NIH Chest X-ray Project Report

Introduction

Computer Aided Diagnosis has been a well sought study field embedding practical value. Increasing access to medical images makes it possible for researchers to develop more accurate and reliable models. In this project, we applied transfer learning to train multiple DenseNet models on NIH Chest X-ray dataset and achieved competitive result.

NIH Chest X-ray dataset is comprised of 112,120 X-ray images with disease labels from 30,805 unique patients.There are fourteen disease labels and one “No Finding” label. With modification on these labels, we could build binary classification models and multi-label classification models. For binary classification task, we used one against all strategy, and the model could provide more accuracy result on detecting a specific disease. For multi-label classification task, a single model could be trained to predict the probability of all diseases. Both tasks are supervised machine learning tasks.

The imbalanced nature of the dataset is the major challenge in this project. There are 60,353 images with “No Finding” label (53.8% of all images) and 19,891 images contains “Infiltration” label (17.7% of all images) dominating the dataset. There are also significantly smaller classes such as “Hernia” (227 images / 0.2% of all images), “Pneumonia” (1,430 images / 1.2% of all images) and “Fibrosis” (1,686 images / 1.5% of all images). Due to the disparity of classes, models are largely inclined to dominated classes. In addition, there are 20808 images (18.5% of all images) with multiple labels further complicate the classification.

The benchmark model in this project is the CheXNet [1] . We aim to achieve a similar result as CheXNet with some improvements. Inspired by the CheXNet, we used pre-trained DenseNet model with some modification on fully connected layer. We experimented with different model such as VGG16, DenseNet with different number of layers and Inception V3. We chose DenseNet 169 at last since it yielded the best AUROC scores comparing to other pretrained model. In addition to modifying pre-trained model, we tried to alleviate data imbalance problem by resampling dataset and assigning weights in the training process.

CheXNet Evaluation

There are two main experiments presented in the stanford paper: a binary classification experience on detecting pneumonia and a multi-label classification experiment on detecting 14 pathology classes. The results are promising but there are a few underlying problems.

- First, lack of consideration in the imbalance nature of the dataset result in poor generalization ability. Despite high AUROC scores, the model could hardly classify under-represented classes without careful preprocessing. For example, the AUROC score for detecting ‘Hernia’ is 0.9164 and the score for detecting ‘Infiltration’ is 0.7345. Notice there are 109 times more

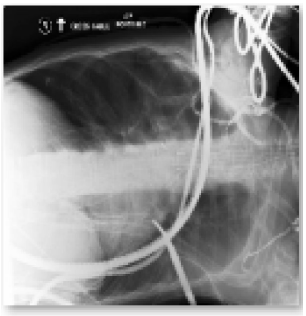
images for ‘Infiltration’ than ‘Hernia’, it is not likely that the model is more sensitive to the present of Hernia. There are very few ‘Hernia’ samples in the test dataset. Even though the model failed to detect ‘Hernia’, predicting all samples to be non-Hernia still yield high AUROC scores. For a more balanced dataset, the model would not have the generalization ability to product great classification results.

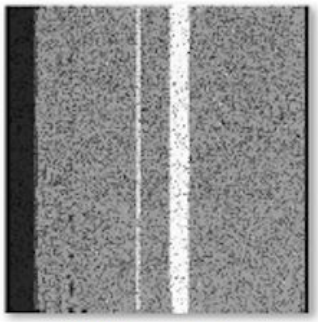
- Second, the model predicts the probability of pathology classes only, but could not predict the probability of ‘No Finding’ class.

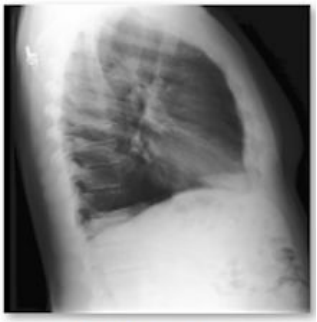
Preprocess

Preprocessing dataset is a crucial step before training models. Below are three major factors that improvement classification accuracy.

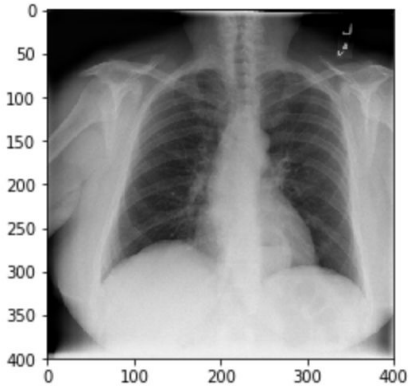
- First, there are some images with poor quality, rotation and different view point in the dataset. A blacklist of these images are provided on AzureChestXRay github[2]. Removing these images improved classification result. Column 5 and 6 shows that, on 10% dataset, removing blacklist yielded better AUROC scores.

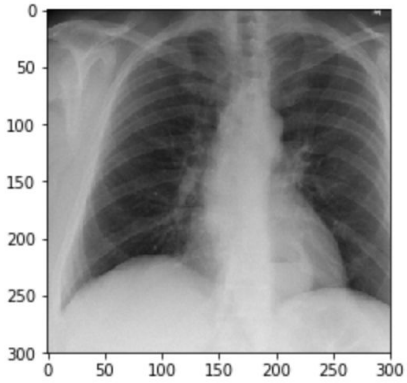

00011460_066.png


00007160_002.png


00007454_001.png

- Second, we cropped and save the center 75% images to better focus on the lung area. This will remove clavicles, top of vertebral column, medical wires, and some writing on original images. Column 6 and 7 shows cropping images or not yielded similar result. But we still experienced with both on full dataset to see the differences. Result will be shown later.





- Third, we loaded image as size 300*300 for better accuracy. Large image dimensions are expected to give more details despite longer training time. Column 3 and 4 shows that, on full dataset, training on images with size 300*300 yielded better result than training on images with size 224*224.

