Multi-Label Classification for Chest X-Rays

Andrea Dunn Beltran (PID: 730296640), Ibraheem Alsaghier (PID: 730354497)

December 2022

1 Introduction

Lung diseases are one of the leading causes of death and hospitalization in the world. Lung cancer causes the most cancer deaths for both men and women in the United States. Medical imaging is critical for the detection of lung diseases like cancer. Chest X-rays are the most common form of imaging used to detect lung diseases. These diseases include tuberculosis, lung cancer, pneumonia, and chronic obstructive pulmonary disease (COPD). The issue is that the number of radiologists does not meet the demand from patients. This is especially true for developing regions of the world. Computer-aided diagnosis and detection have the potential to help ameliorate this problem[7][8]. This issue is even more important due to human error in diagnostics. A review on error in the field of radiology showed some concerning results. Radiologists had an interpretive error rate of between 3 to 5 percent. Interpretive error rate relates to whether other experts view a result in the same way. When it came to retrospective error they had an error of around 30 percent[1]. It is important to have high fidelity. It is especially important in this setting. A false positive may be very expensive; however, a false negative could cost someone their life.

Additionally, applying machine learning techniques to the medical domain comes with its own challenges. Medical imaging data sets are unique for several reasons. First of all, they are typically small because they are time-consuming and expensive to accumulate. Additionally, these data sets are very unbalanced since there are more 'healthy' patients, and therefore more images showing 'healthy' looking lungs. Sick patients whose images would show some pathology are significantly outnumbered. It is common for a patient to show signs of many diseases at once, so we must use a multi-label classifier to identify the pathologies. Given that the data is unbalanced, we will implement a multi-label classifier and explore how different thresholds affect the model's accuracy.

2 Related Work

The increasing prominence of deep learning in 2013 encouraged increased research into image processing. The new methods it spawned caused significant developments in computer vision. Convolutional neural networks specifically have been critical in this field. One of their applications is the analysis of medical imaging. Convolutional neural networks have been used in the segmenting of organs and the detection of abnormalities in them and diseases. They have also shown promising results in the detection of breast cancer, lung cancer, and Alzheimer's[6].

Prior to 2017, the use of deep learning for lung radiology was limited in large part due to the absence of large labeled data sets. Large labeled data sets are critical to producing high-performance machine learning models. In 2017, the NIH established a large-scale chest X-ray image database. It included tens of thousands of images [9]. Other large-scale lung X-ray data sources such as CheXpert and MIMIC-CXR[7] have also made the training of deep learning models easier for lung medical imaging processing. These large-scale data sources were used to demonstrate the efficacy of deep learning in detecting chest abnormalities[7][10] and segmentation of the lungs [8]. This will be critical to help in the use of deep learning networks for the diagnosis and detection of lung diseases.

3 Data Set

We used the National Institutes of Health Chest X-Ray data set to test our convolutional neural network. The data set includes **112,120** X-Ray images with disease labels coming from **30,805** different patients. The labels for the images were created by using natural language processing on the radiology reports.

There are a total of 14 possible disease labels, however, image can have single or multiple labels. The data set contains over 60 thousand images labelled 'No Finding' and over 50 thousand images with at least two disease labels. Additionally, not all diseases are represented equally, with 'Hernia' only holding 0.2% of the total labels and 'Infiltration' holding 14.1%. More information on the label distributions can be found in *Figure 1*.

3.1 Preliminary Processing

We downloaded the data set from Kaggle[4] which came with all 112,120 images divided into 12 folders, a csv file containing all thee metadata, and two text files indicating the testing and training sets. The images were separated into training images and testing images folders based on the provided text file which indicate a 70/30 split.

Most images were single channel images however there was a small subset (less than 0.5%) of images that were saved as 4 channel images, These images were ignored. Additionally, the meta data was separated into smaller csv files corresponding to the designated split.

4 Model

In this section, we describe the the basic model used and our evaluation metrics.

4.1 ResNet

The ResNet model was used to address the issue of degradation in very deep models. The ResNet model has a distinct structure from a plain convolutional neural network. It is composed of a stack of residual blocks. Each residual block is composed of two or three 3x3 convolutional layers. The residual block also has skip connections. These skip connections add the input to the residual layer to its output[3].

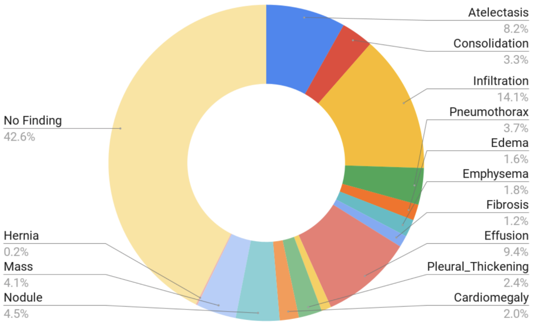

Figure 1: This pie chart shows the proportion for the occurrences of each label type.

