Chest X-ray Thoracic Disease Diagnosis with Convolutional Neural Networks

Wenqin You, Jing Zhao, Xiaojie Du, Jingyao Zhu
Georgia Institute of Technology, Atlanta, GA, USA

https://github.gatech.edu/jzhao365/BD4H-2020Fall-Project
https://www.youtube.com/watch?v=i59dJBxOh-I

Abstract

This project aims to build an automated tool for diagnosing thoracic disease using X-ray images. We have developed an end-to-end deep convolutional neural network architecture to detect and classify thoracic diseases in the lung region facilitated with transfer learning techniques. The performance of our model has been compared with state-of-art in the literature and achieved achieve satisfactory performance evaluated with NIH Chest X-ray Dataset.

1 Introduction

Automated medical image analysis has been a very important field in applying cutting-edge computer vision techniques for the visionary goal of intelligent healthcare. The goal of the automated medical image analysis is to alleviate the burden of physicians and improve the accuracy of clinical diagnosis.[1] The recent progress of deep learning techniques and the success of computer vision algorithms have allowed many researchers to join this field to develop efficient models and architectures that outperform the diagnosis of practicing physicians and radiologists[2]. In this project, we are aiming to build a simplified end-to-end deep convolutional neural networks (CNN) model that diagnose 13 different kinds of thoracic diseases presented in the NIH Chest X-ray dataset with state-of-art performance.

2 Related Work

The chest X-ray has been the major diagnostic approach for thoracic disease for decades. Wang *et al* first demonstrated the application of an unified weakly-supervised multi-label image classification and disease location deep learning CNN framework for detecting the occurrence of thoracic disease[3]. They have established the hospital-scale X-ray image dataset which can be used for training the deep neural networks for automated image analysis and annotation. Rajpurkar *et al* developed the CheXNet algorithm which consists of a 121-layer deep CNN trained on ChestX-ray14 dataset that outperforms the diagnosis from practicing physician and radiologist. They have used dense connection and batch normalization in the network architecture and trained the neural network with Adam optimization[2]. Li *et al* presented a unified approach that simultaneously performs disease classification and localization. They developed the workflow to slice the image into grid patches and then conducted either supervised learning or multiple instance learning (MIL) for the patches[4]. Ge *et al* proposed a novel loss function based on multi-label softmax loss (MSML) to address the problem of multi-label and imbalance between disease's categories. They introduced a fine-grained cross-entropy loss layer for the training[5]. Liu *et al* presented a segmentation-based deep fusion network (SFDN) that take advantages of both domain knowledge in thoracic disease and high resolution images in the lung regions. They have developed a lung region generator that crop the critical region from X-ray image for feature extraction which later used for CNN training[6]. Guan *et al* proposed a category-wise residual attention learning (CRAL) framework that consists of the feature embedding module using CNN and an attention learning module for exploring the assignment scheme of different categories[7]. Jaiswal *et al* has applied a segmented Mask-RCNN for creatively processing the merges bounding boxes on each region of interests[8]. Moreover, Irvin *et al* released the CheXpert dataset to public as a standard benchmark for framework training. They have used different methods to learn the uncertainty labels in the dataset that leads to the better diagnosis than 3 radiologists for 4 clinically relevant diseases[9].

3 Experimental Setup

3.1 *Data*

There are three available datasets - NIH Chest X-ray Dataset, CheXpert Dataset, and JSRT Database. In this project, data was mainly collected from NIH Chest X-ray Dataset, which contains 112,120 total images with size 1024×1024 anonymous chest X-ray images from 30,805 unique patients[10]. And two csv files are "Data_entry_2017.csv", which

