Research article

Monkeypox Lesion and Rash Stage Classification for Self-screening on Mobile Application Using Deep Learning Technique

Parinda Labcharoenwongs, Duangjai Noolek and Orawan Chunhapran*

Department of Computer Science, Faculty of Business Administration and Information Technology, Rajamangala University of Technology Tawan-Ok, Bangkok, Thailand

Curr. Appl. Sci. Technol. 2024, Vol. 24 (No. 6), e0257989; https://doi.org/10.55003/cast.2024.257989

Received: 20 February 2023, Revised: 5 March 2024, Accepted: 24 May 2024, Published: 2 August 2024

Abstract

Keywords

monkeypox;

rash stage;

deep learning techniques;

EfficientNet;

self-screening

The early diagnosis of pox symptoms is an important part of preventing a global pandemic. In addition, the estimation of the disease stage and time period of pox rash is of interest. Computer Aided Diagnosis systems have been developed for identification of suspected cases based on machine learning techniques. In recent years, many deep learning approaches have been developed to classify monkeypox disease. In this study, a monkeypox lesion and rash stage classification for self-screening on mobile applications using deep learning techniques was introduced. The datasets consisted of skin lesion and pox rash images. Data augmentation methods were used to increase the sample size and split data into training and testing sets in the experiment setup. When comparing overall accuracy, EfficientNet achieved an accuracy of 0.95 for pox and 0.97 for the pox rash stage. EfficientNet was selected for conversion and implementation on the mobile application. The study determined that the use of deep learning techniques for monkeypox lesions and rash stage classification on a mobile application enabled early identification of patients and effective control of community spread.

*Corresponding author: Tel.: (+66) 922703824
E-mail: orawan_ch@rmutto.ac.th

1. Introduction

The monkeypox virus became a global public health concern due to its rapid spread to more than 40 countries outside Africa. Monkeypox is a contagious disease caused by a virus called Zoonotic Orthopoxvirus, which is closely related to cowpox and smallpox, and belongs to the Poxviridae family [1]. A person with monkeypox can spread the virus from the time symptoms first appear until the rash has fully healed. Monkeypox rash can be mistaken for chickenpox, shingles, or herpes [2-11]. Diagnosis of monkeypox can be done by observing the unusual characteristics of skin lesions, as well as by taking into consideration the patient's history of exposure. It can also be confirmed through the use of a lab test called polymerase chain reaction (PCR) [12, 13-16]. It can be difficult for individuals to self-diagnose monkeypox based on the appearance of the rash alone, as it may appear similar to other conditions. Early detection of the disease is challenging, even for people who are familiar with the characteristic rash patterns.

Monkeypox, a rare but potentially severe zoonotic disease, was first discovered in 1958 and shares clinical similarities with smallpox and chickenpox, thus making its accurate diagnosis a challenging endeavor. With recent reports of monkeypox outbreaks in various parts of the world, there is a growing need for timely and accurate diagnostic tools to aid in its early detection and management. The traditional methods of diagnosis, relying on clinical symptoms and laboratory tests, can be time-consuming and may not always provide definitive results.

The advent of computer-aided diagnosis (CAD) systems has revolutionized the field of healthcare by leveraging advanced technologies such as artificial intelligence (AI) and machine learning. CAD systems have proven invaluable in assisting healthcare professionals in the accurate and early diagnosis of various diseases, from cancer to infectious diseases. In the context of monkeypox, where prompt diagnosis can play a pivotal role in limiting its spread and guide appropriate clinical interventions, the integration of CAD systems into the diagnostic process holds great promise. The development and implementation of a computer-aided diagnosis system for monkeypox that has the potential to significantly enhance the accuracy and speed of diagnosis will ultimately improving patient outcomes and aid in the containment of outbreaks. This research represents a critical step towards harnessing the capabilities of modern technology to combat the challenges posed by emerging infectious diseases like monkeypox.

Machine learning (ML) is a growing field of artificial intelligence (AI) with established applications in medical diagnosis. In recent years, Deep Convolutional Neural Networks (DCNNs) have been successful applied in various domains of medical science. For instance, Jullapak and Yampaka [17] assessed the performance of three different deep learning models (VGG19, ResNet50, and InceptionV3) to detect coronavirus pneumonia in patients using chest X-ray images. The results suggested that the best performance was achieved with the VGG19 pre-trained model, which had an accuracy of 0.97, sensitivity of 0.97, specificity of 0.93, and an F1-score of 0.97. Yampaka *et al*. [18] proposed regression mutual information with DCNNs to extract the very large image features for COVID-19 classification. Their results showed that the RMI Deep-CNN approach significantly improved computation time and model accuracy for COVID -19 classification.

Data augmentation is a powerful technique that can help improve the performance and robustness of machine learning models. Data augmentation techniques are commonly used in machine learning and computer vision to artificially increase the size of a training dataset by applying various transformations to the existing data. These transformations create new, slightly modified versions of the original data, which helps improve the model's generalization, robustness, and performance. Data augmentation techniques are particularly useful when the available training data is limited. There are ten techniques of image augmentation commonly used [19]: 1) Rotation of images by various degrees (e.g., 90, 180, 270 degrees), 2) Flipping: Horizontally or vertically flipping images, 3) Scaling: Scaling images by resizing them to different dimensions, 4) Translation: Shifting images horizontally or vertically, 5) Shearing: Applying shearing transformations to the image, 6) Zooming: Zooming in or out of the image, 7) Crop and Pad: Randomly cropping and padding images, 8) Color Jitter: Modifying brightness, contrast, saturation, and hue, 9) Gaussian noise: Adding random Gaussian noise to the image, and 10) Blur: Applying Gaussian or motion blur to the image.

