Capstone 2: Machine Learning as a Rapid Radiological Screening Tool

……Application of a CNN to Thoracic X-rays for Classification

Final Report (Blog Post)

" " " This write-up is meant to be a concise summary of my second capstone project. What follows is a long winded, and likely multi-part, simulated blog post of the model, results, and validation. " " "

I. The Problem

Many thousands of x-rays are performed daily. Within the medical community the assessment of an x-ray especially a chest x-ray (CXR), at least in the rapid screening since is a highly common if not ubiquitous task amounts many medical specialties. Official review is done by a Radiologist (MD or DO). To them this task is bordering on trivial in most cases due to the fact that they have evaluated so many. However, to get to that point they have had to undergo many years of school and residency. In addition to this despite being a minor task the sheer number of them makes them require time which taxes the system.

Thus, the idea of an automated system that could take on some of the burden. At the least flag potentially, worrisome scans and eventually if it becomes consistent enough relied upon as an autonomous system for medical professional to simply check the results of. The NIH themselves state that they envision a "virtual radiology resident" i.e. a system that can screen x-rays and report to a senior radiologist for confirmation. Another and possibility more pressing use for something like this comes from areas without access to a radiologist (rapid or at all) either a rural area with only a small clinic.

Given that US health spending represents 18% of the GDP and there is a ubiquitous movement to find ways to increase efficiency, decrease costs, and reduce medical errors. There are many potential clients for such a service from HER companies or directly to hospitals. The overall best implementation would be direct integration into an existing EHR system such as EPIC or Athena so that the results could be presented right alongside the actual scan and directly linked to the associated patient medical record for efficient review. Medical professional already has automated warning built into certain EHRs to inform them about a patient being overdue or in a risky range for lab values. The ultimate goal in my mind for a system such as this would be for it to integrate seamlessly into an existing or many existing electronic platforms for ease of use. Rather than simply adding to the complexity and requiring the use of yet one more piece of technology that hinders interaction with patients.

II. The Approach

The plan is to take advantage of the rapid progress that has been made in computer vision in recent years. I will be using a deep neural network (deep learning) to process and classify CXR images and produce am estimate on which category each image belongs. In this case of course the classes with be pathology types.

In order to speed this process, I will also take advantage of another concept called transitional learning. To do this I will use a pre-trained model that has already learned to identify certain elements in images. This will cut the required training and finetuning required tremendously. The fine detailed are discussed below.

III. The Data

The data for this project originates from the Nation Institutes of Health (NIH). It is a publicly available set of x-ray images and associated labels with some cursory meta-data e.g. age and gender. The data set consists of 112,120 images from over 30,000 patients.

As I will be taking a deep learning approach to this problem and it is a "curated" dataset there is little in the way of classical data wrangling to be done. There are no missing values present within the data and all images have the requisite labels.

Some issues do exist however. Over half of the images fall under the no finding category leading to some imbalance from the start. Also, as seen below we have moderate imbalance among the individual label categories as well. While this is somewhat representative of reality i.e. certain findings will be more common than others due to higher incidence.

Another principal difficulty is the sheer size of the data set. Coming in a about 42 Gb it is sizable for non-industrial standards. Storage itself is only a moderate issue, processing so many images can be time consuming on consumer grade hardware. For the purposes of this project I will be employing a cloud-based solution through Floyd Hub which is based on AWS and provides professional grade graphics cards for data science uses.

Mini glossary of terms

In order to save all of you a little time on google/Wikipedia I have included a few definitions here though they are not necessary to following the dialogue on the model itself. Cardiomegaly may be a little obvious e.g. oversized heart but some the other are no so much.

Atelectasis: collapse of a section of lung.

Effusion: a buildup of fluid.

Edema: swelling, also linked to fluid but presents differently.

Emphysema: damage to the small functional unit of the ling resulting in impaired air exchange. Grouped under COPD.

Infiltration: once again fluid, but is more specifically something more dense than air that is where it should not be. Can be blood, protein , pus, etc.

Pneumothorax: a collection of air where it should not be that results in a collapse of a small to large part of the lung. Think the scenes in moves where they call for a needle to stick in someone's chest….and no that is not how it work in reality.

