Design of a Skin Cancer Diagnosing Web Application Based on Convolutional Neural Network Model and Chatterbot Application Programming Interface

Qi An[1]

[1]Institute for Mathematics, Technical University of Berlin, Berlin, Germany

*Corresponding author's e-mail: an@campus.tu-berlin.de

Abstract. Skin cancer has become a great concern for people's wellness. With the popularization of machine learning, a considerable amount of data about skin cancer has been created. However, applications on the market featuring skin cancer diagnosis have barely utilized the data. In this paper, we have designed a web application to diagnose skin cancer with the CNN model and Chatterbot API. First, the application allows the user to upload an image of the user's skin. Next, a CNN model is trained with a huge amount of pre-taken images to make predictions about whether the skin is affected by skin cancer, and if the answer is yes, which kind of skin cancer the uploaded image can be classified. Last, a chatbot using the Chatterbot API is trained with hundreds of answers and questions asked and answered on the internet to interact with and give feedback to the user based on the information provided by the CNN model. The application has achieved significant performance in making classifications and has acquired the ability to interact with users. The CNN model has reached an accuracy of 0.95 in making classifications, and the chatbot can answer more than 100 questions about skin cancer. We have also done a great job on connecting the backend based on the CNN model as well as the Chatterbot API and the frontend based on the VUE Javascript framework.

1. Introduction

Malignant melanoma is the deadliest form of skin cancer. Yet, melanoma can be easily treatable if detected in the early stages. Hence, as emphasizes Hameed the importance of skin cancer classification [1] in his survey on image-based diagnosis systems for skin cancer. Image-based methods have achieved great performance in diagnosing skin cancers in recent years. For example, Nunnari and Sonntag have designed a software toolbox for tuning the CNN model in the domain of skin cancer classification, which displays the capability of different neural networks on the skin cancer dataset [2]. Furthermore, Ji et al. developed a novel 3D CNN model for action recognition, which can extract features from spatial and temporal dimensions by performing 3D convolutions and achieving promising performance on object recognition tasks [3]. It can be found that the CNN model is advanced at addressing certain classification tasks such as image classification and emotion analysis. We also note that Nagender and Patil have mentioned the python library Chatterbot as a tool to construct their WhatsApp chatbot [4], which has achieved good performance. Due to the excellent achievement of the foundations mentioned above, we would like to implement the skin cancer classification with the CNN model and the chatbot with the Chatterbot API [5-7]. We have compared 3 other state-of-the-art applications featuring skin cancer diagnosing and would like to discuss their pros and cons.

FirstCheck: A mobile application that lets users photograph a mole and send it to a doctor for a professional. However, not an instant opinion [8].

UMSkinCheck: A mobile application intended for self-skin cancer exams. It allows users to storages photos for baseline comparison and provides informational videos and lectures on skin cancer prevention. However, the application is time investing by involving self-studying on skin cancer and does not utilize machine learning [9].

SkinVision: A professional web and mobile application that uses machine learning to detect and diagnose skin cancer, one of the mainstream skin cancer applications. It is fast and easy to use. However, it requires paying [10].

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

1. The construction of the neural network model to make it achieve good performance.
2. The data is collected from the internet to feed to the chat bot.
3. The assembling of all the functions into a web application.

The rest of the paper is organized as follows. In Section II, the structure of the web application will be introduced in detail. In Section III, we will introduce the dataset used for training and experiment with the CNN model on the introduced dataset. Finally, Section IV will conclude this paper.

2. Model Formulation

The application is a frontend-backend web application, where the VUE application serves as the frontend, and the deep learning models serve as the backend. The frontend takes input from users in terms of uploaded images or texts. The backend computes the output with given input, which is then displayed to users by the front end.