In addition, a number of studies involved the use of deep learning technique to classify monkeypox skin lesions. For example, a newly developed "Monkeypox2022" dataset was published and a modified VGG16 deep learning model for monkeypox classification was proposed and evaluated [20]. The study showed that the proposed model was able to accurately identify patients with monkeypox at a rate of 97%. In another study that employed deep learning models to classify monkeypox lesions, it was found that advanced AI deep models on skin images for monkeypox detection could be used for detecting monkeypox from digital skin images, with a precision of 85% [21-23].

Although previous studies were able to classify monkeypox using deep learning technique, only a few studies can develop a self-screening system. For example, a study by Ali and colleagues [24] created a dataset called the "Monkeypox Skin Lesion Dataset" (MSLD), which included skin lesion images of monkeypox, chickenpox, and measles. The best overall accuracy they achieved by ResNet50 model was 82.96%. In addition, they also developed an online monkeypox screening tool. It is not only the classification of monkeypox that is important, but also the knowledge of the evolution of the stage of the lesions is necessary because this information can be used to predict the period of symptoms. Monkeypox can be challenging due to its similarities with skin rashes caused by other Orthopoxviral infections. There are fine differences between the rashes which can be helpful in their differentiation, although laboratory analysis is required for a definitive identification [25]. Many studies reported that most people recover without treatment, primary care clinicians may be the first point of contact for those affected. Prompt assessment, diagnosis, isolation, treatment and prophylaxis will reduce the risk of community transmission [26, 27].

This research aimed to explore the potential of computer- aided diagnosis in the identification and classification of monkeypox cases. By harnessing the power of AI and machine learning algorithms, we sought to develop a CAD system that could analyze clinical data, medical images, and other relevant information to provide rapid and accurate monkeypox diagnoses. Such a system could assist healthcare practitioners, particularly in resource-constrained settings, where timely access to specialized expertise may be limited.

The main contribution of this study is the provision of a diagnosis tool for monkeypox that includes rapid tests that can help with early detection and containment. This study is concerned with developing effective public health interventions and strategies to prevent monkeypox outbreaks, including surveillance, contact tracing, and isolation measures.

In recent years, few studies in monkeypox classification focus on of the stage of the monkeypox lesion. Therefore, the steps of our study were outlined below:

(1) Skin lesions (monkeypox, chickenpox, measles, and normal) were classified using Deep Convolutional Neural Networks (DCNNs).

(2) Monkeypox rash stages were classified into five stages: macules (1-2 days), papules (1-2 days), vesicles (3-5 days), pustules (5-7 days), scabs (7-14), and normal and classified using Deep Convolutional Neural Networks (DCNNs).

(3) Two datasets (skin lesions and monkeypox rash stages) were trained using two light-weight deep models for edge devices such as MobileNetV2 and EfficientNet and the performance of the models were compared.

(4) The best performing model was selected and converted to Tensorflow Lite for implementation in a mobile application.

2. Materials and Methods

2.1 Dataset

During the spread of COVID-19, we observed that AI-based image processing was able to diagnose and prevent the disease [14, 16, 28]. Healthcare professionals have become increasingly concerned about the most recent monkeypox outbreak. The entire dataset was developed by the Department of Computer Science and Engineering, Islamic University, Kushtia-7003, Bangladesh [29]. This dataset consisted of four classes of images: 279 monkeypox, 107 chickenpox, 91 measles, and 293 normal. Figure 1 shows some example images from our experiment dataset. Our focus was to distinguish cases of monkeypox from cases that could be mistaken for it, such as chickenpox. Hence, several data augmentation methods including blur, noise, brightness, darkness, and crop were applied to improve the classification performance. Table 1 shows the data augmentation techniques that were used in this study. In addition, Table 2 shows the sample sizes of each class.

Additionally, many individuals are concerned about any lump or bump on their skin. To help with identification of uncertain cases of monkeypox lesion, health agencies such as the U.S. Centers for Disease Control and Prevention (CDC) and the U.K.'s National Health Service (NHS) have shared images to help distinguish monkeypox lesions at any stage of the disease. These images can be used to identify the illness and to estimate the stages of the disease and the duration of the stages. Therefore, a monkeypox rash stage dataset was included in this study. Sample of the lesions through five stages are shown in Table 3. Figure 2 shows example skin rash or lesions of monkeypox and the duration of the stages.

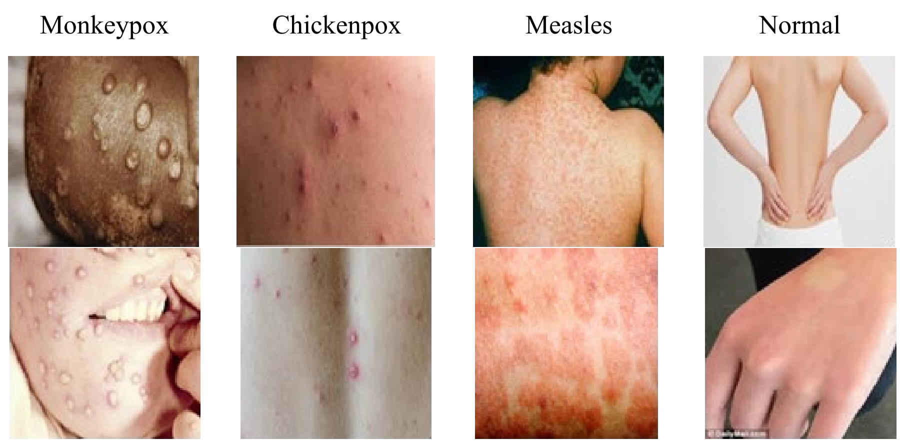


Figure 1. Examples of skin images of monkeypox, chickenpox, and healthy cases from the experiment dataset