Below are my experiment results:

Label Index	Label	100%224NoCrop HasBlack	100%300NoCrop HasBlack	10%224 Crop Has Black	10%224 Crop Remove Black	10%224 NoCrop Remove Black	10%300 Crop Remove Black
Micro Ave		0.78	0.80	0.74	0.75	0.75	0.75
Macro Ave		0.66	0.70	0.60	0.62	0.63	0.62
0	Atelectasis	0.69	0.73	0.64	0.68	0.63	0.62
1	Cardiomegaly	0.68	0.71	0.54	0.57	0.60	0.58
2	Consolidation	0.66	0.68	0.63	0.62	0.64	0.59
3	Edema	0.76	0.77	0.63	0.60	0.68	0.64
4	Effusion	0.74	0.79	0.72	0.72	0.75	0.69
5	Emphysema	0.66	0.68	0.64	0.65	0.57	0.62
6	Fibrosis	0.62	0.67	0.54	0.62	0.60	0.63
7	Hernia	0.57	0.59	0.45	0.50	0.52	0.60
8	Infiltration	0.64	0.67	0.63	0.63	0.64	0.65
9	Mass	0.63	0.69	0.58	0.58	0.60	0.55
10	Nodule	0.63	0.66	0.57	0.59	0.59	0.59
11	Pleural_ Thickening	0.66	0.71	0.59	0.64	0.66	0.65
12	Pneumonia	0.63	0.64	0.53	0.62	0.58	0.56
13	Pneumothorax	0.73	0.79	0.69	0.69	0.68	0.72

Multi-Label Classification on 14 classes

In this section, we present the multi-label classification results on 14 pathology classes. We adopt binary encoding for labelling. The label for 'No Finding' class is [0,0,0,0,0,0,0,0,0,0,0,0,0,0]. Corresponding pathology classes has label 1.

- Fine-tuning DenseNet Model

  We aim to find a model that could outperform the CheXNet. There are many factors need to be considered when fine-tuning a model. In my experiment, I tried different **model type**, **fully connected layer** and **optimizer**. The evaluation matrix is loss value and AUROC

scores. We fixed other parameters when fine-tuning a certain parameter. For speed consideration, following tests are conducted on 10% dataset.

1. Model Selection

   We tested 4 models and they are VGG16, Inception V3, DenseNet 169 and DenseNet 121. DenseNet 169 has better performance than DenseNet 121 and was selected for training. Both DenseNet models outperform VGG16 and Inception V3.

Label Index	Classes / Labels	AUROC Scores for 14 Labels Classification Using VGG16	AUROC Scores for 14 Labels Classification Using InceptionV3	AUROC Scores for 14 Labels Classification Using DenseNet 169	AUROC Scores for 14 Labels Classification Using DenseNet 121
Loss Value		0.15654	0.15943	0.15421	0.15703
Micro Average		0.73	0.73	0.75	0.74
Macro Average		0.59	0.59	0.62	0.60
0	Atelectasis	0.56	0.56	0.61	0.64
1	Cardiomegaly	0.52	0.53	0.58	0.52
2	Consolidation	0.64	0.61	0.59	0.61
3	Edema	0.72	0.66	0.64	0.69
4	Effusion	0.67	0.64	0.69	0.69
5	Emphysema	0.56	0.57	0.62	0.65
6	Fibrosis	0.53	0.52	0.63	0.50
7	Hernia	0.47	0.55	0.60	0.40
8	Infiltration	0.67	0.63	0.65	0.65
9	Mass	0.64	0.52	0.55	0.55
10	Nodule	0.55	0.55	0.59	0.55
11	Pleural_ Thickening	0.56	0.62	0.65	0.62
12	Pneumonia	0.61	0.63	0.56	0.58

13	Pneumothorax	0.65	0.64	0.72	0.70

2. Fully Connected Layer

CheXNet used a single Sigmoid layer. I added two ReLU layers to increase training speed and potentially alleviate vanishing gradient problem. Experiment shown ReLU layer would result in smallest loss value than using a single Sigmoid layer.

Label Index	Label	Dense(14, activation='sigmoid')	Dense(256, activation='relu')+ Dense(50, activation='relu')+ Dense(14, activation='sigmoid')	Dropout(0.2)+ Dense(256, activation='relu')+ Dropout(0.2)+ Dense(50, activation='relu')+ Dropout(0.2)+ Dense(14, activation='sigmoid')
Loss value		0.15783	0.15049	0.15421
Micro Ave		0.76	0.76	0.75
Macro Ave		0.65	0.64	0.63
0	Atelectasis	0.64	0.66	0.63
1	Cardiomegaly	0.73	0.65	0.60
2	Consolidation	0.62	0.64	0.64
3	Edema	0.71	0.70	0.68
4	Effusion	0.73	0.73	0.75
5	Emphysema	0.69	0.59	0.57
6	Fibrosis	0.66	0.66	0.60
7	Hernia	0.56	0.55	0.52
8	Infiltration	0.60	0.64	0.64
9	Mass	0.64	0.60	0.60
10	Nodule	0.61	0.63	0.59
11	Pleural_ Thickening	0.66	0.65	0.66

12	Pneumonia	0.56	0.57	0.58
13	Pneumothorax	0.68	0.64	0.68

3. Optimizer

SGD has slower convergence rate than Adam, but it could potentially jump to local minima and yield better results. Comparing to Adam, SGD could compute a smaller loss value and gave better generalization ability [3].

In the experiment, I set epochs to 20. Using Adam took half of the time as using SGD. Loss values stopped improving after 13 epochs using Adam. However, loss value still improving after 20 epochs using SGD. Therefore, I used Adam to obtain initial weights. Then I loaded the weight and applied SGD to for further training. The combination of two optimizer yielded smaller loss value.

