

# Development of Android Application for Banana Leaf Disease Detection

Chong Mun Zheng<sup>1</sup>, Nik Shahidah Afifi Md Taujuddin<sup>1\*</sup>, Suhaila Sari<sup>1</sup>

<sup>1</sup> Department of Electrical Engineering, Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Johor, MALAYSIA

\*Corresponding Author: shahidah@uthm.edu.my

DOI: <https://doi.org/10.30880/eeee.2024.05.02.010>

## Article Info

Received: 1 July 2024

Accepted: 29 September 2024

Available online: 30 October 2024

## Keywords

Banana, Banana Leaf Disease, Android Application, Image Processing, Convolutional Neural Network, MobileNet

## Abstract

Banana is one of the important commercial fruits and one of the favourite fruits in the world because of its high nutrition content and low cost. However, banana production has declined sharply in the last few years, with about 50% yield loss due to late disease detection. In order to help the farmers detect diseases, an android application for banana leaf disease detection is proposed. This application is not intended to replace the farmer but to assist the farmer in detecting the banana leaf disease efficiently. The method used in this project is the CNN MobileNet algorithm as the core to detect the disease and Android Studio as a platform to develop an application based on the algorithm. In image processing, there are a few steps: image pre-processing, feature extraction, and classification. This proposed system is believed to classify the banana leaf disease efficiently, thus helping the farmers in detecting banana leaf disease once completed.

## 1. Introduction

Bananas are one of the most important fruits and one of the most popular fruits in the world due to their excellent nutritional value and low cost [1], [2], [3]. However, banana production has declined sharply in the last few years, with about 50% yield loss due to diseases [1], [2], [3], [4], [5], [6], [7]. In addition, the diseases contracted during the production process have a major negative impact on the quality of bananas produced [1], [2], [3], [4], [6], [8], [9], [10]. In the end, it can cause significant economic losses to the farmers and impede the growth of the banana industry, so it is essential to detect the diseases by detecting them to prevent the loss [1], [2], [3], [4], [10]. In addition to being costly, time-consuming and labour-intensive, the traditional method of detecting banana disease detection may produce inconsistent results due to subjective variables [1], [2], [3], [4], [6], [7]. Crop disease detection based on image processing using computer technology can be used to calculate and process the information of the plant disease images to achieve the purpose of intelligent detection, thereby increasing the efficiency and accuracy of disease detection [4], [8].

In order to fulfil the need for sustainable agriculture, artificial intelligence technologies such as machine learning and deep learning are becoming common in the agriculture industry [1], [7], [10]. Image segmentation is a crucial stage in crop disease detection that segments the original image into target and background sections, and the performance of this process has a significant impact on image feature extraction and target recognition [1], [3], [4], [7], [10].

### 1.1 Problem Statement

This is an open access article under the CC BY-NC-SA 4.0 license.



Nowadays, the banana plant disease has become a major concern for the farmers, who need to sustain the quality of the bananas. If the disease isn't found in the early stage of the plantation, it may lead to a banana epidemic on the farm, and it can reduce the production and quality of bananas. The current method to detect banana disease is through the farmer's experiences by checking the banana leaf with their eyes, as the banana leaf disease detection system isn't available in the market yet.

## 1.2 Objective

The objective of this project is to develop an Android application to help farmers detect banana leaf disease by implementing CNN MobileNet.

## 2. Literature Review

This section will discuss the previous and recent research based on the topic. It consists of the fundamental steps in image processing and the algorithm used in this project, MobileNet.

### 2.1 Fundamental Steps in Image Processing

#### 2.1.1 Image Acquisition

In image processing, the first step is image acquisition. This step aims to convert an optical image into an array of numerical data to be later processed on a computer. This implies it captures an image using a camera and converts it into a computer-manageable entity. The process uses the sensors in the cameras to convert light into electrical charges by measuring the reflected light energy on the scene being imaged. The value of each cell in the image will be turned into a digital value in binary form using an Analog-to-Digital converter (ADC). Finally, this digital image can be used for the following image processing steps [11].

#### 2.1.2 Image Enhancement

One of the most critical and complex steps in image processing is image enhancement, which is used to enhance the visual appearance of an image and provide a better representation of the image for image processing. High-visibility input images are necessary for all image processing algorithms in order to produce accurate outputs. However, the results might turn out to be inaccurate if the images are typically shot in some particular conditions, such as dim light. Hence, the images should be enhanced before further processing [12]. The most common image enhancement technique is enhancing the contrast [12], [13].

