Pneumonia Detection by observing Lung Opacity from the Chest X-Ray Images

Final Capstone Project Report

Contents

INTRODUCTION

1.1 What is Pneumonia?

Pneumonia is an infection in one or both lungs. Bacteria, viruses, and fungi cause it. The infection causes inflammation in the air sacs in your lungs, which are called alveoli.

Pneumonia accounts for over 15% of all deaths of children under 5 years old internationally. In 2017, 920,000 children under the age of 5 died from the disease. It requires review of a chest radiograph (CXR) by highly trained specialists and confirmation through clinical history, vital signs and laboratory exams.

1.2 What Does a Normal Image Look Like?

This is an illustration of the chest anatomy with the lungs highlighted

It is observed that there is a mass of tissue surrounding the lungs and between the lungs. These areas contain skin, muscles, fat, bones, and also the heart and big blood vessels. That translates into a lot of information on the chest radiograph that is not useful for detecting the lung opacity.

1.3 Chest Radiographs Basics

In the process of taking the image, an X-ray passes through the body and reaches a detector on the other side. Tissues with sparse material, such as lungs which are full of air, do not absorb the X-rays and appear black in the image. Dense tissues such as bones absorb the X-rays and appear white in the image. In short -

- Black = Air

- White = Bone
- Grey = Tissue or Fluid

The left side of the subject is on the right side of the screen by convention. It can also be observed that there is a small L at the top of the right corner. In a normal image we see the lungs as black, but they have different projections on them - mainly the rib cage bones, main airways, blood vessels and the heart.

PROBLEM STATEMENT, DATA AND FINDINGS

2.1 Problem statement

Pneumonia usually manifests as an area or areas of increased opacity on CXR. However, the diagnosis of pneumonia on CXR is complicated because of a number of other conditions in the lungs such as fluid overload (pulmonary edema), bleeding, volume loss (atelectasis or collapse), lung cancer, or post-radiation or surgical changes. Outside of the lungs, fluid in the pleural space (pleural effusion) also appears as increased opacity on CXR.

So lung opacities do not literally point to pneumonia, but there can be many other issues or abnormalities as well. When available, comparison of CXRs of the patient taken at different time points and correlation with clinical symptoms and history are helpful in making the diagnosis.

CXRs are the most commonly performed diagnostic imaging study. A number of factors such as positioning of the patient and depth of inspiration can alter the appearance of the CXR, complicating interpretation further. In addition, clinicians are faced with reading high volumes of images every shift.

2.2 Business Domain Value

Automating Pneumonia screening in chest radiographs, providing affected area details through bounding box. Assist physicians to make better clinical decisions or even replace human judgement in certain functional areas of healthcare (eg, radiology).

With the help of AI techniques and also by relevant clinical support, we can unlock clinically relevant information hidden in the massive amount of data, which in turn can assist clinical decision making.

2.3 Objective of the work

- To build a model to identify whether CXR images have lung opacity or not.
- To build an algorithm to automatically locate lung opacities on chest radiographs.

2.4 About the given Data Set

2.4.1 The initial set of data

Medical images are stored in special format called DICOM files (*.dcm). They contain a combination of header, metadata as well as underlaying raw image arrays for pixel data.

The data and datasets are downloaded from https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/data

The data and the dataset contains:

1) **stage_2_detailed_class_info.csv** – contains attributes patientId and class information
2) **stage_2_train_labels.csv** – contains attributes patientId, x, y, width, height and Target
3) **stage_2_train_images** – 22684 images in .dcm format, to be used for training the model
4) **stage_2_test_images** – 3000 images in .dcm format.

2.4.2 Details about the two given csv files

Sl.No	File Name	Total no. of information's (rows)	No. of Attritubes	Attributes
1	stage_2_detailed_class_info.csv	30227	2	patientId, Class
2	stage_2_train_labels.csv	30227	6	patientId, x, y, width, height, Target

2.4.3 Details about the given images

Training Images : 26,684 images and supported by two csv files

Testing Images : 3,000 images

Observation : All the images are named with the patient-id

- **The testing images doesn't contain any information. So the training images should be splitted into training data and validation data in the ratio of 70:30**

2.5 Findings from the given dataset

Differences observed between training images and no. of information's given in the .csv files

- No of training images : 26,684
- No. of rows given in csv file (No. of patientId) : 30,227
- Differences Observed : 3,543

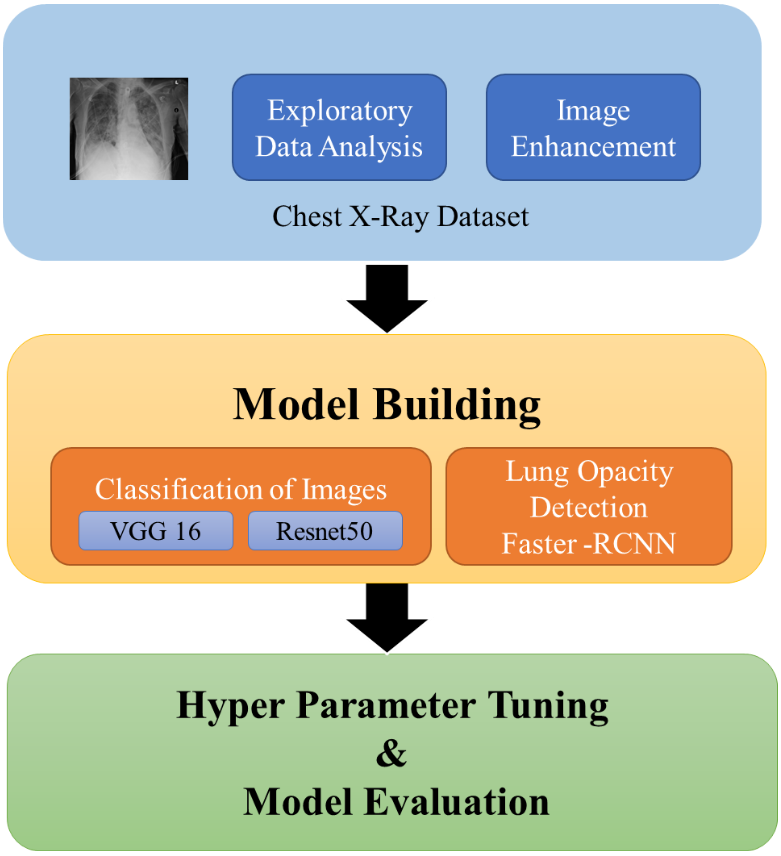
OVERVIEW OF THE FINAL PROCESS

3.1 Problem Methodology

Pneumonia is a life-threatening infectious disease affecting one or both lungs in humans commonly caused by bacteria. Pneumonia accounts for over 15% of all deaths of children under 5 years old internationally. In 2017, 920,000 children under the age of 5 died from the disease. It requires review of a chest radiograph (CXR) by highly trained specialists and confirmation through clinical history, vital signs and laboratory exams. Thus, developing an automatic system for detecting pneumonia would be beneficial for treating the disease. Due to the success of deep learning algorithms in analyzing medical images, Convolutional Neural Networks (CNNs) have gained much attention for disease classification. In addition, features learned by pre-trained CNN models on large-scale datasets are much useful in image classification tasks. In this work, we appraise the functionality of pre-trained CNN models utilized as feature-extractors followed by different classifiers for the classification of abnormal and normal chest X-Rays. Moreover, we also applied Faster RCNN over the pre-trained network to find the area belonging to lung opacity. We analytically determine the optimal model for the finding the boundary of the Lung Opacity.

3.2 Workflow

- Exploratory Data Analysis
- Visualizing the Chest X-Ray Images
- Image Pre-processing
- Model Building
- Hyper parameter tuning and Model Evaluation



3.3 Exploratory Data Analysis

Steps Involved in Exploratory Data Analysis

- **Analysing the Given .csv files**
- **Extracting Metadata information from the training Images**
- **Performing EDA from the Metadata Information**
- **Visualizing the given Images.**

3.3.1 Findings from the given two csv files after concatenating

Merging the two dataframes

```
[ ] dataframe1.set_index("patientId", inplace = True)
    dataframe2.set_index("patientId", inplace = True)
    combined_df = pd.concat([dataframe1, dataframe2], axis=1, join='inner')
    combined_df.reset_index(inplace=True)
    combined_df
```

	patientId	class	x	y	width	height	Target
0	0004cfab-14fd-4e49-80ba-63a80b6bddd6	No Lung Opacity / Not Normal	NaN	NaN	NaN	NaN	0
1	00313ee0-9eaa-42f4-b0ab-c148ed3241cd	No Lung Opacity / Not Normal	NaN	NaN	NaN	NaN	0
2	00322d4d-1c29-4943-afc9-b6754be640eb	No Lung Opacity / Not Normal	NaN	NaN	NaN	NaN	0
3	003d8fa0-6bf1-40ed-b54c-ac657f8495c5	Normal	NaN	NaN	NaN	NaN	0
4	00436515-870c-4b36-a041-de91049b9ab4	Lung Opacity	264.0	152.0	213.0	379.0	1
...	...	...	...	...	...	...	...
30222	c1ec14ff-f6d7-4b38-b0cb-fe07041cbdc8	Lung Opacity	185.0	298.0	228.0	379.0	1
30223	c1edf42b-5958-47ff-a1e7-4f23d99583ba	Normal	NaN	NaN	NaN	NaN	0
30224	c1f6b555-2eb1-4231-98f6-50a963976431	Normal	NaN	NaN	NaN	NaN	0
30225	c1f7889a-9ea9-4acb-b64c-b737c929599a	Lung Opacity	570.0	393.0	261.0	345.0	1
30226	c1f7889a-9ea9-4acb-b64c-b737c929599a	Lung Opacity	233.0	424.0	201.0	356.0	1

30227 rows × 7 columns

- **The dataframe contains 30227 informations**
- **Attributes: class, x, y, width, height and Target**

Information about the missing values

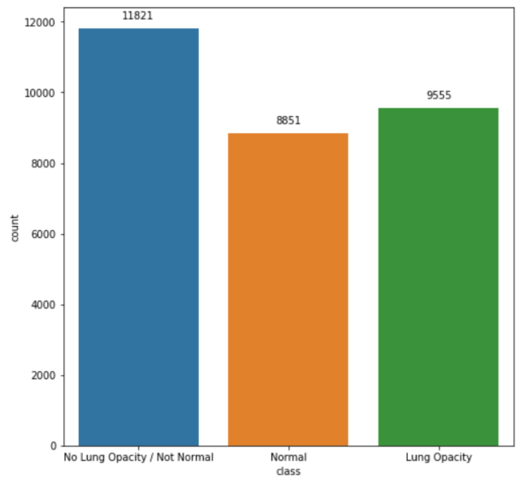
Attributes	Non-Null Count	Null Values count	Dtype
patientId	30227	0	object
class	30227	0	object
x	9555	20672	float64
y	9555	20672	float64
width	9555	20672	float64
height	9555	20672	float64
Target	30227	0	int64

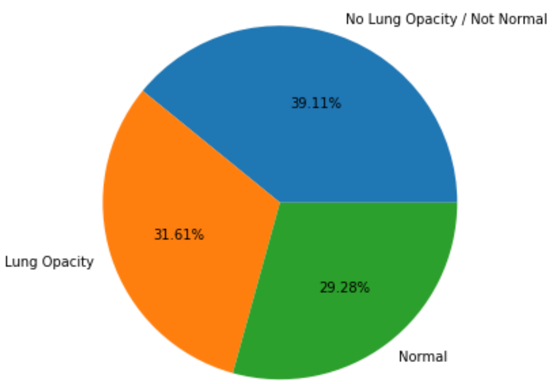
- **No missing values observed in patientId, class and Target**
- **Equal no. of missing values observed in the columns x, y, width and height**

3.3.1.1 Exploring the attribute "class"

- Three different classes of images were observed
  - No Lung Opacity/ Not Normal – 11821 information
  - Lung Opacity – 9555 information
  - Normal – 8851 information
- All the three classes have almost equal distributions

