A simple mobile application that use CNN to classify chest x-rays

Xiaodan (Judy) Zhu
University of Toronto
Toronto, Ontario
judyry.zhu@mail.utoronto.ca

ABSTRACT

Medical imaging, especially x-rays, is the most popular and cheapest technology in medical diagnosis. Chest x-rays can detect many diseases, such as pneumonia, infiltration and effusion. Seeing the necessity and importance of medical imaging in our lives, it would be meaningful if there were technology that can assist in x-ray diagnosis and lower the cost of the procedure. My project is using a CNN model trained from scratch, and using datasets from a publicly available source to do a binary classification of images that can detect a disease called infiltration. Then, the model is saved and converted to coreML and inference is done on a iOS platform. The accuracy of the model reached 78% percent for training accuracy but only over 50% on validation accuracy.

Author Keywords

CNN; medical imaging; CT and MRI; x-ray; AlexNet; GoogleNet

INTRODUCTION

Health is one of the most important areas of concern in our lives. Every year, governments all over the world spend millions of dollars on healthcare. In the U.S. alone, yearly spending can be as high as 980 billion.[1] Around 10% of this cost is spent on medical imaging and diagnosis, such as X-Rays, CT and MRI.[2] In recent years, the number of medical images taken per year is also increasing at a high rate. In the UK for example, there were 40 million imaging tests reported in the year of 2016 and in one month, there can be up to 1.91 million x-rays taken.[2] Many of these images have been used to diagnose cancer. Of all the different types of diagnostic tool, chest x-ray seems to be the most interesting one. It is by far, the most economical, basic and popular medical imaging diagnostic tool. Blood test and chest x-ray are the two mandatory exams one have to go through at a hospital's emergency room. Seeing the necessity and importance of medical imaging in our lives, it would be meaningful if there were technology that can assist in x-ray diagnosis and lower the cost of the procedure.

In recent years, the vast development of Artificial Intelligence and Machine Learning has made this a possibility. Deep Convolutional Neural Network for example, has proven to be very successful in image classification and object recognition. Alex Net, VGG net, and Google Net are some very recent papers in the research field that tries to use CNN to solve problems.[3]-[5] In the field of medical imaging, it has also been used to help detection of nodule, brain disease and skin cancer. [6]-[8]

In terms of X-ray diagnosis, in November 2017, Stanford professor Andrew Ng posted on twitter "Should radiologists be worried about their jobs? Breaking news: We can now diagnose pneumonia from chest X-rays better than radiologists."[9] He cited a paper called, "CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning", which described a 121 layer convolutional neural network trained on ChestXray14, currently the largest publicly available chest Xray dataset and claimed that this model can detect 14 diseases with a performance exceeding average radiologists. [10] The dataset contained over 100,000 images with 14 different diseases labeling.

While been amazed by the astonishing advancement in technology, I was somewhat doubtful about the result, since the dataset claimed that it got the labeling from NLP and the correctness was over 90%.[11] It did not cite if the labeling of the 100,000 images were a consent amongst experienced radiologists or if these results came from less experienced radiologists. Also, were these NLP results manually verified by radiologists? It seems that 100,00 images, if not labelled correctly, can cause a big problem, even if the CNN cam produce good classification accuracy.

With these questions in mind, I was interested in doing a project using these images, to see if I can somehow reproduce the results in the ChestXray14 paper to some extent. Moreover, since the topic of the course is on mobile, I thought of an interesting use case for x-ray classification on mobile. In many third world countries, the radiologists are less trained and inexperienced. Therefore, if their diagnostic result can be cross-checked with the diagnostic results of an accurate machine learning model, it can help save the lives of many people who do not have access to expensive medical resource. With a mobile application, the patients can take a picture or upload their x-ray result and see if that result matches the result given by doctors.

In my project, I used the dataset downloaded from kaggle and processed the data to do a binary classification on one disease. Then, I used a CNN network and created a model. The training was done on my laptop's CPU. The mobile interface was designed for iOS so I used coreML to do the conversion. The results are discussed in later sections.

