A Skin Cancer Detection Interactive Application Based on CNN and NLP

Xuping Gong¹,‡ Yuting Xiao²,†, *

¹ Department of Marine Science, Ocean University of China, Qingdao, Shandong Province, 266000, China
² Department of Data Science (Statistics), The Chinese University of Hong Kong, Shenzhen, Guangdong Province, 518000, China
*Corresponding author’s e-mail: 118010346@link.cuhk.edu.cn
†These authors contributed equally.

Abstract. Skin cancer is the most common cancer with several different types. According to current estimations, one in five Americans will develop skin cancer in their lifetime. Therefore, early diagnosis and treatment of it is of crucial significance. Several advanced image processing methods have been applied to predict skin cancer. However, few researchers utilize those methods to build an interactive application. In this work, we implemented an interactive skin cancer diagnosis website, combining the convolutional neural network (CNN) and natural language processing (NLP) technology. The neural network model uses four convolutional layers and dense layers respectively to improve the accuracy. Two max-pooling layers were used to reduce redundant information. To address the severe overfitting problem, we chose to utilize the batch normalization along with dropout layers. Based on our results, 0.9935 in accuracy and 0.0225 loss is realized for training data, and accuracy of 0.8393 and 0.6648 loss for testing data. Natural language processing (NLP) was used to implement a chatbot for interaction with users. We crawled skin cancer related questions and answers from Quora and used them to train our chatbot. Lastly, we combined CNN and NLP to build an interactive skin cancer diagnosis website. VUE.js and Django were used to build the front-end and back-end of our website. These results offer a guideline for combining artificial intelligence with not only medicine but also interactive network, which enables people to get medical care more easily.

1. Introduction
Skin cancer is a skin malignant tumor [1] with high incidence, which is the most common of all cancers. According to the different sources of tumor cells, it can be divided into several types. The ultraviolet radiation from sunlight is the foremost cause of the skin cancer. Contemporarily, this disease is becoming increasingly common in the young generation because of tanning and other cancer-causing elements. As a result, it is vital to diagnose and treat the skin cancer in the early stage, especially for some high fatality rate types. Since different kinds of skin cancers distinguish themselves from their appearances, it is possible to predict them using some advanced image processing methods, e.g., convolutional neural network (CNN), due to their excellent performance in different tasks (e.g., medical image classification [2] and segmentation [3] etc.).

Many scholars have carried out some work about skin cancer detection or establishing an application for the popularization of the skin cancer. In Ref. [4], the authors focus on the most dangerous type of
skin cancer namely Malignant melanoma, and pay more attention to avoid the misdiagnosis between the Malignant melanoma and the ordinary skin diseases. The unique symptoms of skin cancer are used, e.g., Asymmetry, Border irregularity, Color variation and Diameter, to increase the accuracy of diagnosis. However, this algorithm can only detect one type of skin cancer. In Ref. [5], the authors present an optimized convolutional neural network, which utilizes the improved whale optimization algorithm to enhance a CNN with higher efficiency for the diagnosis usages. Even though many researchers have used the CNN to build the detection of the skin cancer, most of the articles haven’t combined the convolutional neural network and the natural language processing (NLP). Generally, it can build an interactive website to help the users learn some relevant information about the skin cancer and predict the picture uploaded by them.

To overcome the limitations mentioned above and improve the existing application for skin cancer detection, we built a new application based on CNN and NLP technology. To login the application, users only need to upload photos of their skin where there is likely to be a skin cancer mottle. Then, the application will then classify the mottle and the chatbot will interact with the user and give advice.

2. Method

2.1. Data Description and preprocessing
Our dataset is collected from Kaggle [6], which is an open-source data that anyone can access. This dataset consists of 3 GB of skin cancer images used for training. The classifier aims to make predictions about the kind of a cancer mottle photo provided by the user. The dataset used to train the model is a 3GB collection of photos that contains 7 different classes of skin cancer. Some of statistical characteristics of our dataset are shown below:

![Image](image1.png)

Figure 1. Disease distribution in terms of location.

![Image](image2.png)

Figure 2. Disease distribution in terms of class.

![Image](image3.png)

Figure 3. Disease distribution in terms of sex.

For pre-processing, the dataset is first split into training and testing datasets. Next, the distribution of the training and testing dataset is to be balanced by the SMOTE method [7]. Finally, the images are
28 by 28 pixels and then we added a feature dimension to fit the model's first layer based on Tensorflow2.X package.

In the dataset, the numbers of different labels are not equal, which means that the data is not balanced. We balanced our dataset to make the learning ability of our CNN in every label be good. In the data processing process, observing and reshaping the data is crucial. The input of CNN should be 4D in Tensorflow2.x, i.e., we converted our dataset from 3D to 4D to satisfy the input requirement of our model. Furthermore, we also carried out the data normalization in the data pre-processing stage to distribute our dataset evenly.