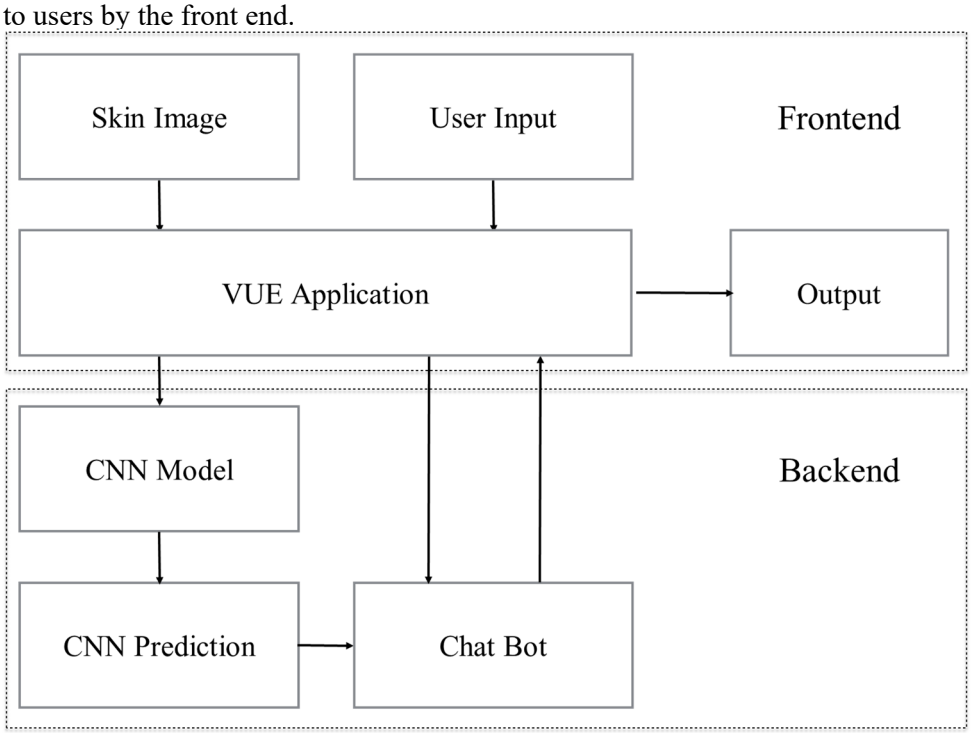

Figure 1. Web application structure

2.1. VUE Application

A VUE application that compiles all the functions, including the CNN model and the chat bot. The application is created to manage all the user's inputs and the outputs presented to the user. The application uses the JavaScript framework Vue.js, a state-of-the-art foundation for developers to create neat and clean user interfaces for their frontend applications.

2.1.1. Skin Image

The user uploads an image file. It is expected to be a photo of the user's skin where there is suspected to be skin cancer. The photo is supposed to be taken by the user themselves and uploaded to the VUE application. The VUE application will send the image to the backend.

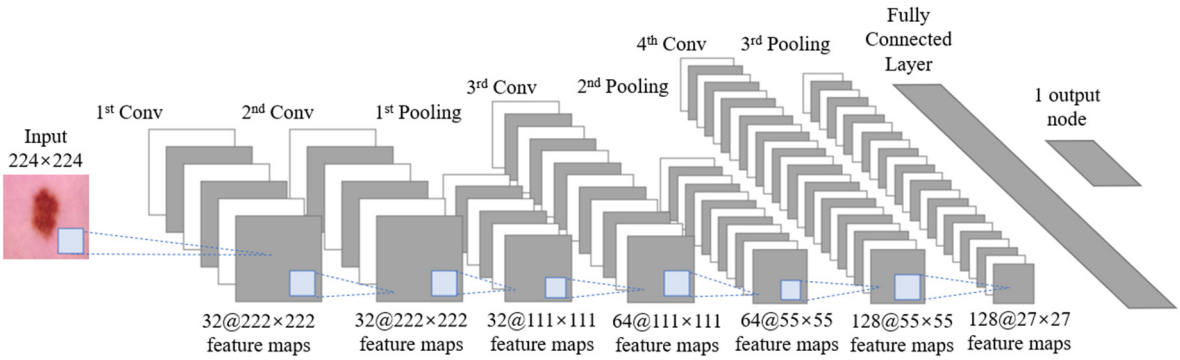

Figure 2. The architecture of CNN Model

2.1.2. CNN Model

A common CNN model consists of multiple convolutional layers followed by multiple dense layers. The layers are consecutively: 2d-convolutional layer (28x28x32), max pooling (14x14x32), 2d-convolutional layer(14x14x64), pooling(7x7x64), 2d-convolutional layer(7x7x128), 2d-convolutional layer (7x7x256), flatten layer (12544), dense layer (128), dense layer (64), dense layer (32), and dense layer (7).

The CNN model will be used to classify the skin image uploaded by the user and will output a 7-dimensional vector.

$$s(i,j) = (X * W)(i,j) = \sum_m \sum_n (xi + m, j + n)\omega(m,n) \quad (1)$$

2.1.3. CNN Prediction

A 7-dimensional vector output by the CNN model. Indicates the probability of each category of skin cancer. The vector will be sent to the chat bot by the backend to help it respond to the user.

2.2. User Input

A text input by the user is expected to be a question about skin cancer. The text is supposed to be input by the user in a text box and sent to the backend through the VUE application.

2.3. Chat Bot

A python NPL model based on the Chatterbot API that interacts with the user. Trained by 270 Q&A's about skin cancer asked on "quara.com". For collecting the data, a python web crawler is made to extract information from HTML pages. And for each question, the five top answers the question are selected. The chat bot will be trained with the crawled questions and answers. Finally, it will find the closest question to the question asked by the user from all the crawled questions and output the answer to the found question.