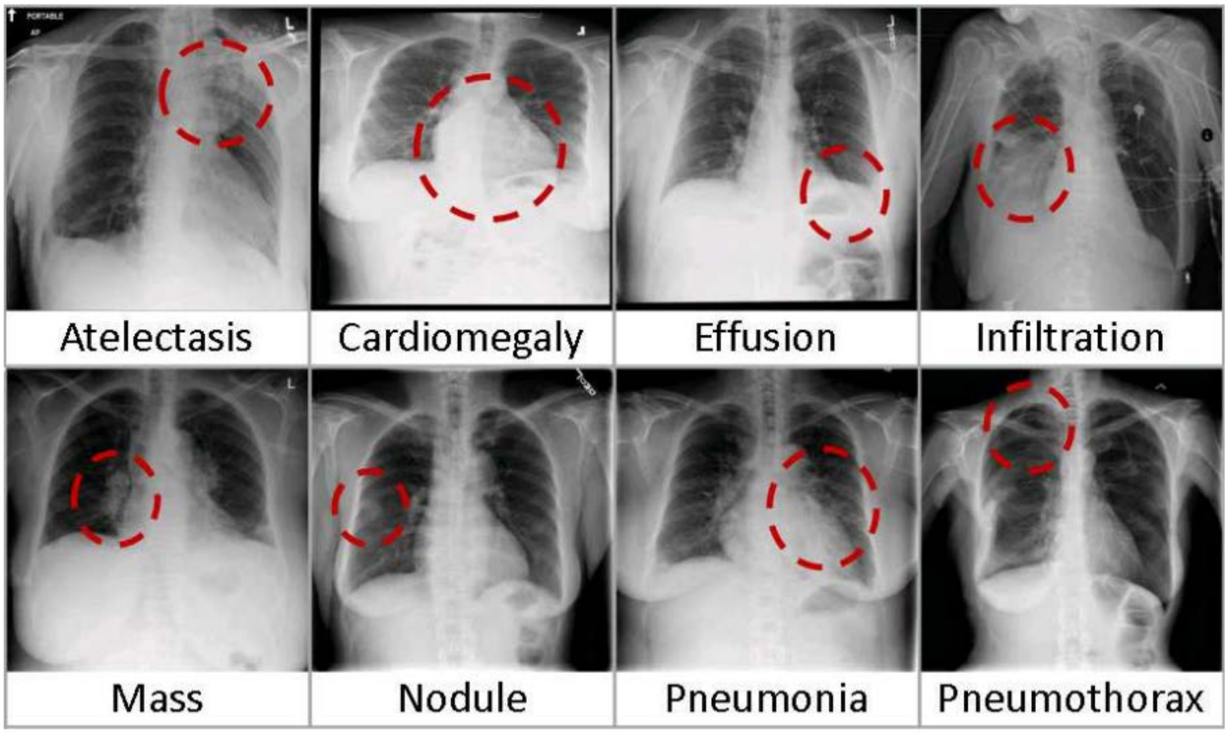


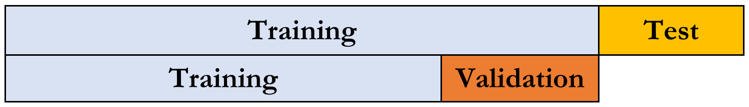
Figure 1 Adapted from Wang et al.