Label Index	Label	Adam	SGD
Micro Average		0.76	0.74
Macro Average		0.62	0.60
0	Atelectasis	0.68	0.64
1	Cardiomegaly	0.54	0.54
2	Consolidation	0.62	0.63
3	Edema	0.70	0.63
4	Effusion	0.72	0.72
5	Emphysema	0.63	0.64
6	Fibrosis	0.53	0.54
7	Hernia	0.40	0.45
8	Infiltration	0.67	0.63
9	Mass	0.61	0.58
10	Nodule	0.64	0.57
11	Pleural_Thickening	0.63	0.59

12	Pneumonia	0.59	0.53
13	Pneumothorax	0.67	0.69

After the experiment, we chose the model below:

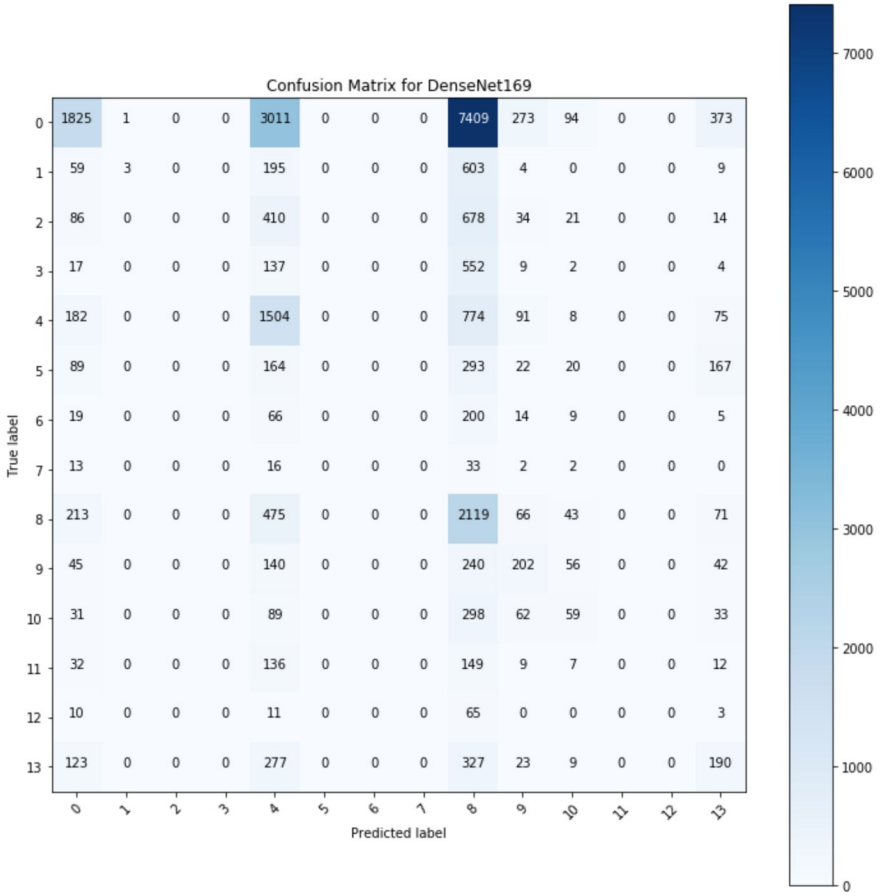
Model	Loss function + activation	Optimizer	Fully Connected Layer	Weight
DenseNet 169	binary_crossentropy + sigmoid	Adam + SGD	Dropout(0.1)+ Dense(256, activation='relu')+ Dropout(0.1)+ Dense(50, activation='relu')+ Dropout(0.1)+ Dense(14, activation='sigmoid')	ImageNet

Below are the training results comparing to CheXNet:

Label Index	Label	CheXNet Result	CheXNet Setting	NoCrop HasBlack	Crop NoBlack
0	Atelectasis	0.8094	0.54	0.69	0.73
1	Cardiomegaly	0.9248	0.55	0.68	0.72
2	Consolidation	0.7901	0.55	0.66	0.70
3	Edema	0.8878	0.59	0.76	0.80
4	Effusion	0.8638	0.57	0.74	0.80
5	Emphysema	0.9371	0.54	0.66	0.80
6	Fibrosis	0.8047	0.59	0.62	0.70
7	Hernia	0.9164	0.48	0.57	0.64
8	Infiltration	0.7345	0.58	0.64	0.69
9	Mass	0.8676	0.52	0.63	0.78
10	Nodule	0.7802	0.54	0.63	0.73
11	Pleural_Thickening	0.8062	0.57	0.66	0.72

12	Pneumonia	0.7680	0.50	0.63	0.66
13	Pneumothorax	0.8887	0.54	0.73	0.84

Using the DenseNet 121 model with a single Sigmoid layer and Adam gave pretty bad result after the actual training. However, our way to preprocess the dataset and select the model indeed improved the classification, despite we did not achieve the high score as CheXNet.



The confusion matrix showed that under-represented classes were not correctly classified. We tried to oversample these classes to see if any improvement would happen.

- Oversample under-represented classes

We oversampled class 1,2,3,5,6,7,11 and 12 since they were not detected in the confusion matrix. We found that AUROC scores were boosted as we enlarged the samples. For ‘Hernia’, as the samples sizes increased 17 times, AUROC scores increase significantly. For under-represent classes like ‘Hernia’, larger samples enabled will improve model’s classification ability quickly. But for dominated classes, increasing size would improve the classification ability by a little . Also, noticed test dataset was imbalanced as well. Therefore,

improvement for under-represent classes will be more obviously than improvement made on dominated classes.

Label Index	Label	Total occurence of a label in original dataset	Total occurence of a label after balanced	Best model before balanced	Best model After balanced
0	Atelectasis	8279	same	0.73	0.72
1	Cardiomegaly	1707	3500	0.72	0.80
2	Consolidation	2852	3500	0.70	0.71
3	Edema	1378	3500	0.80	0.82
4	Effusion	8659	same	0.80	0.80
5	Emphysema	1423	3500	0.80	0.82
6	Fibrosis	1251	3500	0.70	0.78
7	Hernia	151	2500	0.64	0.84
8	Infiltration	13779	same	0.69	0.69
9	Mass	4032	same	0.78	0.78
10	Nodule	4708	same	0.73	0.73
11	Pleural_ Thickening	2242	3500	0.72	0.72
12	Pneumonia	876	3500	0.66	0.70
13	Pneumothorax	2637	same	0.84	0.85
	No Finding	50492	30,000	N/A	N/A