2.2. Algorithm (CNN Model)
The CNN model is chosen to implement our skin cancer diagnosis due to its excellent feature extraction ability compared to other traditional image analysis methods. CNN stands out since its parameters sharing and sparsity of connections. In the CNN model, different areas share the same filter which means they share the same parameters. In this way, less parameters are needed to train a model which can kind of help to solve the overfitting problem. In addition, due to the parameter sharing of the filter, even though the input images undergo translation, the CNN model can still recognize the features. Such property is called “Translation Invariance”, which can make the model more stable. In terms of the sparsity of connection, it means that one unit in the output layer only relates to part of units in the input layer. In contrast, in traditional NN models, one unit in the output layer will be affected by any unit in the input layer due to fully connection, which will degrade the effect of image recognition. Instead, since different areas in the input image have distinct features, CNN model can prevent one area in the input image from being influenced by the other areas.

We also used the pooling layer between two consecutive convolutional layers. The pooling layer will then simply perform down-sampling along the spatial dimensionality of the given input, further reducing the number of parameters within that activation [8]. Pooling layer can resize the image by compressing the number of data and parameters. Therefore, redundant information could be eliminated and only useful features are left. It can also help with the overfitting problem to some extent. Besides, batch normalization and dropout technologies which let the network randomly die some nodes based on a parameter [9] are also used to alleviate the overfitting problem in our study. The architecture of our model is shown below:

Table 1. The architecture of our model.

| Model: “Sequential” |
|---------------------|
| Layer (type) | Output Shape | Param # |
| conv2 | (None, 28, 28, 32) | 320 |
| max_pooling2d | (None, 14, 14, 32) | 0 |
| batch_normalization | (None, 14, 14, 32) | 128 |
| conv2 1 | (None, 14, 14, 64) | 18496 |
| max_pooling2d 1 | (None, 7, 7, 64) | 0 |
| batch_normalization 1 | (None, 7, 7, 64) | 256 |
| conv2 2 | (None, 7, 7, 128) | 73856 |
| batch_normalization 2 | (None, 7, 7, 128) | 512 |
| conv2 3 | (None, 7, 7, 256) | 295186 |
| batch_normalization 3 | (None, 7, 7, 256) | 1024 |
| flatten | (None, 12544) | 0 |
| dropout | (None, 12544) | 0 |
| dense | (None, 128) | 1605760 |
| dropout 1 | (None, 128) | 0 |
| dense 1 | (None, 64) | 8256 |
| dropout 2 | (None, 64) | 0 |
| dense 2 | (None, 32) | 2080 |
| dropout 3 | (None, 32) | 0 |
| dense 3 | (None, 7) | 231 |
| Total params: 2,006,087 |
| Trainable params: 2,005,127 |
2.3. Implementation detail for model training

In the early days of our study, the initial structure performed well on the training dataset but failed on the test dataset. Subsequently, we did the conduction of a few experiments and found the model structure that has the best performance on the dataset. We tried to remove the batch normalization (BN) layer which is used not fairly appropriate in the model, but it aggravated the problem of the overfitting. Furthermore, we also tried to change the dense layers in the end of the model, but the result is not accurate. The final structure is three convolutional layers with kernel size 3 by 3 and Re-Lu as activation function, each followed by a 2 by 2 pooling layer and a BN layer. A flattening layer is then added to the sequential model, and afterward are three dense layers, each followed by a dropout layer. At last, a dense layer was added to the model to reduce the number of neurons to 7, which matches the number of output classes.

In the training phase of the model, the model is trained for 50 epochs on our training dataset. In addition, the optimizer, learning rate, loss function and batch size are Adam, 0.0001, sparse categorical cross entropy and 64, respectively. During the making of the CNN classifier, we encountered the problem of overfitting. The model had a good performance on the training dataset but failed on the testing dataset. The solution turned out to be balancing the dataset before training and adding dropout layers between dense layers.

2.4. The application for model deployment

In addition, we built a chatbot in our application. A chatbot is an artificially intelligent creature which can converse with humans. It is a conversational agent that interacts with users in a certain domain or on a particular topic with input in natural language sentences. Chatbots can be referred as software agents that pretend as a human entity. These are the agents with AI embedded and using NLP they can answer to user questions [10]. Thus, we also utilize NLP to implement our chatbot. The data on which the bot is trained in the 270 questions about skin cancer asked on Quora [11] (a forum for knowledge-based Q&A) and their corresponding answers. For collecting the data, a Python web crawler is made to extract information from HTML pages. We first make use of the python built-in model GoogleSearch to search for skin cancer related questions asked on Quora. Then the Python built-in model urllib.request is used to send a request to the website and obtain the result of the request. The result is decoded into HTML in utf-8 form and one can access the model from Ref. [12]. Beautifulsoap module which can be accessed from Ref. [13] is then used to extract questions and answers from obtained HTML. For each question, the five top answers to the question are crawled down to be the potential responses to the corresponding questions. The last step is to use the Python built-in module Chatterbot to train the chatbot. With using ListTrainer, it allows the chatbot to be trained using a list of strings where the list represents a conversation. Eventually, the robot answers the user's question based on the training set's reply logic. The Chatterbot module can be accessed from [14].

