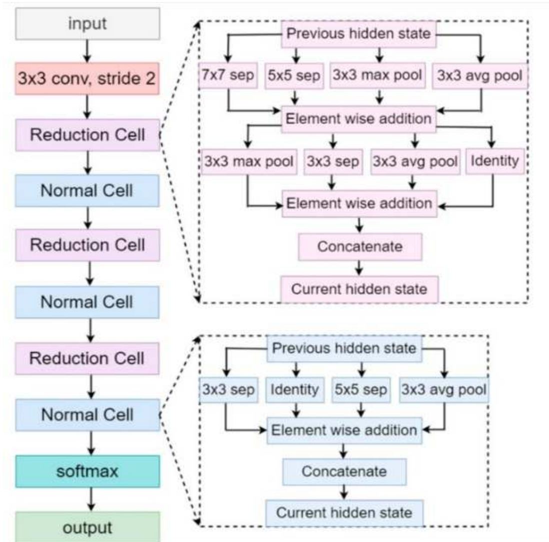

Fig. 4 NASNetMobile Architecture [23]

The reason for using the NASNetMobile model is that it can maintain accuracy even though the number of datasets used is limited [24]. NASNetMobile can accurately classify objects using several datasets [25]. The NASNetMobile architecture used in this study has been modified by adding padding and dropout parameters, as shown in Figure 5.

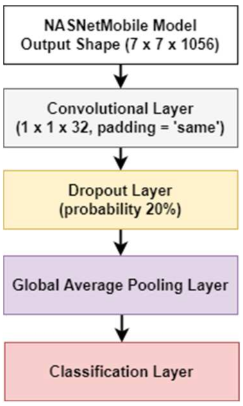

Fig. 5 Purpose CNN Architecture using NASNetMobile Pre-Trained Model

The padding parameter used is the same, and the dropout value used is 20%. After building the CNN architecture, feature extraction and training will be carried out to obtain the model. If the training model has low accuracy, the padding and dropout parameter values can be changed for optimal accuracy.

E. Feature Extraction

The feature extraction stage aims to take the skin disease's shape, color, and pattern characteristics for further analysis [26]. At this stage, the image has been converted into a one-dimensional matrix and is ready to enter the training process. The primary process of feature extraction is image classification based on disease characteristics. Several parameters were entered, including activation, batch size, and loss functions. The value of these parameters can be changed to get the best accuracy value.

F. Model Training

The model is generated after going through the training process. In order to avoid overfitting the model, a callback function is used. The callback function mechanism checks the accuracy value when it reaches a sure accuracy [27]. In this study, the training process will stop if the accuracy has reached 98%.

G. Model Evaluation

At this stage, testing the model that has been produced is carried out. The test uses a confusion matrix to determine accuracy, precision, and sensitivity. The data used for testing the confusion matrix is testing data, which consists of 36 images. Calculating the value of accuracy, precision, and sensitivity is shown in equations 1 and 3 [28].

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

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

$$Sensitivity = \frac{TP}{(TP + FN)} \tag{3}$$

Based on the confusion matrix equations 1 to 3, the model with the best performance was obtained for the classification of skin diseases. The model with the best performance is deployed to an Android device, then the percentage of successful skin disease classification is calculated. We calculated the model's accuracy on an Android device using Equation 4. The percentage of accuracy (p) is generated by comparing the number of successful experimental data (n) with the amount of observation data (m).

$$p = \frac{n}{m} \, x \, 100\% \tag{4}$$

H. Convert Model to TFLite

TensorFlow Lite is a library from TensorFlow that deploy machine-learning models to mobile devices [29]. The model format generated by TensorFlow lite is a file with the *.tflite extension. The training model on google collab is converted into tflite format to be deployed to Android devices. Furthermore, testing is carried out to determine the effect of environmental resources on the development of mobile applications on the model's accuracy. The test takes direct data from the smartphone camera and images stored in the gallery.

III. Results and Discussion

Code writing, training, and testing are done on Google Collab. The training process on google collab uses GPU mode to make it faster. There is a difference in training duration between MobileNetV2 models and NASNetMobile. MobileNetV2 has a shorter duration than NASNetMobile in the training model process. MobileNetV2's training duration is faster than NASNetMobile's, and MobileNetV2 has fewer parameters than NASNetMobile. Because the greater the number of parameters, the more complex the computations performed by the machine will be.