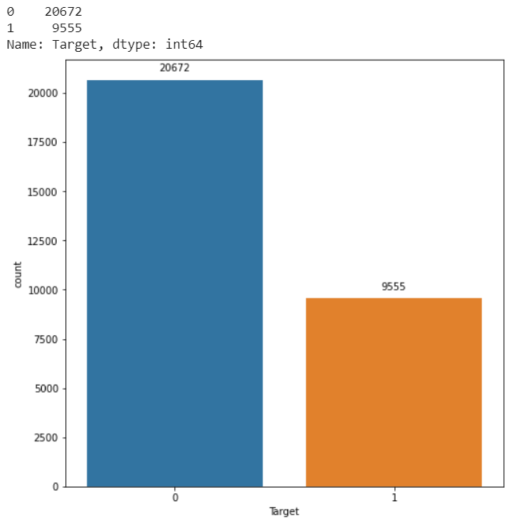
```
No Lung Opacity / Not Normal    11821
Lung Opacity                     9555
Normal                           8851
Name: class, dtype: int64
```

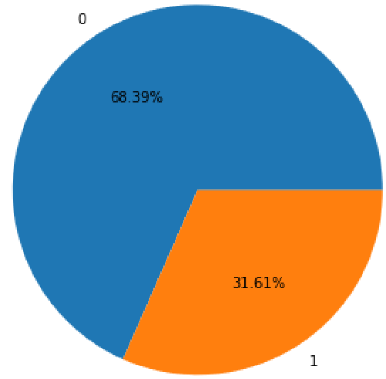




3.3.1.2 Exploring the attribute "Target"

- Two different targets of images were observed
    - Target= 0 –20672 information
    - Target=1 – 9555 information
- The given information is more biased towards Target=0

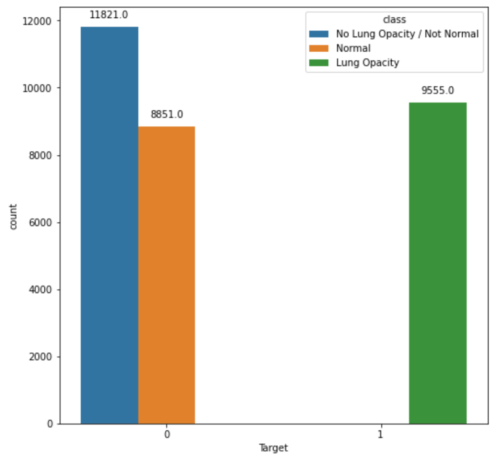




3.3.1.3 Relationship between "Target" and "class"

- Target = 0 includes two classes " No Lung Opacity/ Not Normal" and "Normal"
- Class "Lung Opacity" is categorized as Target=1

Target : 0 **20672 counts**		**Target: 1** **9555 counts**
Class: **No Lung Opacity/ Not Normal** **11821 counts**	**Class:** **Normal** **8851 counts**	**Class: Lung Opacity** **9555 counts**



3.3.1.4 Analyzing the target attribute separately: Target=0

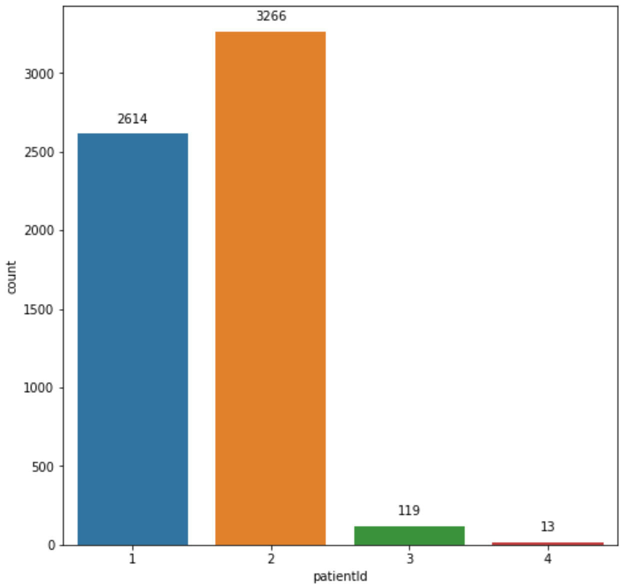
- Total no.of Unique patientId: 20672
- Each patientId contains only one sets of information
- When Target = 0 ➔ Bounding box values are not available [x, y, width, height]

Column	Non-Null Count	Null Values count	Dtype
patientId	20672	0	object
class	20672	0	object
x	0	20672	float64
y	0	20672	float64
width	0	20672	float64
height	0	20672	float64
Target	20672	0	int64

3.3.1.5 Analyzing the target attribute separately: Target=1

- Total no.of Unique patientId: 6012
- Each patientId contains one and more than one informations (upto 4)
- When Target = 1 ➔ No missing values observed

Column	Non-Null Count	Null Values count	Dtype
patientId	9555	0	object
class	9555	0	object
x	9555	0	float64
y	9555	0	float64
width	9555	0	float64
height	9555	0	float64
Target	9555	0	int64



- PatientId with only one sets of bounding box information – 2614
- PatientId with two sets of bounding box information – 3266
- PatientId with three sets of bounding box information – 119
- PatientId with four sets of bounding box information – 13

- Total no unique patient Id - 6012

3.3.2 Exploring the metadata information

The given dicom files contains the following metadata information

```
(0008, 0005) Specific Character Set              CS: 'ISO_IR 100'
(0008, 0016) SOP Class UID                       UI: Secondary Capture Image Storage
(0008, 0018) SOP Instance UID                    UI: 1.2.276.0.7230010.3.1.4.8323329.24506.1517874454.871180
(0008, 0020) Study Date                          DA: '19010101'
(0008, 0030) Study Time                          TM: '000000.00'
(0008, 0050) Accession Number                    SH: ''
(0008, 0060) Modality                            CS: 'CR'
(0008, 0064) Conversion Type                     CS: 'WSD'
(0008, 0090) Referring Physician's Name          PN: ''
(0008, 103e) Series Description                  LO: 'view: AP'
(0010, 0010) Patient's Name                      PN: 'f7909c0c-c9f0-4c93-be7f-113926850ac3'
(0010, 0020) Patient ID                          LO: 'f7909c0c-c9f0-4c93-be7f-113926850ac3'
(0010, 0030) Patient's Birth Date                DA: ''
(0010, 0040) Patient's Sex                       CS: 'F'
(0010, 1010) Patient's Age                       AS: '81'
(0018, 0015) Body Part Examined                  CS: 'CHEST'
(0018, 5101) View Position                       CS: 'AP'
(0020, 000d) Study Instance UID                  UI: 1.2.276.0.7230010.3.1.2.8323329.24506.1517874454.871179
(0020, 000e) Series Instance UID                 UI: 1.2.276.0.7230010.3.1.3.8323329.24506.1517874454.871178
(0020, 0010) Study ID                            SH: ''
(0020, 0011) Series Number                       IS: "1"
(0020, 0013) Instance Number                     IS: "1"
(0020, 0020) Patient Orientation                 CS: ''
(0028, 0002) Samples per Pixel                   US: 1
(0028, 0004) Photometric Interpretation          CS: 'MONOCHROME2'
(0028, 0010) Rows                                US: 1024
(0028, 0011) Columns                             US: 1024
(0028, 0030) Pixel Spacing                       DS: [0.168, 0.168]
(0028, 0100) Bits Allocated                      US: 8
(0028, 0101) Bits Stored                         US: 8
(0028, 0102) High Bit                            US: 7
(0028, 0103) Pixel Representation                US: 0
(0028, 2110) Lossy Image Compression             CS: '01'
(0028, 2114) Lossy Image Compression Method      CS: 'ISO_10918_1'
(7fe0, 0010) Pixel Data                          OB: Array of 122960 elements
```

- All the given images have been of same size (1024, 1024)
- The Patient's Age, Patient's Sex, View Position, Study Date is extracted from the dicom file for analysis

	patientId	Sex	Age	Position	Date
0	f7a37b72-fda5-4adc-b3b0-968c923bc1c6	F	35	PA	19010101
1	f78f155c-0caf-466b-ae36-1f365861b01d	M	33	PA	19010101
2	f77b0afe-0085-4ee0-afad-a1e9fda8fe65	F	29	AP	19010101
3	f760c946-a103-4991-a6e5-ff60c24cd99f	M	46	AP	19010101
4	f6f6cb82-f83b-4abd-88b1-4ad5e9436bfd	F	38	PA	19010101
...	...	...	...	...	...
26679	095aecad-1618-4b23-b7b9-342c25c4666b	M	40	PA	19010101
26680	090abf67-66f4-413b-ac4a-bfad36e07e1a	F	31	AP	19010101
26681	094eed38-9c5b-4042-936d-344ccec4c3cc	F	65	AP	19010101
26682	091e90d0-bdb7-4815-935f-5b52b93ddbe6	F	45	PA	19010101
26683	091cc2b7-8ba6-4fce-8ae2-7547117dbddf	M	62	AP	19010101

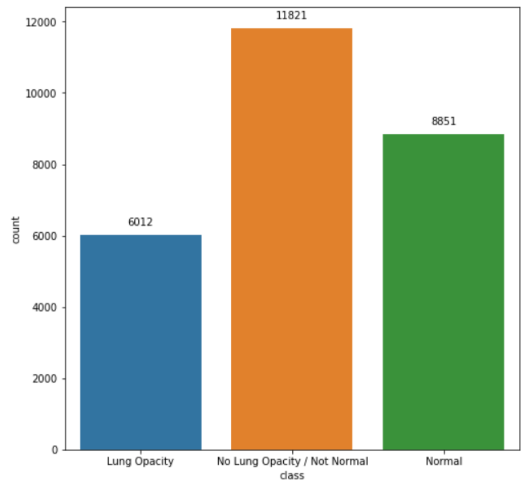
26684 rows × 5 columns

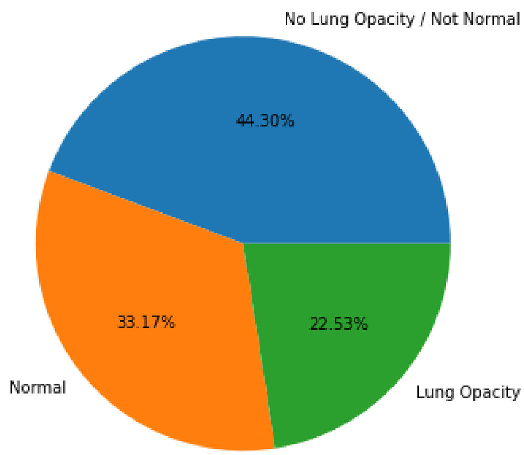
- It is observed that no.of images provided is 26684 and no.of unique patient id is 26684.
- So each patient have one cxr image. Mutiple chest X-ray for a single patient is not available. So ignoring the Study Date from the analysis.