#### 2.1.3 Image Restoration

The process of image restoration in image preprocessing is to recover an unknown true image from its degraded measurement. In other words, it means that this technique restores the image back to its optimal form from degradations such as additive noise, space invariant or variant blur, aliasing, and compression artifact. These degradations mostly come from the process of image formation, transmission, and storage [14].

#### 2.1.4 Colour image processing

Besides image enhancement, colour image processing is another crucial technique in the image processing. This is due to the fact that the images dealt with everyday life consisting of intensity of brightness and chromaticity. The components that makeup chromaticity are hue and saturation. However, the image enhancement technique can only be applied to enhance the intensity but can't be used to enhance the chromaticity. Hence, this technique is crucial for the accuracy of the model [15].

#### 2.1.5 Image Compression

Image compression in image processing is a process to encode the image and reduce the size of the image for storage and transmission. This technique is mainly used in the medical field, such as X-ray films, MRI films and so on, without losing the details of the images. This is because, in order to perform compression on medical images, the effectiveness of resolution and perceptual quality must be restored [16].

#### 2.1.6 Wavelets and multi-resolution processing

In image processing, wavelet is a mathematical function that is used in the compression of images. It serves as a foundation to represent images at various resolutions. Different degrees of detail can be made for multiple objects using multi-resolution analysis, and the rendering time can be reduced by ignoring fine details. It is an efficient tool that can assist in data compression [17].

### 2.1.7 Morphological processing

Morphological processing is a process that analyses the images from the perspective of set theory and geometry. The fundamental operations of morphological processing are dilation, erosion and algorithms. Hence, morphological processing can be used to detect edges of objects and mark and count objects based on the image [18].

### 2.1.8 Image Segmentation

One of the most important processes in image processing is image segmentation. This process divides or partitions the images based on similar characteristics such as colour, intensity, and texture. The aim of this process is to simplify the representation of an image into a more meaningful and easily analysable one [19].

### 2.1.9 Representation and description

After the image segmentation process, representation and description is a process that describes and represents the segmented region in a way that is acceptable for further computer processing. While description works by extracting fundamental attributes to distinguish one class of objects from another, representation determines whether the segmented region should be depicted as a boundary or as a complete region [20].

### 2.1.10 Object recognition

In image processing, object recognition is one of the crucial processes. Object recognition is a process of determining the identity of an object being observed in the segmentation, representation, and description process. In other words, it labels the segmented region in the image segmentation process in order to allow the user to know what has been detected. However, object recognition will become less effective due to the positioning, scaling, alignment, and occlusion of objects that are not fully visible in an image [21].

## 2.2 MobileNet

Among the Convolutional Neural Network (CNN), MobileNet is also one of the CNN image processing algorithms that is an efficient and lightweight model that consists of a smaller number of parameters. In the first few layers of the MobileNet algorithm, it uses depthwise separable convolutions to decrease the computational task which results in lighter weight [22].

In the structure of the MobileNet, It consists of two different kinds of core layers, which are depthwise convolutions and pointwise convolutions. The depthwise convolutions are used to 16 filter the input without creating new features, whereas the pointwise convolutions are the function to create new features. Hence, the depthwise separable convolutions are the combination of both of the functions. Each depthwise separable convolution layer started with the depthwise convolution layer and was followed by the batch normalisation layer and ReLU activation layer. After ReLU activation, pointwise convolution layer is used to create the new features and followed by normalisation layer and ReLU activation [22].