In this study, two models were obtained from the training process of two pre-trained models. Both machine learning models were tested using several test scenarios to get the model with the best performance. The test scenarios of this research are testing the use of learning rates, activation functions, and the model's accuracy on Android devices.

A. Testing The Use of Learning Rate

The learning rate is a machine learning hyperparameter used to calculate weight correction during training. The use of the learning rate must be optimal because if it is too little, it will prolong the training process, while if it is too much, it will cause instability in the training process [30]. This test selects the learning rate value to get the model with the best performance. The test scenario is to change the learning rate value and then calculate the confusion matrix for each resulting model.

Tables 2 and 3 are the results of comparing models produced from 5 different learning rates. Based on the data in Table 2, it is known that the learning rate value of 0.0005 in the MobileNetV2 model has the best performance.

TABLE II
LEARNING RATE COMPARISON MOBILENETV2 MODEL

Learning Rate	MobileNetV2		
	Accuracy (%)	Precision (%)	Sensitivity (%)
0.005	83.3	84	83.3
0.001	77.8	80	78
0.0005	83.3	86	83.6
0.0001	80.5	84.3	80.6

While in the NASNetMobile model, the learning rate that produces the best performance is 0.0001. Table 3 shows the results of the comparison of learning rates on NASNetMobile.

TABLE III
LEARNING RATE COMPARISON OF NASNETMOBILE

Learning Rate	NASNetMobile		
	Accuracy (%)	Precision (%)	Sensitivity (%)
0.005	77.8	81.6	77.6
0.001	77.8	80.6	78
0.0005	77.8	80.6	78
0.0001	80.5	83.6	80.3

B. Testing The Use of Activation Function

Based on the learning rate test results, it is known that the MobileNetV2 model has the best performance when using a learning rate value of 0.0005. While in the NASNetMobile model, the learning rate that produces the best performance is 0.0001. Therefore, in the activation function test, the learning rate is set at 0.0005 for MobileNetV2, and the learning rate is 0.0001 for NASNetMobile.

The training process is carried out using a callback function to avoid overfitting. The model accuracy limit is 98%, and if it has passed the set accuracy limit, the training will stop automatically. The number of epochs for each activation function is different. The number of epochs for each activation function when the training accuracy reaches 98% is shown in Figure 6.

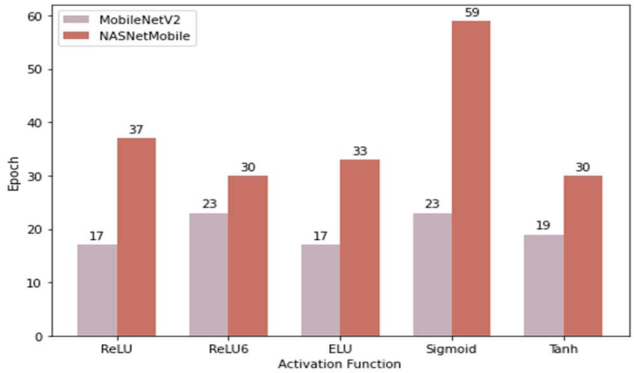

Fig. 6 The Number Of Epochs In Each Activation Function

Figure 6 explains that MobileNetV2 has a faster duration than NASNetMobile in generating models. The epoch value can represent the duration of the training. The higher the number of epochs, the longer the training duration will be. The cause of MobileNetV2 has a small epoch value because MobileNetV2 has fewer layers and parameters than NASNetMobile. These factors make processing faster because the computations are light. The visualization of the training model process on MobileNetV2 is shown in Figure 7. While the visualization of the training model on NASNet is shown in Figure 8.