4.2 Evaluation Metrics

The ROC (Receiver Operating Characteristics) curve plots the true positive rate against the false positive rate using various thresholds. The AUC (Area Under Roc) is the area under the ROC curve. It is frequently used to assess the performance of machine learning models. The AUC provides some advantages over other measures of performance such as accuracy. It works well with unbalanced data[5].

The F1 score is another measure of performance. The F1 score like the AUC works well with imbalanced data. It is composed by two variables. They are precision and recall. The formula for precision is shown below:

$$\frac{TruePositive}{TruePositive + FalsePositive}$$

The formula for recall is shown below:

$$\frac{TruePositive}{TruePositive + FalseNegatives}$$

The formula for the F1 score is shown below:[2]

$$2 * \frac{Precision * Recall}{Precision + Recall}$$

Given the importance of high fidelity, we look at both the AUC and F1 score to get a better picture of the model's ability.

5 Experiments and Results

We used the model described above with some modifications and set hyperparameters. To start, we had to create custom data loaders to parse through the pathology labels and format the data. This was a surprisingly difficult step. Additionally, we made slight modifications to the convolutions in the ResNet model. We define the first convolution to take in a one-channel image with a kernel size=$(7, 7)$, stride=$(2, 2)$, padding=$(3, 3)$, and no bias. Additionally, the final fully connected layer is linear with 2048 in features, and 14 (the total number of pathologies) out features and bias. Additionally, we use the BCEWithLogitsLoss as our criterion and a sigmoid function is applied to the predicted scores before being subjected to a threshold.

All the experiments were run for 50 epochs, with a learning rate of 0.01, and a batch size of 64. We used a cosine learning rate scheduler and the Adam optimization algorithm for training.

We ran three experiments over the unbalanced data. For each experiment, we trained the model and calculated the AUC and F1 Score over the testing set after each epoch. During evaluation, the model predicts the labels for each image and returns the scores for each label. A sigmoid is applied to the scores and then a threshold is applied to create an array of binary indicators for each label. This label is then used to calculate the AUC and F1 Scores. The parameter we changed to predict the output labels was the threshold.

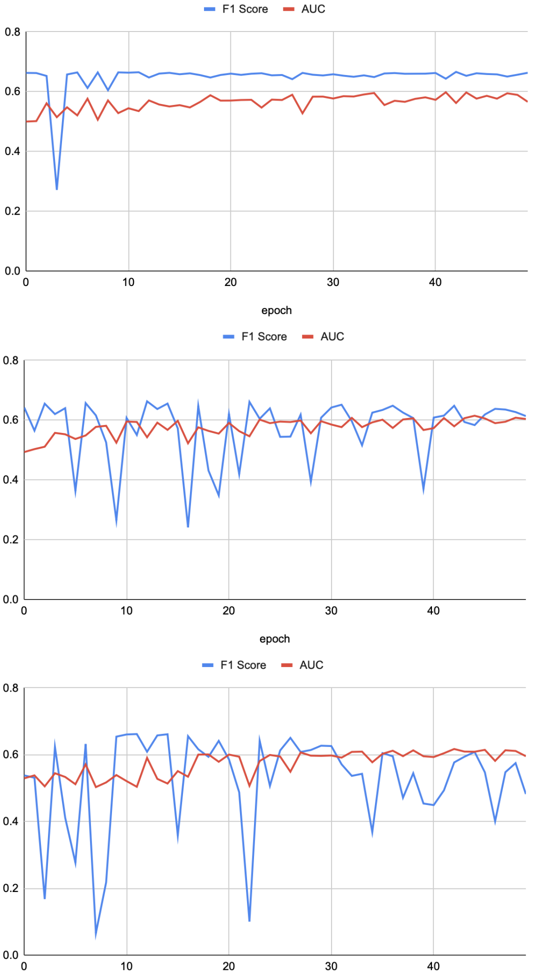

Figure 2: These graphs show the evaluation metrics of the results for each of the three experiments. From top to bottom, the thresholds are 0.4, 0.5, and 0.6.

The resulting AUC and F1 Scores for each of the 3 experiments are shown in *Figure 2.* As seen in the plots, as the threshold increases, the AUC tends to increase; however, the F1 score seems to tend downwards as the number of epochs increases and the F1 score becomes more volatile. In all three plots, the AUC is much more consistent and less dynamic than the F1 score. The AUC scores across the board are low (final AUCs around 0.6for each experiment) given that an AUC of 0.5 is what we would expect from a random guessing model. However, it is important to keep in mind that since this is a multi-label task for labels that do not have uniform probability distributions. The F1 scores are not much better but seem to get worse the larger the threshold.

Intuitively, it makes sense that F1 scores would be higher when the threshold is lower since F1 scores more accurately reflect the imbalanced data. Some labels have very low representation in the data set which may reflect as lower scores than the labels with high representation. At a lower threshold, these labels may be detected while these low scores would be ignored at a higher threshold.

