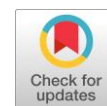


Android skin cancer detection and classification based on MobileNet v2 model



Adi Wibowo ^{a,1,*}, Cahyo Adhi Hartanto ^{a,2}, Panji Wisnu Wirawan ^{a,3}

^a Department of Informatics, Faculty of Sciences and Mathematics, Diponegoro University, Semarang, Indonesia

¹ bowo.adi@live.undip.ac.id; ² cahyo.adhi97@gmail.com; ³ panji@lecturer.undip.ac.id

* corresponding author

ARTICLE INFO

Article history

Selected paper from The 2019 2nd International Symposium on Advanced Intelligent Informatics (SAIN'19) (Fukuoka-Japan, 2-4 September 2019) (<http://sain.ijain.org/2019/>). Peer-reviewed by SAIN'19 Scientific Committee and Editorial Team of IJAIN journal.

Received April 9, 2020

Revised June 08, 2020

Accepted June 18, 2020

Available online July 12, 2020

Keywords

Deep learning
Skin cancer
Android
MobileNet v2

ABSTRACT

The latest developments in the smartphone-based skin cancer diagnosis application allow simple ways for portable melanoma risk assessment and diagnosis for early skin cancer detection. Due to the trade-off problem (time complexity and error rate) on using a smartphone to run a machine learning algorithm for image analysis, most of the skin cancer diagnosis apps execute the image analysis on the server. In this study, we investigate the performance of skin cancer images detection and classification on Android devices using the MobileNet v2 deep learning model. We compare the performance of several aspects; object detection and classification method, computer and Android based image analysis, image acquisition method, and setting parameter. Skin cancer actinic Keratosis and Melanoma are used to test the performance of the proposed method. Accuracy, sensitivity, specificity, and running time of the testing methods are used for the measurement. Based on the experiment results, the best parameter for the MobileNet v2 model on Android using images from the smartphone camera produces 95% accuracy for object detection and 70% accuracy for classification. The performance of the Android app for object detection and classification model was feasible for the skin cancer analysis. Android-based image analysis remains within the threshold of computing time that denotes convenience for the user and has the same performance accuracy with the computer for the high-quality images. These findings motivated the development of disease detection processing on Android using a smartphone camera, which aims to achieve real-time detection and classification with high accuracy.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



1. Introduction

Cancer is one of the many leading causes of human death in the world [1]. A type of cancer that is well known is skin cancer. There were 68,130 cases of skin cancer in America, which caused 8,700 deaths in 2010 [2]. Two examples of skin cancer types are Melanoma and Actinic Keratosis. World Health Organization (WHO) reported 2-3 million cases of non-melanoma skin cancers, and 130 thousand cases of melanoma sufferers each year [3]. In order to reduce the number of late diagnoses, cancer detection using technology assistance is needed.

A form of technology that has a rapid development today is a smartphone or mobile phone. The development of technology in smartphones in the current era introduces new possibilities for disease detection only through smartphones. One part of a smartphone component that has the most significant role in cancer detection is a camera [4]. Images captured by smartphone cameras can be processed by smartphones to detect objects in the picture. In addition, a smartphone as a detector has wireless connectivity, the ability to perform high-resolution photography, and excellent portability. Nowadays,

the detection of objects in imagery can be done using one technique from the branch of machine learning, namely deep learning [5]. Deep learning is a novel machine learning method that is growing rapidly now. Deep learning has a higher level of sensitivity compared to other machine learning methods [6]. Deep learning has been used to analyze biomedical data, such as medical images, biological sequences, and protein structures [5]. Research related to skin cancer classification was carried out by developing a deep neural network with a Google base of inception v3 using a dataset of 2000 images with a division of 374 melanoma, 1372 nevus, and 254 seborrheic keratosis [7]. In 2018, Haenssle *et al.* [8] used Google Inception v4 CNN architecture for the classification of Melanoma.

Skin cancer has an average size of 6 mm [9], which is exceedingly small as compared to the total skin area captured by the camera. Research on skin cancer segmentation was carried out in 2017 using Fully Convolutional Network to decipher the objects of skin cancer [10]. One of the disadvantages of segmentation is only marked the object without object recognition. Therefore, it requires object detection and classification to determine the type of skin cancer.

The latest developments in the smartphone-based skin cancer diagnosis application allow simple ways for portable melanoma risk assessment and diagnosis for early skin cancer detection [11]. Several mobile apps are ready on the app store, such as SkinVison [12], SpotMole [13], Deep Learning for Melanoma [14], and DermIA [15]. These apps provide a simple way of performing a Melanoma risk assessment on outer skin by taking a photo of the skin spot with a smartphone camera. Risk assessment was generated by performing a similarity check between the photo taken and photos of skin cancer. However, several applications (SpotMole, Deep Learning for Melanoma, DermIA) do not seem to have a comprehensive study. Therefore, the accuracy of the applications is unknown.

Moreover, the application also often fails to recognize Melanoma, which is a very high-risk skin problem. Different from other apps, the SkinVision app is more trusted for early detecting skin cancer. The SkinVision app uses a comprehensive study to perform detection. Initially, Skin-Vision is using a rule-based fractal algorithm [16], then followed by analyzing pigmented and non-pigmented lesions [17]. In the image's analysis process, SkinVision requires an internet connection for sending images to the server. The client-server architecture was also offered in the eSkin [18] application to not to burden the computation of image analysis on smartphones and the algorithm updates. The main reason for running the image analysis on the server is the time complexity and error rate when performing the analysis on the smartphone device. However, Melanoma is more common in rural areas where the internet connection is limited. Hence performing image analysis on the smartphone device is required in this situation [19].

MobileNet implements a simple architecture using depth-wise separable convolutions to lightweight deep neural networks [20] for mobile vision applications. MobileNet v2 is the addition of bottleneck layers and shortcut connections [21]. MobileNet is commonly used for object detection. Nevertheless, it is also possible to use MobileNet for classification. In this study, we investigate how the performance of MobileNet v2 running on the smartphone for image analysis of skin cancer detection and classification. Object detection is used to allow the image analysis process with full skin background, while the classification is used to classify the objects that have been cropped according to the training size. The learning rate and epoch parameters were selected to handle the overfitting problem [22]. The code is publicly available through <https://github.com/bowoadi/Melanomax/>. We compare the performance of several aspects; object detection and classification method, computer (Jupyter notebook) and Android-based images analysis, and images acquisition method. Skin cancer Actinic Keratosis and Melanoma are examined to test the proposed method performance. Accuracy, sensitivity, specificity, and running time of the testing methods are used for the measurement.

