Chest X-ray Disease Diagnosis Classification

Team #24: Lanyin Zhang, Peng Yi, and Qingguo Chen
Date: 4-24-2018

I. Abstract

Human diagnosis of X-ray images is both time consuming and prone to errors. In order to expedite the process and to improve the accuracy, we developed an automatic diagnosis tool based on chest X-ray images using big image database and convolutional neural network (CNN) methods, with a web interface implementation for interactive usage by both caregivers and patients. Adopting a transfer learning strategy significantly expedites the training process, yet still achieves satisfactory prediction accuracies, as compared to the state-of-the-art in the literatures, which mostly used full training method. (Youtube demo link: https://youtu.be/xr1dN6l0fco)

II. Introduction

Chest X-ray image analysis is one of the most common diagnosis techniques. Chest X-ray produces images of heart, lung, airways, blood vessels and the bones of spine and chest, utilizing the fact that different organs of the body absorb X-rays differently. It requires only a very small dose of ionizing radiation, and is generally considered noninvasive and safe. It is also cost effective and easy to operate. It is often the first examination to perform to the patients, especially those with symptoms like difficulty in breathing, coughing, fever or chest pain, and is heavily relied upon for diagnosis and treatment.

Although X-ray is a very regular examination, analyzing X-ray images for diagnosis is not easy. It requires significant amount of time, and its accuracy strongly depends on medical knowledge and clinical experience. Fortunately, the development of big data methods and increasingly available image databases in the recent years has made it possible for automatic image diagnosis through machine learning. In this project we used the image recognition techniques to develop and examine an automatic diagnosis model based on deep-learning on a large experimental image dataset. This model can help provide first level understanding of the chest X-ray images, and accelerate the diagnosis process so that more attention and resource can be devoted to treatment and research. We believe that artificial intelligence has a bright future in facilitating X-ray image analysis for clinical applications, and hope our work in this project can demonstrate it.

A large X-ray dataset, ChestX-ray14, from the National Institutes of Health (NIH) Clinical Center has allowed the development and testing of deep learning algorithms to tackle this problem. Wang et [1] introduced this dataset and has used convolution neural networks (CNNs) to classify the 14 diseases types labeled in this dataset to provide an initial benchmark. Triggered by Wang et al.'s work, Yao et al.[2] presented a combination of CNN and a recurrent neural network (RNN) to exploit label dependencies, using a DenseNet model that is trained on X-ray data. Li et al.[3] presented a framework for pathology classification and localization using CNNs. Rajpurkar et al.[4] has significantly improved on the modeling techniques and achieved state-of-the-art results using an 121-layer DenseNet architecture. Notably, the results by their network is more accurate than the prediction provided by their collaborating radiologists. More recently, Baltruschat et al.[5] have performed a more systematic evaluation on the effect of different learning setup, as well as feature construction using non-image data like patient age, etc. Their work has outperformed Rajpurkar et al.[4] in 4 out of the 14 diseases. However, even with these efforts and improvements, there is still a substantial variability of the classification results for the 14 diseases. Part of the variability might come from the quality of the labels of the dataset [6], which in turn points to the advantage and opportunities for the applications of deep learning in the clinical practices.

In this study, we applied transfer learning based on freezed layers of ResNet50 [7] model to the ChestX-ray14 dataset using distributed computing by PySpark. A new three layers neural network was trained, and the performance is measured and compared with literatures using the Area Under Curve (AUC) metric. A web interface was developed for live diagnosis.

III. Methods

1. Data and General Statistics

The NIH X-ray dataset is downloaded from Kaggle.[9] The total size of the dataset is 45.1 GB. It contains 112,120 images of 30,805 unique patients from 1 to 414 (seems like a mistake in the dataset) years old. There are 63,340 images for male patients, and 48,780 for female patients. All the images were labeled with either "No Finding", one out of 14 diseases, or a combination of 14 diseases. Among them, 60,361 images were labeled as "No Finding", which means no pattern of the 14 diseases was found from the image. A disease name is contained in the label of an image if the corresponding patient is diagnosed with this disease. In the following barchart (Figure 1), we summarized the counts of each disease. The sum of all counts is greater than 112,120, because a patient can have more than one diseases. We also randomly sampled one picture for each disease and control, and presented them in Figure 2. All images have resolution of 1024 x 1024.

