*ICIC Express Letters*
*Part B: Applications*
Volume **12**, Number **8**, August **2021**  ISSN 2185-2766
pp. **715–721**

# CLASSIFICATION OF SKIN DISEASES AND DISORDERS USING CONVOLUTIONAL NEURAL NETWORK ON A MOBILE APPLICATION

Nico Bernando Setiawan[1], Friska Natalia[1,*]
Ferry Vincenttius Ferdinand[2], Sud Sudirman[3] and Chang Seong Ko[4]

[1]Department of Information Systems
Universitas Multimedia Nusantara
Scientia Boulevard, Gading Serpong, Tangerang, Banten 15811, Indonesia
nico.setiawan@student.umn.ac.id; *Corresponding author: friska.natalia@umn.ac.id

[2]Department of Mathematics
Universitas Pelita Harapan
Tangerang, Banten 15811, Indonesia
ferry.vincenttius@uph.edu

[3]School of Computer Science and Mathematics
Liverpool John Moores University
Liverpool L3 3AF, UK
s.sudirman@ljmu.ac.uk

[4]Department of Industrial and Management Engineering
Kyungsung University
309, Suyeong-ro, Nam-gu, Busan 48434, Korea
csko@ks.ac.kr

Received December 2020; accepted February 2021

Abstract. *Skin diseases and disorders are common, yet underestimated, in Indonesia. More serious types of them are often left untreated for a long period of time because of the lack of information regarding the therapeutic process of its treatment and available medical support. In this study, we use a deep learning approach using deep convolutional neural network to classify different types of skin diseases and disorders, namely psoriasis, ringworm, and eczema. The algorithm is implemented as an Android-based mobile application app because of the pervasive use of the Android platform in Indonesia. The app development uses the TensorFlow library for its low-level implementation of deep learning and the Android Studio IDE for high-level processes. In addition, the application also provides information on how to treat any skin diseases, it also provides a list of hospitals in the district and the city where the user resides if they wish to get medical treatment immediately. The artefact of this research is available on the PlayStore for people to download and use.*
**Keywords:** Skin diseases and disorders, Machine learning, Deep convolutional neural network, TensorFlow, Android studio

1. **Introduction.** As the outermost part of our body and being the first to come into contact with other objects or organisms, our skin is vulnerable to diseases that cause a skin disorder. Often, these skin diseases disrupt a person's activities and can even damage one's appearance. Many people underestimate the danger of skin disease when they got them. They often think that the disease will disappear by itself without proper medical treatment. As a result, the disease could spread to wider areas of the skin and become more dangerous. Skin disorders are often associated with clogged pores, inflammation, or irritation caused by an infection or immune disorder. Some skin diseases, such as

DOI: 10.24507/icicelb.12.08.715

ringworm, psoriasis, and eczema, have symptoms that are similar to each other including itching and irritation of the skin and thickening of the skin that makes it scaly. These three skin diseases require different treatments and can be a symptom of more serious diseases. They are some of the most common skin conditions in the world. Approximately 20% of children are affected by eczema while psoriasis affects about 1.5% of the British population. Skin disorders are also more prevalent in adults aged 70 and older [1].

Images of skin disease can have vast variations in terms of color, texture, and spread hence as a result, the task of evaluating them is very challenging. There are currently software programs that can analyze skin images taken by smartphone cameras; however, this software and the algorithms they use are proprietary. This hinders any use of the algorithms in other applications and to solve other problems [2]. Several authors have attempted to design an automatic detection system using Deep Convolutional Neural Network (DCNN) [3,4]. DCNN has recently been gaining popularity within the research community due to its ability in recognizing objects in images. This task falls into the general category of the image classification problem. However, the classification of medical images has unique challenges compared to the classification of other more general images. Firstly, they have relatively high intra-class variation and inter-class similarity compared to other types of images and secondly, the size of the medical image dataset is considerably smaller than datasets of other types of images. DCNN has also been successfully used for other medical image processing tasks such as medical image segmentation [5], boundary delineation [6], as well as a physiological measurement of organs [7] using magnetic resonance imaging images.