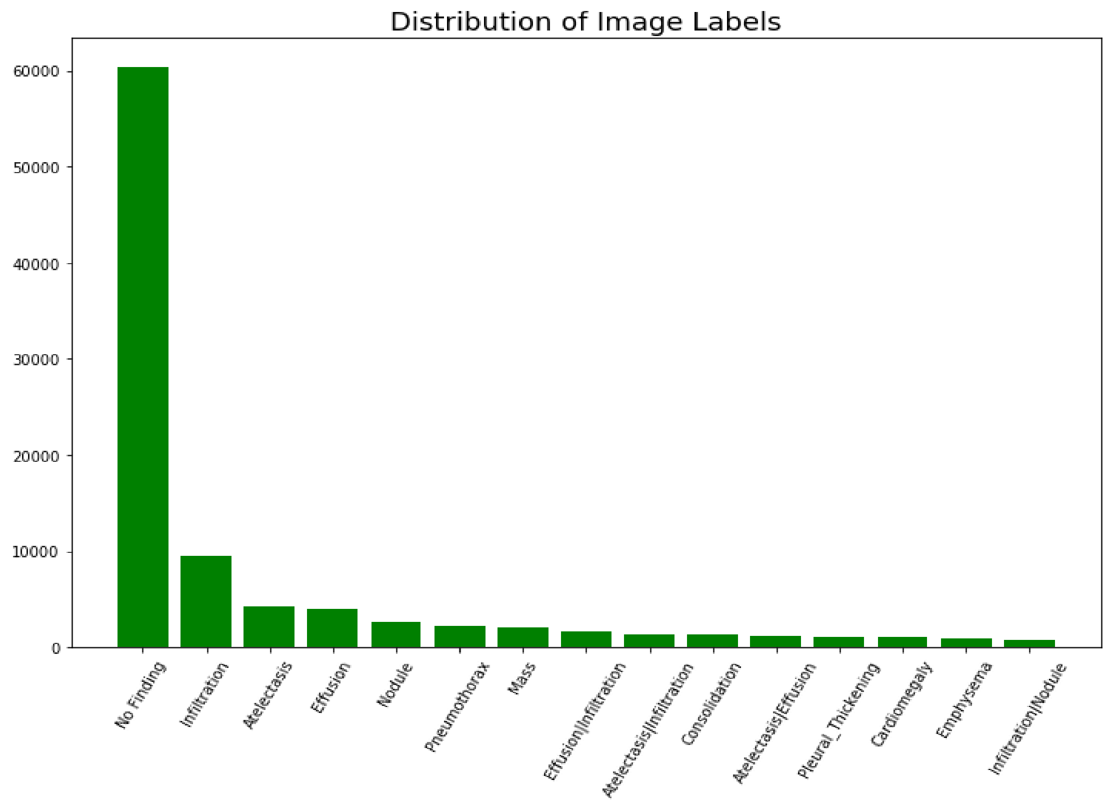
IV. The Implementation

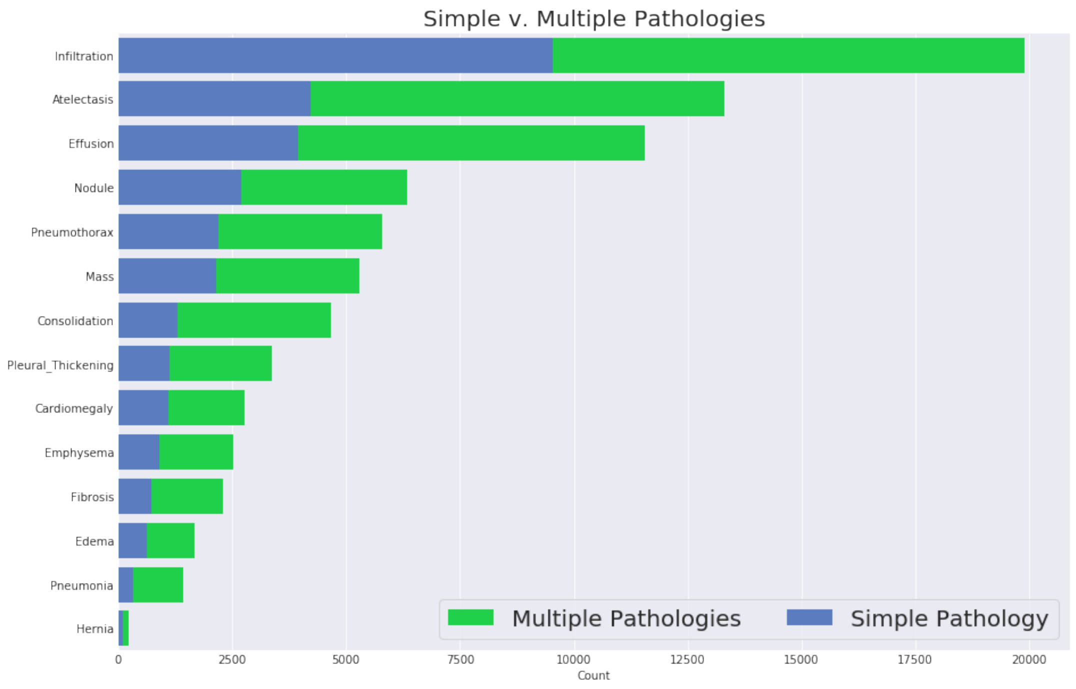
1. Data Analysis and Exploration

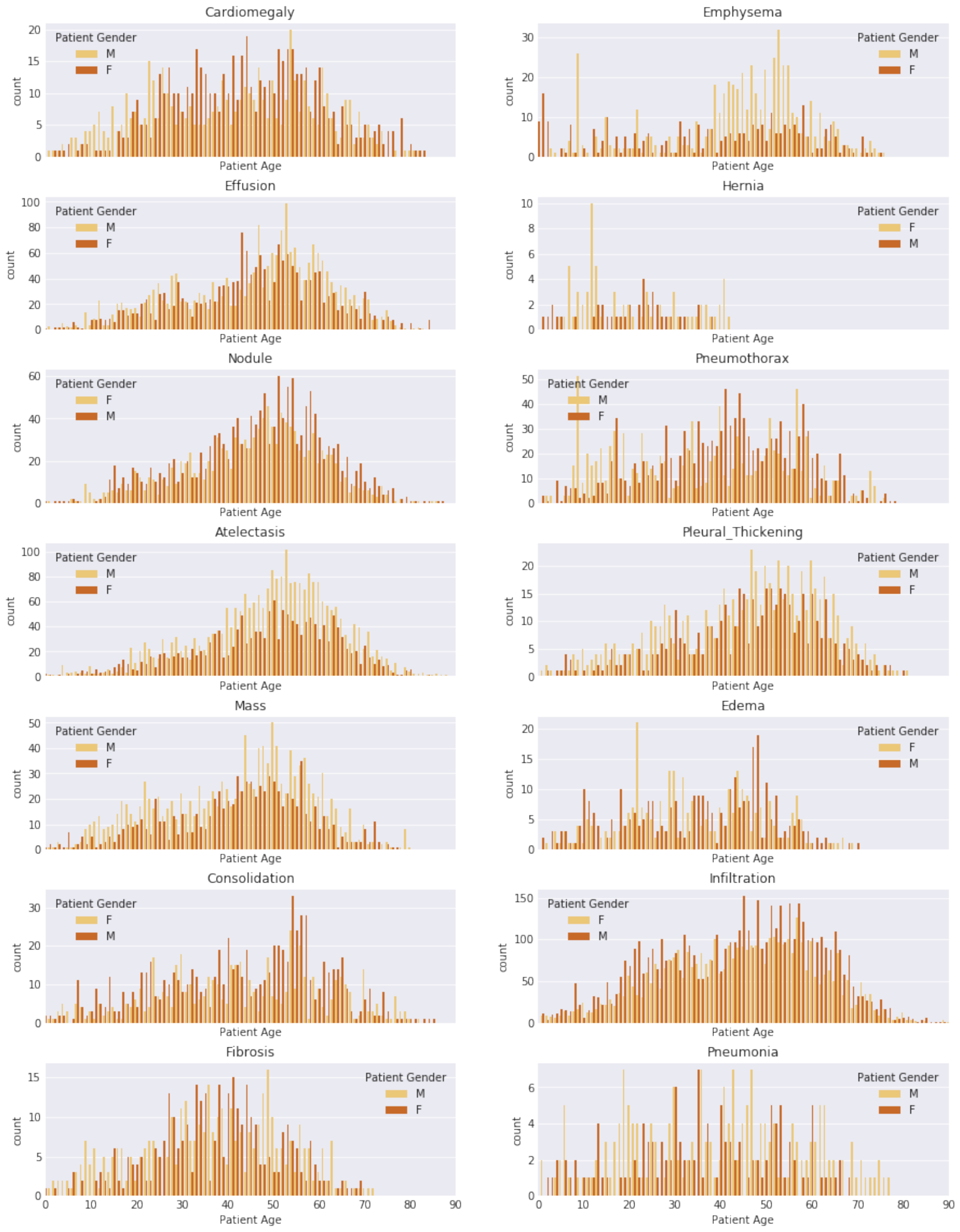
As can be seen from the charts below there is some pronounced categorical imbalance. Further, there are a great many images with more than one label. This is not truly a problem as the reality is that if abnormal many CXRs will have more than one pathological process present to one degree or another. Point being that any functional model will need to identify and distinguish between all of these things in the same image.