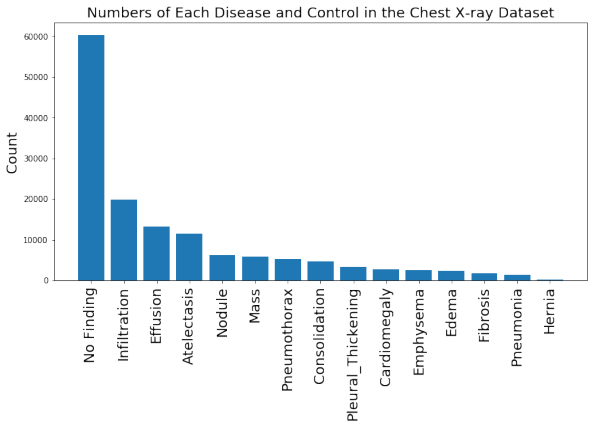


Figure 1. Image counts of diseases and control

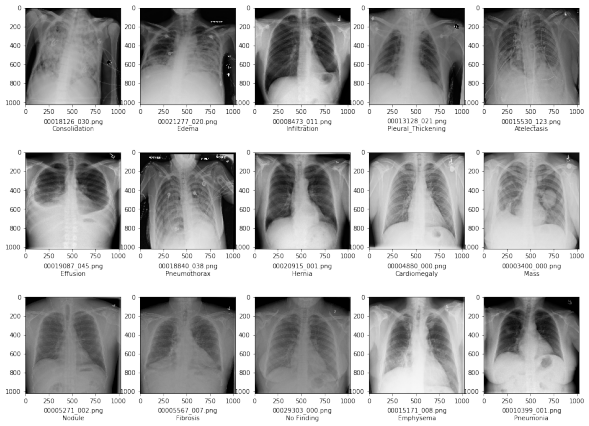


Figure 2. Sample Chest X-Ray images for diseases and control

2. Experimental Setup

In this study, we conducted all experiments in Python environment. PySpark was used for loading metadata, splitting and preprocessing samples, applying freezed layers of ResNet50 model to extract 2048 features from the original 1024x1024 images, and feature saving/loading from/to the local storage. The new three layers neural network model was built and trained using PyTorch. ReLu activation were used for the first two layers, and the last layer was activated by Sigmoid function. Bceloss function was used to calculate binary cross entropy. Since the data are highly imbalanced, we weighted the Bceloss function by measuring the disease frequency in the training set. Adam optimizer was used to optimize model. To find the best learning rate and weight_decay, a grid search process was conducted. And we found with learning rate of 0.0003, weight_decay of 0.00001, betas of (0.9, 0.999), and eps of 1e-08, optimer trained the model having the smallest loss value. ReduceLROnPlateau Scheduler was used with patience of 1, and factor of 0.2. The feature extraction process was conducted in Gatech server machine, and the new model training was performed on a workstation with 16 cores CPU, 48G Memory, and 8G GPU.

3. Work Pipeline

The NIH Chest X-Ray dataset contains 14 disease labels and 1 control label. Since each image may be labeled with multiple diseases, we translated each output into an array with 14 integers of either 0 or 1, with 0 representing "negative" or "No Finding" and 1 otherwise, for each of the 14 disease. The file "Data_Entry_2017.csv", containing image names, patient information and disease labels, were loaded into PySpark as DataFrame, and each row was preprocessed as follows: 1) 14 records with patient ages greater than 123 were filtered out; 2) Dataframe was converted into RDD; 3) Image name was formatted to corresponding file path; 4) Ages were scaled by 100, and gender was one-hot encoded; 5) Each label was converted into one-hot encoded numpy array with 14 elements; and 6) the dataset were splitted into training set (70%), validation set (10%) and testing set (20%).