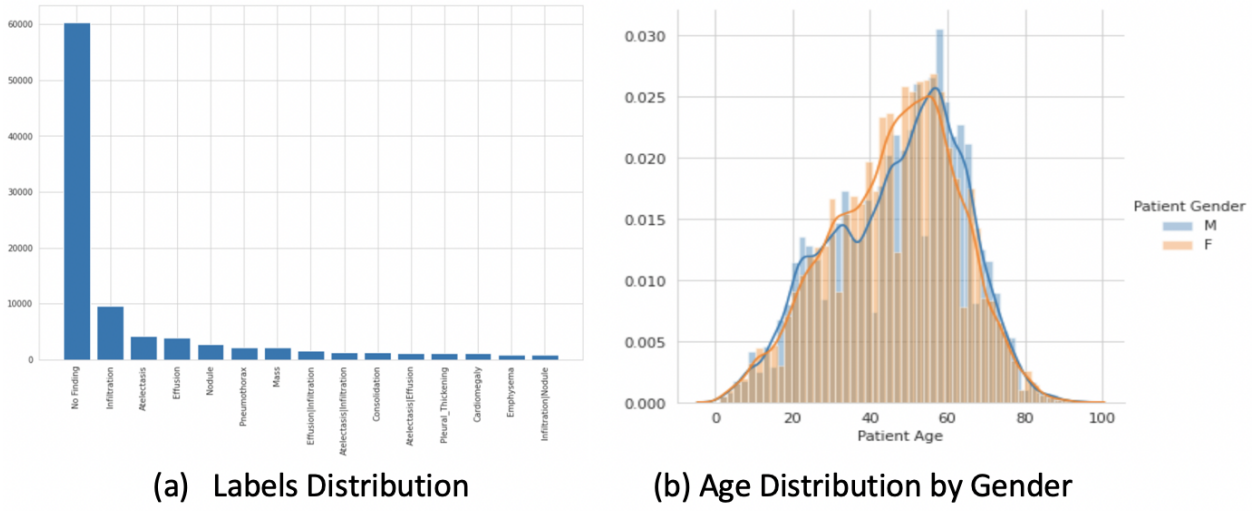

Figure 1: NIH Chest X-Ray statistic view

comprises 709 unique categories present in dataset labels but the labels are hierarchical and "BBox_list_2017.csv", which contains the bounding box coordinates. The whole process will be followed as below. It first starts with data exploration, which includes reprocessing and EDA. Then select the right feature or create new features to help model predict. Model deployment will be coming after feature extraction. Finally, model testing and evaluation will be involve to help pick the best model. From Figure 1 "labels distribution", there are 14 diseases labeled in this dataset including atelectasis, consolidation, infiltration, pneumothorax, edema, emphysema, fibrosis, effusion, pneumonia, pleural thickening, cardiomegaly, nodule, mass, and hernia and before analysis, the hierarchical unique categories is split by bar. Based on the label, "No finding", which does not mean that the patient is healthy, has approximately 60,000 data points. The main goal of this project is to detect the diseases and "No finding" confirms that the patient do not have these specific diseases. We decided to change label "no finding" to 0 at this point. Since one disease has less than 1000 data, 13 diseases will be discussed and reported in this project. Based on different dimensions, we could conclude general statistical analysis about the data set showing in Figure 1 "age distribution by gender". There are two gender - men and women, which includes close 63,000 (56%) men and near 49,000 (44%) women in this data set. The average age of total patient is 46.9 with 16.8 std. However, one thing must be noticed is that about 16 patients are greater than 100 years ago and 400 patients are labeled as months and 1 of 400 is labeled in days. From Figure 2 "simple or multiple labels per image", each patient ID may have at least one image and one follow-up time and the average of follow-up times, which is 8.6 with 15.4 std, may not be accurate with one to multiple relationship. There are 30,963 images associated with only one label that belongs to one of the 14 diseases; the other 20,796 images are associated with more than one label. From Figure 2 "Disease Distribution by Gender", it shows that women are less likely to have such 14 diseases compared with men as data set does have less data points and women does have gender advantages.

3.2 *Software Stack*

Apache Spark 2.3.1 and particularly the PySpark has been used to process the metadata and conducting the exploratory data analysis (EDA). The standalone local mode of Spark is used to read and process the data which is later converted to Pandas dataframe that feed into Tensorflow/Keras framework for training and testing. The program development environment is Python 3.6.8 and the environment management is achieved using Anaconda and Pip. The matplotlib library has been used to conduct EDA and plot experimental results. The details of environment can be found in the *environment.yml* file.