The proposed work aims at recognizing and classifying skin diseases present in the human body using the Deep Convolutional Neural Network (DCNN) method. We developed our dataset containing images of the diseases with the assistance of medical experts. Major skin diseases such as ringworm, psoriasis, and eczema are taken into consideration. The resulting model is then compressed and deployed as a mobile phone application to increase usage and penetration of the research in the wider community. By using this application, the user is expected to be able to recognize the disease more quickly and be informed of all the relevant information including its treatment method and where to get medical assistance if urgently needed. Compared to other similar approaches in the literature, such as [2], the solution that we proposed in this paper contains significant innovation in terms of the compression of the DCNN model to allow deployment in smartphones or small embedded devices without severely impacting the detection accuracy.

This paper is organized as follows. In the next section, we will discuss the methodology that we propose and the image dataset we used. This is then followed by an analysis and discussion of the experiment results before we present the conclusion of the research.

## 2. **Material and Method.**

2.1. **Material.** We used a dataset which is specifically developed for this study. Since we aim to have the final application to be deployed to smartphones and for the application to utilize the phone camera to take a picture of the user's skin condition, we opted not to use any of the pre-existing datasets. The rationale of this decision is based on the fact that the images in these datasets were predominantly taken using professional equipment hence unsuitable for our purpose. We, instead, search the Internet using several search engines for relevant skin images. The skin images are grouped into four categories, namely eczema, ringworm, psoriasis, and healthy. Medical experts are involved in the data verification process to ensure the dataset's accuracy and correctness. The process yielded 2,000 images in total with each category consisting of 500 images.

To significantly increase the size and the diversity of the dataset we employ the data augmentation process to the initial 2,000 images. The purpose of this step is to increase the generality of the classifier that we use in the subsequent step. The more general the classifier, the better it will perform when attempting to infer previously unseen cases. In our case, the data augmentation process randomly performs image rotation with a maximum rotation of 40 degrees, image flipping in either vertical or horizontal direction, and modulation of image brightness by a factor between 0.5 to 1.5. As a final step, the images are then cropped and resized to $224 \times 224$ to maximize the contribution of the skin area in the images to the training and classification processes and to ensure adherence to the requirement of the CNN model used. The distribution of the number of images per category in the final form of our dataset is summarized in Table 1.

Table 1. The distribution of the number of images per category in the dataset

| Class ID | Class name | Original size | Number of augmented images | Total |
|---|---|---|---|---|
| 1 | Eczema | 500 | 1500 | 2000 |
| 2 | Ringworm | 500 | 1500 | 2000 |
| 3 | Psoriasis | 500 | 1500 | 2000 |
| 4 | Healthy | 500 | 1500 | 2000 |
| **Total** | | | | 8000 |

2.2. **Method.** The method that we used as a solution to the problem we posed in Section 2 is based on three concepts, namely a) the transfer learning of learned parameters of a pre-trained CNN model, b) the compression of the model to make it more compact and utilizes less computing power, and c) the deployment of the compressed model to mobile phones. The overview of our proposed method is illustrated in Figure 1.