2. Method

The flowchart of our proposed method is shown in Fig. 1. First, we collected a data set of skin cancer images from isic.org to train MobileNet v2. In this study, we used two types of skin cancer; Actinic Keratosis and Melanoma. Actinic Keratosis is a type of skin cancer that is triggered by the continuous

exposure of ultraviolet radiation. The characteristics of the skin affected by this cancer are crusty, scaly skin with a brownish color, pink, or a combination of these colors [23]. Fig. 2(a) shows Actinic Keratosis originating from the dataset used. Melanoma is a type of malignant cancer that attacks humans in ages ranging from 25 to 50 years old. The cause of Melanoma is known to come from two things, i.e., exposure to UV light and genetic factors [24]. The case of Melanoma is typical in remote areas. The characteristics of Melanoma skin cancer are an irregular shape, consisting of more than one color, itching, and bleeding and can attack anybody parts [19]. Fig. 2(b) shows an example of Melanoma skin cancer. The dataset used in this study was 640 images downloaded from the website <https://isic-archive.com>, five images were downloaded from <https://cancer.org>, and five images from Google Image without regard to age or other factors.



Fig. 1. The flowchart of our proposed method.



Fig. 2. Skin Cancer Images: a) Actinic Keratosis, b) Melanoma.

The second step was splitting the dataset. The 640 images downloaded from the website <https://isic-archive.com> divided into training data, validation data, and test data. Training data contain 200 melanoma and 200 Actinic Keratosis images, while validation data contain 100 Melanoma and 100 Actinic Keratosis images, and Test data contain 20 melanoma and 20 Actinic keratosis images. Training, validation, and test data are different images. Melanoma and Actinic keratosis images with the size information are printed to the paper for Android testing purposes. The printed images were five images downloaded from <https://cancer.org> and five from Google Image.

The third step was ROI determination for manual interpretation to train MobileNet v2. The process produced an XML file containing the coordinates of the object suspected of being cancer. The determination of ROI was done using the LabelImg application that could be downloaded at <https://github.com/tzutalin/labelImg>. Original images with object coordinates were used for object detection training and validation. Then the cropped images based on ROI coordinates were used for the classification training and validation.

The fourth step was the training, where the MobileNet v2 is trained for object detection and classification. The details of both models discussed in the next subsection. The next step was parameter optimization by selecting the learning rate, epoch parameters, activation function, and batch normalization to decrease the overfitting problem. After optimizing the parameters, the achieved model is tested on both computers by using Jupyter notebook and on Android apps. The next step was the MobileNet v2 evaluation. Accuracy, sensitivity, specificity, and running time of the testing data are measured for evaluating the object detection and classification model. The last stage was choosing the most optimized model for Android skin cancer detection and classification.

2.1. MobileNet

MobileNet is a deep learning model that is developed for efficiency and can be implemented on embedded devices or mobile devices such as smartphones without compromising with resources [25]. Fig. 3 defined the MobileNet Architecture for detection and classification. MobileNet is built using the depthwise separable convolution architecture to create a lightweight model [20]. MobileNet v2 was released in two versions, MobileNet v1 and MobileNet v2. The updates on MobileNet v2 are the addition of bottleneck layers and shortcut connections [21]. The bottleneck layers are shown in Fig. 4.

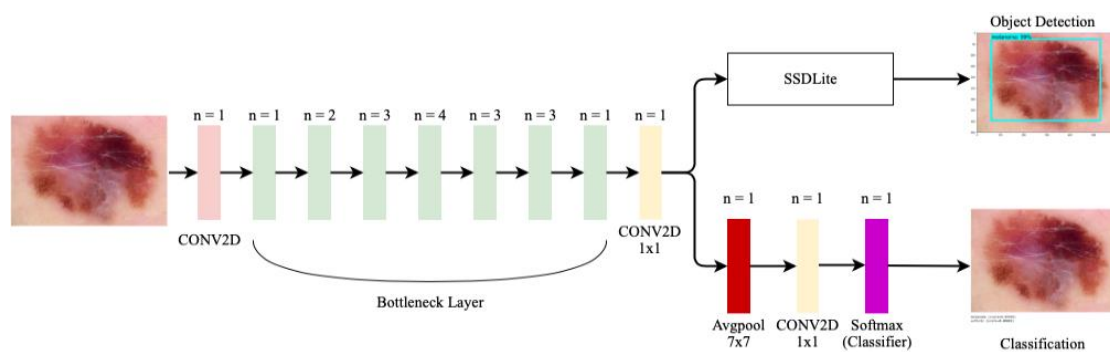


Fig. 3. MobileNet v2 Architecture for object detection and classification.

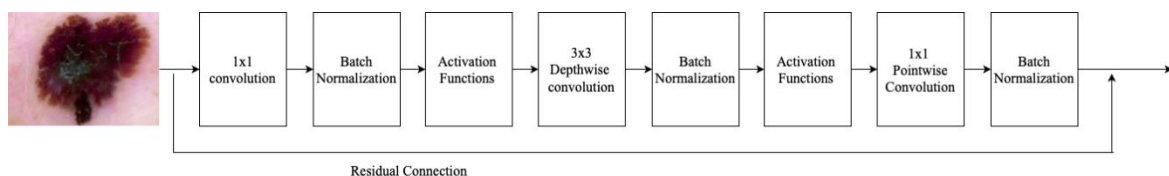


Fig. 4. Bottleneck Layer MobileNet v2.

MobileNet v2 process started with feature extraction using Conv2D with a 3x3 kernel for input images. Then the results from Conv2D entered to 19 bottleneck layers. Bottleneck layers consist of three convolution operations. There are 1x1 convolution, 3x3 depthwise convolution, and 1x1 pointwise

convolution. The 1x1 convolution increases the number of channels to enrich the features. Depthwise separable convolution has two types of convolution layers; depthwise convolution and pointwise convolution, which aim to reduce computing costs. The reduction is executed by separating the feature filtering process at 3x3 depthwise convolution and then the combining feature process at pointwise convolution [20]. Operations on 3x3 depthwise convolution perform by separating all channels on the input, and each channel is convoluted with filters at the 3x3 depthwise convolution layer in the order. Fig. 5(a) shows the example that the red channel in the input is convoluted with the first channel of the depthwise convolution layer first filter. Then, the 1x1 pointwise convolution process is the convolution of all channels on the input that has passed through the 3x3 depthwise convolution layer with all filters at the pointwise convolution layer one by one and in sequence. Fig. 5(b) shows the operation at 1x1 pointwise convolution.