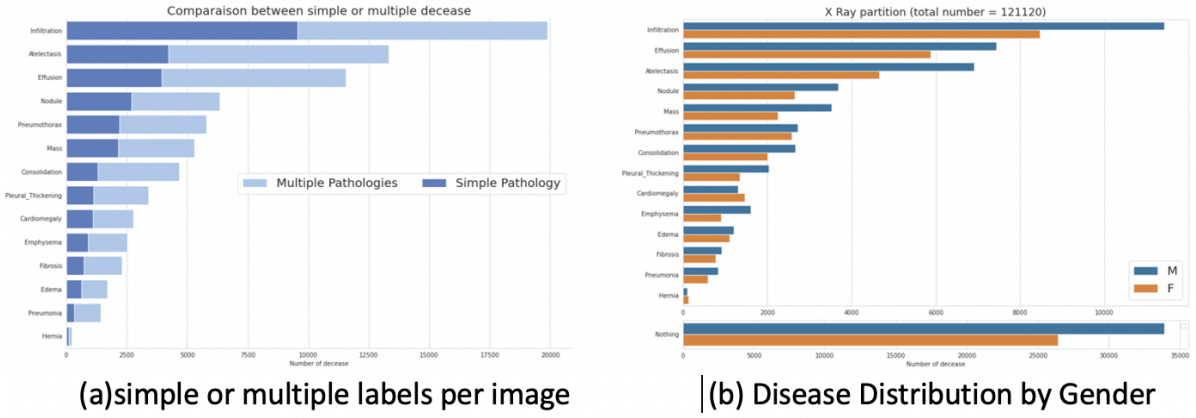

Figure 2: Label Composition of the NIH Chest X-Ray

3.3 *Hardware Specifications*

We have used Amazon Machine Image (AMI) to launch EC2 instance for project development. The operation system of the AMI is Ubuntu 18.04 (version 34.0), and packages such as Tensorflow, PyTorch, Anaconda and Docker are pre-installed. We have chosen p2.xlarge instance with GPU to train our deep learning model. The p2.xlarge instance has 1 NVIDIA K80 GPU and 4 vCPUs and CUDA toolkit is pre-installed for rapid onboarding. To store and NIH Chest X-ray Dataset and exported results and models, additional storage space (500G EC2 volume) is mounted to the instance. We can connect the EC2 instance with SSH client by locating your private key file and its public DNS. All the procedure can be achieved by following this course (AWS Essential Training for Developers)[11] step-by-step.

4 Technical Approach and Evaluation Metric

As shown in Figure 3, a standard modeling pipeline will be utilized for this classification task, i.e., data exploration, feature extraction, model development and model testing. The dataset has been split into training, validation and testing datasets by performing stratified random sampling (without replacement) techniques, which preserves the original ratio of finding labels for each dataset obtained. The training dataset has been used to train the algorithms initially selected and the validation dataset was used to determine the stopping point for each algorithm trained, i.e., to select the final model for each algorithm implemented. Finally, the testing dataset was utilized to evaluate the ultimate model performance and obtain the final winner algorithm, which achieves the best model performance among all the selected algorithms on the testing dataset. To be more specific, we has explored the datasets by using summary statistics and visualization tools such as histograms. As discussed above, we preprocessed the data by changing label "no finding" to 0. For feature extraction, as the outperformance of CNN models in image recognition and the power of transfer learning in knowledge transfer, we leveraged some pre-trained CNN models (i,e., VGGNet16, VGGNet19, ResNet, MobileNet, InceptionV3 and NASNet) to construct the features. Specifically, we has input the images into the convolutional layers of the pre-trained CNN models and extract the weights returned from the layers as our initial features. Then we has fed the extracted features from the pre-trained CNN models to two types of architecture to predict the classes of the input images as illustrated in Figure 4. The first architecture is a feed forward neural network with one average pooling layer and two fully connected layers with dropout applied to prevent over-fitting with two activation functions, i.e., Relu and Sigmoid functions, applied to each fully connected layer respectively, and it has been referred as simple model in the following content. The second architecture utilized the attention models by first feeding the extracted features to a convolution block (i.e., some convolutional layers and average pooling layers), then obtaining the attention weights by product of the outputs of the convolution block and the initially extracted features, and finally training the attention weights through average pooling layers and fully connected layers. This approach has been referred as attention model in the following discussion. For each pre-trained model and architecture utilized, the final model was selected to achieve the smallest loss of the validation dataset. Among different algorithms, we chose the final algorithm with the best Area Under Curve (AUC) values of the Receiver Operating Characteristic (ROC) curves obtained based on testing dataset among all the algorithms.