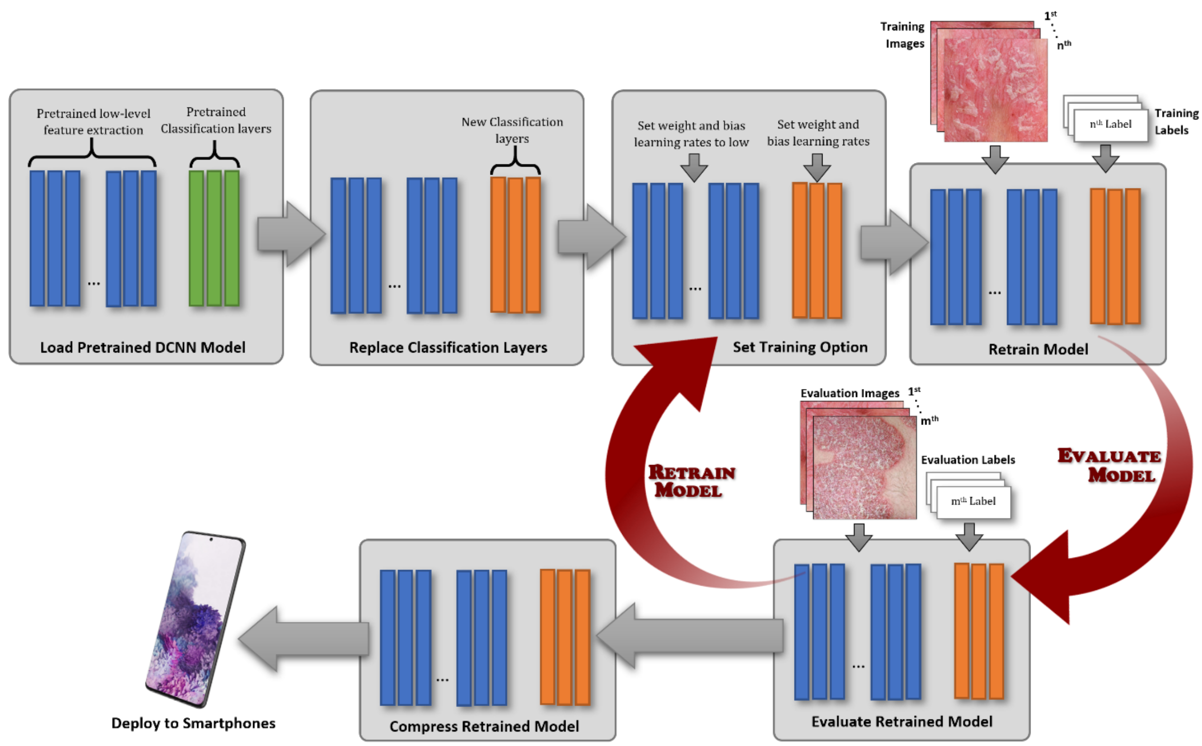



Figure 1. The overview of the proposed method for skin disorder classification on mobile phone

There are several pre-trained DCNN models that can be used for image classification, and each has advantages and disadvantages over the others. The first design decision is to choose the most suitable DCNN model for the task at hand. After reviewing all of the suitable models, we decided to use ResNet50 [8] due to the model's more superior ability to adapt to new datasets/tasks compared to other similar-sized models [9]. In addition, the model was found to be able to handle stronger compression than other models when tested. An experiment conducted in [10] shows that ResNet50 can be compressed by 20× compression factor and still preserves the top-1 accuracy of an uncompressed ResNet-50 on ImageNet [11].

We adopt transfer learning of the chosen model as opposed to retraining the whole network due to the small dataset size. Transfer learning is a process that transfers the learned features from the feature extraction layers of a pre-trained, called the base, network on to another, called the target, network. It is performed by copying the first $n$ layers of the base network onto the first $n$ layers of a target network with any remaining layers of the target network to be randomly initialized. A new classification layer is then added at the end of the network with a new randomly initialized set of weight and bias values. The whole target network is then trained using the new dataset to classify the new image categories by setting the learning parameter too low to avoid overriding the learned features.

The network training is carried out as an iterative process in which we adjust the setting and parameter values of the training set up to increase the validation performance of the network. This includes choosing the best number of epochs, the ratio of the training: validation set, and the k-fold number in the cross-validation setup.

Once the network's training has been completed and the final model is found, it is then compressed using the TensorFlow Lite. The process is designed to allow the model to be deployed onto smartphones or embedded devices such as Raspberry Pis and for the computation to be carried out locally instead of sending data back and forth from a server. This process also improves on some other computing aspects of the system deployment including a) latency, by eliminating the need for round-trip communication with a remote server, b) privacy, by storing all data locally in the device and not in a remote server, c) connectivity by removing the Internet connection requirement and d) power consumption by removing the need for network connections and communication.