3.3.2.1 Exploring the attribute "class" from the metadata

- Three different classes of images were observed
  - No Lung Opacity/ Not Normal – 11821 information
  - Lung Opacity – 6012 information
  - Normal – 8851 information
- All the three classes have almost equal distributions

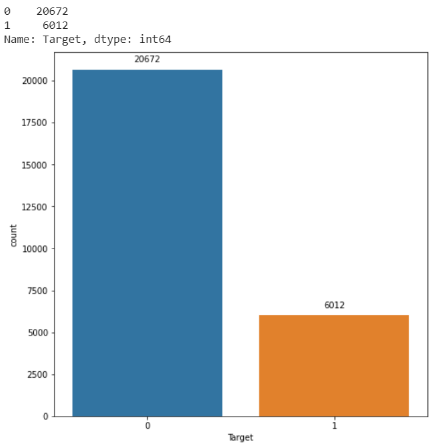
```
No Lung Opacity / Not Normal     11821
Normal                            8851
Lung Opacity                      6012
Name: class, dtype: int64
```

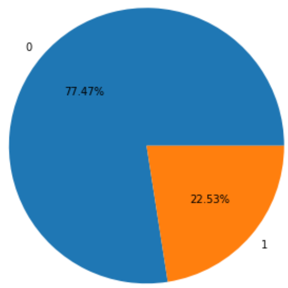




3.3.2.2 Exploring the attribute "Target" from the metadata

- Two different targets of images were observed
    - Target= 0 –20672 information
    - Target=1 – 6012 information
- The given information is more biased towards Target=0

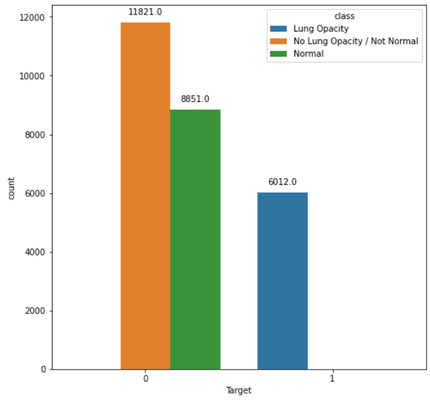




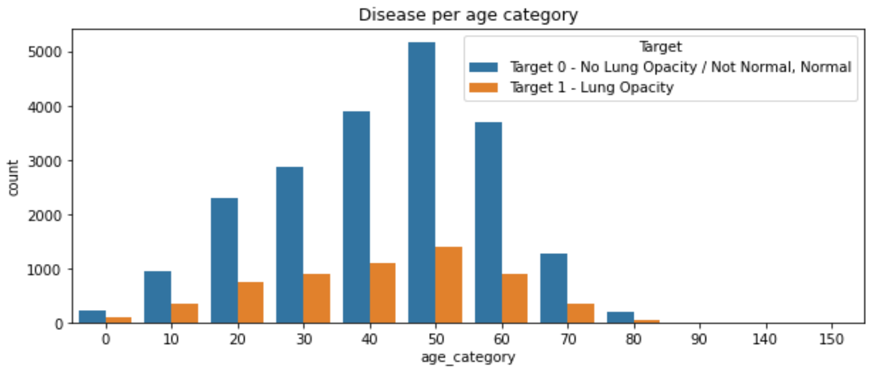
3.3.2.3 Relationship between "Target" and "class"

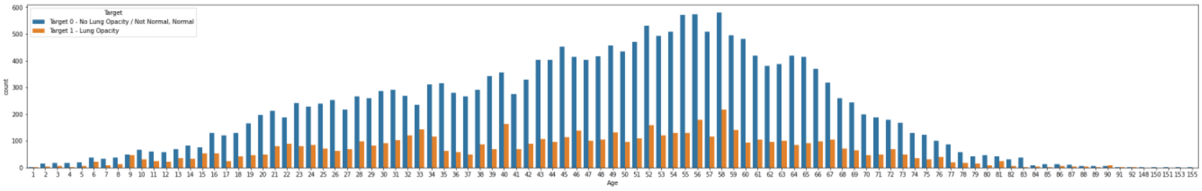
- Target = 0 includes two classes " No Lung Opacity/ Not Normal" and "Normal"
- Class "Lung Opacity" is categorized as Target=1

Target : 0 **20672 counts**		**Target: 1** **9555 counts**
Class: No Lung Opacity/ Not Normal 11821 counts	**Class: Normal 8851 counts**	**Class: Lung Opacity 6012 counts**



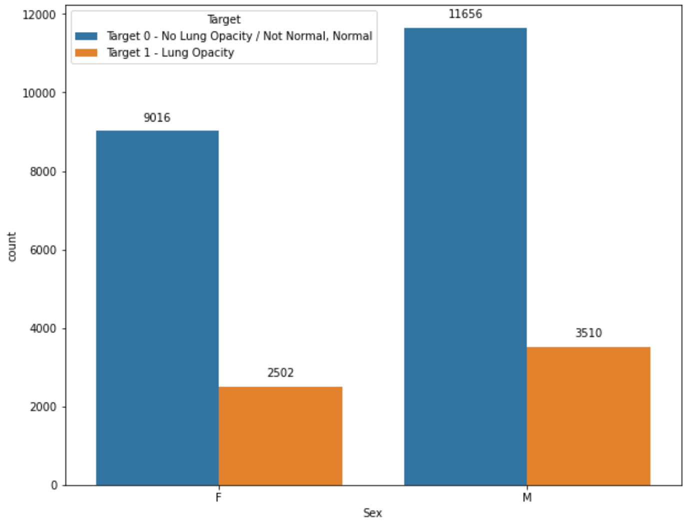
3.3.2.4 "Age distribution" form the given dataset





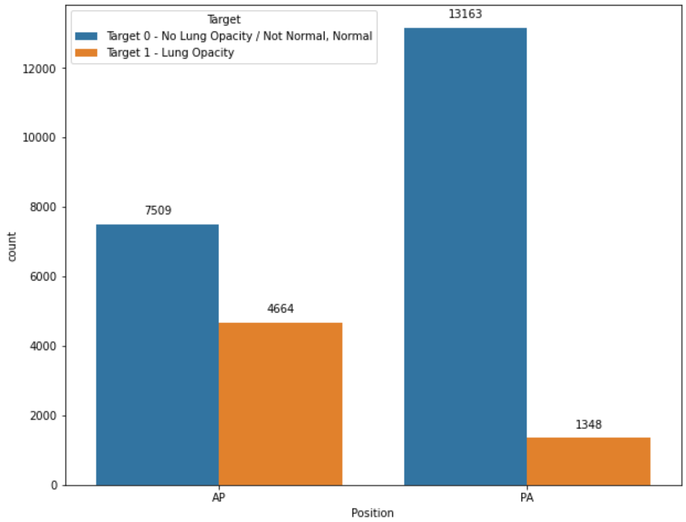
➢ More no. of age distribution lies within the range of 44 to 55.

3.3.2.5 Relationship between “Sex” and “Target”



- Almost equal number of images are provides from Male and Female

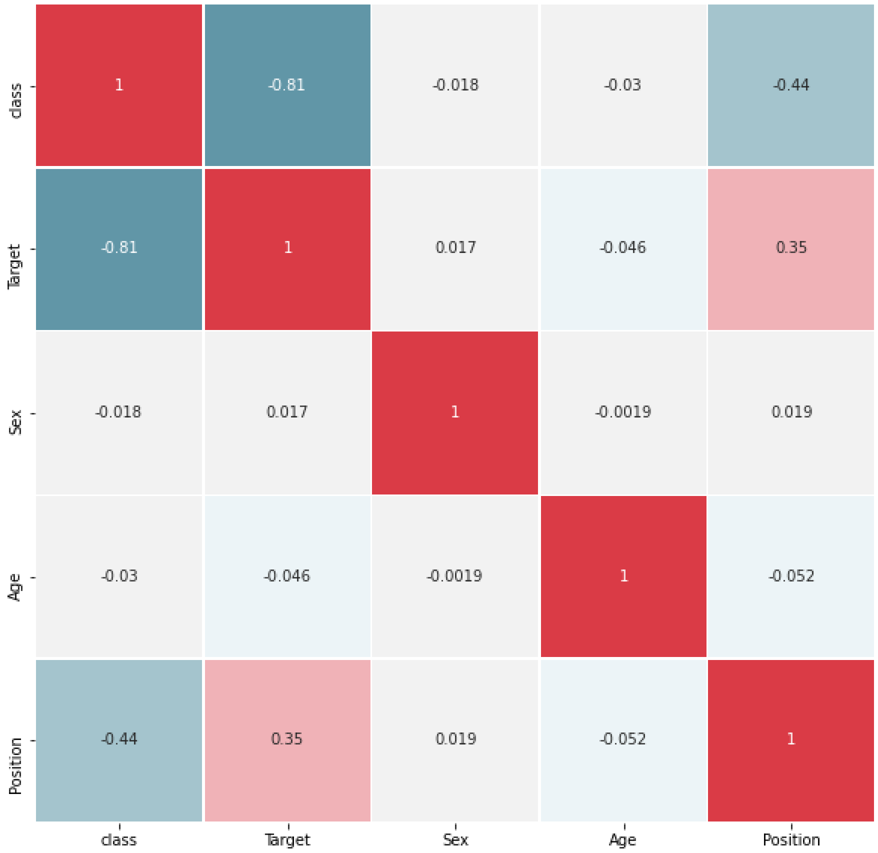
3.3.2.6 Relationship between “View Position” and “Target”



- View postion – AP (Anterior Posterior) having more possible chance of occurring Lung Opacity

3.3.3 Correlation map

- Correlation map between the attributes "Target", "Class", "Age", "Sex" and "View Position" is plotted below