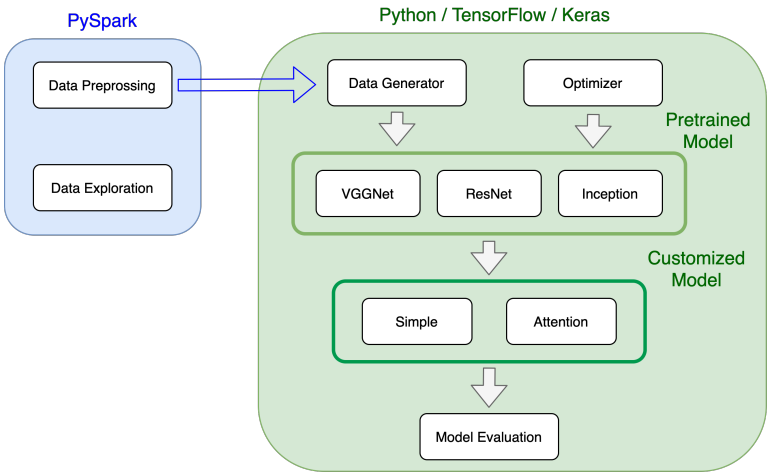

Figure 3: Project Pipeline for Chest X-ray Thoracic Disease Diagnosis

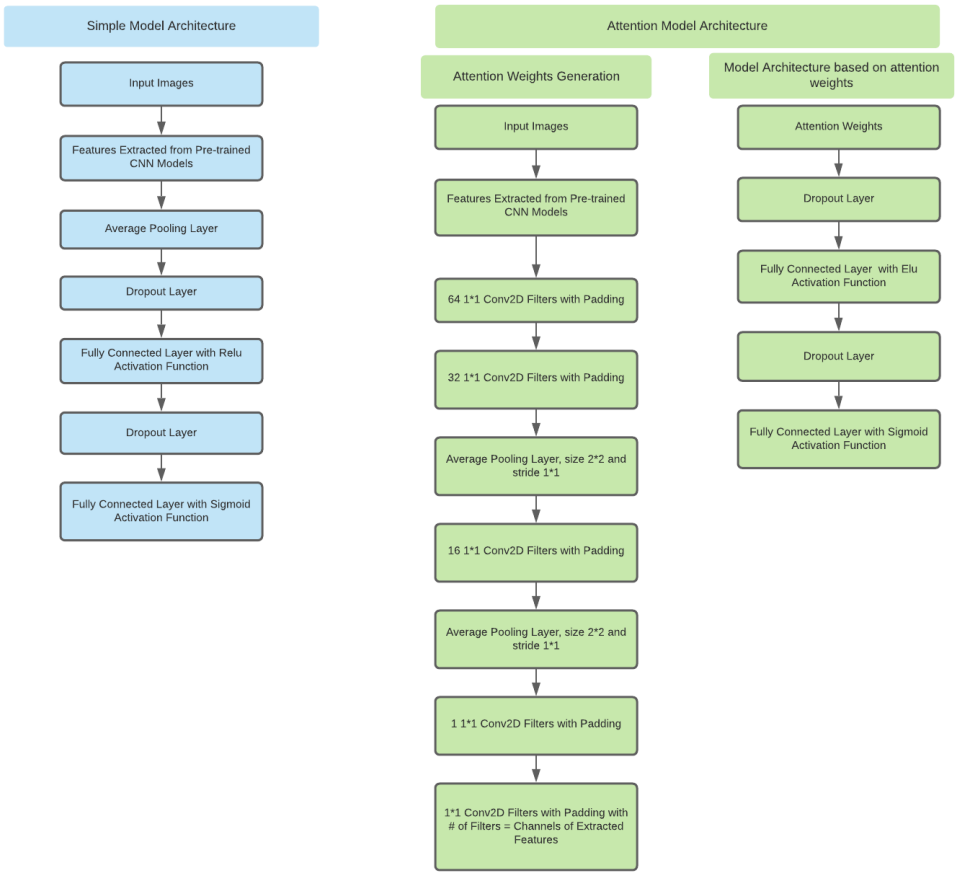

Figure 4: Model Architecture

5 Experimental Results

The training and testing are randomly sampled from stratified dataset. The accuracy and loss trajectories for 40 epochs of both training and testing by using VGGNet16 as the pre-trained CNN model are illustrated in Figure 5. The param-

eters for training the CNN framework includes: SAMPLE_SIZE=80000, BATCH_SIZE=16, STEPS_PER_EPOCH = 1000 and LEARNING_RATE = 0.0005. The loss function is defined using binary cross-entropy, which is described as $Loss = \sum_{i=1}^{N} y_i \log(p(y_i)) + (1-y_i)\log(1-p(y_i))$. We can observe that both training and testing loss are decreasing and accuracy are increasing during the 40 epochs. Additionally, no significant gaps or sudden drops/increases were observed for the accuracy and loss trajectories between the training and testing datasets. Thus, no overfitting or gradient vanishing concerns were raised for the models obtained. Moreover, the testing curve have slightly lower accuracy and higher loss compared to that of training, which are expected for the typical validation trajectories. Overall, the accuracy of our CNN model is stabled at 0.878 after the training.