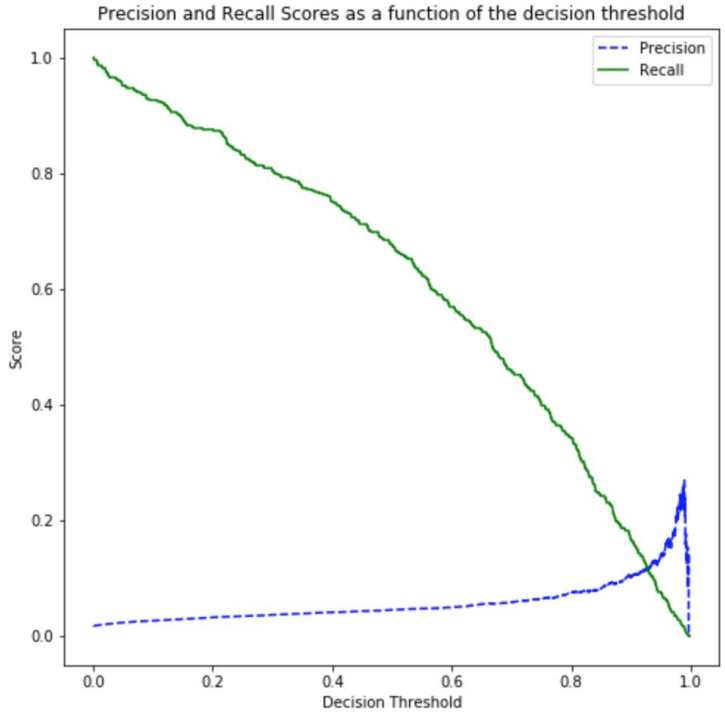
We did not test adding more samples for resource concern. Increasing the sample size slow down the training process a lot. However, we should be able to achieve a similar result as CheXNet if we oversample each class.

Binary Classification

Binary classification could achieve good result on detecting a specific label X. We fixed a label X as the class 0 and the rest of labels as class 1, then balance the dataset accordingly. In my experiment, I tried to detect ‘Fibrosis’. There are 1241 samples contained the label ‘Fibrosis’ in the train dataset and 431 samples contained the label ‘Fibrosis’ in the test dataset. We expected binary classification model could classify single class better. In that case, we could train 15 binary classification models to tackle multi-label classification problem.

- Detect ‘Fibrosis’ without oversampling

  We filtered out all ‘Fibrosis’ samples and mixed them with equal size of random samples from other classes as the train dataset. The model was tested on the same dataset that we used for multi-label classification. From the Precision-Recall vs. Threshold graph, we found the threshold for getting best precision and recall scores is about 0.91.



With the threshold 0.91, we got the confusion matrix:

	Pred_neg	Pred_pos
Neg	23781	1043

Pos	334	100

The confusion matrix yield the following evaluation metrics:

Accuracy	Precision	Recall	F1-score	AUROC
0.9640	0.1073	0.1498	0.125	0.79

Despite the AUROC score and the accuracy are pretty high, both the precision and the recall are low. This means the model did not classify ‘Fibrosis’ well. The lack of ‘Fibrosis’ samples boosted the high accuracy as the model inclined to classify images to non-Fibrosis. A thing worse notice was the AUROC score is very similar to CheXNet. This suggested that CheXNet could not classify under-represented classes neither.

- Detect ‘Fibrosis’ with oversampling

  One possible way to alleviate the influence of dataset imbalance is oversample the ‘Fibrosis’. In the experiment, we oversampled ‘Fibrosis’ samples to size 4,000, and then mixed them with 5,000 random samples from other classes as the train dataset. We test the model on the test dataset. However, there were no improvement in any evaluation metrics. Other size of sampling might make some improvement though.

Multi-Label Classification on 15 classes

In this section, we present the multi-label classification model to classify data into 15 classes (14 pathology classes and 1 'No Finding' class). Our goal is to identify the probability of the 'No Finding' class in addition to predicting the probabilities of 14 pathology classes. The labelling strategy is different from the strategy mentioned in the previous section. For an image in 'No Finding' class, our way of labelling is [0,0,0,0,0,0,0,0,0,1,0,0,0,0], where index 10 indicate 'No Finding'. In our experiment, We trained model on original dataset and balanced dataset. Results are presented below.

- Fine-tuning DenseNet Model

  Again, we experimented with different model type, fully connected layer, optimizer, weights and loss function. The evaluation matrix is loss value. We fixed other parameters when fine-tuning a certain parameter. For speed consideration, following models are conducted on 10% dataset.

  1. Model Selection

Model Type	Loss Value	Optimizer	Fully connected layers	Epochs	Batch size	weights
VGG16	0.20335	SGD	3 densely connected NN layer.	10	32	ImageNet
DenseNet 121	0.19886	SGD	3 densely connected NN layer.	20	32	ImageNet
DenseNet 169	0.19886	SGD	3 densely connected NN layer.	10	32	ImageNet
DenseNet 201	0.20009	SGD	3 densely connected NN layer.	10	32	ImageNet

DenseNet 169 has slightly better performance comparing to other model.

  2. Fully Connected Layer

Fully connected layers	Loss Value	Optimizer	Model	Epochs	Batch size	weights
Dense(256, activation='relu')+ Dense(50, activation='relu')+ Dense(15, activation='sigmoid')	0.20194	SGD	DenseNet 169	5	32	ImageNet
Dense(256, activation='relu')+ Dense(60, activation='relu')+ Dense(15, activation='sigmoid')	0.20102	SGD	DenseNet 169	5	32	ImageNet
Dense(15, activation='sigmoid')	0.20497	SGD	DenseNet 169	5	32	ImageNet

Added 3 dense layers to the pre-trained model/feature extractor for re-training would give pretty good result.

3. Assigning weights

weights	Loss Value	Optimizer	Model	Epochs	Batch size	Fully Connected layers
ImageNet	0.20009	SGD	DenseNet 201	10	32	3 densely connected NN layer
None	0.21176	SGD	DenseNet 201	10	32	3 densely connected NN layer

Training on ImageNet gave a better result than training from scratch.