Table 1. Data augmentation techniques used in this study

Technique	Facility
Blur	5px
Noise	5%
Brightness	+50%
Darkness	-50%
Crop	50%

Table 2. The sample size of each class used in this study

Dataset	Original	Augmentation
Monkeypox	279	780
Chickenpox	107	262
Measles	91	270
Normal	293	441
Total	770	1,753

Table 3. The sample size of lesions through five stages

Dataset	Original	Augmentation
Macules (1-2 days)	71	231
Papules (1-2 days)	52	144
Vesicles (3-5 days)	48	141
Pustules (5-7 days)	120	365
Scabs (7-14)	35	77
Normal	91	293
Total	417	1,251

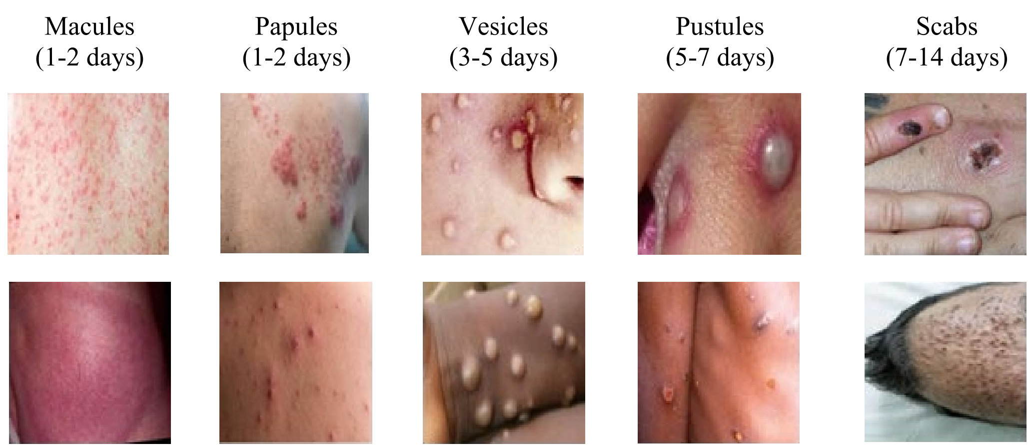


Figure 2. Examples of skin rash or lesions of monkeypox and the duration of the stages

2.2 Experiment

2.2.1 Model architecture

In this study, a prototype of a mobile application was deployed for self-screening. Therefore, an efficient portable model was developed for this work. TensorFlow Lite is a set of tools that enables on-device machine learning by helping developers run their models on mobile, embedded, and edge devices. Pre-trained models like EfficientNet, and MobileNet are available on the TcCensorFlow Hub. These pre-trained models can recognize 1000 classes of images. Transfer learning is a machine learning technique where a model trained on one task is re-purposed on a second related task. This approach is particularly useful in deep learning, where models trained on large datasets can be finetuned on smaller, domain-specific datasets to achieve good performance with less data. By using transfer learning, the knowledge and features learned by the pre-trained model on a large dataset, reducing the amount of training data and time required to train a model for a specific task. This approach is especially useful when working with limited datasets or when training deep learning models, which can be computationally expensive and time-consuming.

The TensorFlow Lite Model Maker library was used in this study for training and converting the lite model for mobile applications. This library supports EfficientNet, and MobileNetV2. In this study, we not only created a prediction model but also compared the performances of two models.

The EfficientNet was developed by Tan and Le [30] that optimized both accuracies and FLOPS when training the adaptive EfficientNet models on ImageNet. Their experiments revealed that the compound scaling method could further enhance accuracy by up to 2.5% compared to other single dimension scaling methods, indicating the significance of the proposed compound scaling method.

The MobileNetV2 architecture was an improvement of MobileNetV1. The improvements resulted in a significant increase in the model's accuracy. The main modifications included the addition of inverted residual blocks and linear bottlenecks, as well as the replacement of ReLU with the ReLU6 activation function.

When EfficientNet was compared with other pre-trained models such as MobileNetV2 in terms of model size and inference time. The MobileNetV2 decreased as expected when conversion to TensorFlow Lite. However, the EfficientNet model tends to achieve higher accuracy than the MobileNet model. Tan and Le [30] also determined that if model size and inference time are a priority over accuracy, then MobileNetV2 should be used, whereas if model accuracy is more important, then the EfficientNet model should be chosen over MobileNetV2. To test this assumption, not only monkeypox classification but also a comparison of the performance of both models were undertaken in this study.

2.2.2 Computation setup and performance evaluation

The digital skin images were divided into training data (80%), validation data (10%) and testing data (10%). Since the original data has an imbalance between classes, different numbers of augmented images per class were used during training to balance the training data. Tables 2 and 3 show the distribution of the number of images.

Training and testing were performed on a Google Colaboratory with Ubuntu 16.04 operating system using a Tesla K80 GPU graphics card. In this work, images of dimensions (224, 224, 3) were used as input and fed into two pre-trained models, one of which was the EfficientNet model and the other was the MobileNetV2 model. The architectures of the two models were different from each other. The output from the last layer was followed by fully connected layers. Lastly, a fully connected (FC) layer was employed with a softmax activation function for the classification task. The number of training epochs was set to 100.