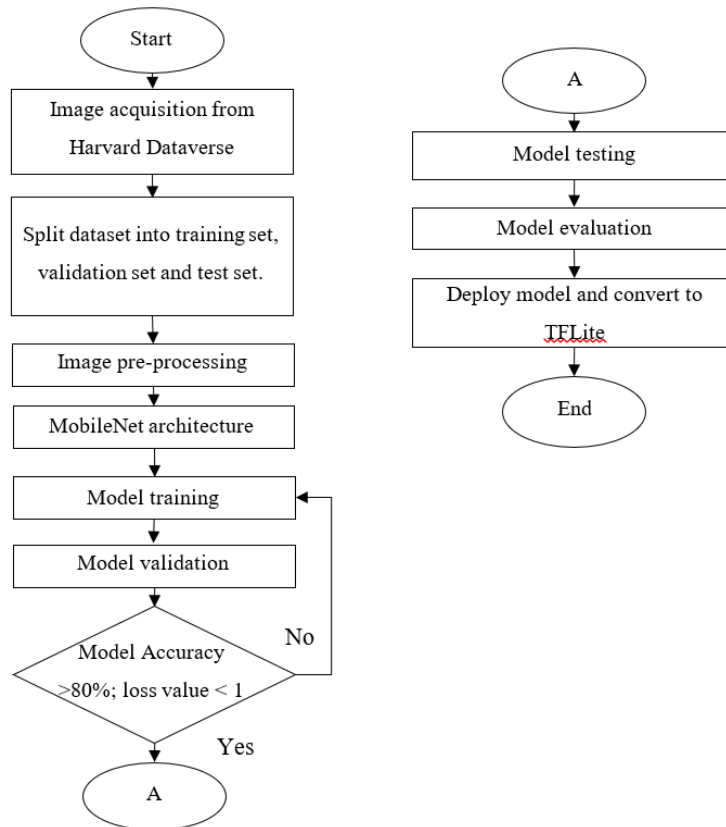
After the depthwise separable convolutional layer, the global average pooling layer will sum up the identical information in the local region generated by the convolutional layers and produce an output of a single value within that region. Then, the reshape layer and dropout are used to prevent overfitting. After that, a convolutional layer is used followed by a softmax activation layer and reshape layer. Finally, the structure of AlexNet is ended with an output layer. This model contains about four million parameters, which is much smaller than other models [22].

## 3. Methodology

Fig. 1 shows a flowchart of the project on developing a banana leaf disease detection system in Jupyter Notebook and Android Studio. The process is started with acquiring image datasets that contained images of healthy banana leaf and banana leaf with Black Sigatoka and Fusarium Wilt from the Harvard Dataverse website. The dataset is split into 3 sets, which are the training set, validation set and testing set with 3000 samples each class.

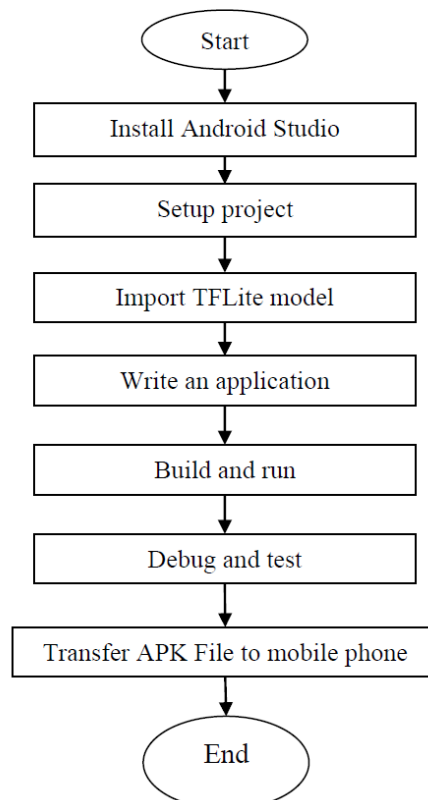
After splitting the datasets, the training set and validation set are pre-processed and are used to train MobileNet for 50 epochs. For this phase, the aim is to achieve an accuracy of 90% and a loss value of less than 1. If the trained model does not achieve the following conditions, the mode will repeatedly train with a bigger epoch until the conditions were fulfilled.

After training, the model is tested and evaluated based on its performance metrics. Then, the trained MobileNet model is deployed and converted to TFLite format so that it can be imported into Android studio.



**Fig. 1** Process flow of MobileNet model development

Fig. 2 shows the process flow for developing an Android application using Android Studio. The process begins with installing Android Studio from their official website and setting up a project environment for developing an application. Following that, the trained MobileNet model in TFLite format is imported into the project.