4. Optimizer

Optimizer	Loss Value	weight	Model	Epochss	Batch size	Fully Connected layers
SGD	0.20101	ImageNet	DenseNet 121	10	32	3 densely connected NN layer
Adam	0.21226	ImageNet	DenseNet 121	10	32	3 densely connected

						NN layer

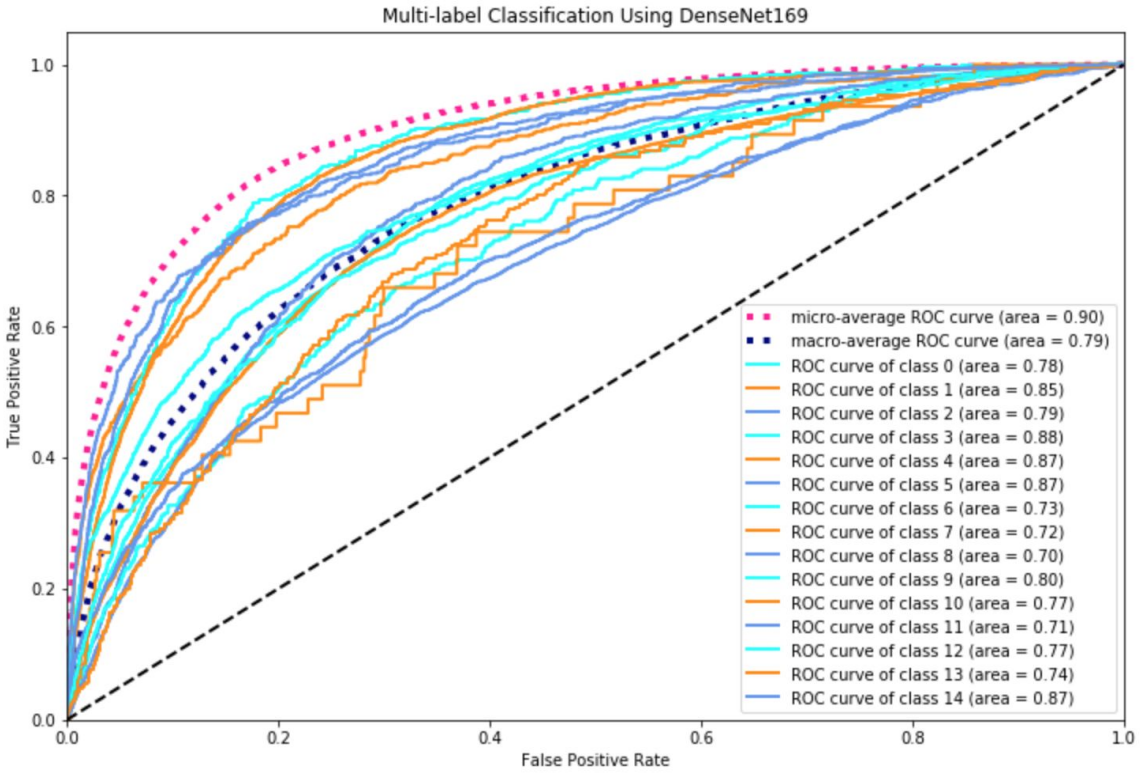
Using SGD as optimizer is much more time-consuming than using Adam, but would yield a better classification result.

- Classification result for benchmark model

After the experiment, we chose the model below as the benchmark.

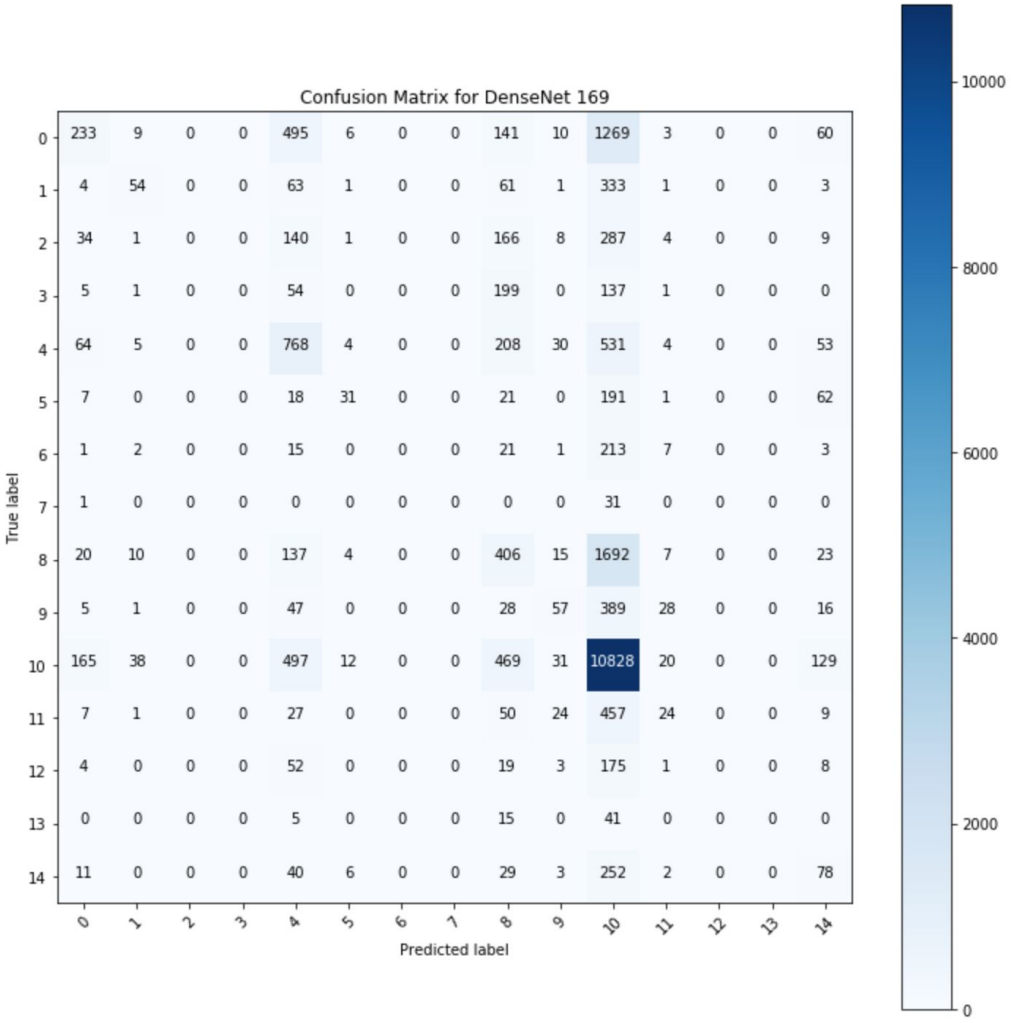
Model	Loss Value	Loss function + activation	Optimizer	Weight	Epochs	Batch size
DenseNet 121	0.18450	binary_crossentropy + sigmoid	SGD	ImageNet	50	16

