Detection of Skin Cancer Image Based on Convolutional Neural Network Model and Website Application

Manye Dong*
College of Letters and Science, University of California, Berkeley, 94709, United States of America
*Corresponding author's e-mail: caradong@berkeley.edu

ABSTRACT

In contemporary society, convolutional neural networks (CNNs) are used mainly for image processing, classification, segmentation, including detecting skin cancer. However, many current CNN models used to analyze skin cancer heavily rely on enough information apart from photos of skin lesions. This need hinders the users from diagnosing themselves. To tackle this issue, the paper proposed a new but simple CNN model. It can be utilized to detect and differentiate benign and malignant skin cancer on the patients' end using image classification. Deep convolutional neural networks, such as CNN, demonstrate the potential to be properly trained and making predictions based on the given training set. In this paper, we show classification of skin cancer using a single CNN model, trained end-to-end from real skin lesion images directly. It only uses pixels and disease labels as inputs. We used the Malignant vs Benign skin cancer dataset generated by Claudio Fancon. Images are provided by the public ISIC (International Skin Image Collaboration) 2018 Skin Lesion Dataset. Furthermore, we resize the original images to a desirable resolution that is ready to be used for training. Then, we build a multiple-layer CNN model and train it on all the data. Finally, we fit the model to a local user interface for future use. The proposed model is found to be successful and promising, achieving testing results with 82% accuracy on the test set. This could significantly reduce human mistakes in the skin cancer diagnosis process. It also obtains an average loss of 0.45, which is a relatively low figure on the test set. The proposed system, therefore, is relatively reliable and robust when detecting potential skin cancer.

Keywords: Cancer detection; Image classification; CNN model; Experimental analysis

1. INTRODUCTION

As society develops, humans are exposed to more various daily activities and thus are more susceptible to skin cancer. Skin cancer can be provoked by some factors, such as smoking, alcohol usage, allergies, infections, viruses, physical activity, environmental change, and ultraviolet light. It can cause death as it deteriorates. The world health organization (WHO) claims that one in every three cancers diagnosed among patients is skin cancer[1]. In the U.S., more than 9,500 people are diagnosed with skin cancer every day, and more than two people die of skin cancer in the U.S. every hour. Nevertheless, the 5-year survival rate for melanoma is almost 99 percent[2]. Therefore, an early diagnosis may lead to treatments that increase the chances of life. Nowadays, image-based diagnostic artificial intelligence (AI) can be used to detect skin cancer. Pathology Deep Learning System (PDLS) is being researched in early 2021, aiming at telling the pathologist the classification of a case. In doing so, it aids the pathologist to prioritize review for a certain type of case. The system thus supports different levels of clinical expertise and many clinical workflows[3]. Yet, faulty AI and PDLS can mislead the whole diagnosis process, and this system is only available to experts, not common patients. Some other methods that help patients diagnose themselves are mobile applications like Skin Vision. Yet, these applications require extra downloads and reduce the accessibility of the handy tool. This study offers a framework based on CNN that operates on users' end in a web browser.

The main aspects of this work can be summarized as follows:

1. We collect and resize the skin lesion images to 80*80, maximizing the result accuracy and saving training time.

2. Two same-size arrays of zeros and ones are created as training examples for predicted accuracy examination.

3. Images are shuffled to reduce the potential bias of the model.

4. We build a CNN model with two convolution layers, two max pooling layers, two dropout layers, and a last flatten layer and dense layer.

5. The model is fit to all data to test the accuracy.

6. The pre-trained CNN model is connected to a user interface that can tell the users their possibility of getting skin cancer based on the uploaded image.

The rest of this paper is represented as follows: Section 2 represents model formulation and UI connection. Section 3 represents dataset processing methods and training and testing performance of our proposed CNN model. Conclusion and future works are summarized in Section 4.

2. METHODOLOGY

In this section, we narrate the following steps and the flowchart of the methodology shown in Figure 1. CNN model is used for the classification task between benign and malignant skin lesions. In addition, we fit the pre-trained model to a user interface for users' convenient use.