3. **Experiment Result, Analysis, and Discussion.** The first training of the network was carried out in 100 epochs using a modest training rate of 0.0001 for the feature extraction layers. The classification layer, on the other hand, learns at a rate ten times as fast. Initially, the dataset is split into 80 : 20 for the training and validation sets, respectively. This resulted in a model that overfits the training data as observed by the training and validation accuracies and losses shown in Table 2.

Table 2. The accuracy and loss of the model after the first training

| No. | Data | Accuracy | Loss |
|---|---|---|---|
| 1 | Validation | 81.7 | 0.64 |
| 2 | Training | 93.0 | 0.22 |

Using the suggested solution to handle an overfitting problem from [12], we adopted the k-fold cross-validation technique when carrying out the second iteration of the training process. To start with we set the value of k-fold to 10 and re-split the dataset into 90 : 10 for the training and validation sets, respectively. The other training setup variables, such as the number of epochs and learning rates were not changed. The validation accuracy and loss were found to be around 0.85 and 0.53, respectively. Although the accuracy has improved slightly, the loss value remains relatively high – slightly above 0.5. However, the improvement in both accuracy and loss values indicates that we are heading in the right direction.

As a result, we not only increase the k-fold to 20 but also decrease the percentage of the validation set. In this training, the dataset is re-split into 95 : 5 ratio for the training

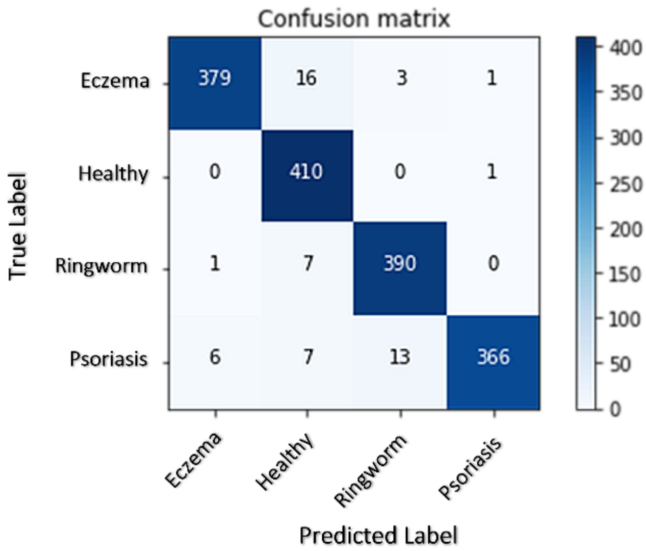



Figure 2. The confusion matrix of the final target model

and validation sets, respectively. Several training options were also changed including the decrease in the number of epochs to 20. At the end of this experiment, the validation accuracy and loss were found to be around 0.90 and 0.39, respectively. The standard deviation of the validation accuracy and loss were found to be 0.11 and 0.46, respectively. The confusion matrix of the model performance is shown in Figure 2.

