Chest X-Ray Diagnosis on NIH Dataset using Deep Learning

Rahul Baid
Georgia Institute of Technology
rbaid3@gatech.edu

Ruhani Suri
Georgia Institute of Technology
rsuri8@gatech.edu

Abstract

Chest X-ray scans are one of the most popular and cost effective diagnosis tools used by doctors worldwide. We wish to train a model that is able to remove the reliance of an x-ray interpretation from error-prone humans to sophisticated machines. We work on the NIH dataset of 14 common Thorax disease categories. We use the DenseNet and VGG 16 models in our project with a baseline of MobileNet. The code repository is https://github.com/rahulb99/ChestXrayDiagnosis.

1. Introduction

Chest X-ray scans are the most frequently conducted type of radiology exams worldwide. It is often used to detect lung cancer or heart failure and monitor other bone and blood vessel abnormalities in patients. It is also the most effective method to diagnose diseases such as Pneumonia. It is also used as a means to track patients' progress when under a particular treatment. We want to create a model that is able to detect such abnormalities and diseases when given an x-ray scan.

Today, most hospitals and doctors employ reading off an x-ray manually, trying to pin point broken ribs, fluids around the lungs, or other signs of cancerous tissues or diseases. Doctors are highly specialized and trained in reading x-ray scans accurately. Then, the findings are communicated to the patient, doctors and other health professionals. This current practice is limited in the way of manual capabilities. Often, hospitals may not have doctors at hand that can ready x-rays or doctors may misread them due to confusion, lack of resources or pressure. The communication after which may be compromised too, such as doctors misunderstanding the diagnosis or treatment requirement. Needless to say, the stakes of this is too high, involving large amounts of money, discomfort and above all, the human life.

If successful, our model can eliminate or greatly reduce the reliance of such medical diagnosis on humans and thus make it largely error free. We do recommend some human interference such as verification before such a technology is used independently. This can help those hospitals which are understaffed or have no trained medic on site. Such a model if successful can help speed up accurate diagnoses of millions of patients around the world and put them on the right kind of treatment at the earliest. Such a model's success brings precision, reliance and speed to the process.

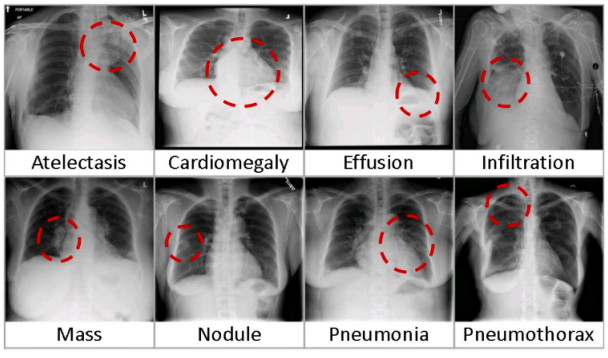
2. Data

We use the NIH Chest X-ray Dataset [4] of 14 Common Thorax Disease Categories, which are:

(1, Atelectasis; 2, Cardiomegaly; 3, Effusion; 4, Infiltration; 5, Mass; 6, Nodule; 7, Pneumonia; 8, Pneumothorax; 9, Consolidation; 10, Edema; 11, Emphysema; 12, Fibrosis; 13, Pleural Thickening; 14 Hernia)

The dataset is extracted from the clinical PACS database at National Institutes of Health Clinical Center and consists of 60% of all frontal chest x-rays in the hospital. It consists of 112,120 frontal-view X-ray images of 30,805 unique patients with the text-mined fourteen disease image labels (where each image can have multi-labels).

Below are some sample images from the NIH dataset, labelled with the ailment, and on the next page is a table showing the salient features of the dataset.