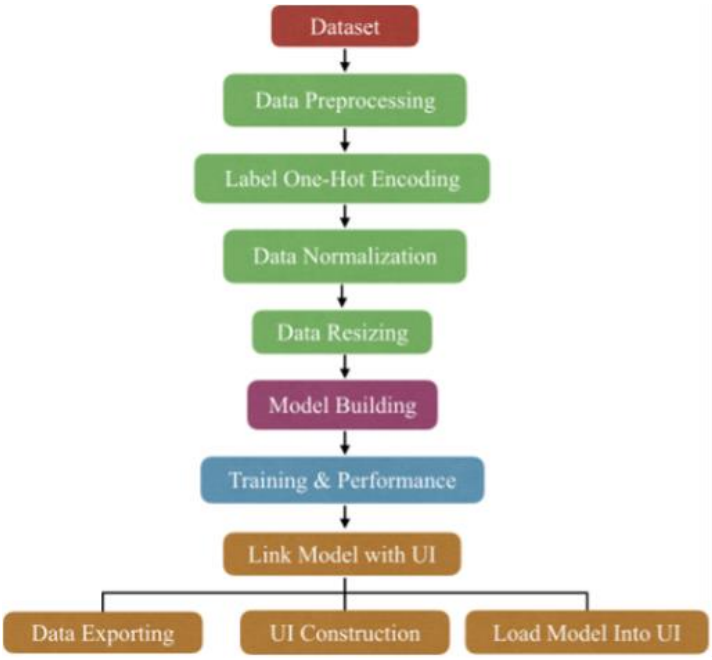


Figure 1. The overall framework of this work.

2.1 Model building

The dataset is split so that 54% is in the training data and the remaining 46% is in the test data. The model's internal layer connection is shown in Figure 2 and the model summary is shown in table 1. We add one layer of CNN at a time to the model. The layers are as follows:

- Conv2D layer: We chose to set 64 filters for the first two conv2D layers with a kernel size of 2x2. Then we switch to a kernel size of 3x3 for the last conv2D layer. Inside all the 64 filters, each filter transforms a part of the given image that is previously "cropped" by the kernel size. The kernel filter matrix is thus applied on the whole image as a transformation. We also added a "relu" activation function as one of the parameters for each convolution layer. They add non-linearity to the model to make it deep. If we do not add such nonlinear functions, the deep CNN model will be simulated by a shallow CNN model. It can bring down the prediction accuracy.

- MaxPool2D layer: This layer serves as a downsampling filter in our CNN model. It takes the maximum of the two nearby pixels, which reduce computational cost. Besides, it helps with reducing overfitting. We choose the pooling size of 2x2. We consider it to be the most suitable dimension of downsampling for our input dataset.

- Dropout layer: Dropout is a regularization method. A part of nodes in the layer are randomly ignored. It is also used to solve the overfitting problem by randomly forgetting some weights for every training sample. We chose a dropout rate of 0.25, which means we scale down the output by 0.25. It saves us from regularizing weights.

- Flatten layer: It is used to convert the final feature maps into a one single one-dimensional vector. It is necessary to implement a flatten layer to use the final fully connected layer. We use this layer to convert the matrix we have to the binary prediction answer (0 for benign and 1 for malignant).
- Fully connected layer (Dense): We used features in one dense layer, just an artificial and neural networks (CNN) classifier.