The parameters of accuracy, sensitivity, and specificity were used to evaluate the performance of the EfficientNet model and MobileNetV2 model. The calculation types of the metrics are shown in equations (2)-(4), where TP, FN, FP, TN represent the number of true positives, false negatives, false positives and true negatives.

$$\boldsymbol{Sensitivity} = \frac{\boldsymbol{TP}}{\boldsymbol{TP + FN}} \tag{2}$$

$$Specificity = \frac{TN}{TN + FP} \tag{3}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4}$$

2.2.3 Model maker and mobile deployment

When deploying a TensorFlow neural-network model for edge device machine learning applications, the TensorFlow Lite Model Maker is a tool that simplifies the process of converting and customizing a model to fit specific input data. Machine learning models that use TensorFlow Lite are created and trained using TensorFlow core libraries and tools, and then converted into a more efficient format called a TensorFlow Lite model. Before building the TensorFlow Lite model, a model using the TensorFlow core libraries was done. Then, this model was converted to Tensorflow Lite. Figure 3 presented the monkeypox detection system integrates TensorFlow Core models, converting them into TensorFlow Lite models, and then uses the mobile application.

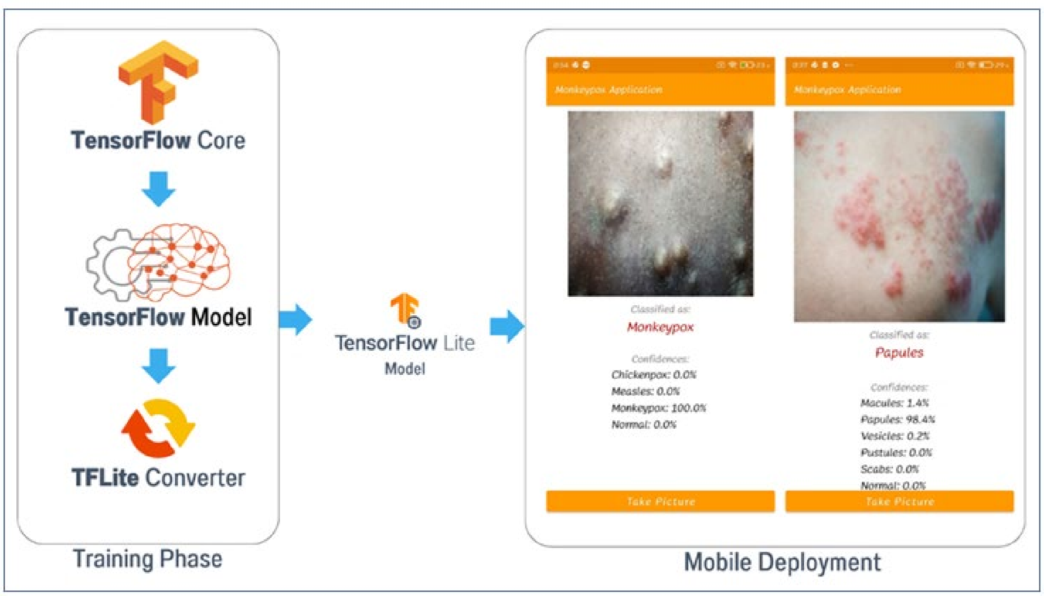


Figure 3. The proposed monkeypox detection system integrates TensorFlow Core models, converting them into TensorFlow Lite models, and then uses the mobile application.

TensorFlow Lite supports EfficientNet models, and MobileNetV2 as pre-trained models for image classification. The step-by-step was explained below.

Step 1: Input data specific to a mobile application were loaded using DataLoader class in keras library. It was assumed that all image files that belonged to the same category were stored in the same subfolder, with the class name as the name of the folder. Images that had been compressed using JPEG or PNG were currently supported. Then, the images were split into training data (80%), validation data (10%) and testing data (10%) sets.

Step 2: The custom image classifier model based on the loaded data was created. The default model used was EfficientNet-Lite0, however, it was possible to switch to MobileNetV2 by simply adjusting the model_spec parameter to the MobileNetV2 model specification.

Step 3: The customized model was evaluated by assessing its results, calculating the loss and accuracy of the model. If the accuracy did not meet the desired level for the application, it was possible to improve the model by considering options such as using a larger model or altering the re-training parameters.

Step 4: The trained model was transformed into the TensorFlow Lite format, which could be used in a mobile application later on. The default file name for the TFLite model was "model.tflite".

2.2.4 Mobile deployment

TensorFlow Lite utilizes TensorFlow models were transformed into smaller, more efficient, and more easily transportable machine learning model formats. When the model works on data, that is when it performs an inference, it produces prediction results in the form of new tensors, and sends them to the Android application. The step by step manner is explained as follows.

Step 1: To start using the TensorFlow Lite machine learning model interpreter, it was necessary to load the model file and set the model parameters. The TensorFlow Lite model was composed of a ".tflite" file that held the model's code and was stored in the src/main/assets directory of the project.

Step 2: The interpretation image was transformed into the Tensor data format. Image frames were extracted to the camera subsystem by ImageAnalysis object. The camera subsystem created a bitmap buffer to hold the data received from the camera. Once a TensorImage object was created, the model could then be run on that data to produce a prediction or inference.

Step 3: Once the image data was processed by the object detection model, it generated a list of predictions for the objects that were detected.