We then applied transfer learning strategy to solve the classification problem. Recent work by Kermany et al.[10] showed excellent results on X-Ray image classification by freezing the layers of pretrained model except the last fully connected layer, and only training the last layer. We decided to apply the same strategy in PySpark. We chose pretrained ResNet50, without the last fully connected layer, as the starting model for transfer learning. Each image was read, scaled, transformed and loaded into the above model, and the output of model was converted into a 2048-element numpy array. Basically, we use the pretrained CNN to extract 2048 features from the images. Then we added another two features (age and gender) into the above output array to a total dimension of 2050. We ran the above processes in PySpark with RDD map function. All the images were resized to 224 x 224. For training set, random horizontal flip was applied to each image. Data were normalized according to Image set statistics. Then for each row of RDD, image data were converted into torch FloatTensor variables and loaded into the freezed ResNet50 model, and the output was collected as numpy array, and appended with encoded age and gender features and encoded ground truth label. Then each row of RDDs was joined as one String by comma, and the RDDs were saved as text files for next training step.

The above steps finally converted 45 GB images into about 4 GB splitted text files. The splitted text files were again loaded by PySpark, and the RDDs were mapped and collected into float numpy array, splitted into input array and target label array (the last 14 columns), and then formatted as tensor dataset with batch size 10. Next, we built another neural network in PyTorch with three fully connected layers (2048 to 128, 128 to 32, and 32 to 14), using ReLu activation for the first two layers and Sigmoid activation for the last one. BceLoss weighted loss function and Adam optimizer were used, and the validation losses were calculated for training. Total 50 epochs were ran and the training losses and validation losses were recorded for learning curve plot. A checkpoint of the model was recorded whenever there was a decrease in validation loss.

The final model was evaluated by calculating roc_auc_score for each disease on testing set. Since the weighted loss function was applied, the outputs of Sigmoid function for each disease would be the probability of the positiveness. We implemented a Flask web application to apply the above strategy to obtain prediction for any single chest X-ray image, and showed the probability of each disease to the users.

The methods pipeline was summarized in the Figure 3.

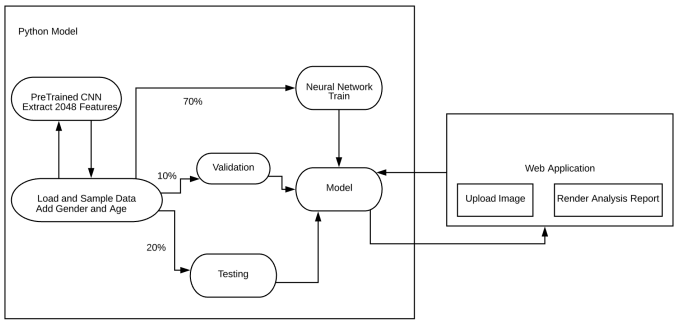
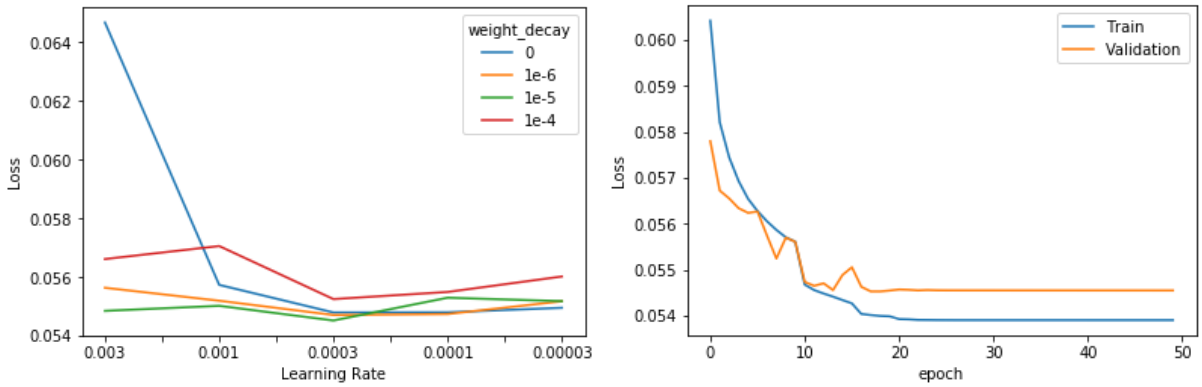