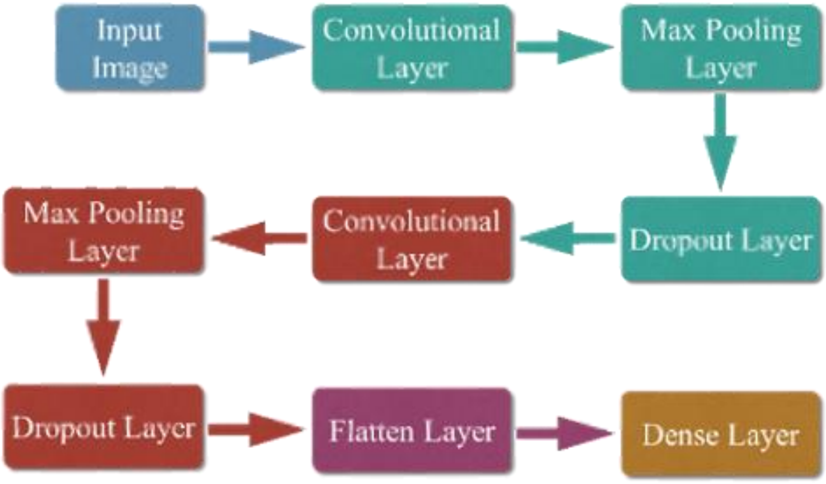


Figure 2. Layers of CNN model.

Table 1. Sequential model summary.

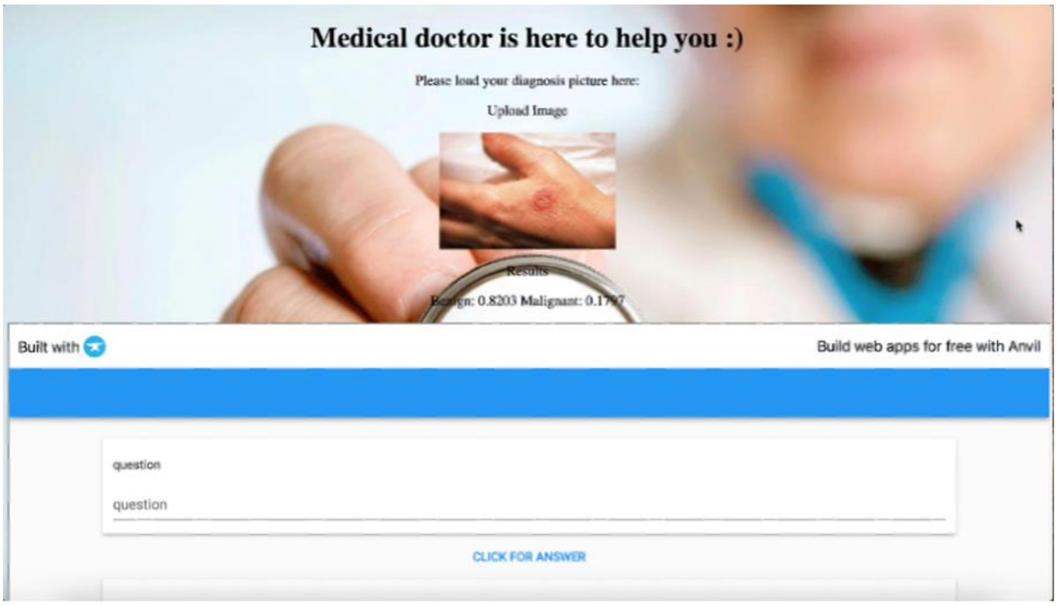
Layer (Type)	Output Shape	Parameters Number
Conv2D	(None, 224, 224, 64)	1792
MaxPooling2D	(None, 112, 112, 64)	0
Dropout	(None, 112, 112, 64)	0
Conv2D	(None, 112, 112, 64)	36928
MaxPooling2D	(None, 56, 56, 64)	0
Dropout	(None, 56, 56, 64)	0
Flatten	(None, 200704)	0
Dense	(None, 128)	25690240
Dense	(None, 2)	258

2.2 Link model with UI

2.2.1 Data exporting. After implementing and testing the CNN model, we saved the model as TensorFlow.js format. It is more compatible with the user interface we will implement than Google Colab notebook. TensorFlow.js is a JavaScript Library for training and deploying machine learning models in the browser. In the last decade, improvements were made in web browsers' processing capacity. Many powerful software libraries like TensorFlow.js were released[4]. It fits to the website source code more easily than all other file formats.

2.2.2 Load model into UI. Out of all coding applications, we chose to use SubLime Text because of its compatibility with the computer hardware systems. We first created an html file that contains all the implementation for the homepage of the website. It includes headings, greetings, and instructions for users to upload their skin image and a disclaimer of our diagnosis results. However, there were some problems while we tried to load the skin cancer detection model into the homepage. We found that the saved TensorFlow.js file cannot be directly used in the html code file. Thus, we added another file in the same directory called model.js to host our pre-trained CNN model. Then it is possible to link the file to the basic homepage file with a simple load in function. Afterwards, we write a prediction file to connect to the homepage. The website can make predictions on the user input image and output results to the homepage.