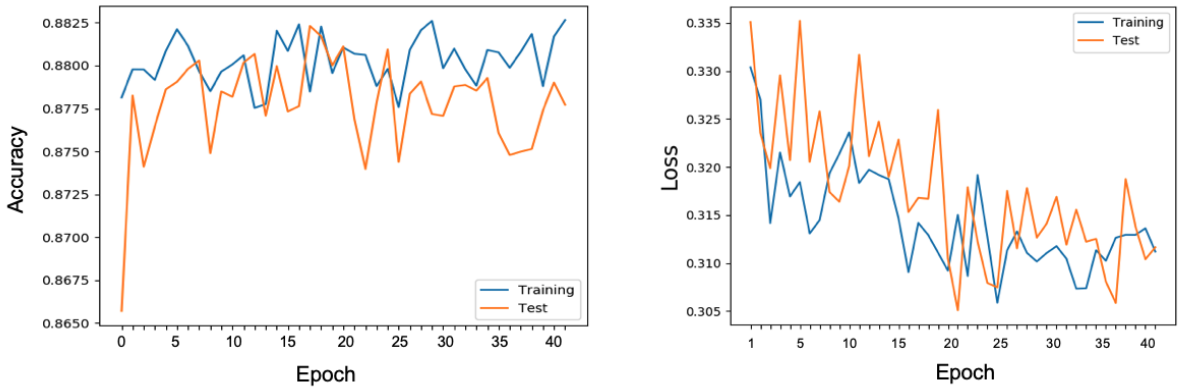

Figure 5: Accuracy and Loss Curves of the Training and Testing using VGGNet16

Table 1: AUC Values of 13 Disease's Categories for CNN Models

Diseases	Wang et al (2017)	VGGNet16 (simple)	VGGNet16 (attention)	VGGNet19 (simple)	VGGNet19 (attention)
Atelectasis	0.716	0.662	0.701	0.664	0.699
Cardiomegaly	0.807	0.669	0.769	0.658	0.772
Consolidation	0.708	0.653	0.664	0.661	0.667
Edema	0.835	0.790	0.793	0.803	0.813
Effusion	0.784	0.718	0.759	0.707	0.750
Emphysema	0.815	0.686	0.742	0.711	0.742
Fibrosis	0.769	0.731	0.738	0.720	0.730
Inltration	0.609	0.609	0.627	0.612	0.639
Mass	0.720	0.623	0.658	0.610	0.638
Nodule	0.671	0.661	0.689	0.660	0.673
Pleural Tickening	0.708	0.640	0.667	0.627	0.643
Pneumonia	0.633	0.587	0.612	0.601	0.606
Pneumothorax	0.806	0.708	0.740	0.704	0.737

In total, we tested 6 pre-trained models and each base model has simple and attention approach. Among these 12 architectures, the AUC values from 4 of them are listed in the Table 1 to compare with the literature paper (Wang et al). The ROC curves from these 4 models are plotted in Figure 6. As illustrated in the Table 1, for each architecture, the model trained with weights extracted from VGGNet16 outperformed the one with weights from Inception_v3 at overall level, i.e., for each label, the AUC value trained with VGGNet16 weights is higher than the one with Inception_v3 weights for the same architecture. This potentially indicates that for this specific task, transfer learning using VGGNet16 model benefits more than using Inception_v3 model. On the other hand, by comparing the different

architecture (i.e., simple vs attention) by leveraging the same pre-trained CNN model, it is found that for most scenarios (Table 1), the attention model performs better than the simple model, which is indicated by the higher AUC value. This is aligned with the expectation that attention model can help to learn where to look in order to find a certain pathology in the images, which will be helpful to classify the disease.