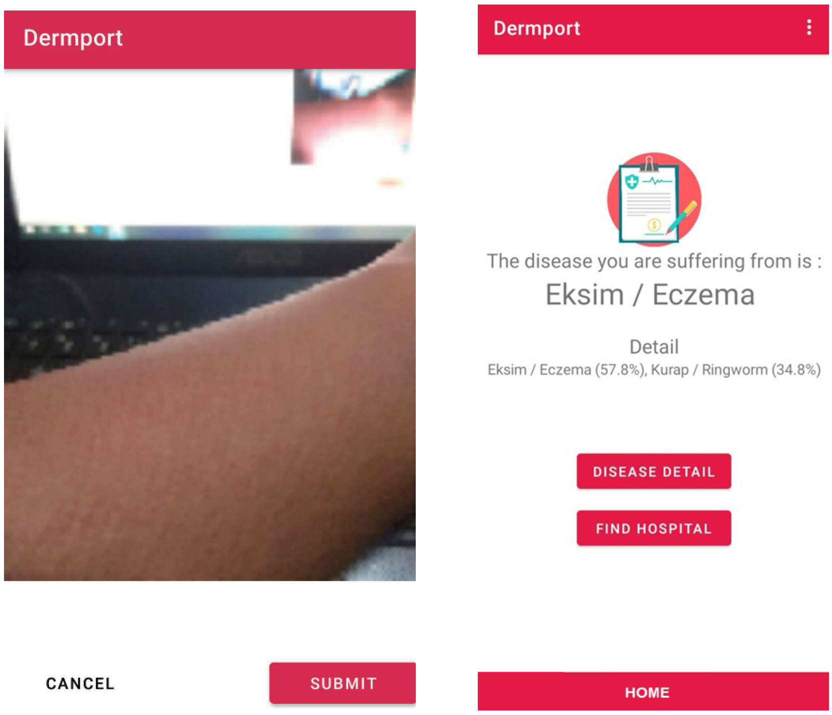
The trained model is then compressed using the TensorFlow Lite API [13]. The compression is carried out with post-training full integer quantization to achieve 4× reduction in network size and over 3× speedup. The compressed model can then be deployed onto a target smartphone. We have developed a mobile application that provides an interface for the user when using the model. This application was developed for the Android operating system using the Android Studio IDE and the Material Component code library. The software development methodology adopted when developing the mobile app is the Rapid Application Development, which is a version of the Agile software development methodology that uses rapid prototyping as a means to obtain and refine users' feedback. A couple of screengrabs of the application are shown in Figure 3.

In addition to the automatic classification of skin disorders, the mobile application also has other features including display of detailed information on many skin disease and disorders, search for the nearest hospital that has a skin disease treatment center, and easy contact with the hospital, based on the information on their official website. The application is designed so that the user can get some additional information and assistance online once he or she receives a positive classification of a disease. The additional information also includes possible self-treatment of the disease such as lotion and ointments, information on how to get them, and direction of use.

The application is then tested on a group of 20 users who would then use it for a few days before being asked to complete a 9-question questionnaire. The participants answer each question by selecting one from possible four scores depending on their level of agreement, which is 25 if they strongly disagree, or 50 if they disagree, or 75 if they agree, or 100 if they strongly agree. The final score $S$ for each question is calculated as:

$$S = \frac{\sum_{i=1}^{4}(s_i \times n_i)}{n} \tag{1}$$

where $s_i$ and $n_i$ are the $i$th score and the number of participants selecting the $i$th answer, respectively and $n$ is the total number of participants. The results were tabulated in Table 3.




(a) (b)

Figure 3. Two screengrabs of the mobile application. The screen (a) shows the graphical interface when the user has taken a picture using the mobile phone's camera and (b) shows the outcome of the classification.

Table 3. The summary of the users' review of the application

| No. | Question | $n_1$ | $n_2$ | $n_3$ | $n_4$ | Score (%) |
|---|---|---|---|---|---|---|
| 1 | Is the language selection good? | 0 | 1 | 5 | 14 | 91.25 |
| 2 | Does the application provide you with useful information on skin diseases and disorders? | 0 | 1 | 12 | 7 | 82.50 |
| 3 | Does the image acquisition feature work well? | 0 | 2 | 9 | 9 | 83.75 |
| 4 | Is the image classification feature accurate? | 2 | 4 | 11 | 3 | 68.75 |
| 5 | Does the location-based hospital search feature work well? | 0 | 1 | 6 | 13 | 90.00 |
| 6 | Does the non-location-based hospital search feature work well? | 0 | 0 | 6 | 14 | 92.50 |
| 7 | Does the additional information shown to you post-classification work well? | 0 | 1 | 10 | 9 | 85.00 |
| 8 | Does the display of the hospital website work well? | 1 | 0 | 10 | 9 | 83.75 |
| 9 | Does the telephone calling system work well? | 1 | 1 | 11 | 7 | 80.00 |