The contents of the dataset are as follows:

1. 112,120 frontal-view chest X-ray PNG images in 1024*1024 resolution

2. Meta data for all images: Image Index, Finding Labels, Follow-up #, Patient ID, Patient Age, Patient Gender,

Description	Training	Dev	Test
Number of images	104266	6336	1518
Percentage of total	93	6	1

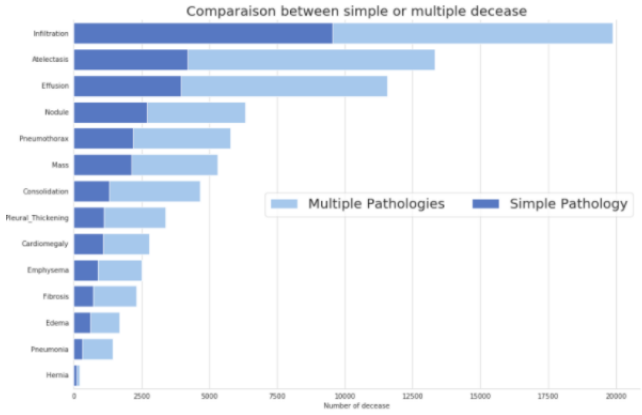
Table 1. Data split for NIH dataset

View Position, Original Image Size and Original Image Pixel Spacing.

3. Bounding boxes for 1000 images :Image Index, Finding Label, Bbox[x, y, w, h]. [x y] are coordinates of each box's topleft corner. [w h] represent the width and height of each box.

4. Two data split files are provided. Images in the ChestX-ray dataset are divided into these two sets on the patient level. All studies from the same patient will only appear in either training/validation or testing set.

A significant number of X-ray images are labelled with multiple conditions as shown below



As per the NIH documentation, following are the limitations of the dataset:

1. The image labels are NLP extracted so there would be some erroneous labels but the NLP labelling accuracy is estimated to be >90%

2. Very limited numbers of disease region bounding boxes

3. Chest x-ray radiology reports are not anticipated to be publicly shared. Parties who use this public dataset are encouraged to share their "updated" image labels and/or new bounding boxes in their own studied later, maybe through manual annotation.

3. Related Work

We did a literature review of some of the existing works in this field to guide us in our model development.

In the past, several groups have been able to train very accurate models to solve the chest x-ray medical imaging classification problem. One of the first attempts was led by Wang et al. , who trained four different networks AlexNet, GoogLeNet, VGGNet-16, and ResNet-50 to classify eight common chest conditions using pre-trained models. Next, Rajpurkar et al. [5] developed a 121-layer convolutional neural network, named CheXNet, to predict the probability of 14 different thoracic diseases. A limitation that is common to some papers we read is that the models trained only considered one x-ray view to train and make predictions. In our project, we tried to see how a training dataset containing both frontal and lateral images affects the model's accuracy and performance.

4. Approach

For this project, we use the existing stable model of DenseNet-121 (or CheXNet) [2] and run it on the NIH dataset. We performed various experiments on this model by tuning various hyperparameters. We train the model on thousands of images and then test our result on a small randomly picked test set. We thought these modules would be successful because they have good feature extraction capabilities and have been pre-trained with similar images. We use MobileNet as our baseline model to compare our performances against. We also use VGG-16 with attention mechanism for our experiments.

The main problem we faced was fitting the entire dataset and training the model on it. Often the training was too slow or just impossible to load so many images. Thus, we resorted on further cutting down our image set into small fractions to train repeatedly. We also had to train on lesser epochs, which could one of the reason for the less accuracy of the model.

5. Models

5.1. DensetNet

It is a Densely Connected Convolutional Network which connects each layer to every other layer in a feed-forward fashion. The convolution networks can be substantially deeper, more accurate, and efficient to train if they contain shorter connections between layers close to the input and those close to the output. For each layer, the feature-maps of all preceding layers are used as inputs, and its own feature-maps are used as inputs into all subsequent layers.

Main advantages of DenseNet 121 includes that they alleviate the vanishing-gradient problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters.

The architecture of DenseNet 121 is given below:

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	[1 × 1 conv, 3 × 3 conv] × 6	[1 × 1 conv, 3 × 3 conv] × 6	[1 × 1 conv, 3 × 3 conv] × 6	[1 × 1 conv, 3 × 3 conv] × 6
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	[1 × 1 conv, 3 × 3 conv] × 12	[1 × 1 conv, 3 × 3 conv] × 12	[1 × 1 conv, 3 × 3 conv] × 12	[1 × 1 conv, 3 × 3 conv] × 12
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	[1 × 1 conv, 3 × 3 conv] × 24	[1 × 1 conv, 3 × 3 conv] × 32	[1 × 1 conv, 3 × 3 conv] × 48	[1 × 1 conv, 3 × 3 conv] × 64
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	[1 × 1 conv, 3 × 3 conv] × 16	[1 × 1 conv, 3 × 3 conv] × 32	[1 × 1 conv, 3 × 3 conv] × 32	[1 × 1 conv, 3 × 3 conv] × 48
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

DenseNets are built from dense blocks and pooling operations, where each dense block is an iterative concatenation of previous feature maps. This architecture can be seen as an extension of ResNets.