1) Multiple category images will be will be entered more than once under each pathology label for which they are reported (multi label model, see below).
2) 15 label categories will be used despite the categorical imbalanced and possibly too low count of certain categories e.g. emphysema.
3) I will attempt to account for this imbalance by using categorical weights when training the model. Other options could be over/under sampling.
4) As the images are currently in one 'giant pile' as it were, and through Keras we have the option to 'flow from directory' when pre-processing the images, I will be sorting the images into subfolders based upon the provided labels. These subfolders will then directly provide the relevant label to the model as it runs.
5) The data will be split into 3 groups. First the entire set will be separated into a 80/20 split for training and testing. Then the training data will be split again in an 80/20 manner for training a validation set for use while training the model.




Distribution of Image Labels
60000
50000
40000
30000
20000
10000
0
No Finding
Infiltration
Atelectasis
Effusion
Nodule
Pneumothorax
Mass
Effusion|Infiltration
Atelectasis|Infiltration
Consolidation
Atelectasis|Effusion
Pleural_Thickening
Cardiomegaly
Emphysema
Infiltration|Nodule


Simple v. Multiple Pathologies
Infiltration
Atelectasis
Effusion
Nodule
Pneumothorax
Mass
Consolidation
Pleural_Thickening
Cardiomegaly
Emphysema
Fibrosis
Edema
Pneumonia
Hernia
Multiple Pathologies
Simple Pathology
0
2500
5000
7500
10000
12500
15000
17500
20000
Count


Cardiomegaly
Emphysema
Effusion
Hernia
Nodule
Pneumothorax
Atelectasis
Pleural_Thickening
Mass
Edema
Consolidation
Infiltration
Fibrosis
Pneumonia
Patient Gender
M
F
count
Patient Age
0
10
20
30
40
50
60
70
80
90

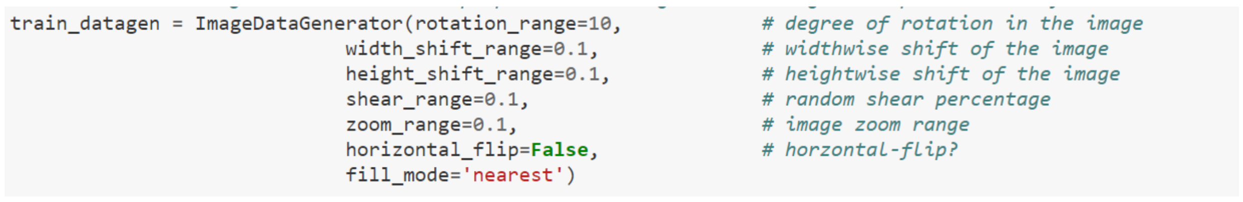
2. Initializing the model:

Getting the model going consists of several steps. Once familiar with Keras syntax this is not difficult. The difficult part comes with picking the 'under the hood' components and fine tuning all of the hyperparameters to achieve the best possible results.

The Keras model is broken down in to several components. First, there is loading and pre-processing the data. As mentioned above the data has been presorted from a single pile into subfolders based on train, validation, and test sets with sub directories under these for each category. There are many alternate ways of doing this part and I experimented with some of them e.g. preprocessing all of the images into np arrays. However, this method made sense to me for several reasons not the least of which was that it allowed me to directly pull images when I wanted to evaluate that the model was assessing during training.

So, we begin by using Keras's built in flow from directory feature and image data generator. This not only assess the categories and loads batches comparable to the available memory on our machine, but also augments the data to provide additional data to the model e.g. it will shift or even flip images prior to feeding them to the model in order to increase the variations the model is able to accurately interpret.