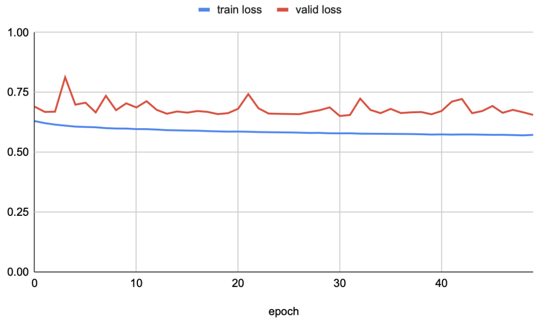

Figure 3: These graphs show the training and validation loss for the experiment with the threshold set at 0.4 over the 50 epochs

During testing, we also kept track of the training and validation loss. As seen in *Figure 3*, both the training and validation loss decrease slowly but remain high as the number of epochs increases. This indicates that the models should train over more epochs until the validation loss reaches its minimum. We did not run these experiments for more than 50 epochs, due to the lack of resources: each experiment took over 5.5 hours to run on one NVidia RTX 3070 GPU.

6 Limitations and Future Work

Given that the evaluation metrics were low, there may still need to be work done tuning the hyper parameters and running over more epochs. Additionally, instead of applying a general threshold across all labels, it may be more accurate to use different thresholds for each possible label. It would most likely improve the accuracy to impose the constraint that if the most likely label is 'No Finding" then no other label can be returned.

Additionally, the data set itself has limitations. Although the data set is large, there are some obvious inconsistencies, such as the channel issue, but there may have been some less obvious inconsistencies since the images were collected from many hospitals over many years. The other issue is not with the data quality itself, but with which groups data is collected from. In the current US health system, medical imaging is expensive and inaccessible; this means the data set may not be the most accurate representation of the general public.

7 Conclusion

From our experiments, we learned that for the case of a model trained on imbalanced data over a relatively small amount of epochs, a lower threshold corresponds to a somewhat low AUC but a higher and more stable F1 score. Higher thresholds correspond to a higher AUC but a lower and more unstable F1 scores. However, due to the size of the data set and the the small number of epochs, our models are not the most accurate or robust. This a good starting point and we were able to implement some of the more challenging aspects off this model; however, there are still some hyper parameters that must be tuned as well as a customized approach to setting thresholds.

References

[1] Adrian Brady. "Error and discrepancy in radiology: inevitable or avoidable?" In: *Insights into Imaging* 8 (Dec. 2016). DOI: 10.1007/s13244-016-0534-1.

[2] Allan Hanbury and Abdelaziz Taha. "Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool." In: *BMC Med Imaging* (2015). DOI: 10.1186/s12880-015-0068-x.

[3] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* June 2016.

[4] National Institutes of Health. *kaggle datasets download -d nih-chest-xrays/data.* URL: https://www.kaggle.com/datasets/nih-chest-xrays/data.

[5] Jin Huang and C.X. Ling. "Using AUC and accuracy in evaluating learning algorithms". In: *IEEE Transactions on Knowledge and Data Engineering* 17.3 (2005), pp. 299–310. DOI: 10.1109/TKDE.2005.50.

[6] Jun Lee June-Goo et al. "Deep Learning in Medical Imaging: General Overview". In: *kjr* 18.4 (2017), pp. 570–584. DOI: 10.3348/kjr.2017.18.4.570. eprint: http://www.e-sciencecentral.org/articles/?scid=1027354. URL: http://www.e-sciencecentral.org/articles/?scid=1027354.

[7] Yu-Xing Tang et al. "Automated abnormality classification of chest radiographs using deep convolutional neural networks". In: *NPJ Digital Media* (2020). DOI: https://doi.org/10.1038/s41746-020-0273-z.

[8] Youbao Tang et al. *XLSor: A Robust and Accurate Lung Segmentor on Chest X-Rays Using Criss-Cross Attention and Customized Radiorealistic Abnormalities Generation.* 2019. DOI: 10.48550/ARXIV.1904.09229. URL: https://arxiv.org/abs/1904.09229.

[9] Xiaosong Wang et al. "ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases". In: *CoRR* abs/1705.02315 (2017). arXiv: 1705.02315. URL: http://arxiv.org/abs/1705.02315.

[10] E.J. Yates, L.C. Yates, and H. Harvey. "Machine learning "red dot": open-source, cloud, deep convolutional neural networks in chest radiograph binary normality classification". In: *Clinical Radiology* 73.9 (2018), pp. 827–831. ISSN: 0009-9260. DOI: https://doi.org/10.1016/j.crad.2018.05.015. URL: https://www.sciencedirect.com/science/article/pii/S000992601830206X.

8 Acknowledgements

Over the summer, Andrea Dunn preformed research at Rice University under Anastasios Kyrillidis with the help of Cameron Wolfe as part of the Data Science REU. Her researched was focused on studying the effectiveness of online learning methods on medical data which are very unbalanced. This model borrows some off the custom data loaders from the summer project.

9 Github

Please check out our work!
https://github.com/asdunnbe/Comp562_Final