The moblie deployment can be done using Android Studio. The TensorFlow website provides code example that can modify and apply them to custom dataset in mobile application.

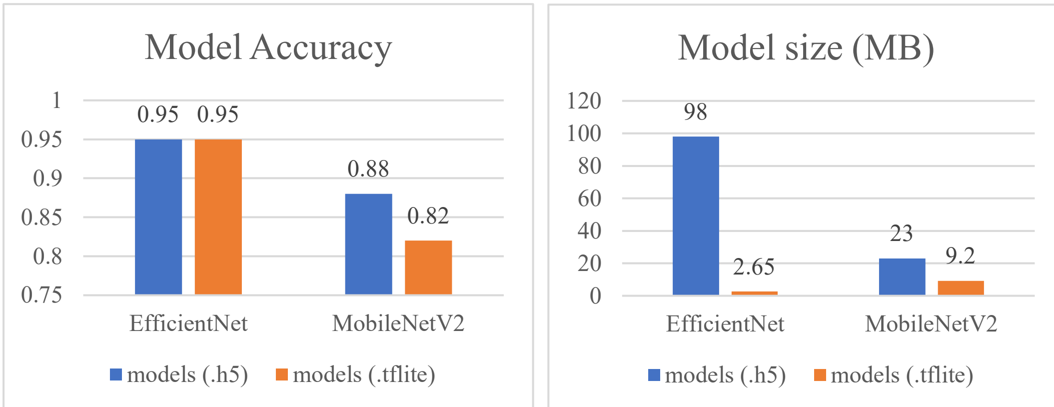
3. Results and Discussion

3.1 Comparison of the accuracy, model size and inference time

When implementing sophisticated deep learning models on mobile devices and edge devices, the constraints on processing power and memory with a good speed tends to be challenging. It is challenging when the trade-off between the lightweight model, priority accuracy, and speed. Figure 4 show the results of the comparison of the initial model and converted models.

3.1.1 Skin lesion classification

In the training of initial models (.h5) for skin lesion classification, the testing accuracy values of all model were recorded. According to the results, while EfficientNet had a higher testing accuracy, MobileNetV2 had a shorter inference time. Specifically, MobileNetV2 had a testing accuracy of 0.88, while the range for EfficientNet was from 0.90 to 0.95. Furthermore, when comparing the inference time of the two models, MobileNetV2 had the lowest time recorded, with little variation between the two models. When considering the model size, the MobileNetV2 had the lowest model size of 23MB. The model sizes for all EfficientNet models ranged from 45MB to 98MB (Figure 4).



(a) **(b)**

Figure 4. The performance comparison of initial models (a) and converted models (b)

In the converted models (.tflite) MobileNetV2 showed slightly lower testing accuracy in comparison with EfficientNet. The results indicated that MobileNetV2 had a testing accuracy of 0.82, while EfficientNet models had a testing accuracy range of 0.89 to 0.95 (Table 4). Additionally, MobileNetV2 was found to have the fastest inference time when compared to EfficientNet. There were non-significant differences in inference time when computed for both models. When considering the model size, the MobileNetV2 had the lowest model size of 2.65MB. The model size for all EfficientNet models ranged from 3.82MB to 9.2MB (Figure 4).

Table 4. the model performance of training of initial models (.h5) for skin lesion classification

models (.h5)	**EfficientNet**	**MobileNetV2**
Accuracy	0.95	0.88
Inference time	Slow	Fast
Model size	98MB	23MB
models (.tflite)	**EfficientNet**	**MobileNetV2**
Accuracy	0.95	0.82
Inference time	Slow	Fast
Model size	2.65MB	9.2MB

3.1.2 Rash stage classification

In the training of initial models (.h5) for rash stage classification, the testing accuracy values of all models were recorded. The study results indicated that while EfficientNet had a slightly higher testing accuracy, MobileNetV2 had a faster inference time. Specifically, MobileNetV2 had a testing accuracy of 0.88, specificity of 0.85, and sensitivity of 0.89. EfficientNet models had a testing accuracy range of 0.94, specificity of 0.93, and sensitivity of 0.95. Additionally, MobileNetV2 was found to have the quickest inference time when compared to EfficientNet. The results revealed that there were minimal variations in the inference time of both models. Additionally, when evaluating the model size, MobileNetV2 was found to have the smallest size of 33MB. The model size for all EfficientNet models ranged from 55MB to 103MB.

In the .tflite versions of the models, EfficientNet had slightly higher testing accuracy than MobileNetV2. Specifically, the testing accuracy for MobileNetV2 was 0.80, specificity of 0.86, and sensitivity of 0.77. The EfficientNet models had a range of testing accuracy of 0.93, specificity of 0.93, and sensitivity of 0.94 (Table 5). The comparison of inference time between MobileNetV2 and EfficientNet showed that MobileNetV2 had the shortest inference time. The differences in inference time between the two models were not considered to be significant. When considering the model size, the MobileNetV2 had the lowest model size of 2.7MB. The model sizes for all EfficientNet models ranged from 2.77MB to 10.40MB.

Table 5. The model performance of training of initial models (.h5) for rash stage classification

models (.h5)	EfficientNet	MobileNetV2
Accuracy	0.94	0.88
Specificity	0.93	0.85
Sensitivity	0.95	0.89
models (.tflite)		
Accuracy	0.93	0.80
Specificity	0.93	0.86
Sensitivity	0.94	0.77

The results made selection for implementation on mobile application clear. If model size and inference time were more important than accuracy, then MobileNetV2 was the preferred choice, but if disease classification required the highest accuracy, then another model needed to be selected. Consequently, the model with the highest lesion classification accuracy, which was EfficientNet-Lite4, was selected for implementation on the mobile application.