**Fig. 2** Process flow of android application development

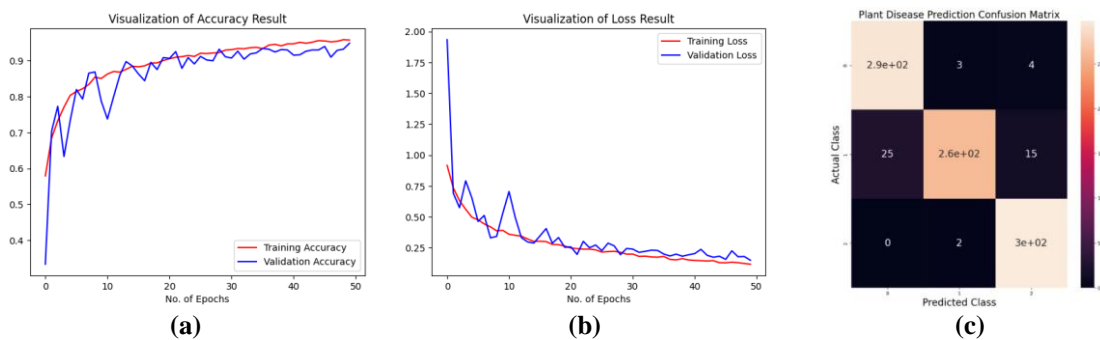
After importing the model, the next step is writing code in the “MainActivity.java” file using the Java programming language to implement the model into the application. Then, the graphical user interface (GUI) is designed and implemented for the application in the “layout.xml” file. The GUI is used as the interaction platform for the user to provide input to the application and provide output to the user. The GUI is designed to allow the user to take a new banana leaf picture for detection or select a picture in the gallery for detection, followed by the result shown in the GUI.

To ensure stability, reliability and accuracy, the application is tested and debugged so that any issues and improvements can be resolved before proceeding to the next step. After that, the application is built and transferred to the mobile phone in Android package kit (.APK) file format.

## 4. Results and discussion

### 4.1 Results

Fig. 3(a) depicts the graphs showcasing the accuracy of the MobileNet model. The graph shows that the training accuracy of the models ended with training accuracy of 0.9578 and validation accuracy of 0.9489. Fig. 3(b) depicts the graphs showcasing the loss of the MobileNet model. The graph shows that the training loss of the models ended with training loss of 0.1167 and validation loss of 0.1484. Fig. 3(c) shows the confusion matrix of the MobileNet model. It shows how accurate the MobileNet model classify the classes.

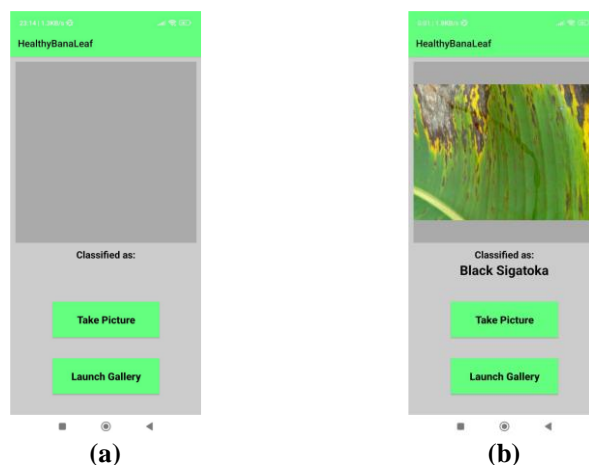


**Fig. 3** Performance of MobileNet model (a) Accuracy; (b) Loss; (c) Confusion Matrix

In the design of the application, the user interface is crucial for the users as it serves as a platform for the user to interact with the application. Fig. 4(a) shows the home page of the application that the user encountered after launching the app. The Android application can be downloaded from this link: [https://drive.google.com/file/d/1Odttml8aWjdvL\\_S5SDPO-gvbD05LBer/](https://drive.google.com/file/d/1Odttml8aWjdvL_S5SDPO-gvbD05LBer/).

On the home page, there are two selections for the user to choose from, which are “Take Picture” and “Launch Gallery”. These selections allow the user to take a new picture of the banana leaf using the camera or select a saved picture of a banana leaf from the phone.








After the user successfully takes a picture of the banana leaf using the camera or selects a picture of the banana leaf from the gallery in the phone, the application will detect and classify the condition of the banana leaf and show the result of the detection to the user. Fig. 4(b) shows the results of the banana leaf disease after detection. In this figure, it shows that the banana leaf is classified as a banana leaf with Black Sigatoka disease.





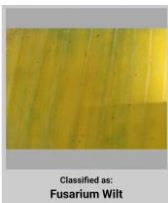


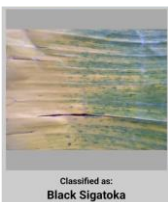


**Fig. 4** HealthyBanaLeaf application (a) Home page; (b) Result after detection

To test the performance of the Android application, the application is tested with 20 samples of banana leaf images with Black Sigatoka, Fusarium Wilt and healthy banana leaf images. Table 1 shows the results obtained from the application after detecting the selected samples of banana leaf images.