We randomized all data and split dataset to 76200 train data, 9478 validation data and 25250 test data. The classification result (AUROC scores) are in the following graph:



Note: indexes 0 to 14 correspond to 15 classes in the following order: 'Atelectasis', 'Cardiomegaly', 'Consolidation', 'Edema', 'Effusion', 'Emphysema', 'Fibrosis', 'Hernia', 'Infiltration', 'Mass', 'No Finding', 'Nodule', 'Pleural_Thickening', 'Pneumonia', 'Pneumothorax'.

Despite relatively high AUROC scores, the confusion matrix shows that the model classified most images to 4 dominant classes (‘Atelectasis’, 'Effusion', 'Infiltration', 'No Finding'). Even worse, images from 6 under-represented classes were completely ignored (all zero), which reflected the model’s poor performance.



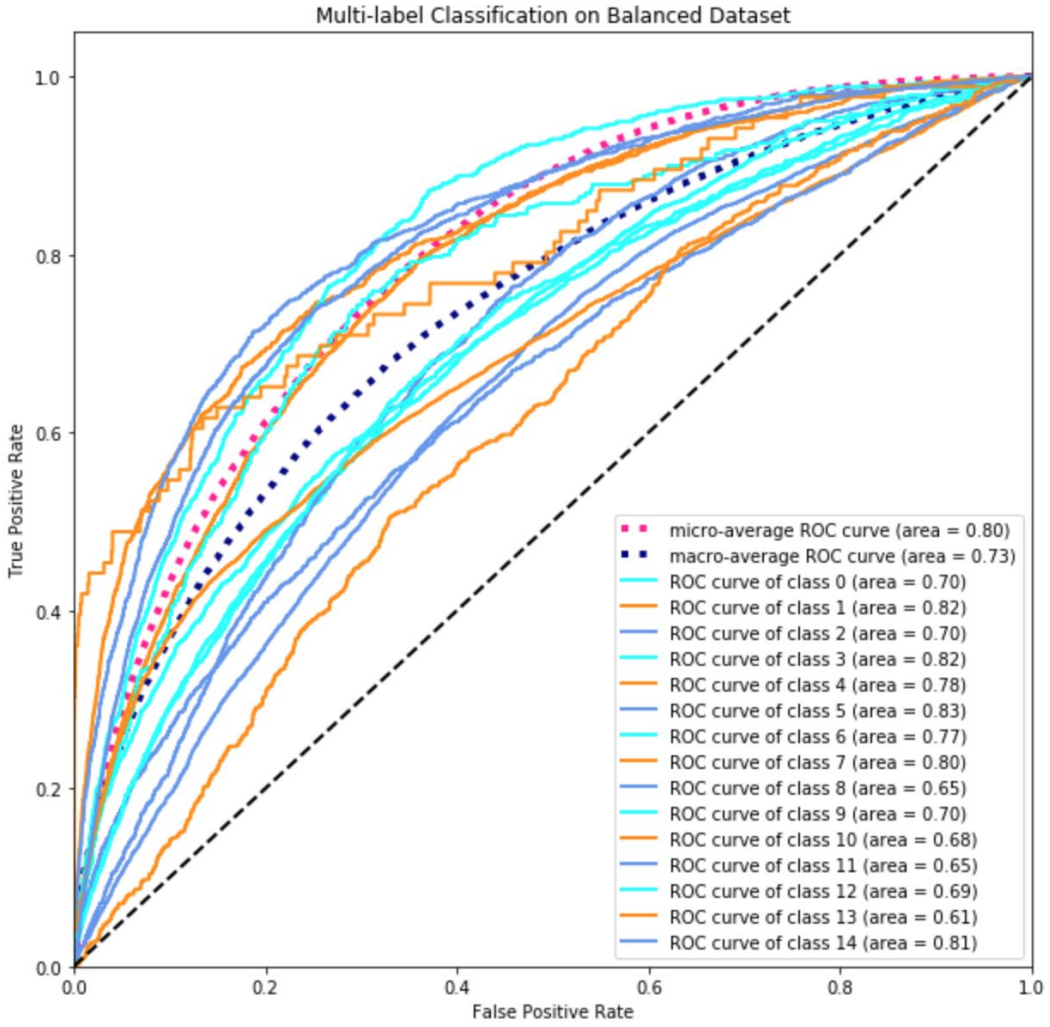
- Weighted DenseNet Model

There are two ways to better classify under-represented classes on imbalanced dataset. The first approach is to resample dataset by oversampling under-represented classes and undersampling over-represented classes. The second approach is to assign the weights to the loss function.