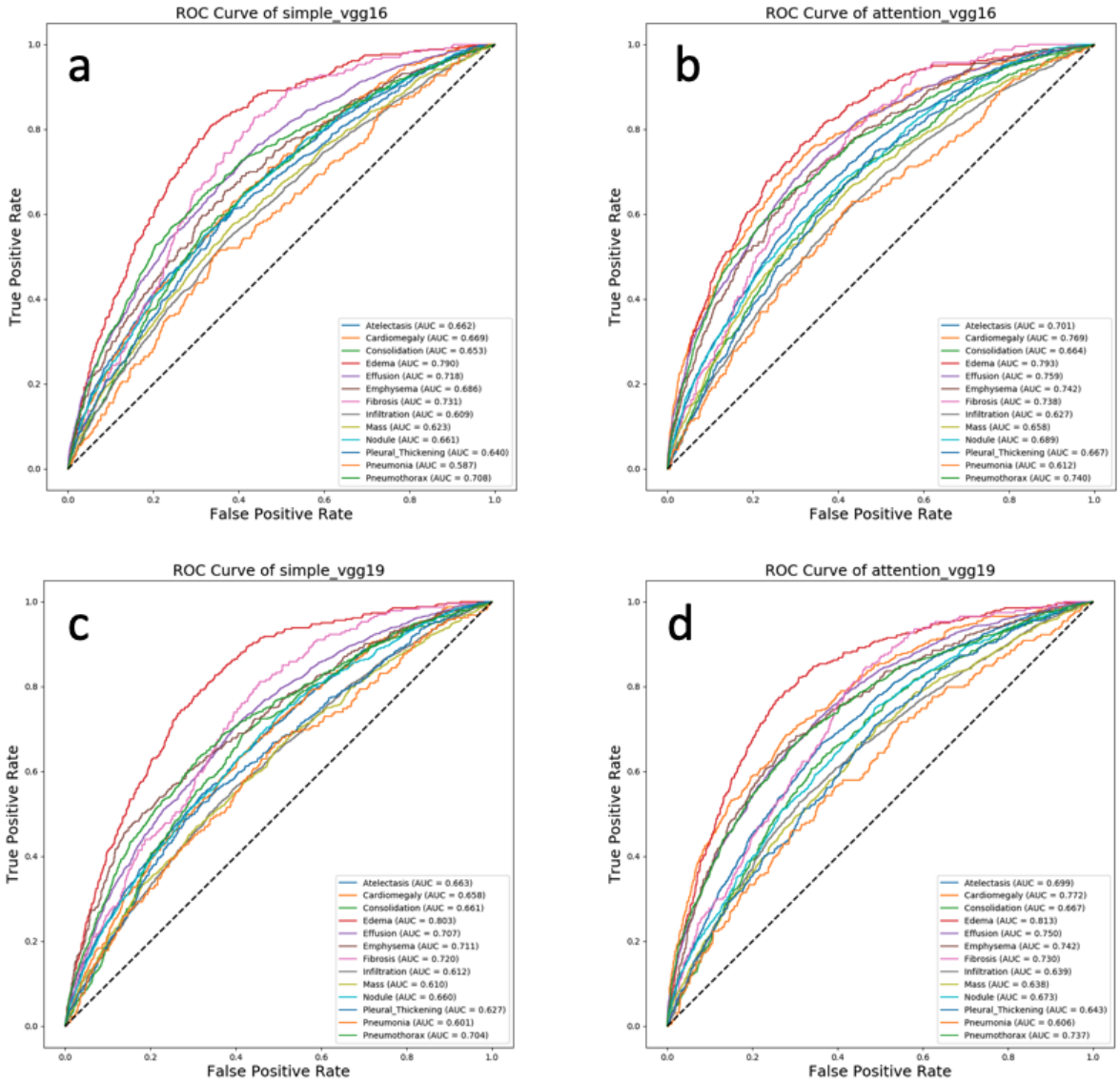

Figure 6: ROC curves of four different models. (a)VGGNet16 + simple model; (b)VGGNet16 + attention model; (c)InceptionV3 + simple model; (d)InceptionV3 + attention model.

6 Discussion

6.1 Model

In the project, we have applied several pretrained imagenet models that can be directly imported from Keras. These models include VGGNet, ResNet, MobileNet, Inception and NasNet. These models are all state-of-art deep learning image classifier that utilize convolutional neural networks. The MobileNet can make CNN faster and lighter, which

provides a great choice for imagine classification on devices with less power . The VGGNet16 and VGGNet19 were introduced by Simonyan and Zisserman at 2014[12]. The network is characterized by its 3×3 convolutional layers stacked on top of each other, which then handeled by maxpooling and two fully-connected layers. In contrast, the ResNet and Inception are not using the traditional sequential model. Instead, they all contains the micro-architecture that can be used for multi-level feature extraction[13,14]. In this project, we haven't seen the benefit in combining the inveption architecture with residual connections (inception_resnet_v2). Our experiment results showed that VGGNet outperforms all the other architectures by at least 20%. Therefore, the relative simple sequential model such as VGGNet is better suited for the multi-label classification task described in this project.