2.2.3 UI construction. Mobile devices can potentially extend the reach of dermatologists outside of the clinic and further to more patients[5]. One technique that collaborates well with mobiles is online websites. It is important to make the CNN model readily available to patients unfamiliar with complicated programming processes and machine learning algorithms. We decided to create a user interface to incorporate this skin cancer detection model, as shown in Figure 3. The website includes a homepage where users can upload the image of their skin. The image will be shown on the website to ensure that users have input the correct information. After a few seconds, the user will see a line of words under the image. It shows the probability of getting benign skin cancer or malignant skin cancer based on their input image in decimal.



Medical doctor is here to help you :)

Please load your diagnosis picture here:

Results

Benign: 0.8203 Malignant: 0.1797

Figure 3. Website overview.

3. EXPERIMENTS

3.1 Datasets

3.1.1 Datasets. To apply machine learning approaches for skin cancer detection, we find it necessary to use a relatively large amount of data. However, there is a lack of trustworthy training data because of the patient record privacy policy. To overcome the problem, we utilized a freely and publicly accessible dataset of dermoscopy pictures named ISIC 2018. It is taken from the International Skin Image Collaboration Archive (ISIC), a global partnership that has organized the world's largest public repository of dermoscopic images of skin[6]. The dataset includes over 12,500 images across 3 different tasks offered for contest participants in 2018. We are using the dataset for task 3. This is an online challenge encouraging teams to classify skin lesion images into either benign or malignant categories. This dataset consists of 1800 dermoscopic level pictures of pre-divided benign classified moles. It also contains 1497 pictures of malignant classified moles.

3.1.2 Data preprocessing. It is crucial that we preprocess the data to enable the user to treat and make use of complex data[7]. In the data preprocessing step, we transform the large-scale raw image data into a format more suitable to create an effective CNN model. The dataset contains images that are visually blurrier than others. These images may undermine the performance of our model. They do not satisfy the standard of appropriate training resolution. Thus, we delete them in the training set but still use them in the test set. It is to consider a larger range of images that could be input to our model by potential users. Moreover, we keep the images that do not show explicit symptoms of skin lesion. It is possible that they represent an early stage of skin cancer that needs to be recognized and diagnosed by our model.

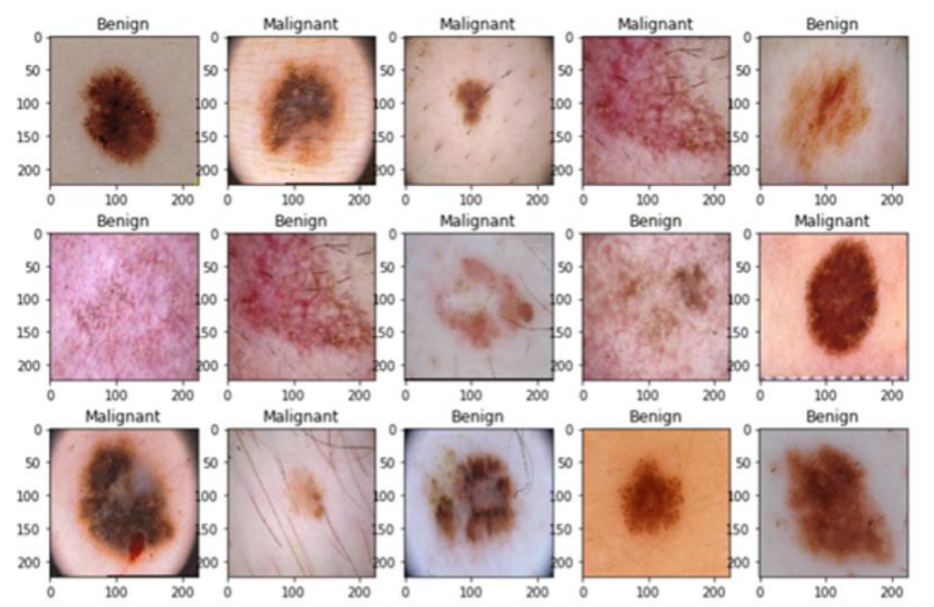


Figure 4. Data after pre-processed.