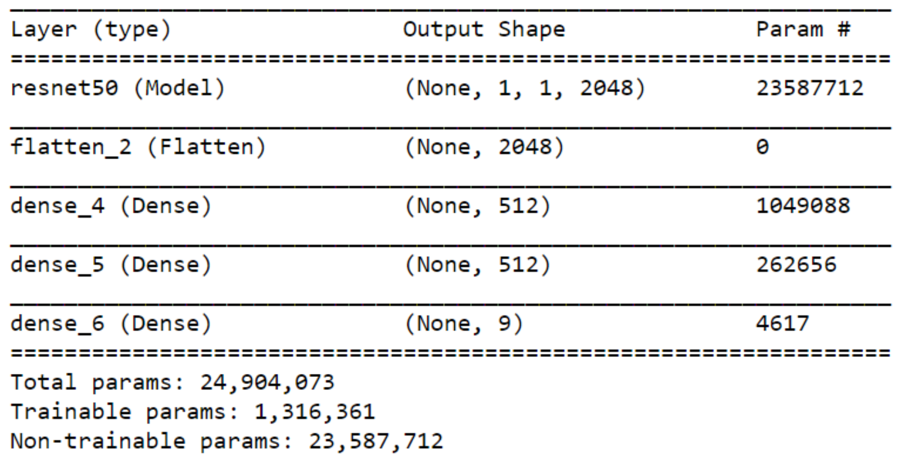
<u>Example of Image Pre-Processing</u>



```
train_datagen = ImageDataGenerator(rotation_range=10,          # degree of rotation in the image
                                   width_shift_range=0.1,      # widthwise shift of the image
                                   height_shift_range=0.1,     # heightwise shift of the image
                                   shear_range=0.1,            # random shear percentage
                                   zoom_range=0.1,             # image zoom range
                                   horizontal_flip=False,      # horzontal-flip?
                                   fill_mode='nearest')
```

After arranging for the data to be brought in we then define the actual structure of the model. This begins by importing the ResNet50 model land imagenet weights, and then locking all of the layers for the initial training runs. I then added some custom layers on top of the ResNet model these include flattening the output of the ResNet model and then adding in two dense layers with ReLu activations and a final Softmax layer which would force the results of the previous layers into an output that would represent categorical probabilities. A summary of the model structure id below.