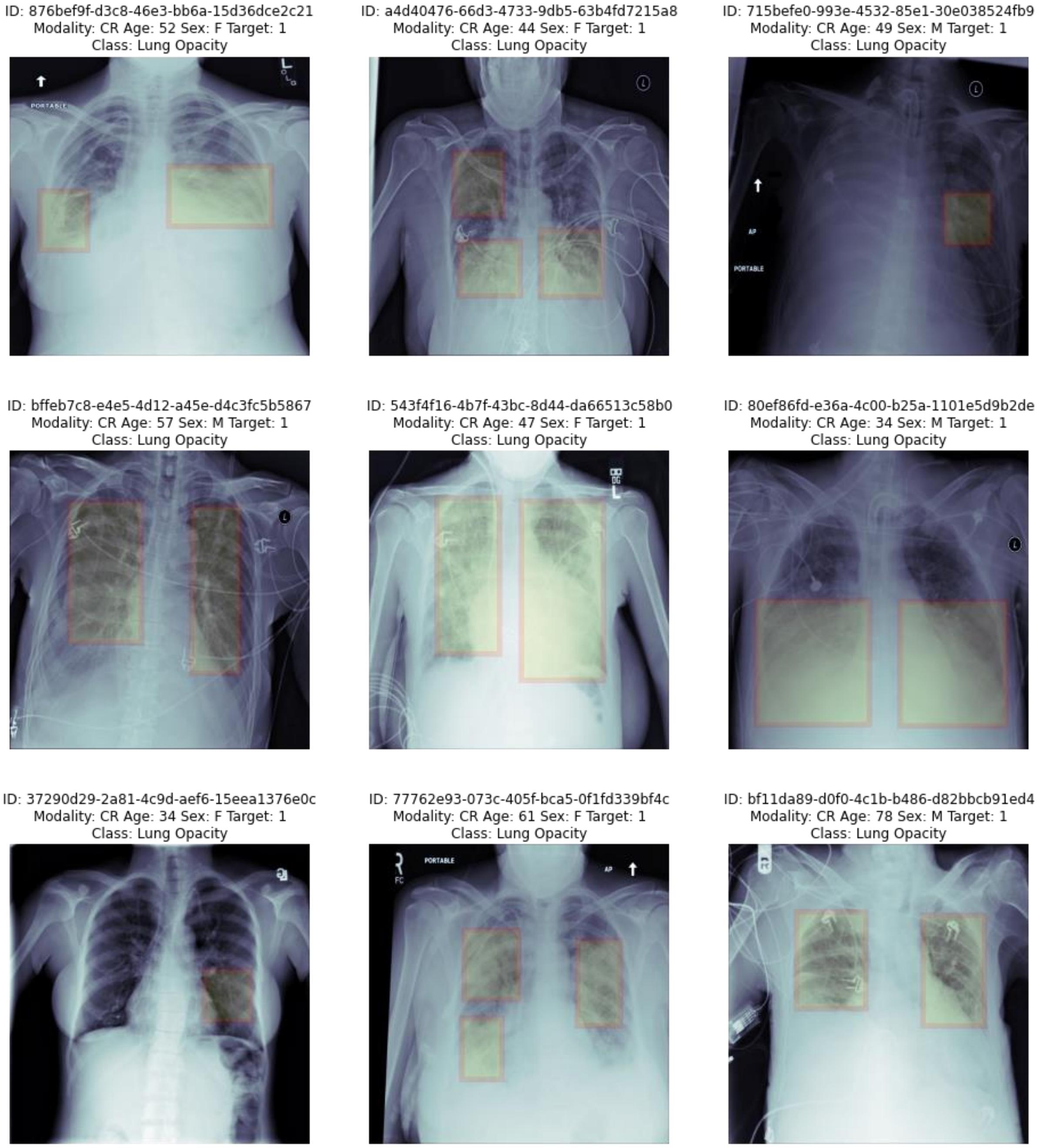


Summary after EDA

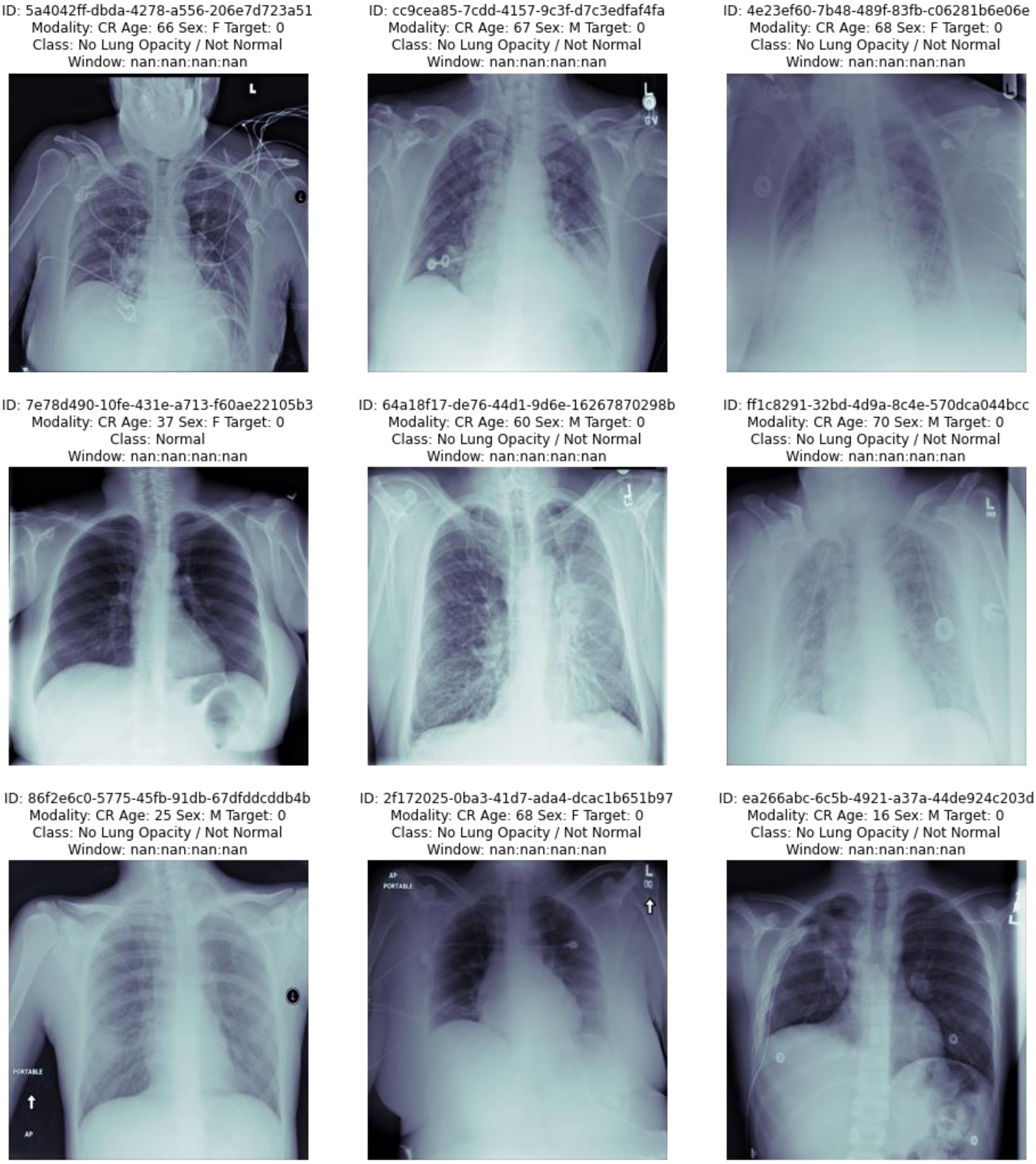
- **3543 difference observed between the information provided in the csv file and the available images.**
- **It is due to patient with lung opacity have multiple bounding box values. (i.e some patients have lung opacity at more than one places in the lungs)**
- **No missing values were observed and the given data can be used as it is for further modelling.**

3.4 Visualizing the given images

- The images with Target =1 and class = Lung Opacity
- Form the given bounding box information, the lung opacity is highlighted with a bounding box
- A patient may have lung opacity in more than one places.



- The images with Target =0 and class = No Lung Opacity/ Not Normal & Normal
- Bounding box information is not provided in this case



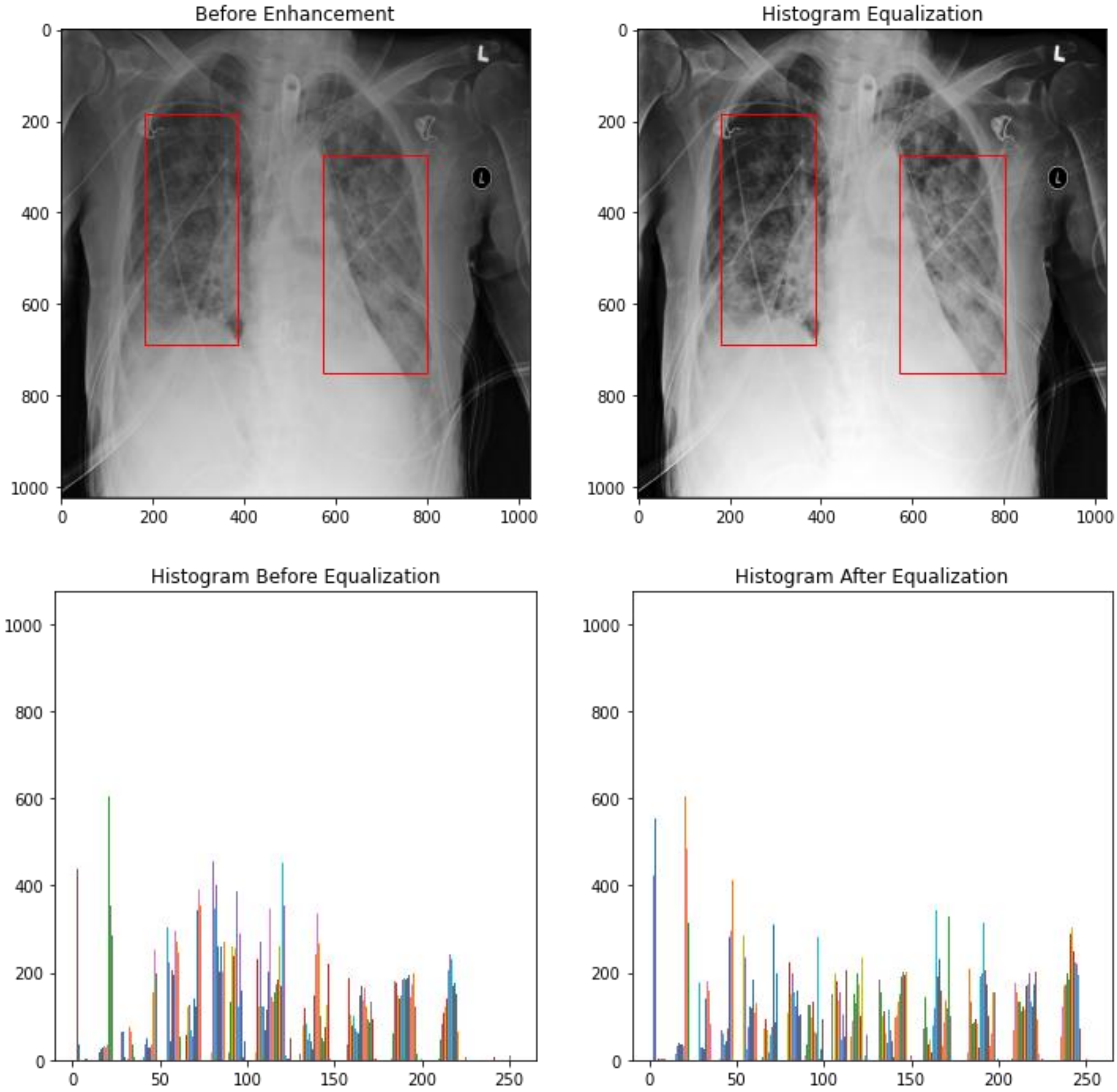
3.5 Image Pre-Processing

The given CXR images are having low contrast. So the contrast of the image is enhanced by using histogram equalization.

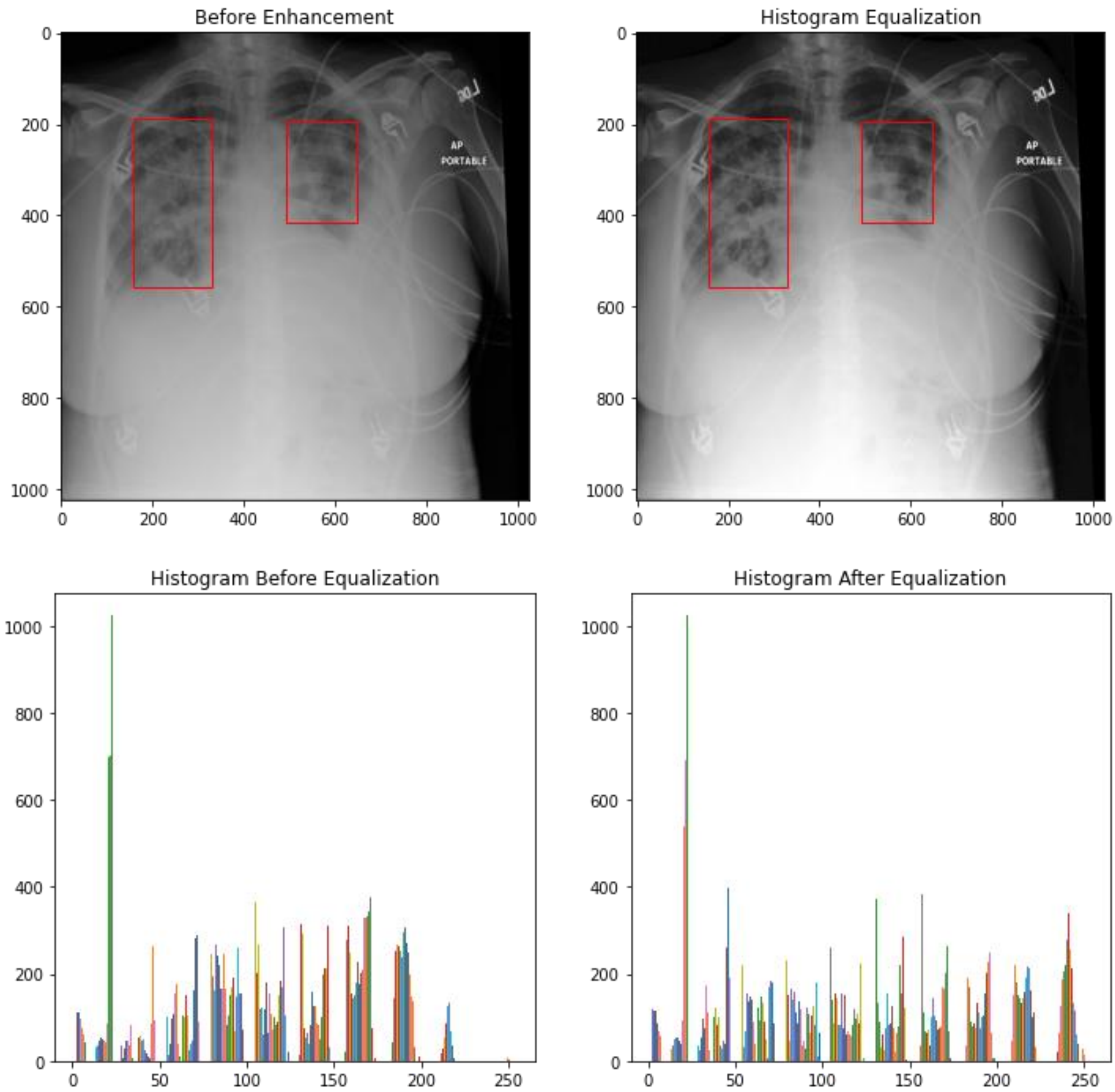
3.5.1 Histogram Equalization

Histogram Equalization is a computer image processing technique used to improve contrast in images. It is a method in image processing for contrast adjustment using the image's histogram. Histogram equalization is simple & best method for image enhancement. It provides better quality of images without loss of any information.