RELATED WORKS

Health is one of the most important areas of concern in our lives. Every year, governments all over the world spend millions of dollars on healthcare. In the U.S. alone, yearly spending can be as high as 980 billion.[1] Around 10% of this cost is spent on medical imaging and diagnosis, such as X-Rays, CT and MRI.[2] In recent years, the number of medical images taken per year is also increasing at a high rate. In the UK for example, there were 40 million imaging tests reported in the year of 2016 and in one month, there can be up to 1.91 million x-rays taken.[2] Many of these images have been used to diagnose cancer. Of all the different types of diagnostic tool, chest x-ray seems to be the most interesting one. It is by far, the most economical, basic and popular medical imaging diagnostic tool. Blood test and chest x-ray are the two mandatory exams one have to go through at a hospital's emergency room. Seeing the necessity and importance of medical imaging in our lives, it would be meaningful if there were technology that can assist in x-ray diagnosis and lower the cost of the procedure.

In recent years, the vast development of Artificial Intelligence and Machine Learning has made this a possibility. Deep Convolutional Neural Network for example, has proven to be very successful in image classification and object recognition. Alex Net, VGG net, and Google Net are some very recent papers in the research field that tries to use CNN to solve problems.[3]-[5] In the field of medical imaging, it has also been used to help detection of nodule, brain disease and skin cancer. [6]-[8]

In terms of X-ray diagnosis, in November 2017, Stanford professor Andrew Ng posted on twitter "Should radiologists be worried about their jobs? Breaking news: We can now diagnose pneumonia from chest X-rays better than radiologists."[9] He cited a paper called, "CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning", which described a 121 layer convolutional neural network trained on ChestXray14, currently the largest publicly available chest Xray dataset and claimed that this model can detect 14 diseases with a performance exceeding average radiologists. [10] The dataset contained over 100,000 images with 14 different diseases labeling.

While been amazed by the astonishing advancement in technology, I was somewhat doubtful about the result, since the dataset claimed that it got the labeling from NLP and the correctness was over 90%.[11] It did not cite if the labeling of the 100,000 images were a consent amongst experienced radiologists or if these results came from less experienced radiologists. Also, were these NLP results manually verified by radiologists? It seems that 100,00 images, if not labelled correctly, can cause a big problem, even if the CNN cam produce good classification accuracy.

With these questions in mind, I was interested in doing a project using these images, to see if I can somehow reproduce the results in the ChestXray14 paper to some extent. Moreover, since the topic of the course is on mobile, I thought of an interesting use case for x-ray classification on mobile. In many third world countries, the radiologists are less trained and inexperienced. Therefore, if their diagnostic result can be cross-checked with the diagnostic results of an accurate machine learning model, it can help save the lives of many people who do not have access to expensive medical resource. With a mobile application, the patients can take a picture or upload their x-ray result and see if that result matches the result given by doctors.

In my project, I used the dataset downloaded from kaggle and processed the data to do a binary classification on one disease. Then, I used a CNN network and created a model. The training was done on my laptop's CPU. The mobile interface was designed for iOS so I used coreML to do the conversion. The results are discussed in later sections.

METHODS

Data collection and processing

The data is collected from "NIH dataset". [13] This NIH Chest X-ray Dataset consists of 112,120 X-ray images with disease labels from 30,805 unique patients. It is the same dataset that Wang et al. picked for their project.[13] Wang's paper focused on multi-labeling. For my project, since resource is limited and I only have my CPU on my laptop, I thought it would be more convenient to just do binary classification. The labels from the dataset are displayed as follows in a CSV file:

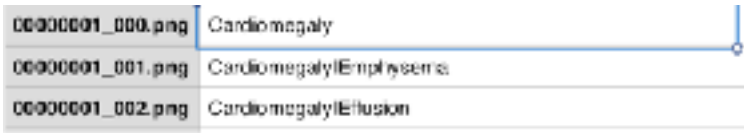

Figure 1. Labeling

Therefore, the first step in the data processing stage is to cut out the pipe that separates the diseases. This is done by creating new columns in the csv file with the header name been the disease name and the content been 1 or 0. After this is done, I calculated the total number of labels with the disease that I wanted to classify:

```
0     92226
1     19894
```