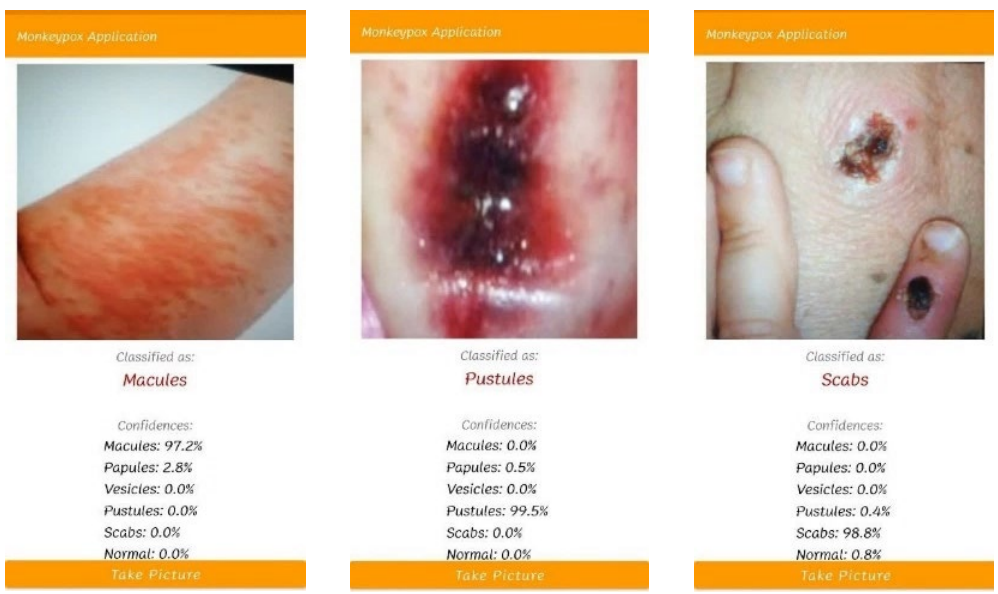
3.2 Mobile application deployment

The most effective model has been implemented in a mobile application prototype. This meant that users could take a picture of their skin lesions and receive an initial diagnosis of the skin lesions. The users could also take a picture of their rash and receive an initial diagnosis of the rash stage. Figure 5 shows the prototype of mobile application.

Because of the limitations of sample size, several data augmentation methods including blur, noise, brightness, darkness, and crop were applied to improve the classification performance. Balancing the training data was achieved by using different numbers of augmented images per class. Then, the training phase was performed followed by conversion of optimized trained model from the initial model (.h5) to TensorFlow Lite model (.tflite). The TensorFlow Lite model makes it easier to adjust and convert a TensorFlow neural-network model to suit specific input data when deploying it for on-device machine learning applications, such as mobile applications. After model maker process was established, the mobile application was implemented using Android Studio. The model performances were measured and discussed and the following issues presented:

(1) The MobileNetV2 model tended to have lower inference time than EfficientNet models, nevertheless, the differences were non-significant. When considering the model size, the MobileNetV2 had the lowest model size. However, most current deep learning models still have large sizes and complex structures, which makes it difficult to perform real-time image classification

(a)



(b)

Figure 5. The prototype of mobile application (a) the interface of the skin lesion classification screening tool (b) the interface of the rash stage lesion classification screening tool

and object detection on mobile devices with limited resources [29]. While MobileNetV2 had the advantage of a smaller model size, it had slightly lower testing accuracy when compared to EfficientNet. These results were in agreement with the finding of Afzaal *et al*. [31], who found that GoogleNet had the quickest inference time compared to EfficientNet and VGGNet. Simonyan and Zisserman [32] conducted a comparison of pre-trained models for disease detection and found that there were minimal differences in performance between VGGNet and EfficientNet. In addition, Bianco *et al*. [33] proposed the use of a smart sprayer for precise application of fungicides in potato fields, which utilized the EfficientNet architecture.

(2) Model for Predictive Performance: During the classification of pox symptoms, the monkeypox, chickenpox, measles, and normal conditions were the main targets for the DCNNs to pick up during the use of DL model for deployment on mobile application for self-screening. In addition, the estimation of the disease stage and time period in macules (1-2 days), papules (1-2 days), vesicles (3-5 days), pustules (5-7 days), and scabs (7-14) is interesting to classify time period in macules. Therefore, the development of a self-screening system in pox symptoms and pox rash stage using a mobile application was performed in this study. An EfficientNet was created and evaluated for its ability to identify patients with monkeypox and to determine the stages and progression of the disease. Our approach achieved an accuracy of 0.95 for pox classification and 0.97 for pox rash stage determination. These results were consistent with the findings of Islam *et al*. [21], who concluded that lighter deep models had the potential for pox classification use on smartphones.