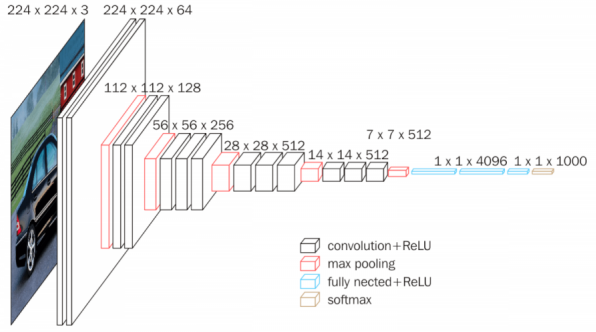
The last layer is a fully connected layer for prediction with sigmoid activation function. We initially start training with pre-trained weights on the ImageNet [6] dataset.

5.2. VGG 16

VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition". It consists of 16 layers in total, including Convolutional layers, Max Pooling layers, Activation layers, Fully connected layers. This network is a pretty large network and it has about 138 million (approx) parameters.

Most unique thing about VGG16 is that instead of having a large number of hyper-parameter they focused on having convolution layers of 3x3 filter with a stride 1 and always used same padding and maxpool layer of 2x2 filter of stride 2.

Below is an image demonstrating the architecture of the VGG 16 model.



5.3. MobileNet

MobileNets are based on a streamlined architecture that uses depth-wise separable convolutions to build light weight deep neural networks. The depthwise convolution applies a single filter to each input channel.

MobileNet has 30 layers, with the architecture attached below. The pointwise convolution then applies a 11 convolution to combine the outputs the depthwise convolution. A standard convolution both filters and combines inputs into a new set of outputs in one step. The depthwise separable convolution splits this into two layers, a separate layer for filtering and a separate layer for combining. This factorization has the effect of drastically reducing computation and model size.

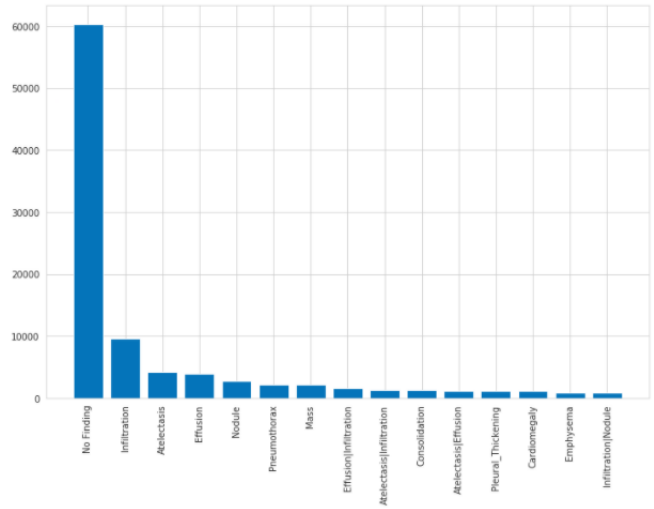
Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	3 × 3 × 3 × 32	224 × 224 × 3
Conv dw / s1	3 × 3 × 32 dw	112 × 112 × 32
Conv / s1	1 × 1 × 32 × 64	112 × 112 × 32
Conv dw / s2	3 × 3 × 64 dw	112 × 112 × 64
Conv / s1	1 × 1 × 64 × 128	56 × 56 × 64
Conv dw / s1	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 128	56 × 56 × 128
Conv dw / s2	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 256	28 × 28 × 128
Conv dw / s1	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 256	28 × 28 × 256
Conv dw / s2	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 512	14 × 14 × 256
5× Conv dw / s1	3 × 3 × 512 dw	14 × 14 × 512
5× Conv / s1	1 × 1 × 512 × 512	14 × 14 × 512
Conv dw / s2	3 × 3 × 512 dw	14 × 14 × 512
Conv / s1	1 × 1 × 512 × 1024	7 × 7 × 512
Conv dw / s2	3 × 3 × 1024 dw	7 × 7 × 1024
Conv / s1	1 × 1 × 1024 × 1024	7 × 7 × 1024
Avg Pool / s1	Pool 7 × 7	7 × 7 × 1024
FC / s1	1024 × 1000	1 × 1 × 1024
Softmax / s1	Classifier	1 × 1 × 1000