3.1.3 Label one-hot encoding. Encoding variables and features is important in preparing a large dataset to be readily trainable for a convolutional neural network model. Categorical encoding has always been the most commonly used technique to accomplish this task. We notice the skin cancer dataset contains categorical variables “benign” and “malignant”. Thus, we chose to apply categorical encoding to all the images. We select one-hot encoding instead of label encoding among categorical encodings. It is because our categorical features are not ordinal. The number of categorical features is also less so one-hot encoding can be effectively applied. Due to its effectiveness and simplicity, one-hot encoding is still the most prevalent procedure for addressing such multi-class classification tasks[8]. It creates additional features based on the number of unique values in the categorical features. Each unique value in the category will be added as a feature. We one-hot encoded both the labels--benign and malignant. Benign skin lesion images is represented as 0 and malignant skin lesion images is represented as 1. The mathematical model is as follows:

$$\begin{aligned} Ax &= 1,\ if\ \ x \in A \\ Ax &= 0,\ if\ \ x \notin A \end{aligned} \tag{1}$$

3.1.4 Data normalization. The success of machine learning algorithms like CNN relies on the suitability of the input dataset[9]. With the help of a certain amount of qualified data, the model can make predictions to multiple classified categories. Therefore, it is essential to apply data normalization to datasets to improve algorithms' performance. There are some current techniques of data normalization, such as decimal scaling normalization, z-score normalization, and min–max normalization. We chose the most popular decimal scaling normalization. We normalize the dataset by dividing 255. It is known as the gray scale value of an image. The number 255 comes from the RGB scale. Each channel (red, green, and blue) of an image is 8 bits, which suggests the upper limit value is 256. It is an industry standard to consider 0.0 as black and 1.0 as white. To convert the numbers of our images from 0 to 255 (inclusive) to the range from 0.0 to 1.0, we need to divide them by 255. Therefore, values are shifted to model readable floating points.

3.1.5 Data resizing. As noted in many other studies, downsizing the resolution of image data affects the model accuracy. Nevertheless, is it not paradoxical to achieve better model performance with lower input image resolutions[10]. In some cases, the model needs reduced features. This is to minimize the number of parameters need to be optimized and constrain the computing runtime with a proper range. Thus, applying such a procedure to a large dataset is necessary to ensure its functionality in the hardware set up. We resize all the images to lower resolution 80x80x3 RGB. It is to normally train the data yet maintain the maximized possible training accuracy. All experiments are conducted on a computer with 2.7 GHz Intel Core i5 processor, 8 GB 1867 MHz DDR3 of Memory, and Intel Iris Graphics 6100 1536 MB GPU card. The program codes of data preprocessing and graphs modeling are written by Python 3.6.5, which is available in a notebook of Google Colab (12 GB RAM).

3.2 Evaluation metrics

We used Keras' built-in function ReduceLROnPlateau. We set a learning rate annealer and train the model with a total of 50 epochs on the training set. Then we apply the CNN model to the test set to check on prediction performance. The model's performance will be tested on the accuracy score and its cross-entropy loss. The accuracy score will be calculated based on the mathematical formula below:

$$Accuracy = \frac{TP + TN}{ALL} \tag{2}$$

Where TP is the true positive rate, TN is the true negative rate, and All is all the observations.

The cross-entropy loss will be calculated based on the mathematical formula below:

$$Cross\_Entropy\_Loss = -[y * \log(p) + (1 - y) * \log(1 - p)] \tag{3}$$

In this calculation, log represents the natural logarithm, y is the indicator variable (0 or 1) if the class label predicted is the correct classification for observation, and p indicates the probability that predicted class label is of the actual observation label. In the context of this study, we have two class labels in total to be classified: benign and malignant.