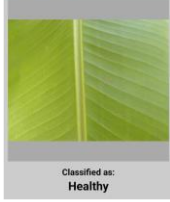



**Table 1** Application test results

Sample	Results	Disease	Detection	Accuracy
1	 Classified as: Black Sigatoka	Black Sigatoka	Black Sigatoka	Accurate
2	 Classified as: Black Sigatoka	Black Sigatoka	Black Sigatoka	Accurate
3	 Classified as: Healthy	Black Sigatoka	Healthy	Inaccurate
4	 Classified as: Black Sigatoka	Black Sigatoka	Black Sigatoka	Accurate
5	 Classified as: Black Sigatoka	Black Sigatoka	Black Sigatoka	Accurate
6	 Classified as: Black Sigatoka	Black Sigatoka	Black Sigatoka	Accurate
7	 Classified as: Fusarium Wilt	Fusarium Wilt	Fusarium Wilt	Accurate

**Table 1** Application test results (continued)

Sample	Results	Disease	Detection	Accuracy
8		Fusarium Wilt	Fusarium Wilt	Accurate
9		Fusarium Wilt	Fusarium Wilt	Accurate
10		Fusarium Wilt	Fusarium Wilt	Accurate
11		Fusarium Wilt	Fusarium Wilt	Accurate
12		Fusarium Wilt	Fusarium Wilt	Accurate
13		Fusarium Wilt	Black Sigatoka	Inaccurate
14		Healthy	Healthy	Accurate
15		Healthy	Healthy	Accurate

**Table 1** Application test results (continued)

Sample	Results	Disease	Detection	Accuracy
16		Healthy	Healthy	Accurate
17		Healthy	Healthy	Accurate
18		Healthy	Healthy	Accurate
19		Healthy	Healthy	Accurate
20		Healthy	Healthy	Accurate

## 4.2 Discussion

In Fig. 2(a), It shows that the MobileNet model has a high validation accuracy of 94.89% and high training accuracy of 95.78%, which are more than 90%. Whereas in Fig. 2(b), It shows that the MobileNet model has a low validation loss of 0.1484 and low training loss of 0.1167, which are less than 0.15. In Fig. 2(c), it shows that the MobileNet model only had trouble in classifying class 1 (Fusarium Wilt), where almost half of class 1 (Fusarium Wilt) was classified as class 0 (Black Sigatoka) and class 2 (Healthy).

Table 1 shows that the application has an overall 90% accuracy in detecting banana leaf diseases. Based on the results, the application has no problem in detecting healthy banana leaves, but it misclassifies Black Sigatoka and Fusarium Wilt into the other, with an accuracy of 83.33% in detecting Black Sigatoka and 85.71% in detecting Fusarium Wilt. Due to the sample size of this test is small, the performance of the application shows that it has similar accuracy with the classification result of the model.

## 5. Conclusion

In conclusion, the CNN MobileNet implemented Android Application, HealthyBanaLeaf has been proven through analysis to be able to detect banana leaf disease with an outstanding performance of 90% accuracy by using a smartphone. Based on the findings and outcomes of this research, there is a recommendation for further enhancing the mobile application by providing banana leaf disease treatment suggestions after detection. This extends the functionality of the mobile application by just providing the farmers with disease treatment recommendations to treat the banana leaf disease.

## Acknowledgement

The authors would like to thank the Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia for its support to make this research possible. Communication of this research is made possible through monetary assistance by Universiti Tun Hussein Onn Malaysia and the UTHM Publisher's Office via Publication Fund E15216.

## Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

## Author Contribution

The authors confirm contribution to the paper as follows: **study conception and design:** Chong Mun Zheng, Nik Shahidah Afifi Md Taujuddin, Suhaila Sari; **data collection:** Chong Mun Zheng, Nik Shahidah Afifi Md Taujuddin; **analysis and interpretation of results:** Chong Mun Zheng, Nik Shahidah Afifi Md Taujuddin; **draft manuscript preparation:** Chong Mun Zheng, Nik Shahidah Afifi Md Taujuddin. All authors reviewed the results and approved the final version of the manuscript.