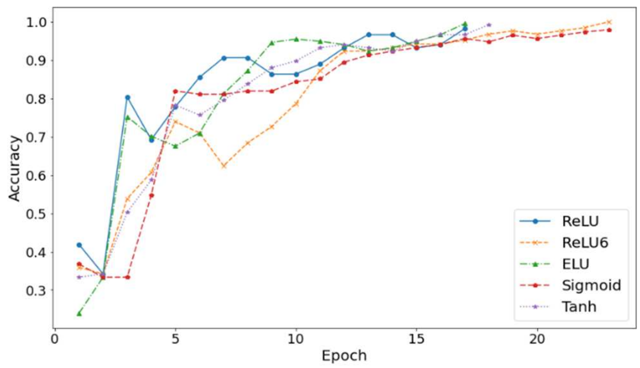

Fig. 7 The Validation Accuracy using MobileNetV2

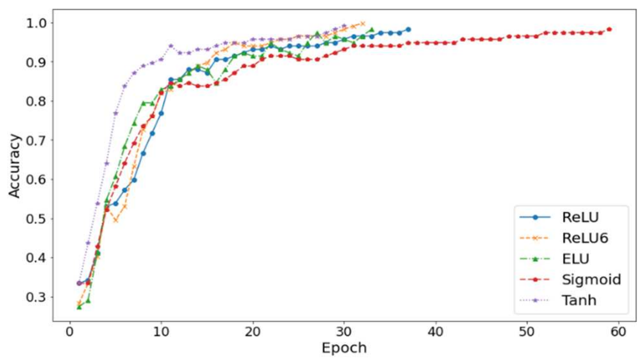

Fig. 8 The Validation Accuracy using NASNetMobile

Figures 7 and 8 show that the more epochs, the higher the accuracy. Then based on the graphs in Figures 7 and 8, it is known that the model built does not experience overfitting and underfitting. The status is known from the accuracy value, which is constantly increasing, and there is no tendency for the accuracy value to decrease [31].

The activation function plays a role in increasing the accuracy of the model. Selecting the proper activation function can increase the model's accuracy [32]. In this study, linear and non-linear activation functions were used. ELU, ReLu, and ReLU6 represent linear activation functions. Meanwhile, non-linear functions are represented by Sigmoid and Tanh. Furthermore, the results of testing the model with the activation function are shown in Table 4 and Table 5.

TABLE IV
ACTIVATION FUNCTION COMPARISON OF MOBILENETV2 MODEL

Activation Function	MobileNetV2, Learning Rate = 0.0005		
	Accuracy (%)	Precision (%)	Sensitivity (%)
ReLU	86.1	87.6	86
ReLU6	83.3	86.3	82.6
ELU	88.9	90.6	89
Sigmoid	75	80.3	75
Tanh	83.3	86.3	82.6

Table 4 is the result of the confusion matrix calculation on MobileNetV2. Based on the data in Table 4, it is known that the ELU activation function has the best performance. So, it can be concluded that the MobileNetV2 model has the best performance for skin disease classification if the Activation Function ELU and Learning Rate are 0,0005. While on NASNetMobile, the model produces the best performance using the ReLU6 activation function. This result shows that linear functions make the dataset used easier to learn. More details about the results of testing the activation function on NASNetMobile are shown in Table 5.

TABLE V
ACTIVATION FUNCTION COMPARISON OF NASNETMOBILE MODEL

Activation Function	NASNetMobile, Learning Rate = 0.0001		
	Accuracy (%)	Precision (%)	Sensitivity (%)
ReLU	83.3	87	83.6
ReLU6	91.6	92	91.6
ELU	86.1	87.6	86
Sigmoid	77.8	81.3	77.6
Tanh	80.5	83	80.6

C. Model Accuracy Testing on Android Devices

Test scenarios on Android devices are carried out using two methods. The first method is to retrieve image data from the gallery. The second method takes an image directly from the smartphone camera. The data used for testing the performance of the model on Android devices is testing data that has been stored in the Android gallery. There are 36 test data, each consisting of images of tinea versicolor, scabies, and ringworm. The test results load images from the gallery on Android devices are shown in Table 6.

TABLE VI
TEST RESULT ON ANDROID DEVICE USING IMAGE DATA FROM THE GALLERY

Model	Correct Prediction	Wrong Prediction	Accuracy (%)
MobileNetV2	32	4	88,9
NASNetMobile	33	3	91,6

Based on the data in Table 6, it is known that the accuracy of the NASNetMobile model is better than MobileNetV2 when it is implemented on Android devices. The accuracy value of the NASNetMobile model is 91.6%, while MobileNetV2 is 88.9%. The display of skin disease classification results on M-Health is shown in Figure 9. Skin disease data in Figure 9 is taken from the smartphone gallery.

Fig. 9 The Result of Classification Using Image from Smartphone Gallery

Based on the tests shown in Figure 9, it is known that the M-Health application works well. The image classification feature from the gallery works well and has high accuracy. The next step is to test the classification of skin diseases using a cell phone camera. The 36 test photographs utilized consisted of 12 scabies images, 12 ringworm images, and 12 tinea versicolor images.