2.4. *Output*

A text is representing the answer to the question asked by the user. The text will be generated by the chat bot and sent to the VUE application. The VUE application will present the text to the front end.

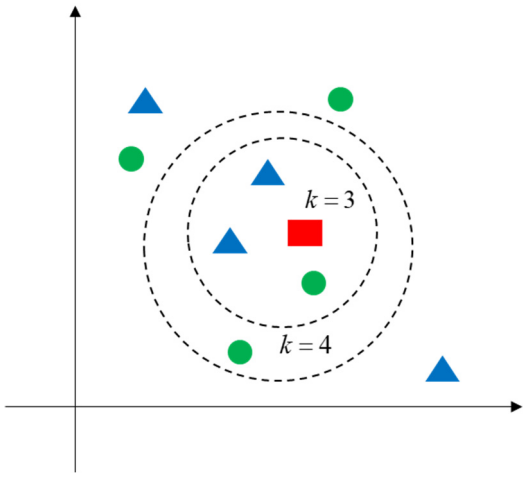

Figure 3. The illustration of image classification

3. Experiments

3.1. *Dataset*

The CNN model is trained on the dataset Skin Cancer MNIST: HAM10000("Human Against Ma-chine with 10000 training images"). The dataset collected 10015 dermatoscopic images from different populations, including a representative collection of 7 important diagnostic categories. We select the file "hmnist_28_28_L.csv" and "HAM10000_metadata.csv". The file "hmnist_28_28_L.csv" is a 10015 by 785 array containing 10015 vectors with length 785 = 28 * 28, each representing a vectorization of a 28 by 28 image. The file "HAM10000_metadata.csv" is a sheet containing information of each image in "hmnist_28_28_L.csv", including the cancer type, the age and gender of the individual, and the location of cancer. The dataset is preprocessed through a couple of procedures. First, the 10015 images are split into the training and testing datasets, with sizes 7511 and 2504. Next, because the number of images from each classification is not balanced, the training datasets should be balanced with the SMOTE method, resulting in the size changing to 35322. Last, the vectorized images are changed back from shape 785 by 1 to shape 28 by 28. The CNN model is then trained on the training dataset for 50 epochs with batch size 256 and learn rate 1e-3 and evaluated on the testing dataset. All procedures are executed on the online platform "kaggle.com".

At each training epoch, all of the input units are gone through the hidden layers to transform into new values, repeating this process until it reaches the output layer. Since we want the CT image to be either positive or negative, we chose the Sigmoid function as our last layer of output. Similarly, if the CNN model is expected to predict multiple categories, a SoftMax function replaces Sigmoid. Furthermore, after receiving the output, the CNN model will perform a loss function, binary cross entropy, in our case since we only have 2 possible outcomes to proceed with backpropagation. It is essential to perform a backpropagation after each epoch because only by doing that can the weights in each hidden layer and feature detector in each convolution layer and pooling layer be adjusted.

The chat bot is trained on a dataset containing a collection of 270 questions and corresponding answers in the google search list searching “skin cancer”. To obtain the data, first, a search list is made with the function search in python library GoogleSearch with the string “\“skin cancer\” stie:www.quora.com”. Next, the HTML file of each link in the search list is downloaded. Last, the HTML file is passed to the python module BeautifulSoup to extract important information, and the question and the answers are saved in txt files for further training. Following are some noticeable examples of questions obtained:

1. What are the symptoms of skin cancer?
2. What is the worst type of skin cancer?

3. What is it like living with skin cancer?
4. Is there any way to cure skin cancer without going to a doctor?
5. Do you feel sick if you have skin cancer?
6. Are white people more prone to skin cancer?

The obtained dataset is finally used to train the Chatterbot bot with a listing trainer.

3.2. Evaluation Metrics

To evaluate the performance of the CNN model, we look at its probability of making correct predictions on the test dataset.

$$Accuracy = \frac{length(where(testdataset) == testlabel)}{length(testdataset)} \quad (2)$$

3.3. *Experimental Results and Analysis*

The next two graphs showed the change of the accuracy and the loss of the CNN model on both the training and the testing dataset. The accuracy on the testing dataset converges after 20 epochs at around 0.85, and the accuracy on the training dataset converges at 0.99. We can see the CNN model has already achieved a considerable result within an acceptable number of epochs.