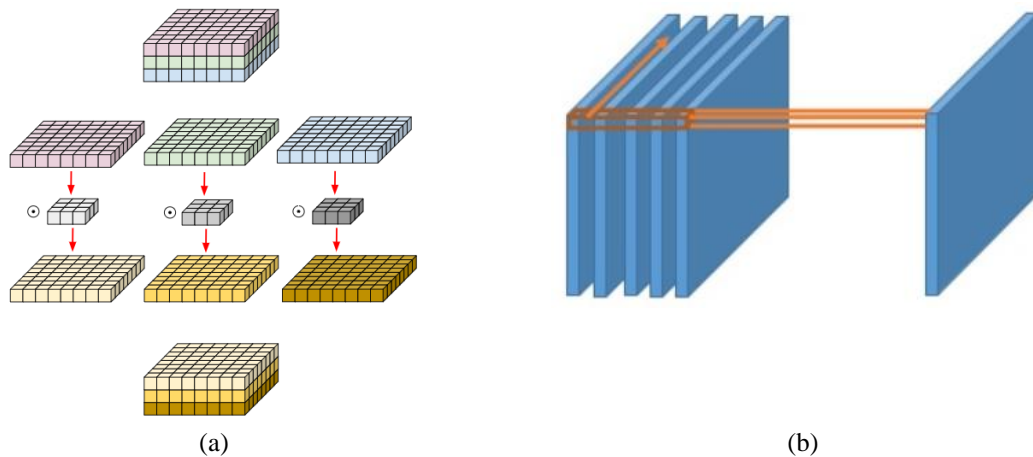


Fig. 5. Depthwise separable convolution and pointwise convolution. a) 3x3 Depthwise Convolution, b) 1x1 Pointwise Convolution.

All layers are followed by batch normalization and activation function. Batch normalization can reduce the gradient dependence on parameter scales. Batch normalization is a normalization process by reducing the average value and dividing it by the standard deviation [26]. In the activation layer, ReLU is default activation on MobileNet v2. ReLU is an activation function that was first introduced by H Sebastian Seung in 2000. The activation function serves to activate and deactivate neurons [27]. Specifically, ReLU6 is used on every layer except in the last convolution layer. The equation for the activation function ReLU6 is shown in (1).

$$f(x) = \min(\max(0, x), 6) \quad (1)$$

where $f(x)$ is a ReLU6 activation result, and x is the value applied to be changed in the range of (0, 6).

ReLU6 has a range between 0 to 6. ReLU6 is used in the MobileNet v2 model because it is stronger than the ReLU activation function [28]. Moreover, ReLU6 has an advantage that is able to retain information from images in low-precision computation [21]. Activation layer unused on the last convolution layer to avoid the elimination of important features. Then the last feature extraction on MobileNet v2 is conv2D 1x1.

The difference between the MobileNet v2 model for detection and classification is in the last layer. The last layer MobileNet v2 for detection contains SSDLite. SSDLite is a modification from regular SSD, which replaced all regular convolution with depthwise separable convolution. The purpose of SSDLite is to make MobileNet v2 more efficient. SSDLite can reduce parameter and cost [21]. The last layer MobileNet v2 for classification contains avgpool, conv2d 1x1, and softmax for image classifier. We used a 300x300x3 image input size, or we used the RGB image for detection and 224x224x3 image input size for classification.

3. Results and Discussion

3.1. The experiment results

The experiments were conducted out using the TensorFlow library and MobileNet v2. There were five scenarios in this study. The first scenario aimed to achieve the best learning rate value using 30,000 epochs; the second scenario aimed to obtain the best learning rate value using 15,000 epochs. The results of these two scenarios were then compared to get the best learning rate value and the most optimum number of the epoch. The third scenario was comparison MobileNet v2 with different activation functions and batch normalization. The fourth scenario was produced to validate the MobileNet v2 model in detecting and classifying skin cancer objects located in the printed real size of skin cancer. In the last scenario, MobileNet v2 was tested with static image input. Images of skin cancer are loaded from the camera gallery into the app.

3.1.1. Device settings in testing scenario

Testing is done by using 40 test data, which are divided into 20 images for Melanoma and 20 images for Actinic Keratosis. Testing was performed on an Android smartphone (Samsung J530G Pro) and Jupyter notebook application (running on NVIDIA GTX 1070Ti) for each model. The distance between the smartphone camera and image to be captured is 10 cm. In Scenario 1 and scenario 2, Android smartphone captures live photos on the screen or the LCD monitor. This scenario is designed to simulate the condition of analyzing skin cancer with a digital dermatoscope, where the dermoscopic image of the lesion is displayed on the screen or the LCD monitor. Scenario 3 was aimed to have performance comparison MobileNet v2 with different activation functions and batch normalization. Jupyter notebook application is used for the testing. In scenario 4, Android smartphone captures live photos on printed images of actual size skin cancer. In scenario 5, Images of skin cancer are loaded from the camera gallery into the Android app. Sketch of parameter testing using the Android smartphone is shown in Fig. 6.



Fig. 6. (a) Sketch of parameter testing using Android smartphone and laptop and (b) sketch of parameter testing using Android smartphone on the printed images of actual size skin cancer.

3.1.2. Scenario 1

In this experiment, the MobileNet v2 model was used to detect and classify between Melanoma and Actinic keratosis skin cancer objects by using 30,000 epochs and four different learning rate values. Each experiment was done in an Android smartphone and a computer using Jupyter Notebook. The results of this scenario are presented in Table 1 and Table 2, where J means experiments in the Jupyter Notebook and S for experiments in the smartphone. Table 1 shows the results of Scenario 1 in skin cancer detection, and Table 2 shows the result of Scenario 1 in skin cancer classification. Fig. 7(a) and (b) are shown the display of the Jupyter Notebook and Android app for skin cancer detection. Fig. 7(c) and (d) are shown the display of the Jupyter Notebook and Android app for skin cancer classification.

Table 1. Skin cancer detection results of the Scenario 1 on Jupyter Notebook (J) and Smartphone (S).

Learning Rate	Accuracy (%)		Sensitivity (%)		Specificity (%)	
	J	S	J	S	J	S
0.005	75	75	50	90	100	60
0.001	92.5	85	90	100	95	70
0.0005	95	87.5	90	100	100	75
0.0001	97.5	90	95	100	100	80

Based on Table 1, the best performance for skin cancer detection for both Jupyter Notebook and the Android app, were produced when using a 0.0001 learning rate. The best accuracy was 97.5% in Jupyter Notebook and 90% in the Android app. The best performance for skin cancer classification by using Jupyter Notebook and the Android app based MobileNet v2 was achieved using a learning rate of 0.005 (Table 2). The average of computing times was less than 2 seconds for object detection and less than 1 second for the classification.