4. **Conclusions.** We proposed a method of recognizing and classifying skin diseases present in the human body using the deep convolutional neural network method. We used a dataset that we developed specifically for this study containing 8,000 images of the ringworm, psoriasis, and eczema diseases plus healthy skin images with the assistance of medical experts. The resulting model is then compressed and deployed as a mobile phone application to increase usage and penetration of the research in the wider community. The mobile version of the model has a slightly lower accuracy than the original model but allows the user to classify the images without an Internet connection. By using

this application, the user should be able to recognize the disease more quickly and be informed of all the relevant information including its treatment method and where to get medical assistance if urgently needed. In the future, we plan to increase the number of skin diseases to classify and deploy the application to the wider public to make a more comprehensive and universal solution to the problem.

## REFERENCES


[1] S.-P. Sinikumpu, J. Jokelainen, A. K. Haarala, M.-H. Keränen, S. Keinänen-Kiukaanniemi and L. Huilaja, The high prevalence of skin diseases in adults aged 70 and older, *J. Am. Geriatr. Soc.*, 2020.

[2] M. Nagano and T. Fukami, Development of a skin texture evaluation system using a convolutional neural network, *International Journal of Innovative Computing, Information and Control*, vol.16, no.5, pp.1821-1827, 2020.

[3] T. Shanthi, R. S. Sabeenian and R. Anand, Automatic diagnosis of skin diseases using convolution neural network, *Microprocess Microsyst*, 2020.

[4] D. A. Gavrilov, A. V. Melerzanov, N. N. Shchelkunov and E. I. Zakirov, Use of neural network-based deep learning techniques for the diagnostics of skin diseases, *Biomed. Eng.*, vol.52, no.5, pp.348-352, 2019.

[5] A. S. Al-Kafri, S. Sudirman, A. J. Hussain, D. Al-Jumeily, P. Fergus, F. Natalia, H. Meidia, N. Afriliana, A. Sophian, M. Al-Jumaily et al., Segmentation of lumbar spine MRI images for stenosis detection using patch-based pixel classification neural network, *2018 IEEE Congress on Evolutionary Computation (CEC)*, pp.1-8, 2018.

[6] A. S. Al-Kafri, S. Sudirman, A. Hussain, D. Al-Jumeily, F. Natalia, H. Meidia, N. Afriliana, W. Al-Rashdan, M. Bashtawi and M. Al-Jumaily, Boundary delineation of MRI images for lumbar spinal stenosis detection through semantic segmentation using deep neural networks, *IEEE Access*, vol.7, pp.43487-43501, 2019.

[7] F. Natalia, H. Meidia, N. Afriliana, J. C. Young, R. E. Yunus, M. Al-Jumaily, A. Al-Kafri and S. Sudirman, Automated measurement of anteroposterior diameter and foraminal widths in MRI images for lumbar spinal stenosis diagnosis, *PLoS One*, vol.15, no.11, 2020.

[8] K. He, X. Zhang, S. Ren and J. Sun, Deep residual learning for image recognition, *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp.770-778, 2016.

[9] K. He, X. Zhang, S. Ren and J. Sun, Identity mappings in deep residual networks, *European Conference on Computer Vision*, pp.630-645, 2016.

[10] P. Stock, A. Joulin, R. Gribonval, B. Graham and H. Jégou, And the bit goes down: Revisiting the quantization of neural networks, *arXiv.org*, arXiv: 190705686, 2019.

[11] J. Deng, W. Dong, R. Socher, L. Li, K. Li and F.-F. Li, ImageNet: A large-scale hierarchical image database, *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp.248-255, 2009.

[12] X. Ying, An overview of overfitting and its solutions, *J. Phys. Conf. Ser.*, DOI: 10.1088/1742-6596/1168/2/022022, 2019.

[13] TensorFlow, *TensorFlow Lite*, https://www.tensorflow.org/, Accessed on Nov. 1, 2020.