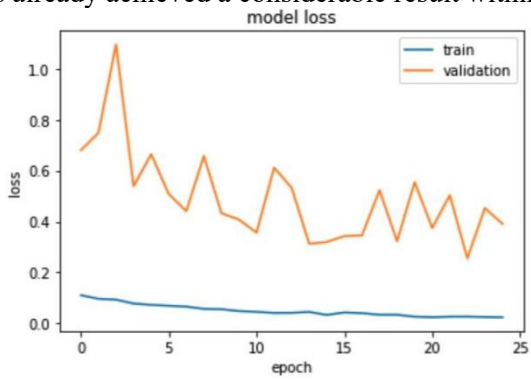

Figure 4. Accuracy on testing and training dataset

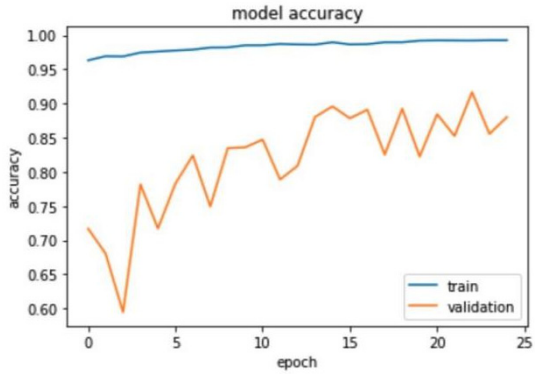

Figure 5. Loss on testing and training dataset.

4. Conclusion

We have presented a basic web application for skin cancer diagnosis by using the CNN model and the Chatterbot API. The CNN model has undertaken the classification of skin cancers, and the Chatterbot API is used to communicate with users. As per our assumption, the CNN model has shown a great capability of processing visual classification tasks. The accuracy of skin cancer type predicting has converged to 0.85 on the testing dataset, meaning that the application can handle a non-professional use of skin cancer diagnosis. The Chatterbot machine learning algorithms have also done great work in producing different responses and making it easy for developers to automate conversations with users. We have utilized the listed trainer of the Chatterbot API to easily train the bot with given questions and answers. Last but not least, the VUE framework makes it possible for developers to build a clean user interface for their web applications with an acceptable amount of codes. The user interface we have built is based on VUE. We have noticed that it is much easier than other JavaScript platforms to realize some advanced functions for the web application, such as letting users upload an image and displaying a chat bot in a floating window.

In the future, a series of meaningful work can be conducted subsequently. For example, we have considered a plugin that allows users to capture their skin with attached cameras and give real-time feedback. Furthermore, we are also conceiving the idea to connect data among multiple users and make a more comprehensive diagnosis by comparing the data. This additional information can be used to build

a better performing NLP system on an existing basis by applying a non-rule-based system performing abstractive instead of extractive text summarization. At the same time, to deal with the problems of overfitting and loss of information, an optimized model can be established by using more advanced oversampling algorithms to solve the issue of the imbalanced dataset. Furthermore, extending the system from patient-web end to web-doctor end can help improve the diagnosis accuracy of dermatologists with AI diagnosis results and summarized reports of patients as support.

References

[1] N. Hameed, A. Ruskin, K. A. Hassan, and M.A. Hossain. (2016) A comprehensive survey on image-based computer aided diagnosis systems for skin cancer. In: The 10th International Conference on Software, Knowledge, Information Management & Applications (SKIMA). Chengdu. pp. 205-214.
[2] F. Nunnari and D. Sonntag. (2019) A CNN toolbox for skin cancer classification.
[3] S. Ji, W. Xu, M. Yang, and K. Yu. (2013) 3D Convolutional Neural Networks for Human Action Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, Issue 1.
[4] Y. Nagender and K. H Pati. (2017) Whatsapp auto responder using natural language processing and AI. International Journal of Computer Engineering & Technology, vol. 8, Issue 5, pp 15-22.
[5] A. Ribas, P. Read, and CL. (2019) Slingluff. Cutaneous melanoma. DeVita, Hellman, and Rosenberg's Cancer: Principles and Practice of Oncology, Ch 92.
[6] T. Majitner, et al.. (2018) Ensemble of convolutional neural networks for dermoscopic images classification. CoRR, abs/1808.05071.
[7] M. A. Hilmy and P. S. Sasongko. (2019) Ensembles of convolutional neural networks for skin lesion dermoscopy images classification. In: 2019 3rd International Conference on Informatics and Computational Sciences. Semarang. pp. 1-6.
[8] K. Simonyan and A. Zisserman. (2015) Very deep convolutional networks for large-scale image recognition, CoRR, abs/1409.1556.
[9] K. He, X. Zhang, S. Ren, and J. Sun. (2016) Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas. pp. 770-778.
[10] V. N. Vapnik and A. Y. Chervonenkis. (2015) On the uniform convergence of relative frequencies of events to their probabilities. Measure of Complexity, pp 11-30.