Table 2. Skin cancer classification results of the Scenario 1 on Jupyter notebook (J) and smartphone (S)

Learning Rate	Accuracy (%)		Sensitivity (%)		Specificity (%)	
	J	S	J	S	J	S
0.005	90	67.5	90	95	90	40
0.001	87.5	50	90	100	85	0
0.0005	87.5	55	90	100	85	5
0.0001	87.5	57.5	95	100	80	15

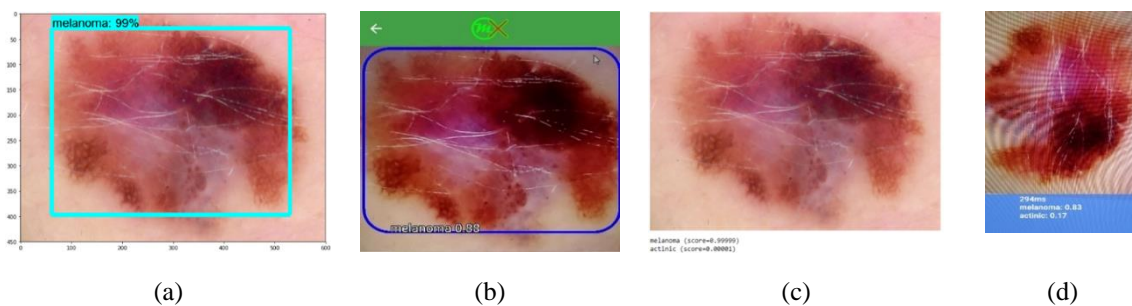


Fig. 7. The display of the Jupyter Notebook and Android app for skin cancer images analysis. Skin cancer detection: a) Jupyter Notebook, b) Android app; classification: c) Jupyter Notebook, d) Android app.

3.1.3. Scenario 2

In this experiment, the MobileNet v2 model was used to detect and classify Melanoma and Actinic keratosis skin cancer objects by using 15,000 epochs and four different learning values. Table 3 shows the results of Scenario 2 in skin cancer detection, and Table 4 shows the result of Scenario 2 in skin cancer classification.

Table 3. Skin cancer detection results of the Scenario 2 on Jupyter Notebook (J) and Smartphone (S).

Learning Rate	Accuracy (%)		Sensitivity (%)		Specificity (%)	
	J	S	J	S	J	S
0.005	85	77.5	90	100	80	55
0.001	100	87.5	100	100	100	75
0.0005	100	92.5	100	100	100	85
0.0001	97.5	95	95	100	100	90

Table 4. Skin cancer classification results of the Scenario 2 on Jupyter notebook (J) and smartphone (S).

Learning Rate	Accuracy (%)		Sensitivity (%)		Specificity (%)	
	J	S	J	S	J	S
0.005	87.5	57.5	90	95	85	20
0.001	87.5	70	90	95	85	45
0.0005	90	67.5	95	100	85	35
0.0001	87.5	65	95	95	80	35

Based on Table 3, the best performance of skin cancer detection in Jupyter Notebook was 100%, and the Android app was 95%. As we mainly focus on the Android app, the best learning rate for the Android app is 0.0001. Table 4 shows the best performance of skin cancer classification using Jupyter Notebook was 90%, and using the Android app was 70%. In this case, we chose a 0.0005 learning rate due to high

sensitivity. It could be seen that, in this scenario, the running time was the same as the previous scenario, less than 2 seconds for object detection, and less than 1 second for classification.

3.1.4. Scenario 3

Scenario 3 aimed to have performance comparison MobileNet v2 with different activation functions and batch normalization. The parameters used were based on the results of the previous scenarios, 0.0001 learning rate, and 15,000 epochs for skin cancer detection and 0.0005 learning rate with 15,000 epochs for skin cancer classification. Scenario 3 used a Jupyter notebook only for the testing. The results of scenario 3 can be seen in Table 5 and Table 6. Based on the result shows that ReLU6 with batch normalization outperformed other setting experiments.

Table 5. Skin cancer detection (D) and classification (C) results of the Scenario 3 based on the activation functions

Activation Function	Accuracy (%)		Sensitivity (%)		Specificity (%)	
	D	C	D	C	D	C
ReLU6	97.5	90	95	95	100	85
ReLU	82.5	87.5	95	95	70	80
ELU	90	82.5	100	90	80	75
Sigmoid	62.5	80	95	90	30	70
Tanh	87.5	82.5	95	90	80	75

Table 6. Skin cancer detection (D) and classification (C) Results of the Scenario 3 based on batch normalization

Batch Normalization	Accuracy (%)		Sensitivity (%)		Specificity (%)	
	D	C	D	C	D	C
Yes	97.5	90	95	95	100	85
No	87.5	87.5	95	95	80	80

3.1.5. Scenario 4

Scenario 4 is aimed to test the MobileNet v2 model in detecting and classifying actual skin cancer images. The parameters used were based on the results of the previous scenarios, 0.0001 learning rate, and 15,000 epochs for skin cancer detection and 0.0005 learning rate with 15,000 epochs for skin cancer classification. Scenario 4 used a smartphone camera that had a zoom feature. The images of skin cancer were printed according to cancer size. Several zoom settings used in this scenario; 1x zoom, 2x zoom, 3x zoom, and 4x zoom. The results of scenario 4 can be seen in Table 7 and Table 8.

Table 7. Skin cancer detection results of the Scenario 4

Zoom	Accuracy (%)	Sensitivity (%)	Specificity (%)
1x	0	0	0
2x	0	0	0
3x	0	0	0
4x	60	80	40

Table 8. Skin cancer classification results of the Scenario 4

Zoom	Accuracy (%)	Sensitivity (%)	Specificity (%)
1x	50	100	0
2x	50	100	0
3x	50	100	0
4x	60	100	20

Ten images of skin cancer printed on actual size were used to test the MobileNet v2 model. For the object detection, the MobileNet v2 model could not detect all skin cancer objects correctly at 1x zoom, 2x zoom, and 3x zoom. The accuracy of object detection on the 4x zoomed obtained 60% with 80%

sensitivity and 4% specificity. Fig. 8 shows the Android application interface when detecting actual skin cancer images. On the other hand, for skin cancer classification, the MobileNet v2 model produces 50% accuracy at 1x zoom, 2x zoom, and 3x zoom with 100% sensitivity but with a 0% specificity value. MobileNet v2 gets the highest accuracy of 60% for direct skin classification on printed paper when zooming 4x. Fig. 9 shows the Android application interface for the classification of skin cancer printed in actual size. The running time of detection and classification was the same as the previous scenario, less than 2 seconds for object detection and less than 1 second for classification.