Figure 2. Number of infiltration labels

In the dataset, there are 19894 labels marked with this disease. Then, I picked 19894 labels from the non-disease labels and concatenate them together. This is because I want the model to see a balanced number of positive and negative results and minimize class imbalance. Then, the labels ares shuffled at random so that ordering is balanced. This can prevent unbalanced classification. Then, according to the image names and labeling, the images are collected and it's size reduced from 1024*1024 to 128*128. It's pixel values are normalized to between 0 and 1. The reason for the normalization and scaling is because in the process of multiplying weights and adding to these inputs in order to cause activations, in this process, it's best that each feature have a similar range so that our gradients don't go out of control.

Model

The model that I used is as follows:

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 126, 126, 32)	1184
max_pooling2d_1 (MaxPooling2	(None, 64, 64, 32)	0
conv2d_2 (Conv2D)	(None, 64, 64, 64)	8256
max_pooling2d_2 (MaxPooling2	(None, 32, 32, 64)	0
flatten_1 (Flatten)	(None, 65536)	0
dense_1 (Dense)	(None, 128)	8388736
dropout_1 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 1)	129

Figure 3. Model

The reason I did not want to use a pre-trained model is because I want to play around with building something from scratch and see if simpler models can also achieve good results like those of AlexNet and GoogleNet. This model that I used has 2 convolution and 2 max pooling layers. The convolution layer is used to create feature maps and the max pooling layer is used to reduced the number of trainable parameters. 2 convolution layers is used for my project because this is a typical structure that simple cnn models use. Then, the result goes through the fully connected layers so the output is desired.

The dataset is split between 0.8 and 0.2. The 20% is used for validation and test and 80% used for training.

Mobile

The model is saved and converted into a coreML model. The important information here is that when setting the coreML model, the image scale needs to be set to 1/255. This is because the model is normalized between 0 and 1 so when the images are actually in inference, the image scale needs to be between 0 and 1 as well.

In the mobile app, coded in Swift, it is important to note that all images need to be converted into a CVPixel format for the model to read. In the app UI, users can choose a photo from the library. These photos are picked from another dataset that is seperate from the ones I used for training. Then, the app will output whether the image is an infiltration or not.

Figure 4. UI

RESULTS

After building the structure, I ran the code using python and trained the model. I set 30 epoches and the training went on for about 8 hours. The result is listed as follows:

```
Epoch 28/30
31030/31030 [==============================] - 916s 29ms/step - loss: 0.3972 - a
cc: 0.7846 - val_loss: 1.4894 - val_acc: 0.5213
Epoch 29/30
31030/31030 [==============================] - 873s 27ms/step - loss: 0.3961 - a
cc: 0.7878 - val_loss: 1.2788 - val_acc: 0.5259
Epoch 30/30
31030/31030 [==============================] - 881s 28ms/step - loss: 0.3931 - a
cc: 0.7881 - val_loss: 1.3878 - val_acc: 0.5256
7950/7950 [==============================] - 43s 5ms/step
```

Figure 5. Results

The total accuracy reached about 78% for training yet the validation is only 53%. I suspected this is the problem with my model so I tried to use a pre-trained model online using