In addition, as illustrated in Figure 6, the attention architecture outperforms the simple one by utilizing the VGGNet pre-trained models, which in general aligns with the expectation. This is because the attention architecture reforms the originally extracted features by multiplying the weights trained through several convolutional and average pooling layers to take the information of other inputs into consideration, which indicates for each X-ray image, the model decides its final classification by considering other X-ray images seen in the training process as well.

6.2 Optimizer

We have applied two gradient descent optimizer in our project. The first one is the Keras default Adaptive Moment Estimation(Adam) optimizer and the other is the accumulated Adam gradient optimizer. Adam uses an adaptive learning for each parameters during the training. Adam not only stores an exponentially decaying average of the past squared gradients v_t like RMSprop but also keeps an exponentially decaying average of past gradients m_t, similar to momentum. In comparison, the accumulated gradients method means that the weights of the model are not updated after every single batch. Instead, the same weights are used for several batches, while the gradients are accumulated and then averaged for a single weight-update action. Our experiment results suggests that the accumulated gradients improved the model performance by 5 to 10% in terms of average AUC. And it is particularly helpful for the attention model compared to the simple model. This is probably because the attention model captures the subset of images with region of interests resulted in a much higher variance, where the accumulated gradient method can significantly lower this variance by using a larger batch to update.