## References

- [1] Kakati, J. B., and Das, T. K., "Classification of healthy and unhealthy banana leaves using deep learning approach: A comparative assessment," 2023. doi: <https://doi.org/10.1109/I3CS58314.2023.10127309>
- [2] Jianqing, H., Qi, Y., Debing, L., and Jiarong, Z., "Research on banana leaf disease detection based on the image processing technology," 2022. doi: <https://doi.org/10.1109/CCET55412.2022.9906377>
- [3] Jianqing, H., Qi, Y., Debing, L., and Hailing, F., "An image segmentation method for banana leaf disease image with complex background," 2022. doi: <https://doi.org/10.1109/DSIT55514.2022.9943850>
- [4] Tuazon, G. L. H., Duran, H. M., and Villaverde, J. F., "Portable Sigatoka spot disease identifier on banana leaves using support vector machine," 2021. doi: <https://doi.org/10.1109/HNICEM54116.2021.9731808>
- [5] Mathew, D., Sathish Kumar, C., and Anita Cherian, K., "Classification of leaf spot diseases in banana using pre-trained convolutional neural networks," 2023. doi: <https://doi.org/10.1109/ICCC57789.2023.10165629>
- [6] Saranya, N., Pavithra, L., Kanthimathi, N., Ragavi, B., and Sandhiyadevi, P., "Detection of banana leaf and fruit diseases using neural networks," 2020. doi: <https://doi.org/10.1109/ICIRCA48905.2020.9183006>
- [7] Vandana, C., and Manoj, P., "Banana leaf disease detection using K-means clustering and feature extraction techniques," 2020. doi: <https://doi.org/10.1109/ICACCM50413.2020.9212816>
- [8] Karthik, G., and Praburam, N., "Detection and prevention of banana leaf diseases from banana plant using embedded Linux board," 2016. doi: <https://doi.org/10.1109/GET.2016.7916817>
- [9] Antonio, C., Miguel, M., Eduardo, S., and Gustavo, V., "Design and evaluation of a convolutional neural network for banana leaf diseases classification," 2020. doi: <https://doi.org/10.1109/EIRCON51178.2020.9254072>
- [10] Banerjee, D., Kukreja, V., Hariharan, S., and Sharma, V., "Precision agriculture: Classifying banana leaf diseases with hybrid deep learning models," 2023. doi: <https://doi.org/10.1109/I2CT57861.2023.10126431>
- [11] Mishra, V. K., Kumar, S., and Shukla, N., "Image acquisition and techniques to perform image acquisition," 2017. doi: <https://doi.org/10.18090/samriddhi.v9i01.8333>
- [12] Ackar, H., Almisreb, A. A., and Saleh, M. A., "A review on image enhancement techniques," 2019. doi: <https://doi.org/10.21533/scjournal.v8i1.175>
- [13] Shashi B. Rana, S., "A review of medical image enhancement techniques for image processing," 2011. doi: <https://doi.org/10.14741/ijcet/22774106/5.2.2015.121>
- [14] Gunturk, B. K., and Li, X., "Image restoration: Fundamentals and advances." 2013. doi: <https://doi.org/10.1201/b12693>
- [15] Wong, S.-L., Aarii, S., Paramesran, R., and Taguchi, A., "Color image processing system based on color/brightness constancy," 2015. doi: <https://doi.org/10.1109/ISPACS.2015.7432729>
- [16] Satone, K. N., Deshmukh, A. S., and Ulhe, P. B., "A review of image compression techniques," 2017. doi: <https://doi.org/10.1109/ICECA.2017.8203651>
- [17] Valette, S., and Prost, R., "Wavelet-based multiresolution analysis of irregular surface meshes," 2004. doi: <https://doi.org/10.1109/TVCG.2004.1260763>
- [18] Sha, H., and Wah, C., "Morphological image processing and its parallel implementation," 1996. doi: <https://doi.org/10.1109/ICSIGP.1996.567321>

- [19] Sahu, S., Sarma, H., and Bora, D. J, "Image segmentation and its different techniques: An in-depth analysis," 2018. doi: <https://doi.org/10.1109/RICE.2018.8509038>
- [20] Ebrahim, Y., Ahmed, M., Chau, S. C., and Abdelsalam, W, "An efficient shape representation and description technique," 2007. doi: <https://doi.org/10.1109/ICIP.2007.4379616>
- [21] Meera, M. K., and Shajee, M. B. S, "Object recognition in images," 2016. doi: <https://doi.org/10.1109/INFOSCI.2016.7845313>
- [22] Sae-Lim, W., Wettayaprasit, W., and Aiyarak, P, "Convolutional neural networks using MobileNet for skin lesion classification," 2019. doi: <https://doi.org/10.1109/JCSSE.2019.8864155>