3.5.2 Results Observed after Histogram Equalization



File Name: 00a05408-8291-4231-886e-13763e103161.png



File Name: 00f08de1-517e-4652-a04f-d1dc9ee48593.png

The Result shows that the histogram equalized images provide better clarity than the original image.

3.6 Model Building

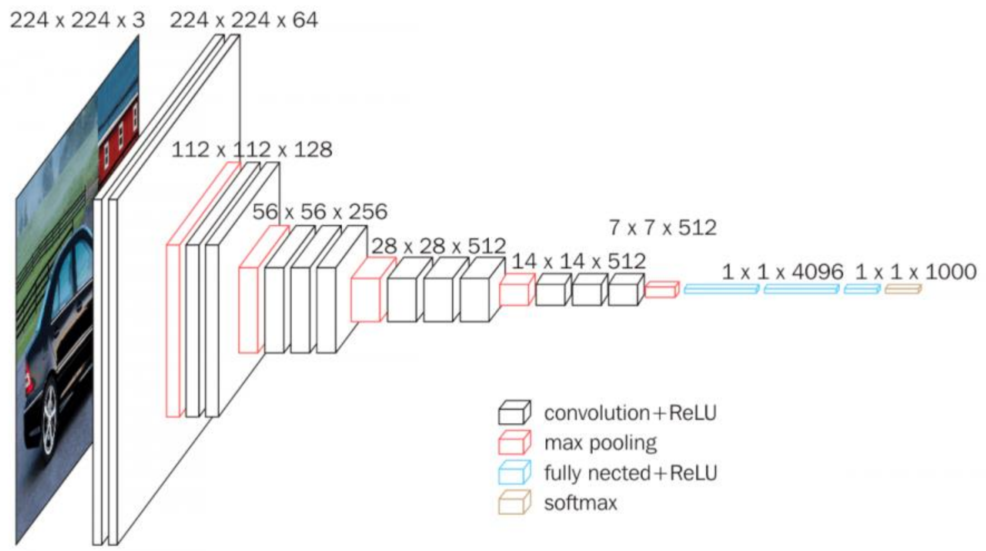
Steps Involved in Model Building

- **Classification using VGG16 and Resnet50**
- **Observing Performance Parameters**
- **Lung Opacity Detection using Faster RCNN Algorithm with VGG16 and Resnet50 as Backbone**

Deep learning is fast becoming a key instrument in artificial intelligence applications. For example, in areas such as computer vision, natural language processing, and speech recognition, deep learning has been producing remarkable results. Therefore, there is a growing interest in deep learning.

3.6.1 VGG16 Architecture

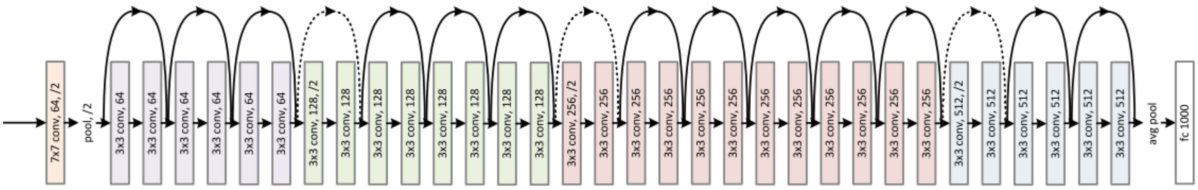
VGG16 is a convolution neural net (CNN) architecture which was used to win ILSVR (Imagenet) competition in 2014. It is considered to be one of the excellent vision model architecture till date.



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. It follows this arrangement of convolution and max pool layers consistently throughout the whole architecture. In the end it has 2 FC(fully connected layers) followed by a softmax for output. The 16 in VGG16 refers to it has 16 layers that have weights. This network is a pretty large network and it has about 138 million (approx) parameters.

3.6.2 Resnet50 Architecture

ResNet-50 is a convolutional neural network that is 50 layers deep. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224-by-224.



ResNet first introduced the concept of skip connection. Resnet stack convolution layers as before but we now also add the original input to the output of the convolution block. This is called skip connection

Advantage of Skip connections work here:

1. They mitigate the problem of vanishing gradient by allowing this alternate shortcut path for gradient to flow through
2. They allow the model to learn an identity function which ensures that the higher layer will perform at least as good as the lower layer, and not worse

3.6.3 Classification of CXR Images

The goal of image classification

- Is to classify the images of the three classes – Normal, No Lung Opacity / Not Normal and Lung Opacity by considering the 'class' attribute.
- Is to classify the images into two classes – Target 0 and Target 1 by considering the 'target' attribute

For classification purpose we used pre-trained model is a saved network that was previously trained on a large dataset, typically on a large-scale image-classification task.

The intuition behind transfer learning for image classification is that if a model is trained on a large and general enough dataset, this model will effectively serve as a generic model of the visual world. Transfer Learning can then take advantage of these learned feature maps without having to start from scratch by training a large model on a large dataset.

We have used VGG16 and Resnet50 for image classification with pretrained "Imagenet" weights

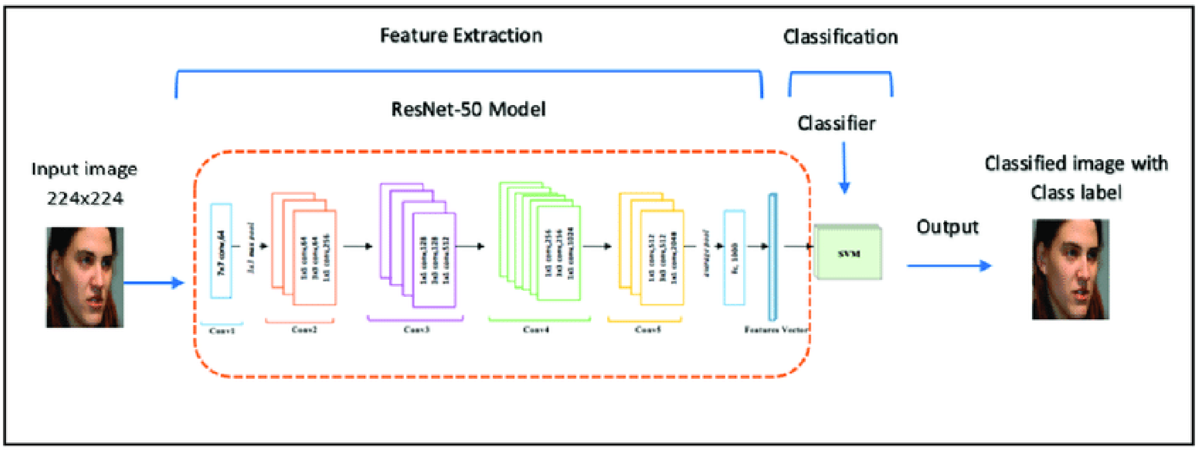
We have customized pretrained model for Lung Opacity Detection with the following steps

1. Feature Extraction: Using the representations learned by a previous network to extract meaningful features from new samples. Since "Image Net classification task" is entirely different from "Medical Images" the last layer from the pretrained network is removed and additional layers we added with the pretrained network.

Added a new classifier, which will be trained from scratch, on top of the pretrained model so that the network can repurpose the feature maps learned previously for the dataset.

You do not need to (re)train the entire model. The base convolutional network already contains features that are generically useful for classifying pictures. However, the final, classification part of the pretrained model is specific to the original classification task, and subsequently specific to the set of classes on which the model was trained.

2. Fine-Tuning: Unfreeze a few of the top layers of a frozen model base and jointly train both the newly-added classifier layers and the last layers of the base model. This allows us to "fine-tune" the higher-order feature representations in the base model in order to make them more relevant for the specific task.



We have done image classification by two methodologies

First Methodology:

1. Converting the input images into enhanced images
2. Then resized each images of the size of 300X300
3. Created a .npy file out of it. The size of the .npy file is (26684x300x300x3)
4. Extracted the target attribute (y) from the given .csv file
5. Composing the model.

    a. Load in the pretrained base model (VGG16 and Resnet50) and pretrained weights ("image net weights")

    b. Stack the classification layers on top

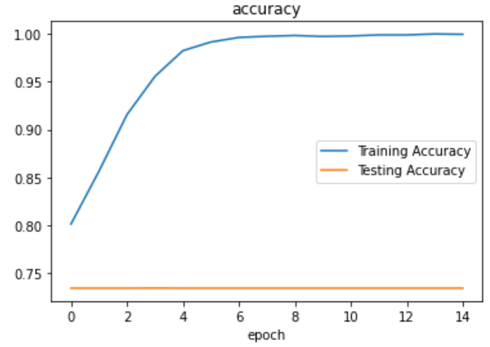
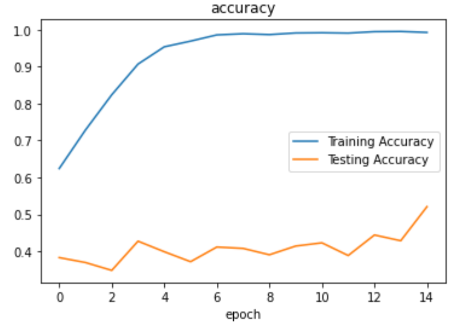
6. Training the model
    - By trying out the two different optimizers (Adam and RMS Prop)
    - With different Learning rate (lr=0.0001, 0.001 and 2e-5 etc)

7. Evaluating the model
    - While evaluating the model it is observed Resnet50 performs better than VGG16, but still the model performs poorly.
    - The model seems to be overfit. The reason behind that is we didn't involve methodologies such as data augmentation. Moreover, the given data set is biased over target_0

Evaluating the Model:

Evaluation Parameters	VGG16 with the attribute "Target"	VGG16 with the attribute "Class"
Training Accuracy	98.82%	96.32%
Validation Accuracy	72.02%	33.42%
Recall	49.07%	34.16 %
Precision	36.56 %	41.42 %
F1 Score	41.90 %	20.06 %
Confusion Matrix	[[5761 120] [2120 4]]	[[308 3005 22] [200 2336 10] [228 1864 32]]

The model is evaluated over the attribute "Target" and "class". The accuracy for the model built over the attribute "Target" was higher compared to the attribute "Class". But still the model performance is since the input images are biased over "Target 0"