4. Conclusions

The use of deep learning techniques for monkeypox lesion and rash stage classification on a mobile application can enable early identification of infected patients and effective control of community spread through self-screening. In our study: 1) Various skin lesions (monkeypox, chickenpox, measles, and normal) were classified using Deep Convolutional Neural Networks (DCNNs). 2) The monkeypox rash stages were classified into five stages such as macules (1-2 days), papules (1-2 days), vesicles (3-5 days), pustules (5-7 days), scabs (7-14), and normal conditions using Deep Convolutional Neural Networks (DCNNs). 3) Two datasets (skin lesions and monkeypox rash stages) were trained using two light-weight deep models for edge devices such as MobileNetV2 and EfficientNet and the performances of the models were compared. 4) The best performance model was selected and converted to Tensorflow Lite for implementation in a mobile application. In addition, not only monkeypox skin lesions were classified in this study but monkeypox rash stages were also investigated and classified using Deep Convolutional Neural Networks (DCNNs). Finally, physician can easily adapt our model, which is cost and time effective and does not require extensive PCR or microscopy testing, for first diagnosis. Although this study provides many advantages, the study had some limitations. First, the training dataset contained a limited number of samples. Second, the accuracy of the system needs to be further improved. Finally, a limitation of classification on mobile applications is that the image must be zoomed in for image sharpness if accurate predictions are to be made.

Monkeypox research is a complex and challenging field. It may require interdisciplinary collaboration and a willingness to explore unconventional ideas. This study can be essential in pursuing novel research directions. Additionally, staying updated with the latest scientific advancements and technologies in related fields is crucial for identifying opportunities for innovation in monkeypox research. This limitation produced many challenges during the analysis. The use of GAN-AI to create syntactic datasets may be helpful to improve in future studies.

References

[1] McCollum, A.M. and Damon, I.K., 2013. Human monkeypox. *Clinical Infectious Diseases*, 58(2), 260-267, https://doi.org/10.1093/cid/cit703.

[2] Farahat, R.A., Ali, I., AL-Ahdal, T., Benmelouka, A.Y., Albakri, K., El-Sakka, A.A., Abdelaal, A., Abdelazeem, B., Anwar, M.M., Mehta, R., Sah, R., Rouniyar, R. and Sah, R., 2022. Monkeypox and human transmission: Are we on the verge of another pandemic? *Travel Medicine and Infectious Disease*, 49, https://doi.org/10.1016/j.tmaid.2022.102387.

[3] Adler, H., Gould, S., Hine, P., Snell, L.B., Wong, W., Houlihan, C.F., Osborne, J.C, Rampling, T., Beadsworth, M.BJ., Duncan, C.J., Dunning, J., Fletcher, T.M., Hunter, E.R., Jacobs, M., Khoo, S.H., Newsholme, W., Porter, D., Porter, R.J., Ratcliffe, L., Schmid, M.L., Semple, M.G., Tunbridge, A.J., Wingfield, T. and Price, M.N., 2022. Clinical features and management of human Monkeypox: A retrospective observational study in the UK. *The Lancet Infectious Diseases*, 22(8), 1153-1162, https://doi.org/10.1016/s1473-3099(22)00228-6.

[4] Li, Y., Olson, V.A., Laue, T., Laker, M.T. and Damon, I.K., 2006. Detection of Monkeypox virus with real-time PCR assays. *Journal of Clinical Virology*, 36(3), 194-203, https://doi.org/10.1016/j.jcv.2006.03.012.

[5] Saijo, M., Ami, Y., Suzaki, Y., Nagata, N., Iwata, N., Hasegawa, H., Ogata, M., Fukushi, S., Mizutani, T., Iizuka, I., Sakai, K., Sata, T., Kurata, T., Kurane, I. and Morikawa, S., 2008. Diagnosis and assessment of monkeypox virus (MPXV) infection by quantitative PCR assay: differentiation of Congo Basin and West African MPXV strains. *Japanese Journal of Infectious Diseases*, 61(2), 140-142.

[6] Antinori, A., Mazzotta, V., Vita, S., Carletti, F., Tacconi, D., Lapini, L.E., D'Abramo, A., Cicalini, S., Lapa, D., Pittalis, S., Puro, V., Capparuccia, M.R., Giombini, E., Gruber, C.E.M., Garbuglia, A.R., Marani, A., Vairo, F., Girardi, E., Vaia, F., Nicastri, E. and the INMI Monkeypox Group, 2022. Epidemiological, clinical and virological characteristics of four cases of monkeypox support transmission through sexual contact, Italy, May 2022. *Eurosurveillance,* 27(22), https://doi.org/10.2807/1560-7917.es.2022.27.22.2200421.

[7] Sklenovská, N. and Van Ranst, M., 2018. Emergence of monkeypox as the most important orthopoxvirus infection in humans. *Frontiers in Public Health,* 6, https://doi.org/10.3389/fpubh.2018.00241.

[8] Grant, R., Nguyen, L.-B.L. and Breban, R., 2020. Modelling human-to-human transmission of Monkeypox. *Bulletin of the World Health Organization*, 98(9), 638-640, https://doi.org/10.2471/blt.19.242347.

[9] Angelo, K.M., Petersen, B.W., Hamer, D.H., Schwartz, E. and Brunette, G., 2019. Monkeypox transmission among international travellers—serious monkey business? *Journal of Travel Medicine*, 26(5), https://doi.org/10.1093/jtm/taz002.

[10] Abdelaal, A., Serhan, H.A., Mahmoud, M.A., Rodriguez-Morales, A.J and Sah, R., 2023. Ophthalmic manifestations of Monkeypox virus. *Eye*, 37(3), 383-385, https://doi.org/10.1038/s41433-022-02195-z.

[11] Response, E., 2023. *Diagnostic Testing for the Monkeypox Virus (MPXV): Interim Guidance, 9 November 2023*. [online] Available at: https://www.who.int/publications/i/item/who-mpx-laboratory-2023-1.