Each image is captured twice, there are 72 shot cameras. The test scenario is to point the phone camera at the disease object printed on the paper. The distance between the phone and the object of the disease is 10 cm. Then a test was carried out using a camera flash to determine the effect of light intensity on the model's accuracy.

TABLE VII
TEST RESULT ON ANDROID DEVICE USING IMAGE DATA FROM THE CAMERA

Scenario	Pre-Trained Model Accuracy (%)	
	MobileNetV2	NASNetMobile
Without Flashlight	72.2	75
With Flashlight	77.8	80.5

Table 7 is the result of testing the classification of skin diseases using a cell phone camera. The test results show that the accuracy of the classification of skin diseases using the camera is lower than if the image data is taken from the gallery. Several factors affect these results, such as camera resolution, light intensity, shooting angle, etc. The use of flash on the camera phone affects the accuracy in classifying skin diseases. The flashlight can make the camera more focused and clarify the features of the detected skin disease. So that the accuracy of the application in detecting the disease is more accurate. The results of the classification of skin diseases using the camera on an android phone are shown in Figure 10.

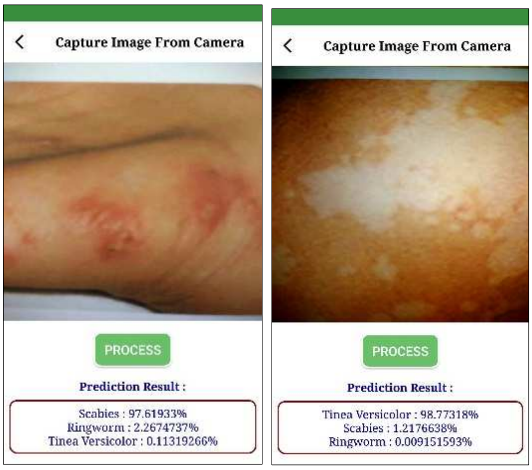

Fig. 10 The Result of Classification Using Image Capture from Camera

The android application that was built can predict skin diseases accurately, as shown in Figure 10. There is a confidence score for each predicted image. The prediction result from the application is the disease with the highest confidence score. Android device specification testing is also performed to assess the application's compatibility. The specifications of five distinct Android devices vary. Table 8 displays the test results for several Android smartphones.

TABLE VIII
TESTING OF ANDROID DEVICE SPECIFICATION

CPU	**RAM**	**Dimensions**	**Operating System**	**Result**
Octa-core 2,84 GHz	8 GB	6.3 inches	Android 12	Success
Octa-core 2,05 GHz	8 GB	6.43 inches	Android 11	Success
Octa-core 2,2 GHz	4 GB	6.6 inches	Android 12	Success
Octa-core 2,2 GHz	6 GB	6.6 inches	Android 11	Success
Quad-core 12 GHz	2 GB	5.45 inches	Android 10	Success

Table 8 shows that the skin disease prediction application that was developed works effectively on Android smartphones. The responsive application's UI adjusts to the screen and can run smoothly with a minimum CPU specification of quad-core. Applications created can operate on the most recent Android version, Android 12.

D. Discussion of The Results

In terms of identifying skin disease, NASNetMobile outperforms MobileNetV2. In future studies, the performance of NASNetMobile should be compared against the most recent version of MobileNet, MobileNetV3. The number of datasets must be increased to improve the model's accuracy. The sorts of skin disorders discovered must also be provided since the apps developed are extremely valuable for preventing and treating skin diseases. The application is now available for download from the Google Play store at the URL https://play.google.com/store/apps/details?id=com.tflite.rice varietiesclassificationmobilenetv1. Furthermore, an iOS version of the skin disease diagnosis program that may be obtained from the Apple Store is required.

IV. CONCLUSION

The test results show that the model has successfully predicted skin diseases through images. The model built using MobileNetV2 has optimal performance using a learning rate of 0.0005 and the ELU activation function. While the model built using NASNetMobile will have optimal performance if the learning rate is 0.0001 and the activation function is ReLU6. Smartphone tests show that NASNetMobile's accuracy is better than MobileNetV2 in predicting skin diseases. NASNetMobile has 91.6% accuracy in predicting skin diseases using images from the smartphone gallery. Meanwhile, MobileNetV2 has 88.9% accuracy in predicting skin diseases using images from the smartphone gallery.