Figure 3. Work pipeline of the automatic diagnosis package

IV. Experimental Results

After distributed computing on feature extraction of all images by PySpark, training, validation and testing data were recorded as files with 2050 columns of features and 14 columns of target array. The training and validation data were loaded for training our neural network model with three fully connected layers. Learning parameters were optimized by monitoring the loss function. Figure 4(a) shows that through grid search a combination of learning rate of 0.0003 and weight decay of 1e-5 result in the best loss performance. Figure 4(b) shows the learning curve during training process to monitor the decreasing trend of the mean of binary cross entropy for both training samples and validation samples. Base on Figure 4, the decreasing trend stops at about 20 epochs for both the training and validation sets. The model with least validation loss was recorded as the best model.



(a) (b)

Figure 4. Optimizing learning parameters

To evaluate our model, the AUC scores for each disease on the testing dataset, using both the model just after initialization and the best model after training, were calculated and plotted in Figure 5. Based on Figure 5, the trained model shows significant improvement for all the diseases compared to the initial model. Our best model also outperforms the model by Wang et al [1] for 9 diseases, and achieves comparable performances for 3 diseases.

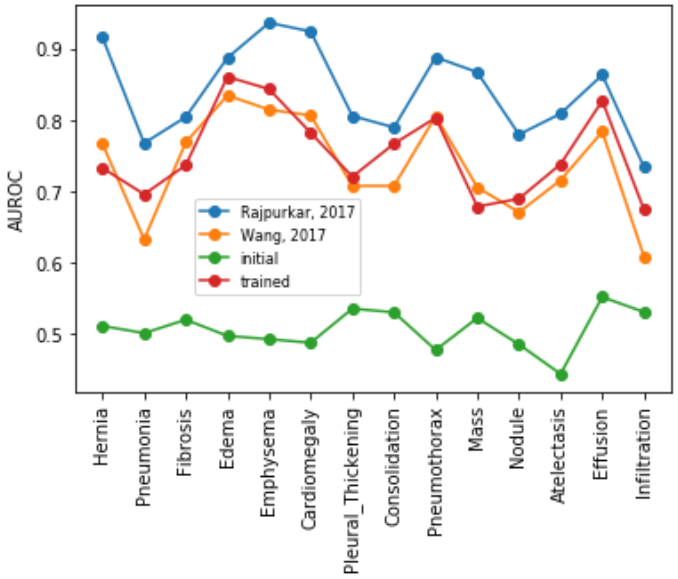


Figure 5. Performance comparison between different models

With the sequential process of the freezed pretrained ResNet50 model and our best model, we successfully built a web application on local server. Figure 6 shows an example of the uploaded image and the model prediction.