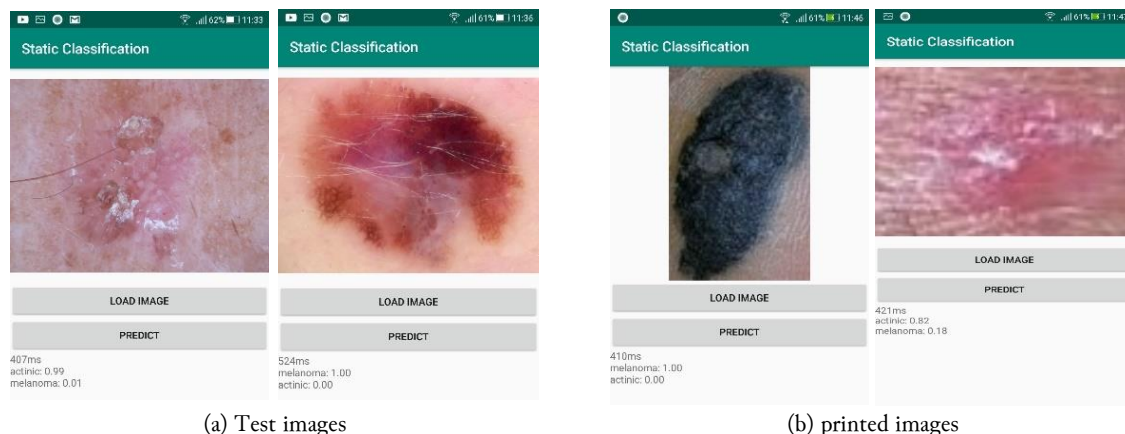
Fig. 8. The result detection using smartphone in actual skin with 4x zoom in Scenario 4.



Fig. 9. The result of classification using smartphone in actual skin with 4x zoom in Scenario 4.

3.1.6. Scenario 5

In scenario 5, the classification of skin cancer with MobileNet v2 is tested with static image input. Images of skin cancer are loaded from the camera gallery into the app. The images loaded are of two types, 40 test images, as shown in Fig. 10(a) and 10 photographs of 4x zoom actual skin cancer images printed on paper, as shown in Fig. 10(b). The skin cancer classification model was chosen in this scenario because the accuracy and sensitivity produced were higher than object detection. The best setting with the learning rate is 0.0005, and epoch value 15000 is used, which has resulted in 60% accuracy with a sensitivity of 100%.



(a) Test images

(b) printed images

Fig. 10. Classification of the Scenario 5.

Based on Table 9, the accuracy value produced by the Jupyter notebook is the same as the value of accuracy produced by the Android app for 40 test images. The running time of computation for image analysis in this scenario higher compare to the previous app. The previous scenario used a live camera to detect and classify skin cancer, but in this scenario, images are loaded to the system the same as the Jupyter notebook system. Using Samsung J530G pro, it was required average in 20 seconds to analyze the skin cancer, and when using Samsung Note 9, it was required only average in 2 seconds to analyze the skin cancer.

Table 9. Classification results of the Scenario 5.

Images types	Accuracy (%)	Sensitivity (%)	Specificity (%)
40 Test images	90	95	85
10 capture printed images	50	40	60

3.2. Discussion

The use of higher epochs generally resulted in higher accuracy, but based on scenario 1 and scenario 2, 30,000 epoch parameters produced lower accuracy values than 15,000 epochs for detection. The loss graph from the training process with 30,000 epochs with learning rate 0.0001 can be seen in Fig. 11(a), and loss graph from the training process with 15,000 epochs with learning rate 0.0001 for detection in Fig. 11(b).

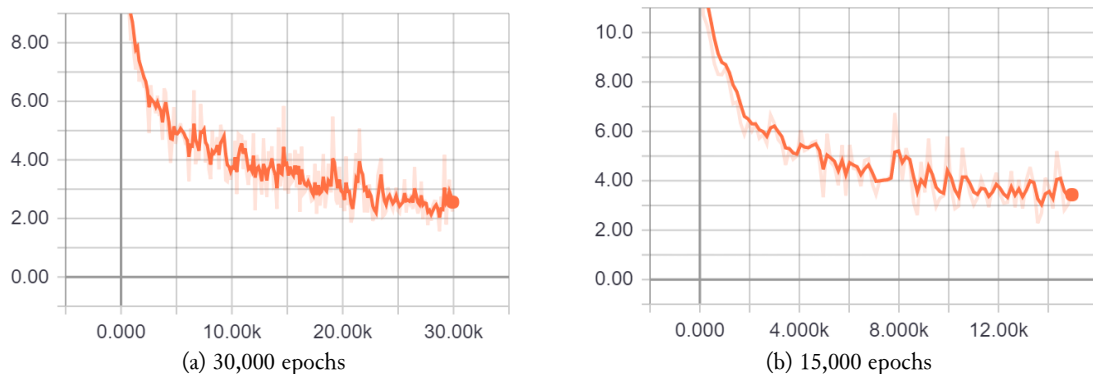


Fig. 11. Loss graphs of skin cancer detection with learning rate 0.0001

Based on Fig. 11(a), when the epoch is more than 15,000, overfitting occurred, and the loss could not decrease any less than 2%. Fig. 11(b) showed a stable loss decrease under 4% and did not indicate an overfitting problem. The accuracy value of the testing process using Jupyter Notebook outperformed the accuracy value because when using a smartphone camera, there were some influencing factors. These influencing factors included the camera's ability to capture images, the amount of light, and noise level [29]. Nevertheless, the best accuracy from scenario 1 and 2 using the Android app was 95% for object detection and 70% for classification. The results show the feasibility of supporting the analysis from the monitor on a digital dermascope.