In addition to taking data from the galley, test data is also taken through the smartphone camera in real time. The test results show a decrease in the accuracy of each model. The MobileNetV2 model has an accuracy of 72.2%, while that of the NASNetMobile is 75%. In order to increase accuracy results, the flashlight is used when capturing objects. As a result, MobileNetV2's accuracy increased to 77.8%, while NASNetMobile's accuracy increased to 80.5%. Data retrieval using a smartphone camera needs to consider the camera's resolution, the angle at which the object is taken, the distance between the camera and the object, and the use of flashlights to make objects easier to identify.

REFERENCES

[1] E. S. S. Daili, S. L. Menaldi, and I. M. Wisnu, *Penyakit Kulit Yang Umum Di Indonesia*. Jakarta: PT. Medical Multimedia Indonesia, 2006.

[2] A. Nadiya, R. Listiawaty, and C. Wuni, "Hubungan Personal Hygiene Dan Sanitasi Lingkungan Dengan Penyakit Scabies Pada Santri Di Pondok Pesantren Sa'Adatuddaren," *Contag. Sci. Period. J. Public Heal. Coast. Heal.*, vol. 2, no. 2, p. 99, 2020, doi: 10.30829/contagion.v2i2.7240.

[3] L. Wijaya;, R. Fernando;, and S. Lembar, *Pemeriksaan Penunjang dan Laboratorium Pada Penyakit Kulit dan Kelamin*, First. Jakarta: Universitas Katolik Indonesia Atma Jaya, 2019.

[4] GSMA, "The Mobile Economy 2022," 2022. [Online]. Available: www.gsmaintelligence.com.

[5] Mednet, "Medgic." 2021, Accessed: Aug. 10, 2022. [Online]. Available: https://play.google.com/store/apps/details?id=co.medgic.medgic.

[6] IDerma, "Model Dermatology - Skin Disease." 2022, Accessed: Aug. 09, 2022. [Online]. Available: https://play.google.com/store/apps/details?id=com.phonegap.whichderm.

[7] K. Studio, "All Skin Diseases and Treatment," 2020. https://play.google.com/store/apps/details?id=com.ruthieapps.all.skin.diseases.with.photo.treatment (accessed Aug. 09, 2022).

[8] S. Ayhan, "AI Skin Disease Detection," 2021. https://play.google.com/store/apps/details?id=com.suna.PlantDisease Classification (accessed Aug. 09, 2022).

[9] A. Wibowo, C. A. Hartanto, and P. W. Wirawan, "Android skin cancer detection and classification based on mobilenet v2 model," *Int. J. Adv. Intell. Informatics*, vol. 6, no. 2, pp. 135–148, 2020, doi: 10.26555/ijain.v6i2.492.

[10] P. N. Srinivasu, J. G. Sivasai, M. F. Ijaz, A. K. Bhoi, W. Kim, and J. J. Kang, "Classification of Skin Disease using Deep Learning Neural Networks with Mobilenet V2 and LSTM," *Sensors*, vol. 21, no. 8, pp. 1–27, 2021, doi: 10.3390/s21082852.

[11] J. Velasco *et al.*, "A Smartphone-Based Skin Disease Classification Using MobileNet CNN," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 8, no. 5, pp. 2632–2637, 2019, doi: 10.30534/ijatcse/2019/116852019.

[12] B. Ahmad, M. Usama, C. M. Huang, K. Hwang, M. S. Hossain, and G. Muhammad, "Discriminative Feature Learning for Skin Disease Classification Using Deep Convolutional Neural Network," *IEEE Access*, vol. 8, pp. 39025–39033, 2020, doi: 10.1109/ACCESS.2020.2975198.

[13] E. Goceri, "Diagnosis of skin diseases in the era of deep learning and mobile technology," *Comput. Biol. Med.*, vol. 134, no. April, p. 104458, 2021, doi: 10.1016/j.compbiomed.2021.104458.

[14] F. Zhuang *et al.*, "A Comprehensive Survey on Transfer Learning," in *Proceedings of the IEEE*, 2021, vol. 109, no. 1, pp. 43–76, doi: 10.1109/JPROC.2020.3004555.

[15] F. Saxen, P. Werner, S. Handrich, E. Othman, L. Dinges, and A. Al-Hamadi, "Face attribute detection with mobilenetv2 and nasnet-mobile," *Int. Symp. Image Signal Process. Anal. ISPA*, vol. 2019-Septe, no. C, pp. 176–180, 2019, doi: 10.1109/ISPA.2019.8868585.