Model Summary



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
resnet50 (Model)             (None, 1, 1, 2048)        23587712  
_________________________________________________________________
flatten_2 (Flatten)          (None, 2048)              0         
_________________________________________________________________
dense_4 (Dense)              (None, 512)               1049088   
_________________________________________________________________
dense_5 (Dense)              (None, 512)               262656    
_________________________________________________________________
dense_6 (Dense)              (None, 9)                 4617      
=================================================================
Total params: 24,904,073
Trainable params: 1,316,361
Non-trainable params: 23,587,712
```

3. Under the hood:

"The idea of knowledge is cumulative —seeing farther by standing on the shoulders of giants"

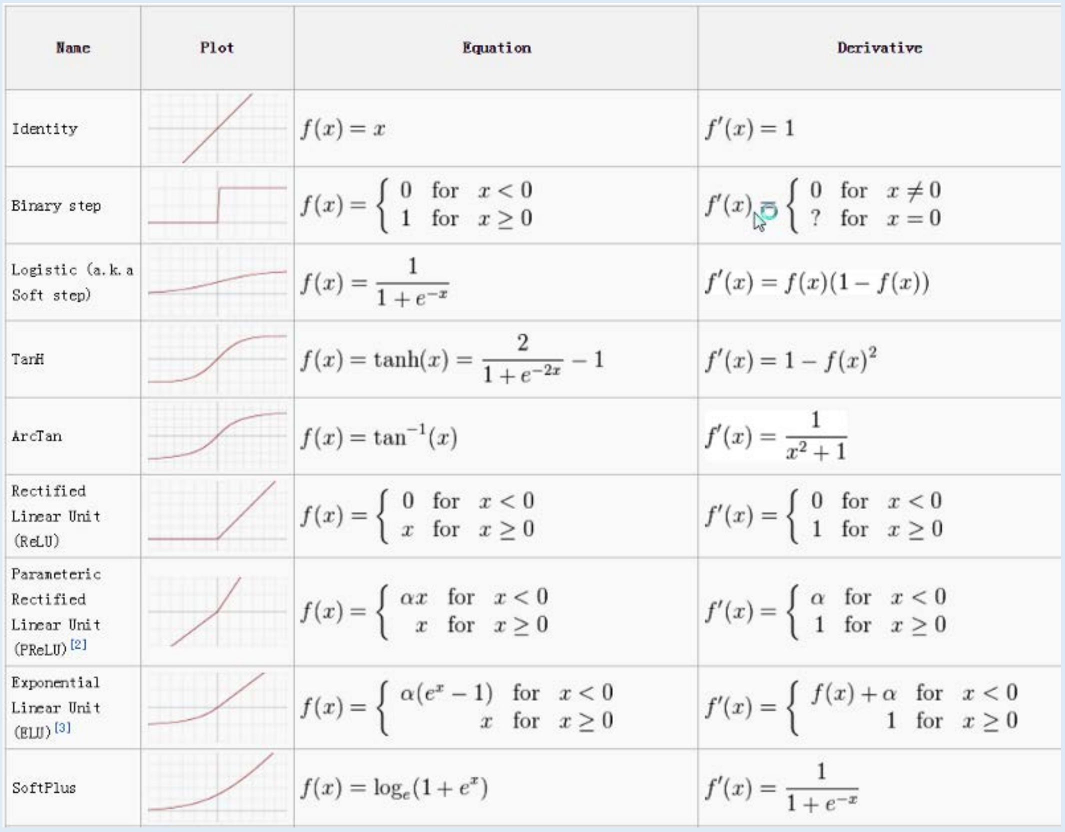
- Isaac Newton

Arguably the principal component of this model was the ResNet50. ResNet is short for Residual Network 50. As the name implies it is a 50 layer type of deep convolutional neural network that employs residual learning to attempt to gain further accuracy by delegating the learning of low/mid/high level features to different layers. E.g. one set of layers may focus on a tiny aspect or combination of aspects while other will focus on more macroscopic features in a given image. The short of is that this type of network has proven to be superior to traditional convolutional neural networks when it comes to image classification. There are also 'lighter' and 'heavier' derivatives of this kind of model which have more, fewer, and differing kinds of layers to them. I chose this particular one because it hit a happy medium between depth e.g. computational power required, speed of training, and accuracy. It has also been employed in a similar capacity by several other groups. More detailed information about this model can be read here.

Several of the other components are discussed below:

ReLu

Otherwise known as rectified linear unit activation. It has become highly popular in deep learning circles due to the fact that it simplifies the math behind the scenes and thus shortens training times without causing accuracy to suffer. I use the term accuracy fairly loosely here. The downside of this function is that we loose the ability to have the model interpret negative inputs as they are all reduced to zero.



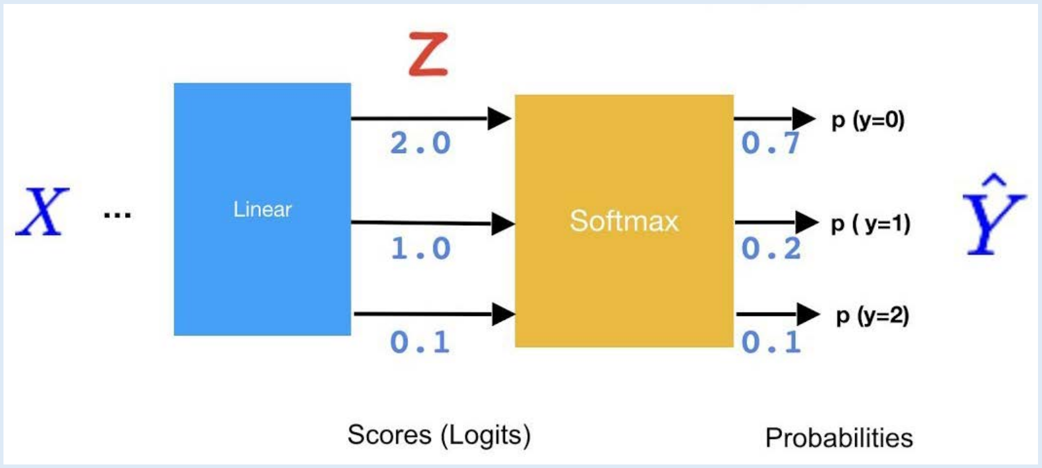
Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2+1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU) [2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU) [3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1+e^{-x}}$

RMSprop

As an alternative to SGD for the optimizer of the model. This function uses a running average of the magnitude of the gradient when finding the next step. It is also well known to pair well with recurrent and residual neural networks. This assists in refining the learning rate and gradient modification of the model. More info can be found here for RMSprop and here for learning rate schedules. FYI it is a highly complicated concept that takes some time to get your head wrapped around if you are not already familiar with the underlying concepts.