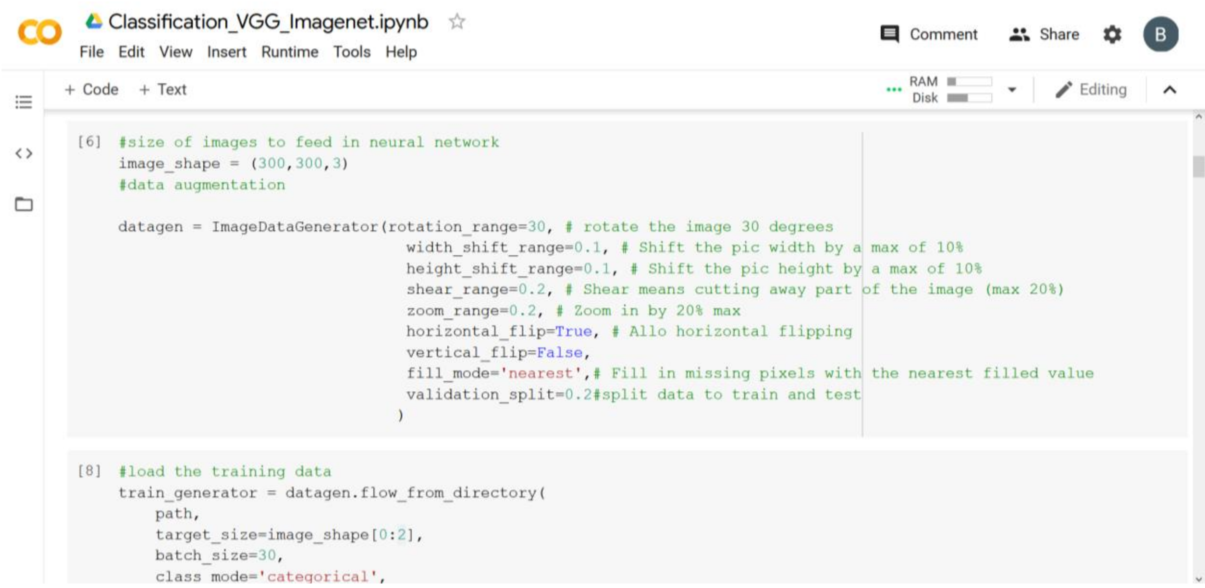
Evaluation Parameters	Resnet50 with the attribute "Target"	Resnet50 with the attribute "Class"
Training Accuracy	99.74%	99.46%
Validation Accuracy	73.47%	34.48%
Recall	50 %	35.68 %
Precision	36.73 %	47.91 %
F1 Score	42.35 %	21.11 %
Confusion Matrix	[[5881 0] [2124 0]]	[[178 3138 19] [1 2545 0] [226 1861 37]]
		

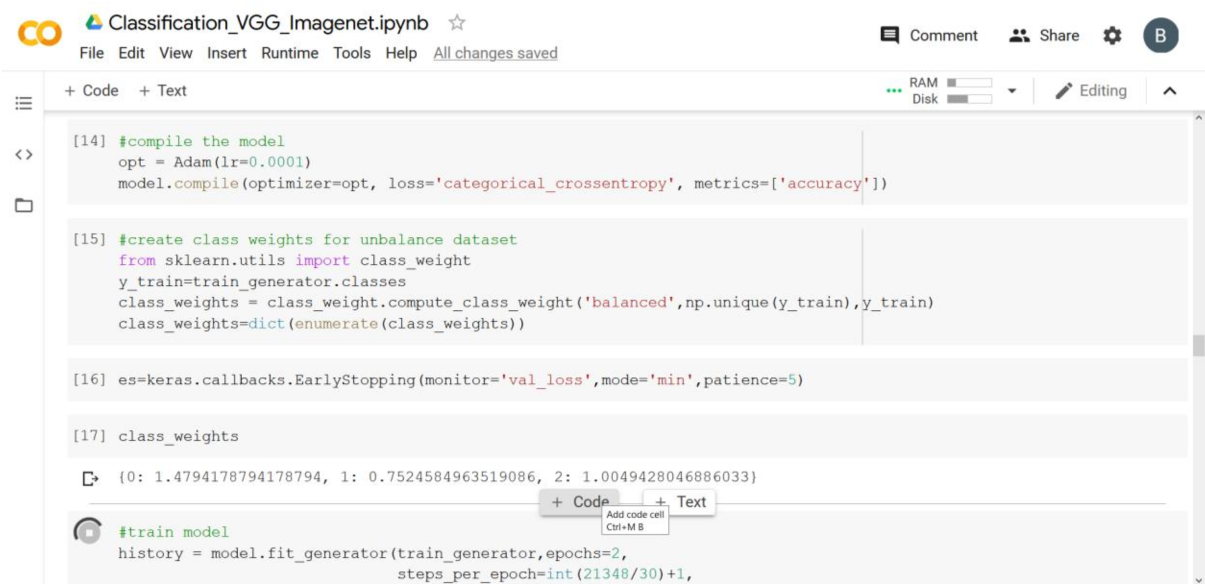
Findings

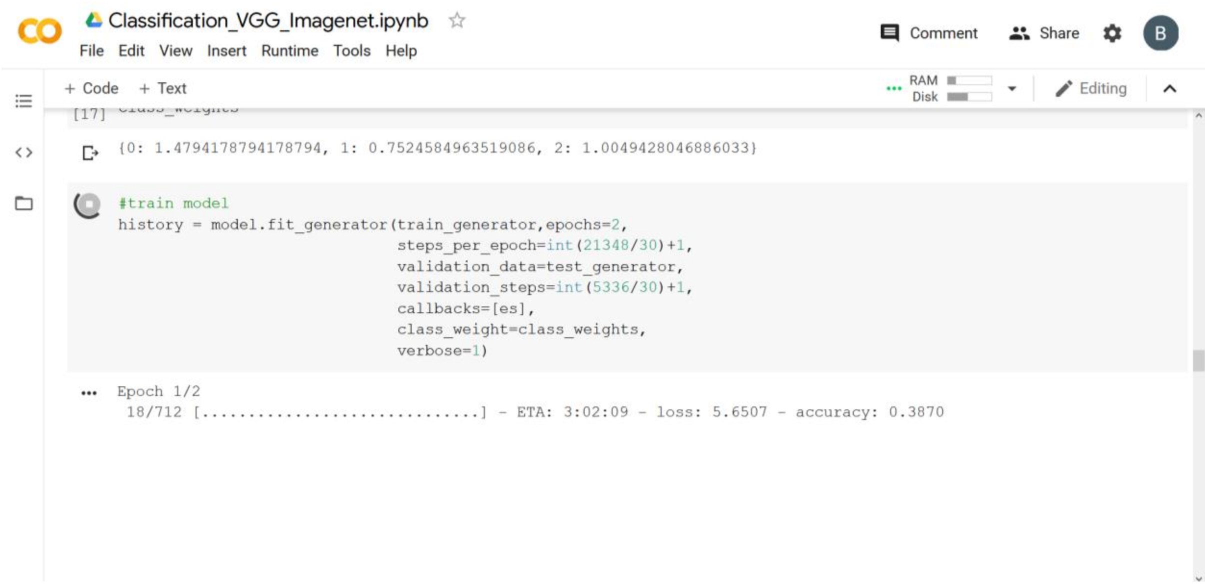
- Resnet50 Preforms slightly better than VGG16
- The model seems to be overfit.
- The performance of the model can't able to improve further if the number of epochs increased further and even by unfreezing the few more top layers of the pretrained network.

Second Methodology:

1. Converting the input images into enhanced images
2. Then resized each images of the size of 300X300
3. Building an input pipeline, in this case using Keras ImageDataGenerator with data augumentation and by balancing the classes.
4. Composing the model.
    - Load in the pretrained base model (VGG16 and Resnet50) and pretrained weights ("image net weights")
    - Stack the classification layers on top
5. Training the model







6. Evaluating the model
    - Can’t able to evaluate because of very high training time

Findings

- Can’t able to evaluate the model due to very high training time.
- This is because of total number of Images 26,684 along with data augmentation
- So it is decided to proceed further for the bounding box detection by considering the images having Lung Opacity (Target 1).

Note : Bounding Box information’s is not available for the images with Target 0 (Normal, No Lung Opacity/ Not Normal)

3.6.4. Faster RCNN for Lung Opacity Detection (Object Detection)

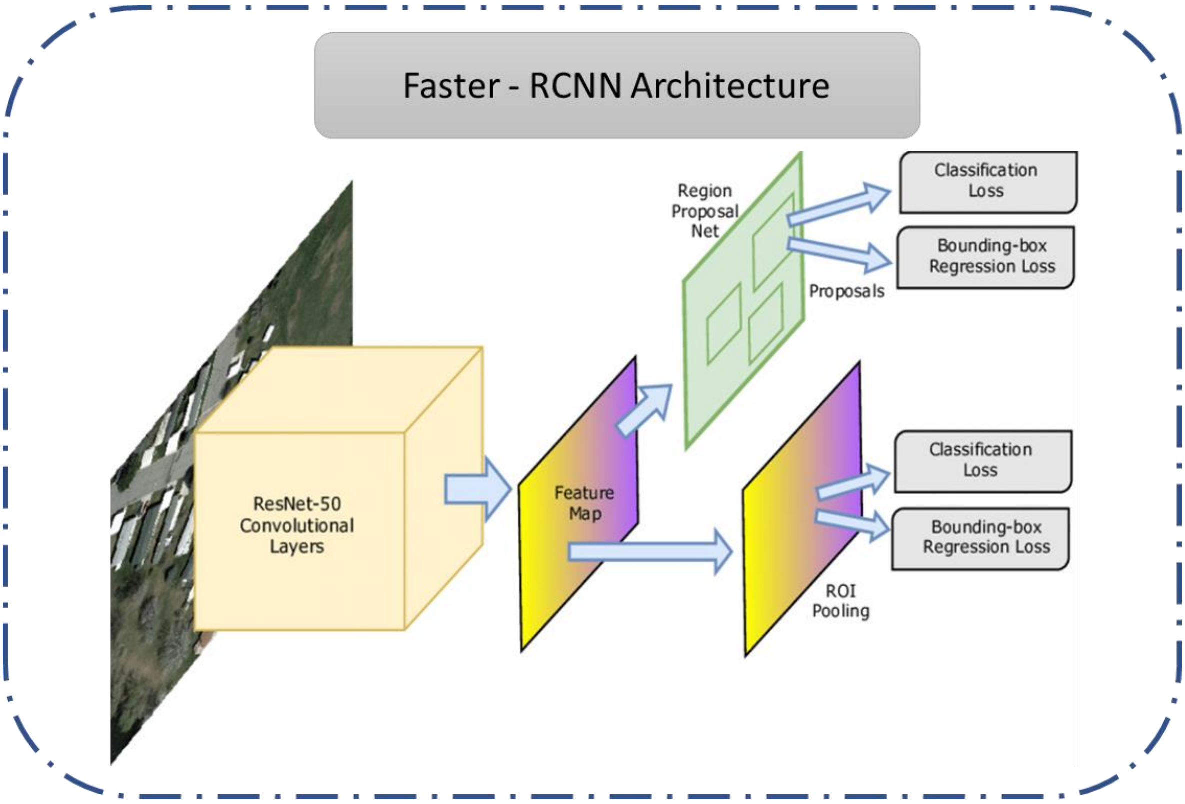
Since Images with Lung Opacity alone have the bounding box values and due to Google Colab usage limits, it is decided to proceed further for Object Detection for the images having "Lung Opacity".

We tried Implementing Faster-RCNN using Keras with VGG-16 and Resnet50 as backbone, owing to high accuracy and faster results in object detection technique.

3.6.4.1 About Faster RCNN

R-CNN is the first step for Faster R-CNN. It uses **search selective** to find out the regions of interests and passes them to a ConvNet. It tries to find out the areas that might be an object by combining similar pixels and textures into several rectangular boxes. The R-CNN paper uses 2,000 proposed areas (rectangular boxes) from search selective. Then, these 2,000 areas are passed to a pre-trained CNN model. Finally, the outputs (feature maps) are passed to a SVM for classification. The regression between predicted bounding boxes (bboxes) and ground-truth bboxes are computed.