3.3 Experimental results and analysis

The accuracy trend over 50 epochs is shown in Figure 5 and the loss trend is shown in Figure 6. At the end of the 50th epoch, the model obtains around 0.82 (82%) overall prediction accuracy on the test set. On the training set , the accuracy is around 0.85 (85%). The loss diminishes to an average of 0.45 on the test set and an average of 0.38 on the training set.

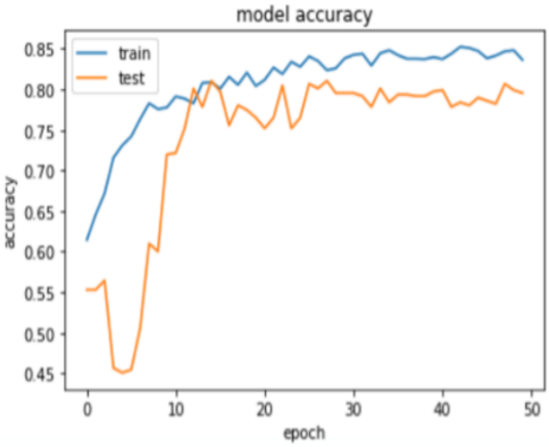


Figure 5. Model accuracy.

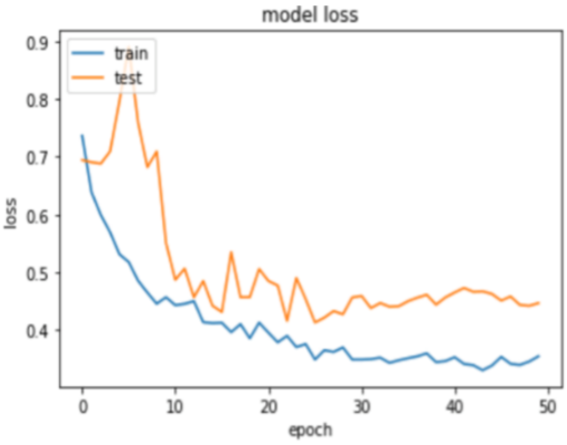


Figure 6. Model loss.

From the model accuracy plot, we can observe there is a jump from 0.55 to 0.45 around epoch 4 or 5 for the test set. The model loss increases dramatically from 0.7 to 0.9 around roughly the same epoch. It can be attributed to an uncommon test sample that happened to be used during that epoch. Across the several last epochs, both the train and test set accuracy and loss begin to converge to a settled number. This trend demonstrates that the more epochs we train, the higher accuracy and lower model loss we can achieve. There is a flattening line towards the end. It means that 50 epoch is approximately the epoch that maximizes the model accuracy and minimizes the model loss.

Another method downsizes input images to 32x32 RGB in the data resizing process. All the rest of the experiments remain the same. 32x32 resolution undermines the testing accuracy to around 0.632 (63.2%). Training accuracy drops to around 0.68 (68%), as shown in Figure 8. The model loss rises to an average of 0.55 on the test set and an average of 0.45 on the train set, as shown in Figure 9. Low resolution as abovementioned omits important features of the images. Therefore, the CNN model can only run with less defining factors from the images to make classification. This results in less accurate categorization. 80x80x3 RGB is by far the optimal resolution. It reaches a relatively high accuracy and remains functional on an acceptable computer hardware.

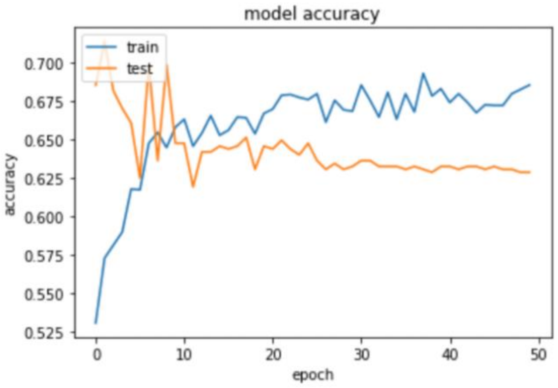


Figure 7. Model accuracy for 32x32 RGB.