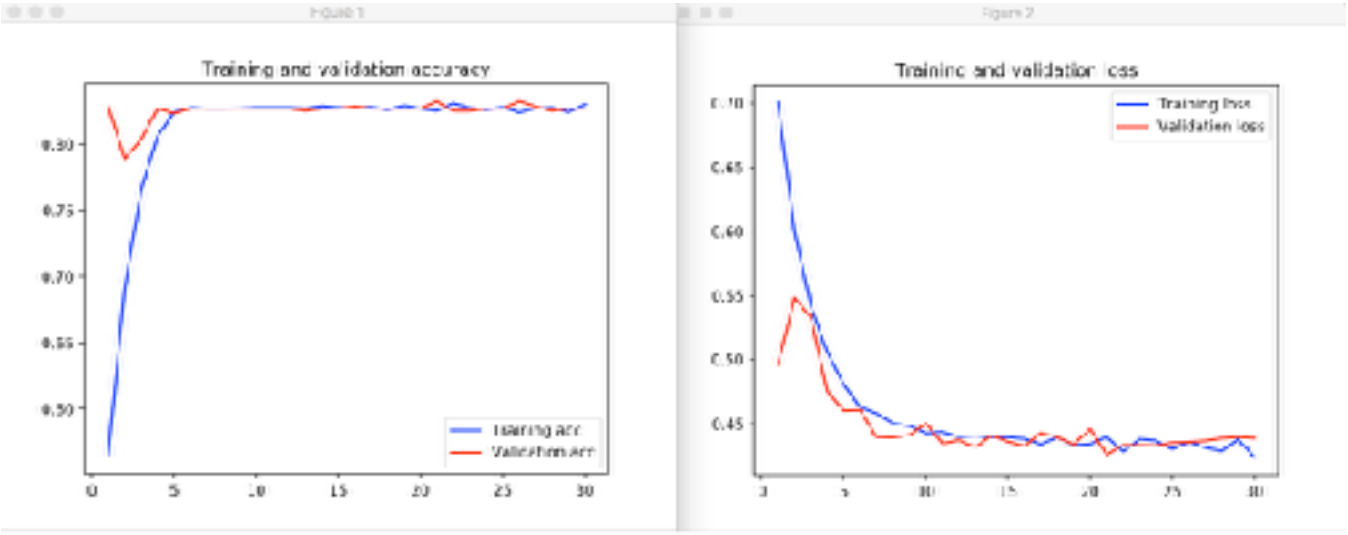

Figure 6. Results 2

MobileNet and see how that performs. The result is much better. The result is listed in Figure 6.

DISCUSSION

I would like to start with analyzing my model. The reason that the training result is 78% but the validation result is only 50% is perhaps due to problems with overfitting. I don't think the problem is not with not enough datasets because training of around 32000 images and validation of around 8000 images should be enough. However, after doing some research, I found that the model does not have a "relu" layer and transformation of images. With these techniques applied, perhaps the overfitting problem will decease.

For the pre-trained model, it gets an accuracy of around 80% for both training and validation. However, the interesting problem I found was that when I test it on mobile with the other dataset I found online by hand, I feel that the result is lower than 80%. During the conversion process, the model parameters should be inputted correctly. The image is rescaled to 1/255 and there were no errors when the inference happens. Since I don't really know how to batch test the images on mobile, through my hand testing, I am not sure whether there really was an error. There could be many areas of problems. One is that my model is still wrong while converting, either when converting from imageUI to CVPixel, or when setting the model parameters. Another is that there are mistakes with the dataset, either the NIH one or the other smaller dataset. However, with no medical knowledge, there is no way of verifying if the labeling is right.

One radiologist wrote in a blog about the problems he saw in this dataset.[14] He claims that he examined about 150 images in the dataset and found a very diverse result in how he labelled them and how the original datasets labels were. Moreover, despite the fact that the AUC reached around 70%, when he used the model to do some predictions, there's a very high amount of wrong results, comparing to his own judgement. This result is consistent with my evaluation. Although the model accuracy looked very good when I printed out the accuracy, when I hand tested and viewed the images, the result seems to be off. The radiologist quote, "I think this is caused by several things; medical images are large, complex, and share many common elements. But even more, the automated method of mining these labels does not inject random noise when it is inaccurate. The programmatic nature of text mining can lead to consistent, unexpected dependencies or stratification in the data." [14] I think that despite the fact that there is very likely a problem with my model and dataset, it is not impossible that the datasets need some improvement as well. Further investigation, especially with the involvement of experienced computer scientist and radiologists, is perhaps needed.