6. Experiments and Results

6.1. Preprocessing

Below we give a graph of the initial distribution of the NIH dataset.



As is visible, one of the category, which unlabelled images labelled 'No Finding', are in vast majority. Such large imbalances, which is due to over-representation of common medical problems and scarcity of rare problems, usually makes it difficult to train a model into identifying diseases and labels other than 'No Finding'. Thus, we remove these unlabelled images from our dataset.

For the step of preprocessing, we cleaned the images by re-scaling by a factor of (1/255) and set the target size to be 224x224. We loaded the data using the `ImageDataGenerator` class to perform data augmentation, to make up for the vast majority of the images we scrapped out for being unlabelled and increase our training and test data size.

6.2. Training

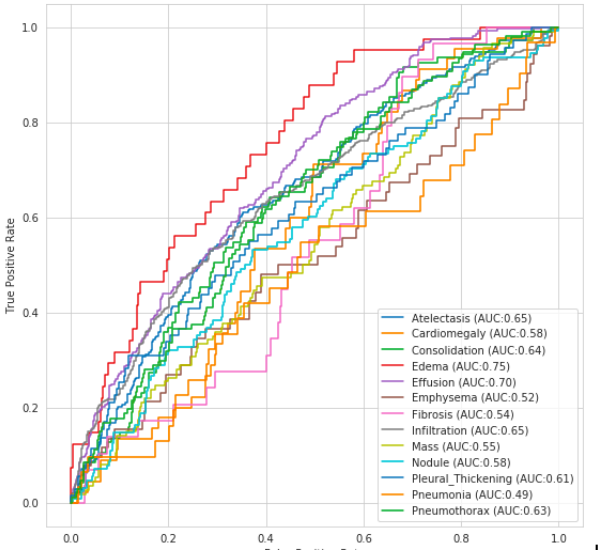
6.2.1 MobileNet

Following are the details of our model training.

1. Batch size: 32, 32, 1024 for train, validation and test respectively
2. Number of epochs: 20
3. Steps per epoch: 100
4. Binary Cross Entropy for loss, using Adam optimizer

Layer (type)	Output Shape	Param #
mobilenet_1.00_128 (Model)	(None, 4, 4, 1024)	3228288
global_average_pooling2d_1 (	(None, 1024)	0
dropout_1 (Dropout)	(None, 1024)	0
dense_1 (Dense)	(None, 512)	524800
dropout_2 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 13)	6669

Total params: 3,759,757
Trainable params: 3,737,869
Non-trainable params: 21,888



The final testing results are as follows:

1. Atelectasis: actual: 22.27%, predicted: 21.43%
2. Cardiomegaly: actual: 4.39%, predicted: 5.98%
3. Consolidation: actual: 9.28%, predicted: 8.33%
4. Edema: actual: 4.00%, predicted: 4.91%
5. Effusion: actual: 26.66%, predicted: 21.41%
6. Emphysema: actual: 5.08%, predicted: 6.16%
7. Fibrosis: actual: 2.83%, predicted: 3.70%
8. Infiltration: actual: 36.82%, predicted: 40.52%
9. Mass: actual: 11.13%, predicted: 10.04%
10. Nodule: actual: 12.50%, predicted: 16.83%
11. Pleural Thickening: actual: 6.93%, predicted: 7.77%
12. Pneumonia: actual: 3.03%, predicted: 3.28%
13. Pneumothorax: actual: 10.45%, predicted: 8.64%