[12] Ahsan, M.M., Gupta, K.D., Islam, M.M., Sen, S., Rahman, M.L. and Hossain, M.S., 2020. Covid-19 symptoms detection based on NasNetMobile with explainable AI using various imaging modalities. *Machine Learning and Knowledge Extraction*, 2(4), 490-504, https://doi.org/10.3390/make2040027.

[13] Ahsan, M.M., Alam, T.E., Trafalis, T. and Huebner, P., 2020. Deep MLP-CNN model using mixed-data to distinguish between COVID-19 and non-covid-19 patients. *Symmetry*, 12(9), https://doi.org/10.3390/sym12091526.

[14] Ahsan, M.M., Ahad, M.T., Soma, F.A., Paul, S., Chowdhury, A., Luna, S.A. and Huebner, P., 2021. Detecting SARS-COV-2 from chest x-ray using artificial intelligence. *IEEE Access*, 9, 35501-35513, https://doi.org/10.1109/access.2021.3061621.

[15] Ahsan, M.M., Nazim, R., Siddique, Z. and Huebner, P., 2021. Detection of COVID-19 patients from CT scan and chest X-ray data using modified mobilenetv2 and lime. *Healthcare*, 9(9), https://doi.org/10.3390/healthcare9091099.

[16] Lecun, Y., Bottou, L., Bengio, Y. and Haffner, P., 2019. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324, https://doi.org/10.1109/5.726791.

[17] Jullapak, R. and Yampaka, T., 2021. Covid-19 classification using DCNNs and exploration correlation using canonical correlation analysis. *Proceedings of the 18th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, Thammasat University, Lampang Campus, Thailand, June 30-July 3, 2021, pp. 1-6, https://doi.org/10.1109/jcsse53117.2021.9493846.

[18] Yampaka, T., Vonganansup, S. and Labcharoenwongs, P., 2022. Feature selection using regression mutual information deep convolution neuron networks for covid-19 x-ray image classification. *International Journal of Advances in Intelligent Informatics*, 8(2), https://doi.org/10.26555/ijain.v8i2.809.

[19] Shorten, C. and Khoshgoftaar, T.M., 2019. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6, https://doi.org/10.1186/s40537-019-0197-0.

[20] Ahsan, M.M., Uddin, M.R., Farjana, M., Sakib, A.N., Momin, K.A. and Luna, S.A., 2022. Image data collection and implementation of deep learning-based model in detecting Monkeypox disease using modified VGG16. *arXiv*, https://doi.org/10.48550/arxiv.2206.01862.

[21] Islam, T., Hussain, M.A., Chowdhury, F.U.H. and Islam, B.M.R., 2022. Can artificial intelligence detect monkeypox from digital skin images? *BioRxiv Preprint*, https://doi.org/10.1101/2022.08.08.503193.

[22] Sitaula, C. and Shahi, T.B., 2022. Monkeypox virus detection using pre-trained deep learning-based approaches. *Journal of Medical Systems*, 46(11), https://doi.org/10.1007/s10916-022-01868-2.

[23] Park, A., 2022. *There's Already a Monkeypox Vaccine. But Not Everyone May Need It*. [online] Available at: https://time.com/6179429/monkeypox-vaccine.

[24] Ali, S.N., Ahmed, M.T., Jahan, T., Paul, J., Sani, S.M.S., Noor, N., Asma, A.N. and Hasan, T., 2023. A web-based Mpox skin lesion detection system using state-of-the-art deep learning models considering racial diversity. *arXiv Preprint*, https://doi.org/10.48550/arXiv.2306.14169.

[25] Petersen, BW. and Damon, I.K., 2020. Smallpox, monkeypox, and other poxvirus infections. In: L. Goldman and A.I. Schafer, eds. *Goldman-Cecil Medicine.* 26th ed. Philadelphia, PA: Elsevier.

[26] Moore, Z. S., 2008. *Monkeypox*. In: S.S. Long, ed. *Principles and Practice of Pediatric Infectious Disease.* 3rd ed. Amsterdam: Elsevier

[27] Altindis, M., Puca, E. and Shapo, L., 2022. Diagnosis of monkeypox virus – an overview. *Travel Medicine and Infectious Disease*, 50, https://doi.org/10.1016/j.tmaid.2022.102459.

[28] Narin, A., Kaya, C. and Pamuk, Z., 2021. Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks. *Pattern Analysis and Applications*, 24(3), 1207-1220, https://doi.org/10.1007/s10044-021-00984-y.

[29] Bala, D, 2022. *Monkeypox Skin Images Dataset (MSID)*. [online] Available at: https://www.kaggle.com/datasets/dipuiucse/monkeypoxskinimagedataset.

[30] Tan, M. and Le, Q.V., 2019. *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. [online] Available at: https://arxiv.org/pdf/1905.11946.pdf.

[31] Afzaal, H., Farooque, A.A., Schumann, A.W., Hussain, N., McKenzie-Gopsill, A., Esau, T., Abbas, F. and Acharya, B., 2021. Detection of a potato disease (early blight) using artificial intelligence. *Remote Sensing*, 13(3), https://doi.org/10.3390/rs13030411.
[32] Simonyan, K. and Zisserman, A., 2014. *Very Deep Convolutional Networks for Large-scale Image Recognition*. [online] Available at: https://arxiv.org/pdf/1409.1556.pdf.
[33] Bianco, S., Cadene, R., Celona, L. and Napoletano, P., 2018. Benchmark analysis of representative deep neural network architectures. *IEEE Access*, 6, 64270-64277, https://doi.org/10.1109/access.2018.2877890.