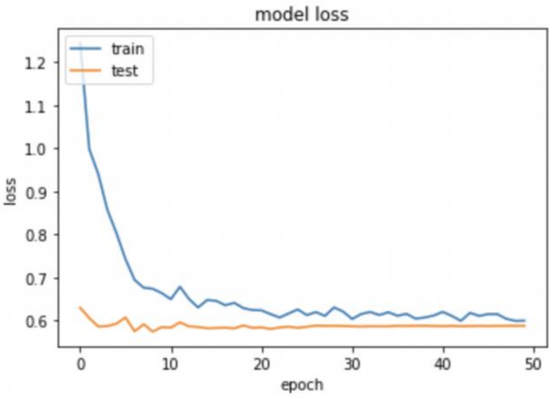


Figure 8. Model loss for 32x32 RGB.

4. CONCLUSION

The research builds a user interface exploiting a CNN model. It detects the possibility of having benign and malignant skin cancer based on users' input images. We resize the images from ISIC 2018 Skin Lesion Dataset to a readily usable resolution. Then, we shuffle it randomly. We move on to build a CNN model and train it on the whole training dataset for numerous epochs. In the next step, the model accuracy is tested on the test set. Finally, the pre-trained model is fit to a local website. Users can easily diagnose the concerned skin condition on their mobiles or laptops. The experimental results demonstrated that the testing accuracy is over 82%, which shows a promising result of the model.

In the future, a series of meaningful work can be conducted subsequently. For example, the research can incorporate more machine learning networks like GAN to improve the accuracy of the existing model. A more precise image-to-text model is also a good help. It can be installed with a discriminator in addition to the generator. It can even display users' skin cancer evaluation based on descriptions of their skin without uploading an image. This possible improvement may boost the reliability of the provided website. Thus, it will take the model to a level that parallels the diagnosis of dermatologists. Furthermore, the user interface can bring in additional NLP (Natural Language Processing) techniques to implement a chatbot. Users will be able to type in questions regarding skin cancer. The interface will automatically output the answers to the given questions. This future work can promote user experience with the website and maintain the professional characteristic of its results.

REFERENCES

[1] Ali S., Miah S., Haque J., Rahman M., Islam K. (2021) An enhanced technique of skin cancer classification using deep convolutional neural network with transfer learning models. Mach. Learn. Appl, vol. 5.

[2] Apalla Z., Lallas A., Sotiriou E., Lazaridou E., Ioannides D. (2017) Epidemiological trends in skin cancer. Dermatol Pract. Concept., vol. 7, no.2.

[3] Tschandl P., Rinner C. (2020) Human computer collaboration for skin cancer recognition. Nat. Med., vol. 26: pp. 1229–1234.

[4] Esteva A., Kuprel B., Novoa R. (2017) Dermatologist-level classification of skin cancer with deep neural networks. Nature, vol. 542: pp. 115–118.

[5] Jianu S., Ichim L., Popescu D., 2019. Automatic Diagnosis of Skin Cancer Using Neural Networks. In: 11th International Symposium on Advanced Topics in Electrical Engineering (ATEE). Bucharest. pp. 1-4.

[6] Codella N., Rotemberg V., Tschandl P., 2019. Skin lesion analysis toward melanoma detection 2018: a challenge hosted by the international skin imaging collaboration (ISIC). https://arxiv.org/abs/1902.03368.

[7] García S., Ramírez-Gallego S., Luengo J., Benítez J., Herrera F. (2016) Big data preprocessing: methods and prospects. Big Data Anal., vol. 1, no. 9.

[8] Rodríguez P., Bautista M., Gonzàlez J., Escalera S. (2018) Beyond one-hot encoding: lower dimensional target embedding. Image Vis. Comput., vol. 75: pp. 21-31.

[9] Singh D., Singh B. (2020) Investigating the impact of data normalization on classification performance. Appl. Soft Comput., vol. 97.

[10] Sabottke C., Spieler B. (2020) The effect of image resolution on deep learning in radiography. Radiol. Artif. Intell., vol. 2, no. 1.