[16] K. Maharana, S. Mondal, and B. Nemade, "A Review: Data Pre-processing and Data Augmentation T echniques," *Glob. Transitions Proc.*, vol. 3, no. 1, pp. 91–99, 2022, doi: 10.1016/j.gltp.2022.04.020.

[17] S. Agustin, H. Tjandrasa, and R. V. H. Ginardi, "Deep Learning-based Method for Multi-Class Classification of Oil Palm Planted Area on Plant Ages Using Ikonos Panchromatic Imagery," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 10, no. 6, pp. 2200–2206, 2020, doi: 10.18517/ijaseit.10.6.12030.

[18] R. L. Galvez, E. P. Dadios, A. A. Bandala, and R. R. P. Vicerra, "Object detection in x-ray images using transfer learning with data augmentation," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 9, no. 6, pp. 2147–2153, 2019, doi: 10.18517/ijaseit.9.6.9960.

[19] 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, doi: 10.1109/CVPR.2018.00474.

[20] Y. J. Cheng, Wei Lin, Lu Sun, and Yuan Zhen Liu, "Classification of skin diseases based on improved MobileNetV2," in *2021 33rd Chinese Control and Decision Conference (CCDC)*, 2021, pp. 298–603, doi: 10.1109/CCDC52312.2021.9602387.

[21] L. A. Wu, Y. Wu, and C. Wang, "MobileNet investigation: its application and reproducing edge detectors using depth-wise separable convolution," in *2nd International Conference on Machine Learning and Computer Application*, 2021, pp. 1–6.

[22] A. O. Adedoja, P. A. Owolawi, T. Mapayi, and C. Tu, "Intelligent Mobile Plant Disease Diagnostic System Using NASNet-Mobile Deep Learning," *IAENG Int. J. Comput. Sci.*, vol. 49, no. 1, pp. 216–231, 2022.

[23] Nillmani *et al.*, "Four Types of Multiclass Frameworks for Pneumonia Classification and Its Validation in X-ray Scans Using Seven Types of Deep Learning Artificial Intelligence Models," *Diagnostics*, vol. 12, no. 3, pp. 1–32, 2022, doi: 10.3390/diagnostics12030652.

[24] M. M. Ahsan, K. D. Gupta, M. M. Islam, S. Sen, M. L. Rahman, and M. Shakhawat Hossain, "COVID-19 Symptoms Detection Based on NasNetMobile with Explainable AI Using Various Imaging Modalities," *Mach. Learn. Knowl. Extr.*, vol. 2, no. 4, pp. 490–504, 2020, doi: 10.3390/make2040027.

[25] S. D. Bimorogo, "A Comparative Study of Pretrained Convolutional Neural Network Model to Identify Plant Diseases on Android Mobile Device," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 3, pp. 2824–2833, 2020, doi: 10.30534/ijatcse/2020/53932020.

[26] T. Shanthi, R. S. Sabeenian, and R. Anand, "Automatic diagnosis of skin diseases using convolution neural network," *Microprocess. Microsyst.*, vol. 76, no. 1–8, p. 103074, 2020, doi: 10.1016/j.micpro.2020.103074.

[27] E. Bisong, *Regularization for Deep Learning. In: Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Berkeley: Apress, 2019.

[28] 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, doi: 10.1016/j.ipm.2009.03.002.

[29] A. Singh and R. Bhadani, *Mobile Deep Learning with TensorFlow Lite, ML Kit and Flutter*. Birmingham: Packt Publishing, 2020.

[30] W. El-Shafai *et al.*, "Efficient deep CNN model for COVID-19 classification," *Comput. Mater. Contin.*, vol. 70, no. 3, pp. 4373–4391, 2022, doi: 10.32604/cmc.2022.019354.

[31] X. Ying, "An Overview of Overfitting and its Solutions," *J. Phys. Conf. Ser.*, vol. 1168, no. 2, pp. 1–6, 2019, doi: 10.1088/1742-6596/1168/2/022022.

[32] X. Wang, H. Ren, and A. Wang, "Smish: A Novel Activation Function for Deep Learning Methods," *Electronics*, vol. 11, no. 4, pp. 1–15, 2022, doi: 10.3390/electronics11040540.