In scenario 3, the batch normalization and ReLU6 outperformed the accuracy value of the experiment. The batch normalization increases the stability of the method by normalizing the input layer [26]. Moreover, ReLU6 requires less computation process compare to other activation then make faster training and convergence [30]. In scenario 4, the Android smartphone could not recognize skin cancer objects at zoom 0x, 2x zoom, and 3x zoom in object detection. Undetected skin cancer due to the size of the objects was inadequate for target detection. Moreover, the low camera's capability factor on Android devices makes insufficient light and adding noise into images that affected the detection process of skin cancer. Android smartphones could detect skin cancer at 4x zoom and produced 60% accuracy, 80% sensitivity, and 40% specificity for object detection and 60% accuracy, 100% sensitivity, and 20% specificity for classification.

In scenario 4, the accuracy rate produced by the Jupyter notebook (the computer) is equal with Android-based images analysis. The performance value of 40 validation images was 90% accuracy, 95% sensitivity, and 85% specificity. MobileNet v2 models continuously produced high sensitivity rates, which means it can classify Melanoma better than Actinic Keratosis. To find out how the model discriminates the two classes, we visualize the convolution results of the multiple layer process on the model using the Matplotlib library, as illustrated in Fig. 12. In Fig. 12, we display only a few images features on a particular layer, to reveal how the feature extraction process. First, after passing the convolution layer, the convolution images input will be processed in the first bottleneck layer. The first bottleneck layer results show that that the featured image of the Melanoma is more visible as compared to Actinic Keratosis. Unclear convolution images on Actinic Keratosis were happened due to its color characteristic. The Melanoma image is completely contrasting on skin color, while the Actinic color image is more predominantly red and less clear between background and cancer. In the middle bottleneck layer, the Melanoma features were also more visible as compared to the Actinic. However, in the last bottleneck layer produces significant features to contrast between the Melanoma and Actinic Keratosis images. These features are reliable to improve classification and object detection performance. The conv2D 1x1 layer becomes the last layer before entering SSDlite for object detection or Avgpool-Conv2d-Softmax for classification. One thousand one features are obtained in this layer and can be more significant if the input image is not resized 224x224 (classification). This process shows that the bottleneck layer produces a small number of features for contrasting the two-class images. Moreover, a small number of features is essential for low computing so that it can run on smartphone devices.

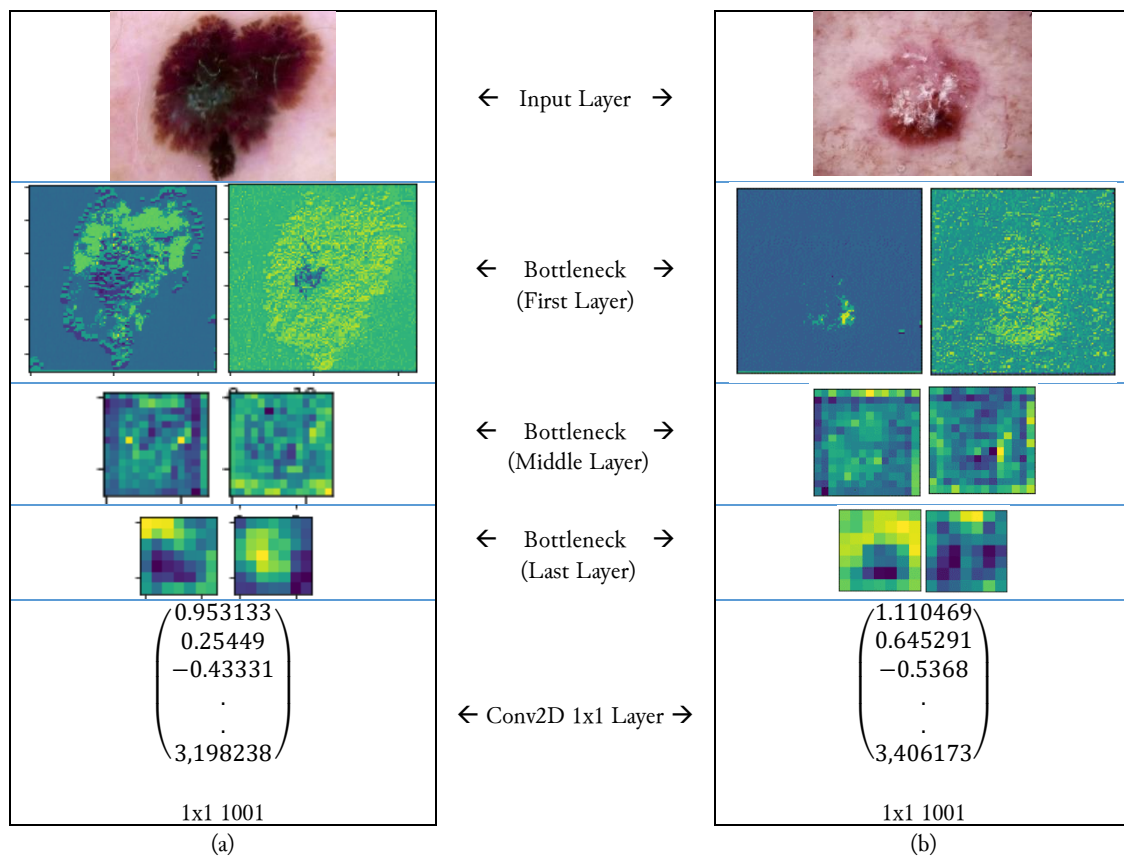


Fig. 12. Visualization of Intermediate Layer in MobileNet v2 Model (a) Melanoma images input, (b) Actinic keratosis images input.

Moderate accuracy was obtained while using photos obtained from an Android camera on printed images, as well as monitor screens. Most problems occur in Actinic images. As an illustration in Fig. 13, image (a) is an Actinic image of validation images, image (b) was obtained from an Android camera on the first shot, and image (c) was obtained from an Android camera on the second shot. The result of the classification, images (a) and (b) were classified as Actinic, while image (c) was classified as Melanoma.

Based on the results of the Conv2D 1x1, it can be observed that the features of images a, b and c still seem similar, although there are variations in some features. The miss classification can occur due to the classification layers are not yet optimal in extracting the best feature.