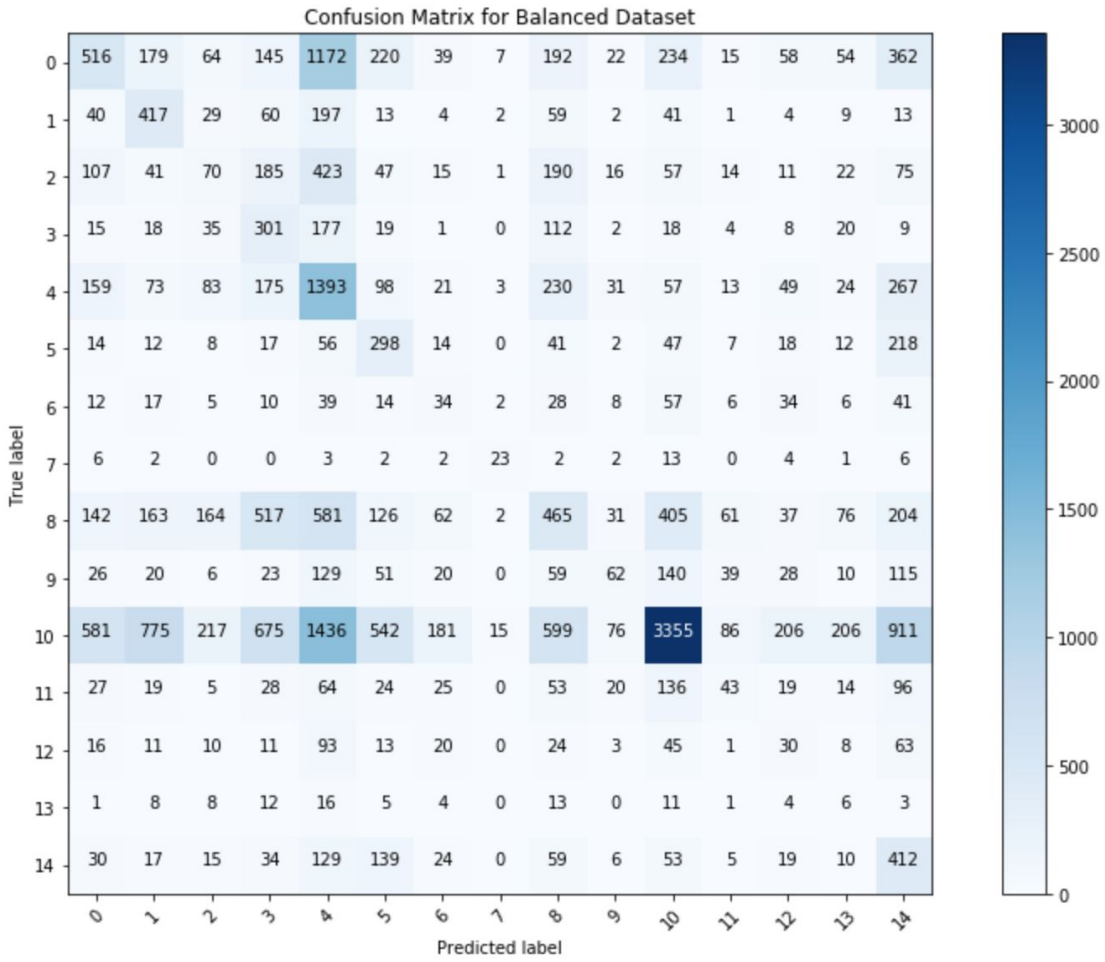
1. Resampling dataset

Resampling dataset alleviated the influence of imbalanced dataset from the source. It allowed the model pay more attention to samples in small classes. However, a difficult task was to choose the number of samples for each class manually. In my case, I set the number of each label as the following. This setting roughly brought up under-represented classes to the same scale as large classes, but still kept the large classes in dominating positions. The training data size drops from 86512 to 69495:

Label Index	Label	Total occurence of a label in original dataset	Total occurence of a unique label in original dataset	Total occurence of a label after balanced
0	Atelectasis	8279	3414	7712
1	Cardiomegaly	1707	777	5304
2	Consolidation	2852	829	5849
3	Edema	1378	397	5366
4	Effusion	8659	2788	8627
5	Emphysema	1423	587	5145
6	Fibrosis	1251	551	4986
7	Hernia	151	65	4118
8	Infiltration	13779	7326	9934
9	Mass	4032	1696	4941
10	No Finding	50492	50492	10000
11	Nodule	4708	2248	5677
12	Pleural_Thickening	2242	817	5993
13	Pneumonia	876	234	4889
14	Pneumothorax	2637	1241	5912



Applied the new model trained on balanced dataset to the same test dataset. We could see the AUROC for under-represent classes (‘Fibrosis’, ‘Hernia’) increased. As a trade-off, AUROC scores for other classes decreased. This indicated that balanced the dataset from the source helped with detecting under-represent classes. However, the overall classification ability did not improved. We needed to experiment with different resampling sizes.



The confusion matrix showed that previously ignored images from 6 under-represented classes were detected, but there are significantly more false prediction for majority class(‘No Finding’, ‘Infiltration’, ‘Edema’).