Softmax

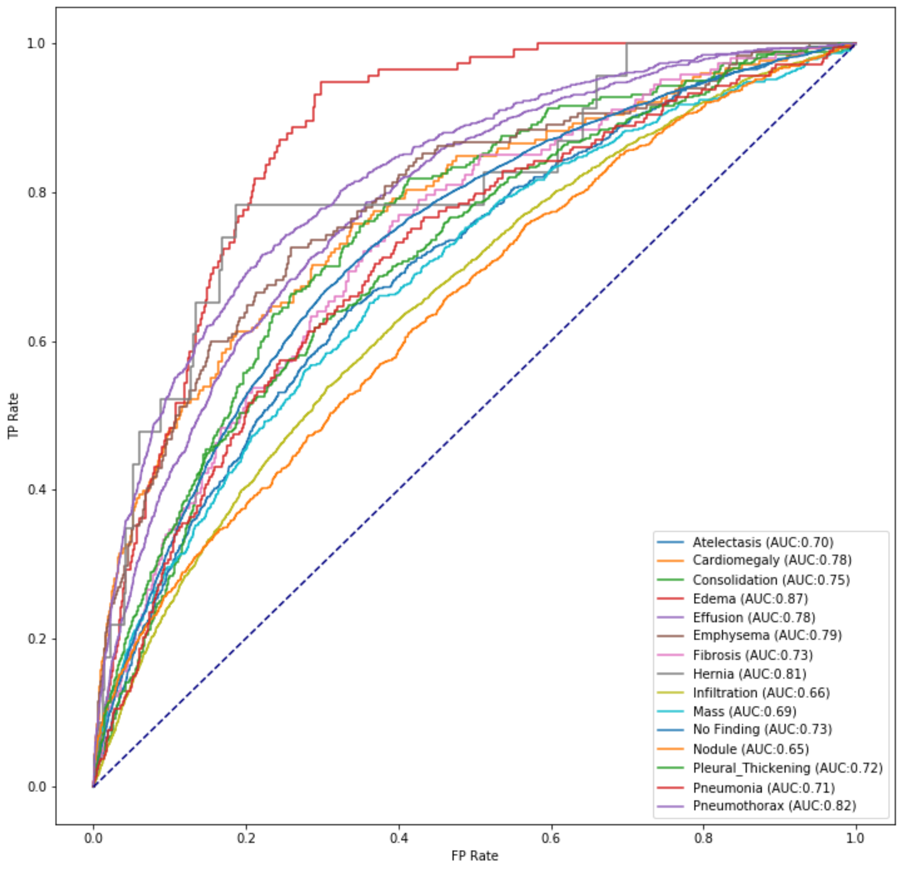
This function works well for multi-class problems as the probabilities of a sample belonging to each class all sum to one. It is used as the activation function in the final layer of the network. More information can be found here.



The principle end goal of all of the above components is to prevent overfitting and to increase the interpretability of the output. As these models are renowned for a) overfitting and b) being mysterious black boxes every little bit helps. There are many subcomponents to each of these that are beyond the scope of this post but which can be used to tune these models till the end of time if you so wish.

4. Training the model

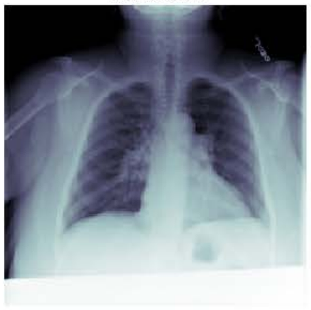
The original plan was to run the model with all of the ResNet50 layers locked and to then to run it for additional epochs with several or even all of the layers unlocked. This would allow the lower layers of the model to 'adapt' to the image set. However, during testing problems with overfitting the second part of this plan turned out to not be feasible within the time and financial constraints of this project. Thus, the results discussed here are from a 30 epoch run with all ResNet layers locked.



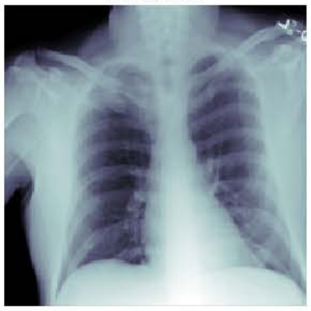
ROC Comparison by Class

Class	**My Model**	**NIH Model**
Atelectasis	0.70	0.70
Cardiomegaly	0.78	0.81
Consolidation	0.75	
Edema	0.87	
Effusion	0.78	0.73
Emphysema	0.79	
Fibrosis	0.73	
Hernia	0.81	
Infiltration	0.66	0.61
Mass	0.69	0.56
No Finding (Normal)	0.73	
Nodule	0.65	0.71
Pleural Thickening	0.72	
Pneumonia	0.71	0.63
Pneumothorax	0.82	0.78

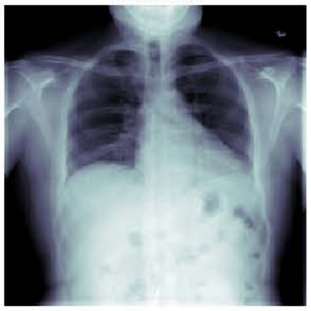
Dx: No Finding
PDx: No F:37%



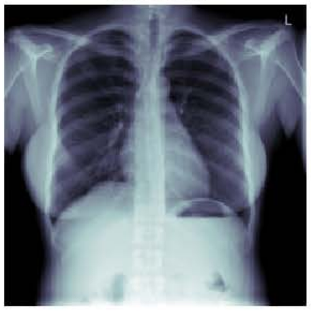
Dx: No Finding
PDx: No F:51%



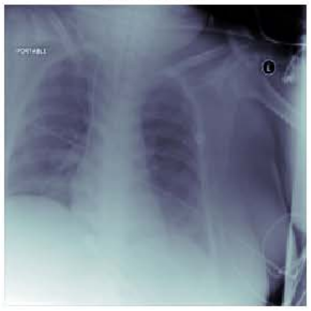
Dx: Cardiomegaly
PDx: Card: 3%



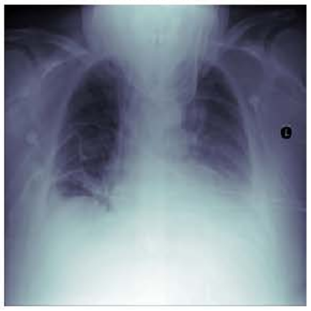
Dx: No Finding
PDx: No F:62%



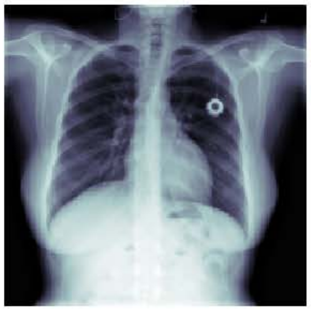
Dx: No Finding
PDx: No F:20%



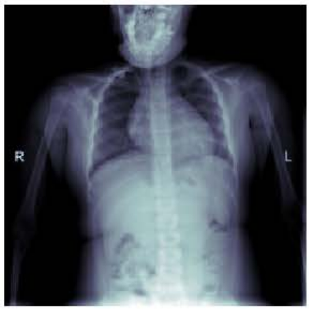
Dx: Infiltration
PDx: Infi:29%



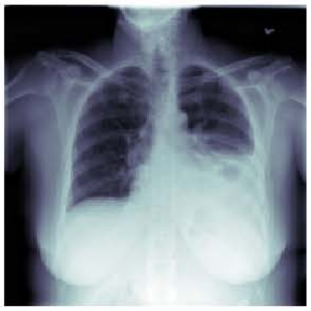
Dx: Infiltration
PDx: Infi:21%



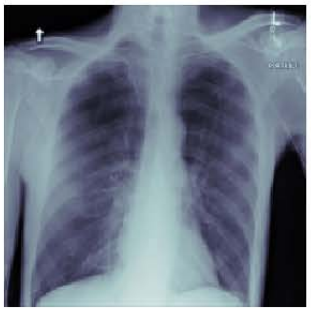
Dx: No Finding
PDx: No F:30%



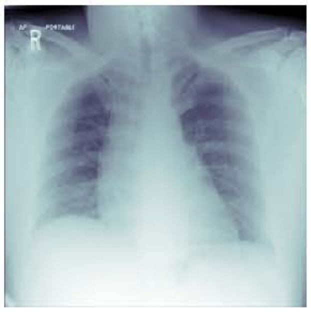
Dx: Effusion
PDx: Effu:48%



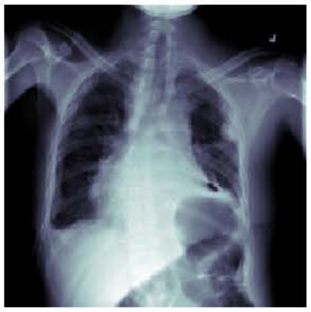
Dx: No Finding
PDx: No F:36%



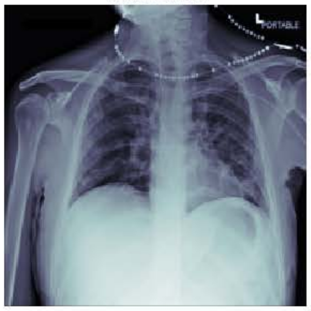
Dx: Pneumothorax
PDx: Pneu: 1%



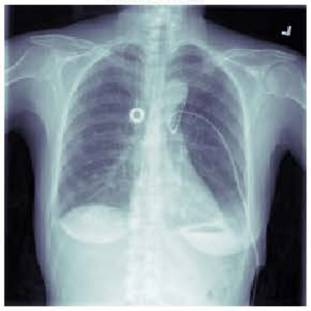
Dx: Pneumothorax
PDx: Pneu:14%



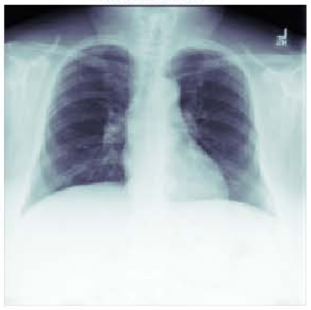
Dx: No Finding
PDx: No F:18%



Dx: Nodule
PDx: Nodu: 4%



Dx: No Finding
PDx: No F:33%



5. Conclusion