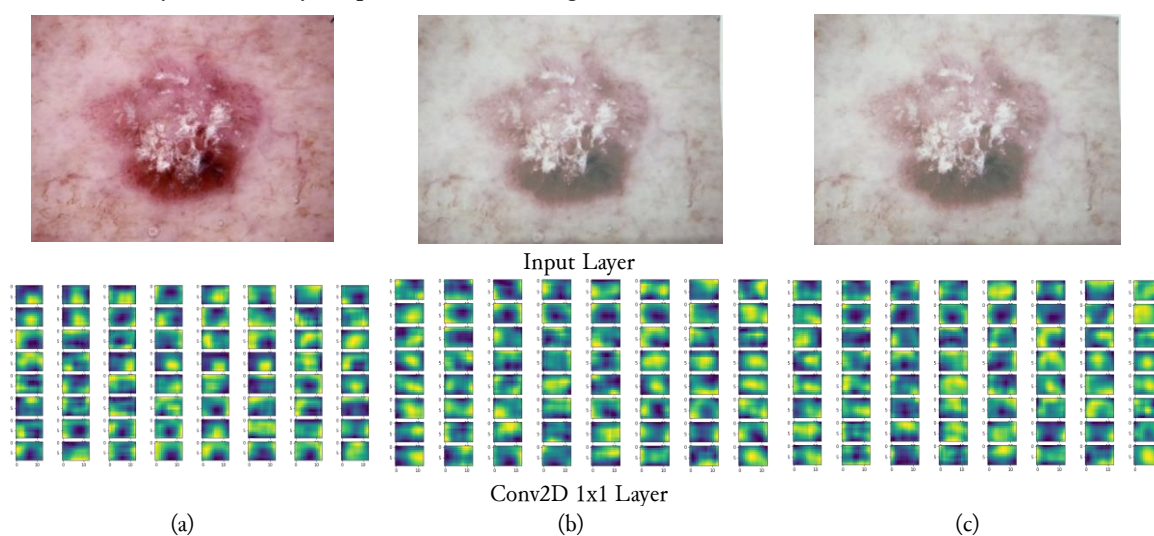


Fig. 13. Visualization of Conv2D 1x1 Last Layer in MobileNet v2 Model (a) Images from Validation, (b) Images from Android camera 1 on printed paper (c) Images from Android camera 2 on printed paper.

4. Conclusion

This study was conducted to determine the performance of skin cancer image detection and classification on Android devices using MobileNet v2 deep learning model. Based on the experiment results, the best parameter for learning rate and epoch were obtained with learning rate 0.0001 and 15,000 epochs for object detection and 0.0005 learning rate with 15,000 epochs for the classification. The Bottleneck layer was improving the feature extraction for input on object detection using SSDlite and Classification using SoftMax. The 95% accuracy for object detection and 70% for classification for smartphone camera as input show the feasibility for supporting the analysis from the monitor on a digital dermascope. The equal high accuracy between computer and Android images analysis was obtained when using a high-quality image of skin cancer that was loaded in the system with 90% accuracy, 95 sensitivity, and 85% specificity. Android-based image analysis is still within the threshold of computing time that is convenient for the user and has the same performance accuracy with the computer for the same quality images, with less than 2 seconds for live camera mode and 20 seconds for loaded image mode. Testing detection scenario and classification using a smartphone on images printed with the actual size of skin cancer resulted in the best accuracy at 4x zoom. However, image acquisition requires improvement in normalization input and getting high-quality images and visibility. Several further potential developments are possible to be performed to enhance the results. In the classification model, it is possible to add a dropout layer to reduce the event of overfitting [31]. Adam's optimization [32] is also possible to set the learning rate more adaptive so that the model can identify more high-grade features. Integrating skin lesions segmentation as the picture preprocessing step also is potentially improved the accuracy of skin cancer classification. Moreover, standardization of images acquisition can be used to capture high-quality images and excellent visibility. The use of compact microscopes for smartphones that can function as dermascope can be implemented by connecting the compact microscope with the Android camera.

Acknowledgment

The authors thank to the Ministry of Research, Technology, and Higher Education (RISTEKDIKTI) of the Republic of Indonesia for supporting this research through "Penelitian Dasar" Research Grant 2019 with grant number 257-22/UN7.P4.3/PP/2019.

Declarations

Author contribution. All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding statement. This work was fully supported by the Ministry of Research, Technology, and Higher Education (RISTEKDIKTI) of the Republic of Indonesia under “Penelitian Dasar” Research Grant 2019 (Grant No. 257-22/UN7.P4.3/PP/2019).

Conflict of interest. The authors declare no conflict of interest.

Additional information. No additional information is available for this paper.

References

- [1] M. Heron and R. N. Anderson, “Changes in the leading cause of death: Recent patterns in heart disease and cancer mortality,” *Natl. Cent. Heal. Stat. Data Br.*, no. 254, 2017, available at: [Google Scholar](#).
- [2] M. C. F. Simões, J. J. S. Sousa, and A. A. C. C. Pais, “Skin cancer and new treatment perspectives: A review,” *Cancer Lett.*, vol. 357, no. 1, pp. 8–42, 2015, doi: [10.1016/j.canlet.2014.11.001](#).
- [3] World Health Organization, “INTERSUN: The Global UV Project. A guide and compendium 2003”, available at: <https://apps.who.int/iris/bitstream/handle/10665/42814/9241591056.pdf?sequence=1>, (accessed 30 Jul. 2019).
- [4] A. Bourouis, A. Zerdazi, M. Feham, and A. Bouchachia, “M-health: skin disease analysis system using Smartphone’s camera,” *Procedia Comput. Sci.*, vol. 19, pp. 1116–1120, 2013, doi: [10.1016/j.procs.2013.06.157](#).
- [5] G. Cao, X. Xie, W. Yang, Q. Liao, G. Shi, and J. Wu, “Feature-fused SSD: Fast detection for small objects,” in *Ninth International Conference on Graphic and Image Processing (ICGIP 2017)*, 2018, vol. 10615, p. 106151E, available at: [Google Scholar](#).
- [6] Y. Yang, Z. Ye, Y. Su, Q. Zhao, X. Li, and D. Ouyang, “Deep learning for in vitro prediction of pharmaceutical formulations,” *Acta Pharm. Sin. B*, vol. 9, no. 1, pp. 177–185, 2019, doi: [10.1016/j.apsb.2018.09.010](#).
- [7] H. Chang, “Skin cancer reorganization and classification with deep neural network,” *arXiv Prepr. arXiv1703.00534*, 2017, available at: [Google Scholar](#).
- [8] H. A. Haenssle *et al.*, “Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists,” *Ann. Oncol.*, vol. 29, no. 8, pp. 1836–1842, 2018, doi: [10.1093/annonc/mdy166](#).
- [9] R. M. Gordon, “Skin cancer: more than skin deep,” *Adv. Skin Wound Care*, vol. 22, no. 12, pp. 574–580, 2009, doi: [10.1097/01.ASW.0000363470.25740.a2](#).
- [10] Y. Yuan, M. Chao, and Y.-C. Lo, “Automatic skin lesion segmentation using deep fully convolutional networks with jaccard distance,” *IEEE Trans. Med. Imaging*, vol. 36, no. 9, pp. 1876–1886, 2017, doi: [10.1109/TMI.2017.2695227](#).
- [11] A. A. Zaidan *et al.*, “A review on smartphone skin cancer diagnosis apps in evaluation and benchmarking: coherent taxonomy, open issues and recommendation pathway solution,” *Health Technol. (Berl.)*, vol. 8, no. 4, pp. 223–238, 2018, doi: [10.1007/s12553-018-0223-9](#).
- [12] SkinVision B.V., “SkinVision - Detect Skin Cancer.”, available at: [Google Play Store](#), (accessed Jul. 01, 2019).
- [13] M. Cristian, “SpotMole.” available at: [Google Play Store](#). (accessed Jul. 01, 2019).
- [14] C. P. Koirala, “Deep Learning for Melanoma.”, CJ63, available at: [Google Play Store](#). (accessed Jul. 01, 2019).
- [15] GeniaLabs, “DermIA.”, available at: [Google Play Store](#). (accessed Jul. 01, 2019).
- [16] T. Maier *et al.*, “Accuracy of a smartphone application using fractal image analysis of pigmented moles compared to clinical diagnosis and histological result,” *J. Eur. Acad. Dermatology Venereol.*, vol. 29, no. 4, pp. 663–667, 2015, doi: [10.1111/jdv.12648](#).