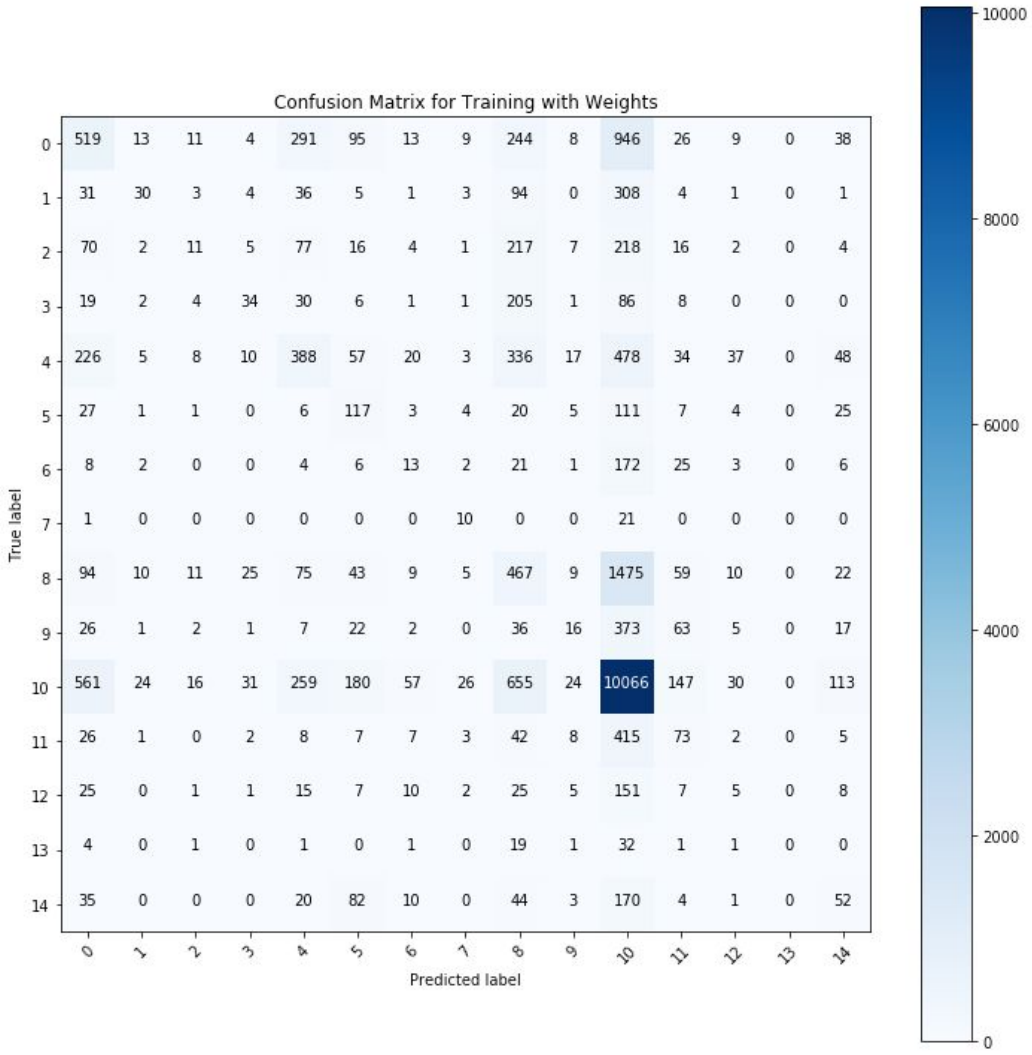
Resampling from the source is a good practice to deal with imbalanced dataset since we actually generated more data. However, it was hard to find a balance of the size of all classes. In addition, it was expensive to regenerate the whole dataset and train on it.

2. Assign weights to loss function

According to the Keras document, we can assign class weights to weight the loss function during training, which tells the model to “pay more attention” to samples from an under-represented class. Despite resampling from the source is a better practice, assigning weights could yield a similar result without too much trouble.

By experimenting with different weights, we found one setting that suited the dataset. The weights are:

Atelectasis: 1.0	Cardiomegaly: 2.0	Consolidation: 2.0	Edema: 2.2	Effusion: 0.3
Emphysema: 3.0	Fibrosis: 7.0	Pneumothorax: 0.5	Infiltration: 0.5	Mass: 1.6
No Finding: 0.5	Pneumonia: 12.0	Pleural_Thickening: 5.0	Nodule: 3.0	Hernia: 14.0



From the confusion matrix above, we could see most predictions are along the diagonal, which is a good sign. Note it's fine to have some predictions not on diagonal for multi-label classification. However, assigning weight would only fit model to specific dataset as it learned distribution of all classes. Therefore, assigning weight is not the best practice.

Multi-Label Classification on 14 classes

In this section, we present the multi-label classification results on 14 pathology classes only. To remove the influence of the 'No Finding' class', we dropped samples in the 'No Finding' class for both train and test dataset. Then splitted train dataset to train data and validation data. There are 32400 train data, 3620 validation data and 15731 test data (with proportion about 9:1:4).

AUROC scores for 14 labels classification model were lower than AUROC scores for 15 labels classification model, except for 'Hernia' and 'Fibrosis', the smallest class and third smallest class. This finding showed the 14 labels classification model was more sensitive to small classes. AUROC scores for the 15 labels classification model were boosted by the 'No Finding' class.

Label Index	Classes / Labels	AUROC scores for 14 labels classification	AUROC scores for 15 labels classification
0	Atelectasis	0.72	0.78
1	Cardiomegaly	0.84	0.85
2	Consolidation	0.65	0.79
3	Edema	0.80	0.88
4	Effusion	0.78	0.87
5	Emphysema	0.85	0.87
6	Fibrosis	0.77	0.73
7	Hernia	0.82	0.72
8	Infiltration	0.67	0.70
9	Mass	0.75	0.80
10	Nodule	0.71	0.71
11	Pleural_Thickening	0.71	0.77
12	Pneumonia	0.60	0.74
13	Pneumothorax	0.83	0.87

Limitation

For multi-label classification, the model could not classify under-represented classes due to lack of samples. The precision and recall are both very low. In addition, the model has limited generalization ability. For a more balanced dataset, the model should not yield satisfying result. Furthermore, there are 18.5% of images with multiple labels, but we ignored the label dependency in the training process.

Conclusion

Preprocessing is an important step before actually training the model. Removing low quality images, cropping images to focus on specific area and loading images as larger size will improve classification. In addition, oversampling data in small classes could potentially bring up the average AUROC scores for multi-label classification. However, the trade off is lower AUROC scores for dominated classes. For binary classification, oversampling makes limited improvement.

Adding ReLU layers will classify the dataset better. In general, SGD could find a smaller loss value than Adam despite slower convergence. Among all tested pretrained models, DenseNet 169 yield best result. Using our model yield better classification result than using CheXNet's setting. Our model did not outperform CheXNet, but building 15 binary classification models could achieve similar results.

Although CheXNet achieved high AUROC scores, the model could not identify under-represent class well. The high scores are boosted by the imbalanced dataset. To improve the classification ability, adding more data in the training step is crucial.

REFERENCES

[1] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya *et al.*, “Chexnet: Radiologist- level pneumonia detection on chest x-rays with deep learning,” *arXiv preprint arXiv:1711.05225*, 2017.

[2] AzureChestXRay Github: Blacklist Images

[3] Wilson, A. C, Roelofs, R., Stern, M., Srebro, N., and Recht, B. The marginal value of adaptive gradient methods in machine learning. *arXiv preprint arXiv:1705.08292*, 2017.

[4] A Github with binary classification example