Another big issue is the fact that 80% is still low for this kind of task. For healthcare, an accurate prediction is very essential. A small margin of error can cause a big emotional influence for the patients. The same applies to the papers that I read. Most of the accuracy there are between 70-90 percent. This is perhaps still too low for this technology to become a clinically applicable technology. More research and improvement is a must.

CONCLUSION

In this paper, I conducted a preliminary study on using CNN to train a model for diagnosing a single disease from chest X-ray images. The dataset is collected from publicly available dataset of 100,000 images. The disease that I chose to classify is infiltration. The CNN model, through 30 epoches, reached an accuracy of 78% percent. Yet the

validation accuracy is only 52%. In addition, by using coreML, a mobile application is created to evaluate a 2D x-ray image that is uploaded by the user. The mobile application can successfully detect some of the x-ray images uploaded to the app. The data that is used to test in the mobile application is also downloaded from kaggle, but belongs to another dataset. The accuracy in the mobile application is lower than expected. This is perhaps due to less training and a simple network. Comparing to the popular GoogleNet and ResNet, my network is too simple. Moreover, the accuracy of the labeling itself can never be verified since I have no medical knowledge. Therefore, to make this application meaningful in the future, the participation of radiologists is a must, at least in the verification stage.

REFERENCES

[1] How much does the federal government spend on health care? (n.d.). Retrieved April 12, 2018, from http://www.taxpolicycenter.org/briefing-book/how-much-does-federal-government-spend-health-care

[2] Imaging Utilization Trends and Reimbursement. (2014, July 24). Retrieved April 12, 2018, from http://www.diagnosticimaging.com/reimbursement/imaging-utilization-trends-and-reimbursement

[3] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84-90. doi:10.1145/3065386

[4] Figure 2f from: Irimia R, Gottschling M (2016) Taxonomic revision of Rochefortia Sw. (Ehretiaceae, Boraginales). Biodiversity Data Journal 4: E7720. https://doi.org/10.3897/BDJ.4.e7720. (n.d.). doi:10.3897/bdj.4.e7720.figure2f

[5] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., . . . Rabinovich, A. (2015). Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2015.7298594

[6] B. V. Ginneken, A. A. A. Setio, C. Jacobs, and F. Ciompi, "Off-the-shelf convolutional neural network features for pulmonary nodule detection in computed tomography scans," in IEEE International Symposium on Biomedical Imaging, 2015, pp. 286–289.

[7] L. Rongjian, Z. Wenlu, S. Heung-Il, W. Li, L. Jiang, S. Dinggang, and J. Shuiwang, "Deep learning based imaging data completion for improved brain disease diagnosis," 2014, pp. 305–12

[8] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," Nature, vol. 542, no. 7639, p. 115, 2017

[9] Ng, A. (2017, November 15). Should radiologists be worried about their jobs? Breaking news: We can now diagnose pneumonia from chest X-rays better than radiologists. https://t.co/CjqbzSqwTx. Retrieved April 12, 2018, from https://twitter.com/andrewyng/status/930938692310482944?lang=en

[10] Figure 2f from: Irimia R, Gottschling M (2016) Taxonomic revision of Rochefortia Sw. (Ehretiaceae, Boraginales). Biodiversity Data Journal 4: E7720. https://doi.org/10.3897/BDJ.4.e7720. (n.d.). doi:10.3897/bdj.4.e7720.figure2f

[11] National Institutes of Health Chest. (2018, February 21). Retrieved April 12, 2018, from https://www.kaggle.com/nih-chest-xrays/data

[12] Dong, Y., & Pan, Y. (n.d.). Learning to read chest-x-ray. Retrieved from http://iiis.tsinghua.edu.cn/~weixu/files/bigdata4health2017_pan.pdf

[13] Wang, Xiaosong, Peng, Yifan, Lu, Le, Lu, Zhiyong, Bagheri, Mohammadhadi, and Summers, Ronald M. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. arXiv preprint arXiv:1705.02315, 2017.

[14] Exploring the ChestXray14 dataset: Problems. (2017, December 18). Retrieved April 12, 2018, from https://lukeoakdenrayner.wordpress.com/2017/12/18/the-chestxray14-dataset-problems/