Finally, to build our skin cancer detection interactive website, we chose the structure of the separation of the front and rear ends which can improve the readability of the code and the development efficiency [15]. Vue.js and Django are used to implement the front-end and back-end respectively. In addition, we used an api called Restful to connect the front-end and back-end.
3. Results and Discussion

3.1. The performance of CNN

![Figure 4. The curve for training & validation loss and accuracy.](image)

The final accuracy for training data is up to 0.9935 and that for validation data is 0.8393. The final loss for training data is 0.0225 and that for validation data is 0.6648. In our training process, the testing dataset was regarded directly as the validation dataset, therefore, the accuracy and loss of the validation dataset is the final testing result.

It can be observed that even though max pooling, batch normalization and dropout have been used to alleviate the overfitting problem, this phenomenon still exists. We think that the main reason of this phenomenon is the excessive number of parameters. However, if we directly reduce the number of parameters, it may lead to the decrease in overall accuracy. Therefore, further improvement for our model is still needed to better solve the overfitting problem and find a better balance between the overfitting and underfitting.

3.2. The result for Web application

The homepage of our website is shown in Fig. 5, which contains five sub-pages. The Skin Cancer Information page tells users what is skin cancer. The Risk Factors page lists six factors that are related to skin cancer. They are tanning, sunburn, ultraviolet radiation, atypical moles, photosensitivity and the skin type. The Prevention page provides some skin cancer prevention methods such as wearing sun-protective cloth and ingesting vitamin-D. The Diagnosis page corresponds to the CNN model works is used to detect skin cancer. As shown in Fig. 6, users can upload their skin cancer related image, then our CNN model will detect the classification of this skin cancer and feedback the result to the webpage. At last, the Treatment page describes the most effective technique of treating skin cancer, Mohs Surgery. On the lower right corner of each webpage locates the chatbot which is corresponding to our NLP application. Fig. 7 displays the detailed chatbot interface.
Figure 5. Homepage of the skin cancer diagnosis website.

Figure 6. Skin Cancer Diagnosis Page.

Figure 7. Chatbot Interface.
4. Conclusion
In summary, the CNN is utilized to detect and classify skin cancer with an interactive application based on NLP, i.e., users can upload pictures on the website to get detection results, treatment recommendations and some related information. Specifically, the constructed CNN model has high-accuracy in image identification, which addresses the problem of skin cancer detection; whereas the NLP creates an environment that users can easily interact with our application. In the future, we will intend to use different powerful neural networks to broaden the categories of diseases detecting with high accuracy. The results of this research provide a guideline for online healthcare using neural network and natural language processing.

References
[1] The Skin Cancer Foundation. (2021) Skin Cancer Information. https://www.skincancer.org/.
[2] M. Amrane, S. Oukid, I. Gagaoua and T. Ensarlı. (2018) Breast cancer classification using machine learning. 2018 Electric Electronics, Computer Science, Biomedical Engineering’s Meeting (EBBT), pp. 1-4, doi: 10.1109/EBBT.2018.8391453.
[3] Y. Qiu, C. S. Chang, J. L. Yan, L. Ko and T. S. Chang. (2019) Semantic Segmentation of Intracranial Hemorrhages in Head CT Scans, 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS), pp. 112-115, doi: 10.1109/ICSESS47205.2019.9040733.
[4] Jaleel, J. Abdul, Sibi Salim, and R. B. Aswin. (2012) Artificial neural network based detection of skin cancer. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering 1.3.
[5] Zhang, Ni, et al. (2020) Skin cancer diagnosis based on optimized convolutional neural network. Artificial intelligence in medicine 102: 101756.
[6] Kaggle. Skin Cancer MNIST: HAM10000. https://www.kaggle.com/kmader/skin-cancer-mnist-ham10000.
[7] Zhang, Hongpo, et al. (2020) An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset. Computer Networks 177: 107315.
[8] O'Shea, Keiron, and Ryan Nash. (2015) An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458.
[9] Chauhan, Rahul, Kamal Kumar Ghanshala, and R. C. Joshi. (2018) Convolutional neural network (CNN) for image detection and recognition. 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC).
[10] Lalwani, Tarun, et al. (2018) Implementation of a Chatbot System using AI and NLP. International Journal of Innovative Research in Computer Science & Technology (IJIRCST) Volume-6, Issue-3.
[11] Quora. https://www.quora.com.
[12] Python Software Foundation. Algorithm. (2001) https://docs.python.org/3/library/urllib.request.html#module-urllib.request.
[13] Currmy. Beautiful Soup. (2020) https://www.crummy.com/software/BeautifulSoup/.
[14] Github. Installation. https://chatterbot.readthedocs.io/en/stable/setup.html.
[15] Quan, Yun. (2019) Design and Implementation of E-commerce Platform based on Vue.js and MySQL. Proceedings of the 3rd International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2019).