Fast R-CNN moves one step forward. Instead of applying 2,000 times CNN to proposed areas, it only passes the original image to a pre-trained CNN model once. Search selective algorithm is computed base on the output feature map of the previous step. Then, ROI pooling layer is used to ensure the standard and pre-defined output size. These valid outputs are passed to a fully connected layer as inputs. Finally, two output vectors are used to predict the observed object with a softmax classifier and adapt bounding box localisations with a linear regressor.



Faster R-CNN makes further progress than Fast R-CNN. Search selective process is replaced by **Region Proposal Network** (RPN). As the name revealed, RPN is a network to propose regions.

3.6.4.2 Region Proposal Network

To begin with, Faster RCNN takes the feature maps from CNN and passes them on to the Region Proposal Network. RPN uses a sliding window over these feature maps, and at each window, it generates K Anchor boxes of different shapes and sizes.

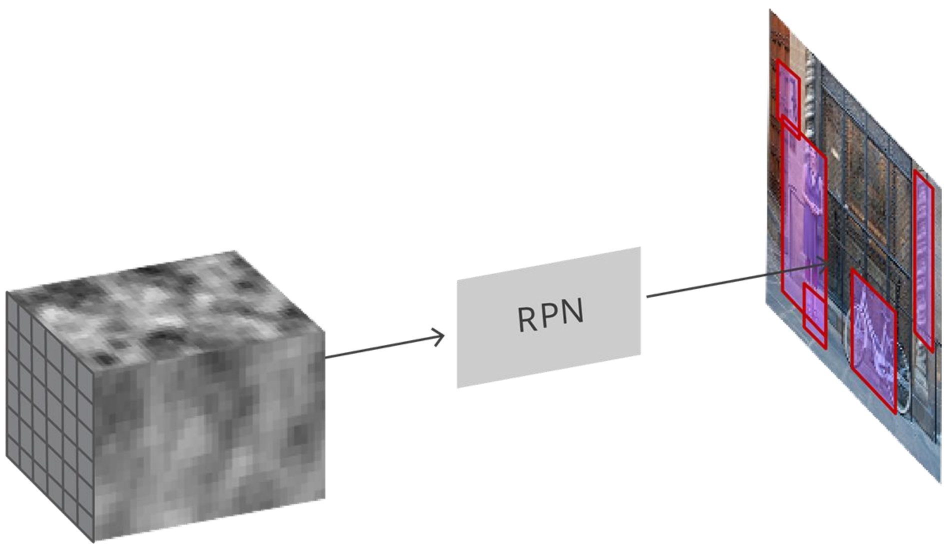
Anchor boxes are fixed sized boundary boxes that are placed throughout the image and have different shapes and sizes. For each anchor, RPN predicts two things:

- The first is the probability that an anchor is an object (it does not consider which class the object belongs to)
- Second is the bounding box regressor for adjusting the anchors to better fit the object

As we mentioned before, the RPN takes all the reference boxes (anchors) and outputs a set of good proposals for objects. It does this by having two different outputs for each of the anchors.

The first one is the probability that an anchor is an object. An "objectness score", if you will. Note that the RPN doesn't care what *class* of object it is, only that it does in fact look like an object (and not background). We are going to use this objectness score to filter out the bad predictions for the second stage. The second output is the bounding box regression for adjusting the anchors to better fit the object it's predicting.

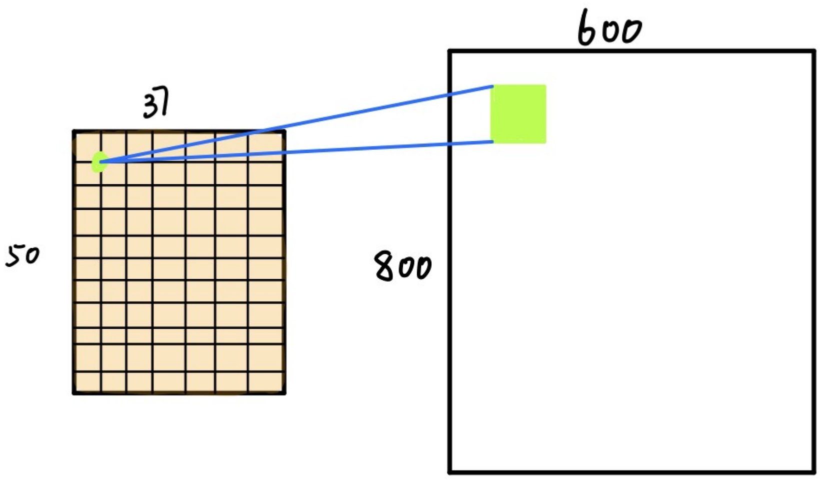
The RPN is implemented efficiently in a fully convolutional way, using the convolutional feature map returned by the base network as an input. First, we use a convolutional layer with 512 channels and 3x3 kernel size and then we have two parallel convolutional layers using a 1x11x11x1 kernel, whose number of channels depends on the number of anchors per point.



The RPN Takes the convolutional Feature Map and Generates Proposals over the image

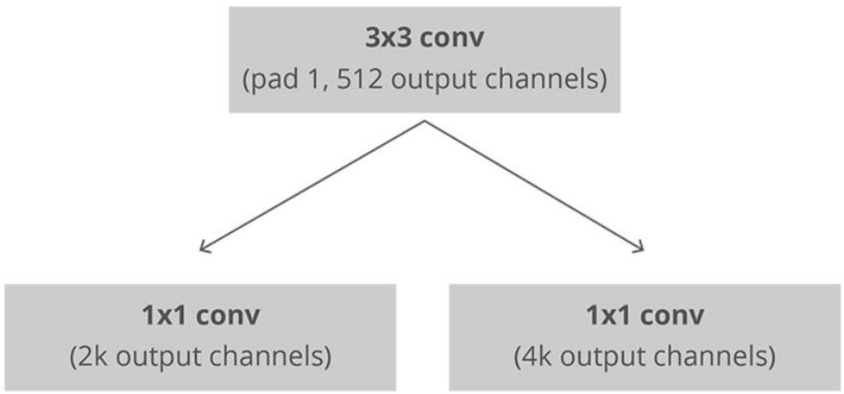
For instance, after getting the output feature map from a pre-trained model (VGG-16), if the input image has 600x800x3 dimensions, the output feature map would be 37x50x256 dimensions.

Each point in 37x50 is considered as an anchor. We need to define specific ratios and sizes for each anchor (1:1, 1:2, 2:1 for three ratios and 128^2, 256^2, 512^2 for three sizes in the original image).



One Anchor projected to the original Image

Next, RPN is connected to a Conv layer with 3x3 filters, 1 padding, 512 output channels. The output is connected to two 1x1 convolutional layer for classification and box-regression (Note that the classification here is to determine if the box is an object or not).



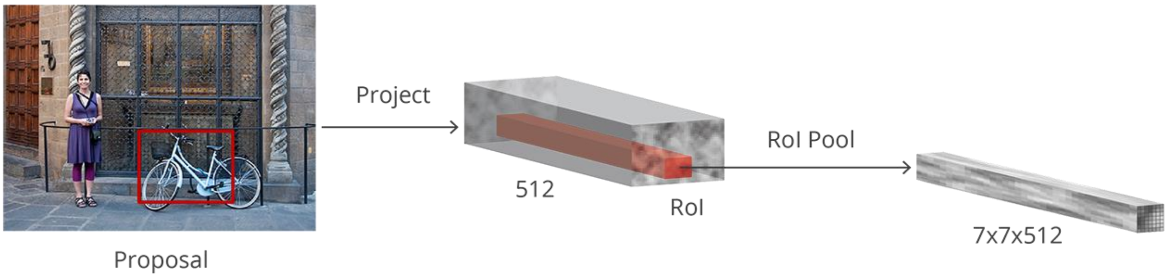
Convolutional implementation of an RPN architecture, where k is the number of anchors.

In this case, every anchor has 3x3 = 9 corresponding boxes in the original image, which means there are 37x50x9 = 16650 boxes in the original image. We just choose 256 of these 16650 boxes as a mini batch which contains 128 foregrounds (pos) and 128 backgrounds (neg). At the same time, `non-maximum suppression` is applied to make sure there is no overlapping for the proposed regions.

3.6.4.3 ROI Pooling Layer

Then we go to the second stage of faster RCNN. Similar to Fast R-CNN, ROI pooling is used for these proposed regions (ROIs).

We now have bounding boxes of different shapes and sizes which are passed on to the RoI pooling layer. Now it might be possible that after the RPN step, there are proposals with no classes assigned to them. We can take each proposal and crop it so that each proposal contains an object. This is what the RoI pooling layer does. It extracts fixed sized feature maps for each anchor. The output is 7x7x512. Then, we flatten this layer with some fully connected layers and passed to a fully connected layer which has a softmax and a linear regression layer. It finally classifies the object and predicts the bounding boxes for the identified objects.



Region of Interest Pooling

For every point in the output feature map, the network has to learn whether an object is present in the input image at its corresponding location and estimate its size. This is done by placing a set of “Anchors” on the input image for each location on the output feature map from the backbone network. These anchors indicate possible objects in various sizes and aspect ratios at this location. The figure below shows 9 possible anchors in 3 different aspect ratios and 3 different sizes placed on the input image for a point A on the output feature map. For the PASCAL challenge, the anchors used have 3 scales of box area 128^2, 256^2, 512^2 and 3 aspect ratios of 1:1, 1:2 and 2:1.

As the network moves through each pixel in the output feature map, it has to check whether these K corresponding anchors spanning the input image actually contain objects, and refine these anchors’ coordinates to give bounding boxes as “Object proposals” or regions of interest.

STEP BY STEP WALK THROUGH THE SOLUTION

1.	Data Preparation	• Downloading the zipped file from Kaggle • Extracting the Dataset • Understanding the data
2.	Exploratory Data Analysis	• EDA from the given CSV files • Extracting Metadata information from the training data • EDA from the metadata information
3.	Classification	• Reshaping the images • Converting the data into .npy file • Splitting the data into training and validation set • Extracting the target information • Using the pretrained model • By unfreezing the last layer of pretrained model and adding additional dense layers. • Training the model • Observing the performance metrics. • It is decided to proceed further with the images having Lung Opacity for Object Detection
4.	Lung Opacity Detection (Bounding Box detection using Faster RCNN)	• Importing Necessary Libraries • Pre-processing the data and bounding box values • Data Augmentation • Setting the Hyper Parameters • Defining RPN Layer (Base Layers and Setting of Anchors) • Defining ROI Layer • Performing Classification and Regression • Model Evaluation (Loss, Accuracy, mAP)

4.1 Bounding Box detection using Faster RCNN

The below steps are typically followed in a Faster RCNN approach:

1. Importing Necessary Libraries
   The libraries keras.models, keras.layers, keras.backend, keras.optimizers were used for performing Bounding Box Detection using Faster RCNN.