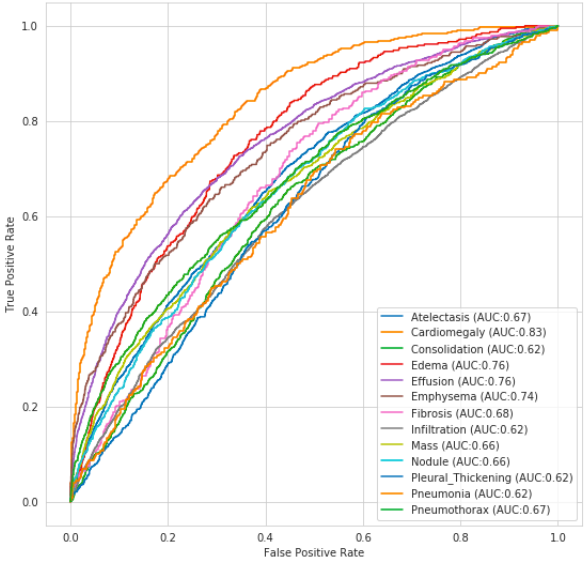
Final roc score = 0.615584234678

6.2.2 DenseNet

The details of the DeseNet 121 is as follows:

1. Last layer is a fully connected layer for prediction with sigmoid activation
2. 32,32,8000 batch size for train, val, test
3. Pretrained weights on ImageNet dataset (transfer learning)

4. optimiser , loss, metrics same as mobilenet.
5. Learning rate = 0.001
6. Number of epochs = 20
7. steps per epoch = 100



The final testing results are as follows

1. Atelectasis: actual: 23.11%, predicted: 19.46%
2. Cardiomegaly: actual: 5.73%, predicted: 4.99%
3. Consolidation: actual: 9.53%, predicted: 12.48%
4. Edema: actual: 4.86%, predicted: 5.84%
5. Effusion: actual: 26.50%, predicted: 32.40%
6. Emphysema: actual: 4.98%, predicted: 4.42%
7. Fibrosis: actual: 3.46%, predicted: 2.35%
8. Infiltration: actual: 38.01%, predicted: 42.11%
9. Mass: actual: 11.53%, predicted: 14.22%
10. Nodule: actual: 12.44%, predicted: 14.17%
11. Pleural Thickening: actual: 7.27%, predicted: 9.43%
12. Pneumonia: actual: 2.90%, predicted: 2.82%
13. Pneumothorax: actual: 10.34%, predicted: 8.42%

Final roc score = 0.68547827422

We also got class activation maps with the help of Grad-CAM++ [1] to interpret the predictions of this model and the location of the pathology.