7 Conclusion

There are two main challenges when doing this project. First, it took a lot of time in reading papers to understand the previous related work done by other researchers and to analyze NIH Chest X-ray data set tables and images. The other challenge is to tune the model hyper parameters. We are spending good amount of time in finding a good parameter for VGGNet16 models and analyzing ROC/AUC curve. This project was a great learning experience for the team to run in full cycle of a deep learning problem implementation. The current pipeline and method could also be used in different x-ray data set and further improvement will be focus more on getting additional clinical notes or more patient identity data to help model predict. We believe that the current work we have done is the cornerstone of the clinic operational AI in the future.

As a summary, the work for the entire project life-cycle was divided into two phases - phase1:data collection, processing and environment setup; phase2: model setup, optimization and evaluation. This final project is dedicated mainly for phase two because it is the most time-consuming portion. Following a guide of previous researches, our approach did take less time on training and optimizing models while achieved satisfactory performance in terms of AUC at the same time. To improve the model performance in future, we will focus on the efforts on these aspects 1) feature engineering, 2)pooling layer (max, min, average).

References

[1] B.V. Ginneken. Fifty years of computer analysis in chest imaging: rule-based, machine learning, deep learning. Radiol. Phys. Technol., 2017, pp. 23-32.

[2] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, M. P. Lungren, and A. Y. Ng. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. arXiv preprint arXiv:1711.05225, 2017.

[3] Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., & Summers, R. M. ChestX-Ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases. CVPR, 2017.

[4] Z. Li, C. Wang, M. Han, Y. Xue, W. Wei, L.-J. Li, and L. Fei-Fei. Thoracic disease identification and localization with limited supervision. CVPR, 2018.

[5] Z. Ge, D. Mahapatra, S. Sedai, R. Garnavi, and R. Chakravorty. Chest x-rays classification: A multi-label and fine-grained problem. CoRR, abs/1807.07247, 2018.

[6] H. Liu, L. Wang, Y. Nan, F. Jin, and J. Pu. Sdfn: Segmentation-based deep fusion network for thoracic disease classification in chest x-ray images. arXiv preprint arXiv:1810.12959, 2018.

[7] Q. Guan and Y. Huang. Multi-label chest x-ray image classification via category-wise residual attention learning. Pattern Recognition Letters, 2018.

[8] Jaiswal, A. K., Tiwari, P., Kumar, S., Gupta, D., Khanna, A., & Rodrigues, J. J. Identifying pneumonia in chest X-rays: A deep learning approach. Measurement, 145, 511-518, 2019.

[9] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. Ball, K. Shpanskaya, J. Seekins, D. A. Mong, S. S. Halabi, J. K. Sandberg, R. Jones, D. B. Larson, C. P. Langlotz, B. N. Patel, M. P. Lungren, and A. Y. Ng. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. arXiv preprint arXiv:1901.07031, 2019.

[10] NIH Clinical Center provides one of the largest publicly available chest x-ray datasets to scientific community. Retrieved from https://www.nih.gov/news-events/news-releases/nih-clinical-center-provides-one-largest-publicly-available-chest-x-ray-datasets-scientific-community, 2017.

[11] Villeneuve, J. AWS Essential Training for Developers - Create an EC2 instance. Retrieved from https://www.linkedin.com/learning/aws-essential-training-for-developers/create-an-ec2-instance?u=2163426.

[12] Simonyan, K. & Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. https://arxiv.org/abs/1409.1556

[13] He, K., Zhang, X., Ren, S. & Sun, J. Identity Mappings in Deep Residual Networks. Computer Vision – ECCV 2016 Lecture Notes in Computer Science, 630-645.

[14] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D. & Rabinovich, A. Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

8 Team Contribution

Jing Zhao and Jingyao Zhu recorded the video for the presentation. Xiaojie Du and Wenqin You set up the AWS EC2 instance and run the models. All team members have contributed to design the model, tune model hyper-parameters, write the report and prepare the PowerPoint slides together.