As can be seen from the results displayed above this model has demonstrated moderate to good performance. However, this must be taken in context. Most of these conditions should also have been evaluated in a clinical context with meta-data e.g. causational patient information to increase the evidence for or against a certain finding. Also, we must take into account the questionable labels. From my own cursory review of several images in the dataset I can say that there are a few that seem mislabeled, but I am no radiologist.

To take another step back we can give this model and other like a little firmer footing though. According to Brady et.al. in a study published in February 2017 there is an estimated daily error rate for radiologists of up to 5% and world wide this can average out at up to 40 million or so report errors per year. This is not a jab at radiologists, rather simply putting error rates in perspective. All fields of medicine make errors on a daily basis to a very uncomfortable degree. It is my hope that models along the lines of this one can be implemented as tools for medical practitioners to dramatically reduce these errors and improve the quality of medical care.

Some foreseeable problems with this venture are listed in the next section. One specific one that I would like to point out here though is the exponential increase in complexity of trying to apply similar techniques to other medical imaging eg. CT and MRI. X-rays are single time point images not innately different from any other image classification problem. The later two types of scans are far more dynamic and can encompasses thousands of slices depending upon the type of scan needed.

6. Next steps

Possible steps for building upon the current work:

1) Further training, +/- may help to increase the accuracy.
2) Further tuning of hyperparameters e.g. activation, regularizes, and image pre-processing.
3) Use of alternate techniques e.g. unsupervised feature detection and/or incorporation of meta data such as patient age, gender, and possibly even past medical variables.
4) Add additional features to create heatmaps and bounding boxes around features of interest in each image.

Possible enhancement for any large-scale deployment:

1) Must train on hand-labeled data with inter-observer cross validation.
2) Must have extensive training and clearly be identifying aspects of disease e.g. not identify a chest tube as a marker for specific disease type.
3) First and foremost, must be able to identify a normal CXR from a abnormal one regardless of category. If the model identifies a pathology incorrectly it can be taken care of by secondary human screening. However, if the model missed a single tumor it could cost a life or at least increase morbidity i.e. would need a very very low false-negative rate.

All of the relevant code for this project can be found here.

Resources

https://www.nih.gov/news-events/news-releases/nih-clinical-center-provides-one-largest-publicly-available-chest-x-ray-datasets-scientific-community

https://medium.com/intuitionmachine/the-brute-force-method-of-deep-learning-innovation-58b497323ae5

https://lukeoakdenrayner.wordpress.com/2017/12/18/the-chestxray14-dataset-problems/

http://www.radisphereradiology.com/wp-content/uploads/Diagnostic-Accuracy-in-Radiology

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5265198/