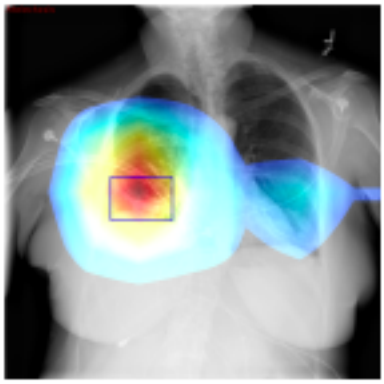


Figure 1. Atelectasis

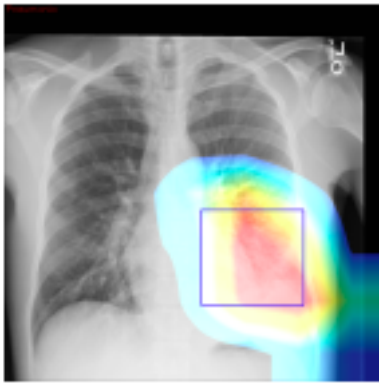


Figure 2. Pneumonia

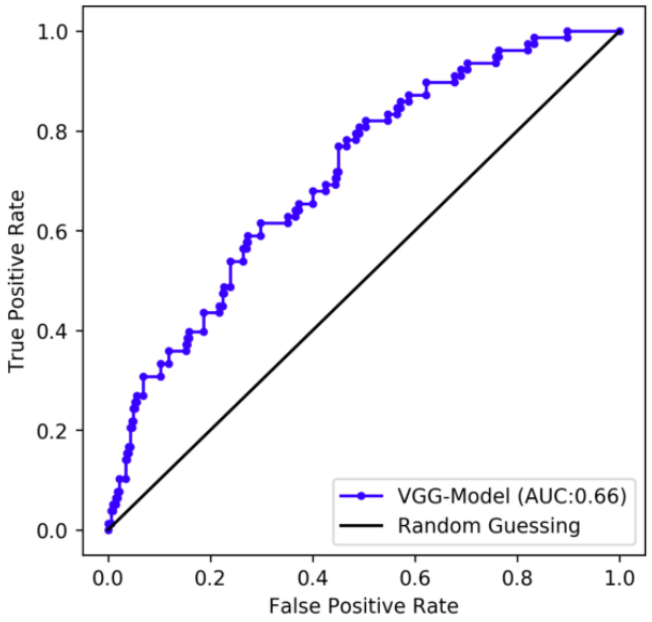
6.2.3 VGG 16

We used transfer learning by taking the first layers of a VGG16 model trained on ImageNet data (classifying color images of dogs, airplanes, cats, ...) and retrained it on grayscale images of chests [3]. One of the major chances we made is that we built an attention mechanism to turn pixels in the global average pooling on and off before the pooling and then rescale the results based on the number of pixels. The model could be seen as a sort of 'global weighted average' pooling. This kind of attention models if often used in the field of Natural Language Processing. It outputs a spatial mask of which regions of the pre-trained feature map we want to use. We rescale the feature dimension back out to the original number of features.

1. Number of epochs = 10
2. Batch size = 24, 32 for train, test
3. We use the same loss, optimizer and learning rate as DenseNet

Layer (type)	Output Shape	Param #
vgg16 (Model)	(None, 16, 16, 512)	14714688
attention_model (Model)	(None, 1)	138690

Total params: 14,853,378
Trainable params: 137,154
Non-trainable params: 14,716,224



7. Analysis

The problem we chose to takle was that of analysing chest x-rays as forms of medical imaging to aid in diagnosis. Our model closesly resembled this structure in the sense that it recieved as input an image and aimed to classify it in one of the 14 ailments labels.

We had to tune our hyper parameters to receive the results that we documented above. We performed some preprocessing tasks such as removing class imbalance and performing data augmentation.

The loss function used was weight binary cross entropy.

$$L(X|y) = -\sum_{c=1}^{14}[w_{+}.-y_c \log p(Y=1|X)+w_{-}.(1-y_c)\log p(Y=0|X)] \quad (1)$$

We reduced our training size to relatively small values so as to avoid over-fitting. We instead test on a large data size so as to predict well.

We used Keras with Tensorflow backend as a framework for this project.

Future work could address the class imbalance problem in medical imaging or chest X-ray in general, by using similar datasets and generating augmented or synthetic images, this would help to improve the performance of some of the pathology classes. Training for more epochs, hyperparamter tuning, and using images with higher resolutions could lead to better performance as well. Furthermore, some limitations with the dataset, example incorrect labelling, could also be a reason for few wrong answers occasionally.

8. Conclusion

Through this small project, we have shown the immense scope of medical image analysis in the field of disease diagnosis and tracking. Though our model does not perform very well, we are certain that more work on this field and better computing resources and richer training data can significantly improve the accuracy. Soon enough, these models will good enough to be used independently in hospitals, thus reducing massive amounts of manual labuor and errors.

9. Work Division

Please see Table 2 for detailed work division. Both members of the team contributed equally.

References

[1] Aditya Chattopadhyay. Gradcam++. 5

[2] Bruce Chou. Chexnet keras. 2

[3] Keras. Models for image classification with weights trained on imagenet. 5

[4] National Institute of Health. Nih chest x-ray dataset of 14 common thorax disease, 2017. 1

[5] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Yi Ding, Aarti Bagul, Curtis Langlotz, Katie S. Shpanskaya, Matthew P. Lungren, and Andrew Y. Ng. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *CoRR*, abs/1711.05225, 2017. 2

[6] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 3

Student Name	Contributed Aspects	Details
Rahul Baid	Training and Tuning	Trained the models on dataset and hypertunning of parameters
Ruhani Suri	Dataset and Analysis	Prepared the data for training, testing and performed analysis on the result

Table 2. Contributions of team members.