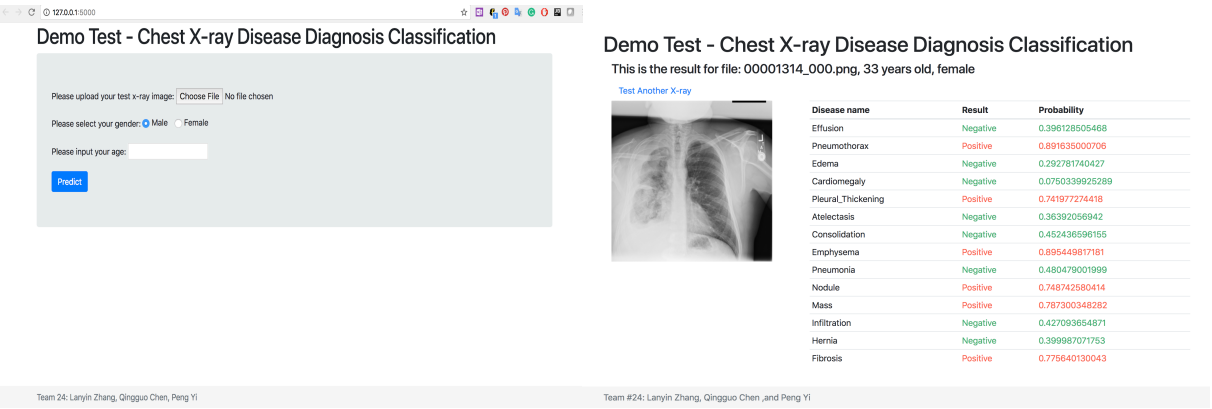


Figure 6. Web interface for Chest X-ray Disease Diagnosis

V. Discussion

Our models outperformed the CNN model by Wang et al. 2017 [1] for 9 diseases, and achieved similar performance in another 3. One possible reason for this outperformance could be the inclusion of gender and age as features. It is a reasonable approach since people with different gender and age would have different organ sizes. These demographic features should be taken into consideration in similar health data analysis.

Another advantage of our transfer learning strategy is that the training of new model is not time consuming comparing to the full training of a CNN model. Our model training usually took less than 20 minutes for all 50 epochs, but full training of a CNN model usually took more than 20 minutes for each epoch. Even after taking into account of the feature extraction time of about 8 hours by PySpark, the whole training process with transfer learning is still much shorter than the full training of CNN model.

Furthermore, since the ResNet50 model is pretrained, fixed and available from many resources, we only need store our three layer neural network model. The size of the full ResNet50 model is about 100 MB, but our model is only 1 MB in size. It is much easier to store and share our model.

Comparing to the more recent 121 layers DenseNet model [4], however, we underperform in every disease class. There are several possible reasons: 1) We used ResNet50 model, but DenseNet121 could be better for X-ray classification; 2) A fine-tune ResNet50 is necessary for solving the X-ray classification problem, rather than freezing the ResNet50 model; 3) The settings of training process, like learning rate of optimizer, decay rate of the learning rate, and batch size, have not been fully optimized. We plan to conduct fine tuning of ResNet50 model in the next phase of study. The fine tuning has another advantage that we can perform class activation mapping to localize the area for diagnosis.

VI. Conclusion

We successfully implemented an integrated system with distributed computing and deep learning for chest disease diagnosis based on X-ray image with a web interface. We found that using a pretrained model can significantly reduce the training time and still achieve satisfactory prediction results. The web interface can provide instant diagnosis information for caregivers and patients.

References:

[1] X. Wang et al., Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3462–3471. IEEE, 2017.
[2] L. Yao et al., Learning to diagnose from scratch by exploiting dependencies among labels. In: CoRR (2017)
[3] Z. Li et al., Thoracic disease identification and localization with limited supervision. In: CoRR (2017)
[4] Pranav Rajpurkar et al., CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning, arXiv:1711.05225 [cs.CV]
[5] Ivo M. Baltruschat et al., Comparison of Deep Learning Approaches for Multi-Label Chest X-Ray Classification,
[6] L. Oakden-Rayner, Exploring the ChestXray14 dataset: problems (2017), https://lukeoakdenrayner.wordpress.com/2017/12/18/
[7] K. He et al., Deep Residual Learning for Image Recognition, arXiv:1512.03385 [cs.CV] (2015)
[8] Radiological Society of North America (RSNA) and American College of Radiology (ACR). (n.d.). X-ray (Radiography) - Chest. Retrieved March 10, 2018, from https://www.radiologyinfo.org/en/info.cfm?pg=chestrad
[9] Kaggle, 2018 NIH X-Ray Dataset, from https://www.kaggle.com/nih-chest-xrays/data/data
[10] Daniel S. Kermany et al., Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning, Cell, 172, 1122-1131 (2018)