- [17] M. Thissen, A. Udrea, M. Hacking, T. von Braunmuehl, and T. Ruzicka, "mHealth app for risk assessment of pigmented and nonpigmented skin lesions—a study on sensitivity and specificity in detecting malignancy," *Telemed. e-Health*, vol. 23, no. 12, pp. 948–954, 2017, doi: [10.1089/tmj.2016.0259](https://doi.org/10.1089/tmj.2016.0259).
- [18] J. Jaworek-Korjakowska and P. Kleczek, "Eskin: study on the smartphone application for early detection of malignant melanoma," *Wirel. Commun. Mob. Comput.*, vol. 2018, 2018, doi: [10.1155/2018/5767360](https://doi.org/10.1155/2018/5767360).
- [19] N. Z. Tajeddin and B. M. Asl, "Melanoma recognition in dermoscopy images using lesion's peripheral region information," *Comput. Methods Programs Biomed.*, vol. 163, pp. 143–153, 2018, doi: [10.1016/j.cmpb.2018.05.005](https://doi.org/10.1016/j.cmpb.2018.05.005).
- [20] A. G. Howard *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv Prepr. arXiv1704.04861*, 2017, available at: [Google Scholar](https://scholar.google.com/citations?view_op=view_citation&hl=en&user=aghoward&citation_for_view=aghoward:arXiv:1704.04861v1).
- [21] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520, doi: [10.1109/CVPR.2018.00474](https://doi.org/10.1109/CVPR.2018.00474).
- [22] L. N. Smith, "A disciplined approach to neural network hyper-parameters: Part 1--learning rate, batch size, momentum, and weight decay," *arXiv Prepr. arXiv1803.09820*, 2018, available at: [Google Scholar](https://scholar.google.com/citations?view_op=view_citation&hl=en&user=lnsmith&citation_for_view=lnsmith:arXiv:1803.09820v1).
- [23] W. Jerjes, Z. Hamdoon, A. A. Abdulkareem, and C. Hopper, "Photodynamic therapy in the management of actinic keratosis: Retrospective evaluation of outcome," *Photodiagnosis Photodyn. Ther.*, vol. 17, pp. 200–204, 2017, doi: [10.1016/j.pdpdt.2016.04.017](https://doi.org/10.1016/j.pdpdt.2016.04.017).
- [24] M. Rastrelli, S. Tropea, C. R. Rossi, and M. Alaibac, "Melanoma: epidemiology, risk factors, pathogenesis, diagnosis and classification," *In Vivo (Brooklyn)*, vol. 28, no. 6, pp. 1005–1011, 2014, available at: [Google Scholar](https://scholar.google.com/citations?view_op=view_citation&hl=en&user=mrastrelli&citation_for_view=mrastrelli:In+Vivo+(Brooklyn):+2014).
- [25] B. Singh, D. Toshniwal, and S. K. Allur, "Shunt connection: An intelligent skipping of contiguous blocks for optimizing MobileNet-V2," *Neural Networks*, vol. 118, pp. 192–203, 2019, doi: [10.1016/j.neunet.2019.06.006](https://doi.org/10.1016/j.neunet.2019.06.006).
- [26] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv Prepr. arXiv1502.03167*, 2015, available at: [Google Scholar](https://scholar.google.com/citations?view_op=view_citation&hl=en&user=sioffe&citation_for_view=sioffe:arXiv:1502.03167v1).
- [27] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv Prepr. arXiv1803.08375*, 2018, available at: [Google Scholar](https://scholar.google.com/citations?view_op=view_citation&hl=en&user=afagarap&citation_for_view=afagarap:arXiv:1803.08375v1).
- [28] B. Szal, "On the uniform convergence of weighted trigonometric series," *Banach Cent. Publ.*, vol. 1, no. 92, pp. 339–350, 2011, doi: [10.4064/bc92-0-23](https://doi.org/10.4064/bc92-0-23).
- [29] Y. Makihara, M. Takizawa, Y. Shirai, and N. Shimada, "Object recognition under various lighting conditions," in *Scandinavian Conference on Image Analysis*, 2003, pp. 899–906, doi: [10.1007/3-540-45103-X_119](https://doi.org/10.1007/3-540-45103-X_119).
- [30] M. Liu, Z. Zhou, P. Shang, and D. Xu, "Fuzzified image enhancement for deep learning in iris recognition," *IEEE Trans. Fuzzy Syst.*, 2019, doi: [10.1109/TFUZZ.2019.2912576](https://doi.org/10.1109/TFUZZ.2019.2912576).
- [31] 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. 1, pp. 1929–1958, 2014, available at: <https://dl.acm.org/doi/abs/10.5555/2627435.2670313>.
- [32] S. De, A. Mukherjee, and E. Ullah, "Convergence guarantees for RMSProp and ADAM in non-convex optimization and an empirical comparison to Nesterov acceleration," *arXiv Prepr. arXiv1807.06766*, 2018., available at: [Google Scholar](https://scholar.google.com/citations?view_op=view_citation&hl=en&user=sde&citation_for_view=sde:arXiv:1807.06766v1)