2. Downloading the weights
   The weights were downloaded from the github https://github.com/fchollet/deep-learning-models/releases/download/v0.2/resnet50_weights_tf_dim_ordering_tf_kernels.h5

3. Extracting the images having Lung Opacity alone and creating annotation files
   - Due to Google Colab Usage Limitations it is decided to train the images having Lung Opacity.
   - The images were splitted up into training and testing data (80:20) ratio
   - Bounding Box Values are extracted from the given .csv file and a separate txt file was created for training and testing data

4. Resizing the images to the shape of 300*300 and Resizing the bounding box values
   - The input images of size 1024*1024 is enhanced by histogram equalization
   - Then the images are reshaped into the size of 300*300
   - Since the images are reshaped, the bounding box values should be also rescaled

5. Data Augmentation
   Data Augmentation were performed by considering the following
   - horizontal_flips = True
   - vertical_flips = True
   - rot_90 = True

6. Setting the Hyper Parameters
   Through Literature Survey and over few trials the following hyperparameters considered

Scales of Anchor Boxes	self.anchor_box_scales = [64, 128, 256]
Image Size	self.im_size = 300
Channel Mean and Scaling Factor	self.img_channel_mean = [103.939, 116.779, 123.68] self.img_scaling_factor = 1.0
Region of Interest	self.num_rois = 4
RPN Stride	self.rpn_stride = 16
RPN Minimum and Maximum Overlap	self.rpn_min_overlap = 0.3 self.rpn_max_overlap = 0.7
Classifier Minimum and Maximum Overlap	self.classifier_min_overlap = 0.1 self.classifier_max_overlap = 0.5

Lambda Values	lambda_rpn_regr = 1.0 lambda_rpn_class = 1.0 lambda_cls_regr = 1.0 lambda_cls_class = 1.0
Epsilon	epsilon = 1e-4
Optimizer	optimizer_classifier = Adam(lr=1e-5)
Backbone	Resnet50

7. Defining RPN Layer

- We take an image as input and pass it to the ConvNet which returns the feature map for that image.
- Region proposal network is applied on these feature maps. This returns the object proposals along with their objectness score.
- Defining Base Layers and Setting of Anchors

8. Defining ROI Layer
- A RoI pooling layer is applied on these proposals to bring down all the proposals to the same size.
- Finally, the proposals are passed to a fully connected layer which has a softmax layer and a linear regression layer at its top, to classify and output the bounding boxes for objects.

9. Performing Classification and Regression
- Classification is done to find out whether the pixel belongs to foreground or Background. Activation function used Classification is “Sigmoid”
- The offset between the proposal and its corresponding ground-truth box is predicted using Regression. Activation function used for Regression is “Linear”

10. Model Evaluation.

The following Metrices were considered for Model Evaluation
- Mean Overlapping Bounding Boxes (IoU)
- Loss
- Accuracy
- Mean Accuracy Precision (mAP)
- Elapsed Time

MODEL EVALUATION

The following Metrices were considered for Model Evaluation

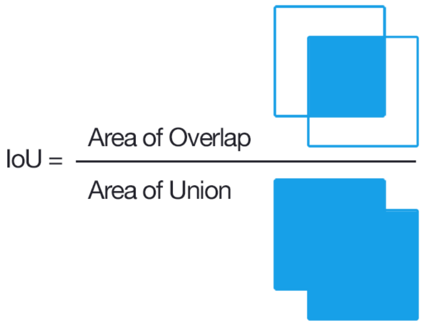
- Mean Overlapping Bounding Boxes (IoU)
- Loss
- Accuracy
- Mean Accuracy Precision (mAP)

5.1 Mean Overlapping Bounding Boxes (IoU)

Intersection over Union is an evaluation metric used to measure the accuracy of an object detector on a particular dataset.

More formally, in order to apply Intersection over Union to evaluate an (arbitrary) object detector we need:

1. The ground-truth bounding boxes (i.e., the hand labeled bounding boxes from the testing set that specify where in the image our object is).
2. The predicted bounding boxes from our model.



5.2 Loss

The multi-task loss function combines the losses of classification and bounding box regression:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{box}}$$

$$\mathcal{L}(\{p_i\}, \{t_i\}) = \frac{1}{N_{\text{cls}}} \sum_i \mathcal{L}_{\text{cls}}(p_i, p_i^*) + \frac{\lambda}{N_{\text{box}}} \sum_i p_i^* \cdot L_1^{\text{smooth}}(t_i - t_i^*)$$

Where$\mathcal{L}_{cls}$ is the log loss function over two classes, as we can easily translate a multi class classification into a binary classification by predicting a sample being a target object versus not. L_1^{smooth} is the smooth L1 loss.

$$\mathcal{L}_{\text{cls}}(p_i, p_i^*) = -p_i^* \log p_i - (1 - p_i^*) \log(1 - p_i)$$

5.3 Accuracy

Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right. Formally, accuracy has the following definition:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ number\ of\ predictions}$$

5.4 Mean Average Precision (mAP)

mAP (mean average precision) is the average of AP. In some context, we compute the AP for each class and average them. But in some context, they mean the same thing. For example, under the COCO context, there is no difference between AP and mAP. Here is the direct quote from COCO:

AP is averaged over all categories. Traditionally, this is called "mean average precision" (mAP). We make no distinction between AP and mAP (and likewise AR and mAR) and assume the difference is clear from context.

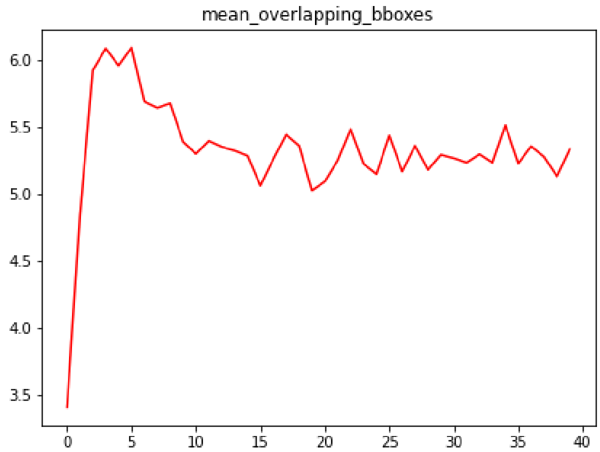
5.5 Results and Discussions

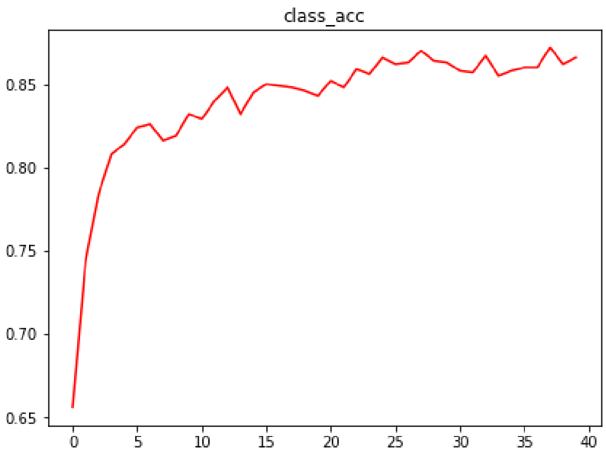
The model is trained for 40 Epochs and the results were saved in a .csv files

Epochs	mean_overlapping_bboxes	class_acc	loss_rpn_cls	loss_rpn_regr	loss_class_cls	loss_class_regr	curr_loss	elapsed_time	mAP
1	3.41	0.656	4.04	0.185	0.549	0.353	5.127	5.645	0.252
2	4.844	0.744	2.445	0.186	0.531	0.393	3.554	4.584	0.259
3	5.926	0.784	1.841	0.181	0.461	0.405	2.888	4.582	0.259
4	6.089	0.808	1.685	0.185	0.419	0.394	2.683	4.557	0.26
5	5.961	0.814	1.464	0.18	0.409	0.394	2.447	4.533	0.263
6	6.096	0.824	1.454	0.166	0.387	0.381	2.388	4.499	0.264
7	5.691	0.826	1.369	0.162	0.379	0.372	2.281	4.521	0.272
8	5.645	0.816	1.418	0.172	0.397	0.357	2.343	4.515	0.273
9	5.678	0.819	1.39	0.165	0.391	0.345	2.291	4.466	0.273
10	5.392	0.832	1.262	0.17	0.374	0.338	2.144	4.474	0.273
11	5.302	0.829	1.236	0.157	0.377	0.324	2.094	4.538	0.274
12	5.399	0.84	1.215	0.152	0.364	0.316	2.046	4.523	0.274
13	5.354	0.848	1.304	0.157	0.341	0.307	2.109	4.482	0.275
14	5.326	0.832	1.224	0.161	0.372	0.319	2.076	4.451	0.275
15	5.288	0.845	1.101	0.155	0.341	0.304	1.901	4.448	0.278
16	5.064	0.85	1.242	0.148	0.341	0.295	2.027	4.504	0.281
17	5.262	0.849	1.11	0.151	0.343	0.296	1.899	4.471	0.282
18	5.445	0.848	1.185	0.155	0.328	0.297	1.965	4.51	0.283
19	5.362	0.846	0.945	0.159	0.343	0.289	1.736	4.486	0.285

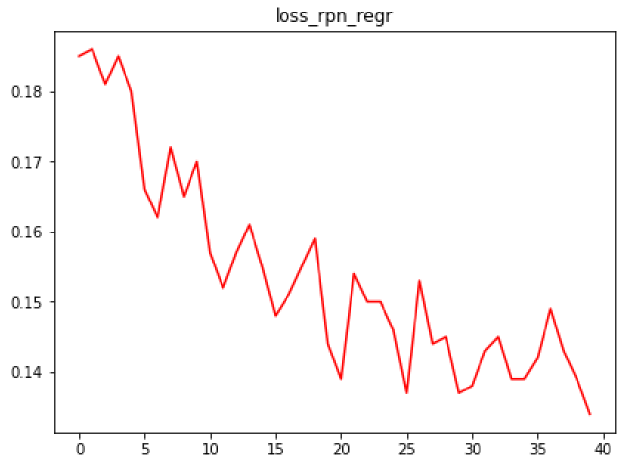
20	5.028	0.843	1.101	0.144	0.336	0.289	1.87	4.458	0.288
21	5.095	0.852	1.117	0.139	0.335	0.283	1.873	4.463	0.289
22	5.253	0.848	0.942	0.154	0.349	0.285	1.73	4.46	0.289
23	5.487	0.859	1.077	0.15	0.319	0.293	1.839	4.544	0.293
24	5.227	0.856	1.056	0.15	0.322	0.275	1.803	4.521	0.295
25	5.151	0.866	1.078	0.146	0.297	0.281	1.801	4.495	0.296
26	5.438	0.862	1.042	0.137	0.315	0.288	1.782	4.461	0.305
27	5.17	0.863	0.925	0.153	0.308	0.275	1.661	4.421	0.307
28	5.362	0.87	1.163	0.144	0.302	0.273	1.881	4.459	0.302
29	5.182	0.864	0.962	0.145	0.305	0.274	1.685	4.402	0.305
30	5.295	0.863	1.016	0.137	0.311	0.278	1.743	4.396	0.311
31	5.269	0.858	0.985	0.138	0.314	0.274	1.711	4.386	0.314
32	5.236	0.857	0.968	0.143	0.318	0.273	1.702	4.399	0.318
33	5.299	0.867	0.967	0.145	0.309	0.273	1.694	4.422	0.361
34	5.235	0.855	0.987	0.139	0.325	0.282	1.733	4.412	0.325
35	5.516	0.858	0.914	0.139	0.316	0.272	1.64	4.438	0.372
36	5.23	0.86	1.013	0.142	0.322	0.259	1.736	4.429	0.322
37	5.358	0.86	1.161	0.149	0.314	0.252	1.874	4.403	0.354
38	5.28	0.872	0.99	0.143	0.301	0.263	1.697	4.482	0.373
39	5.134	0.862	0.896	0.139	0.313	0.26	1.609	4.441	0.375
40	5.337	0.866	0.936	0.134	0.303	0.259	1.633	4.437	0.383

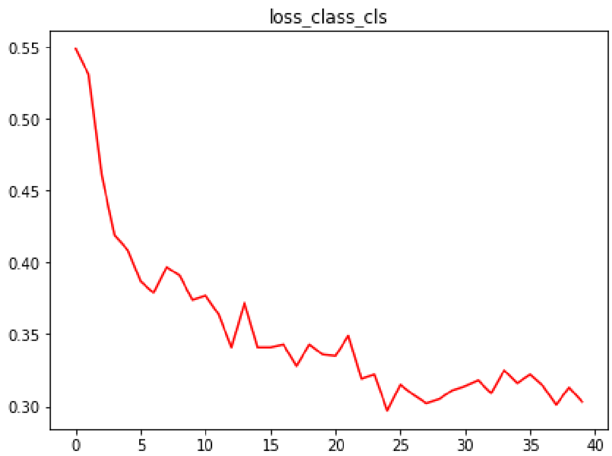
The Observed results were plotted for better understanding

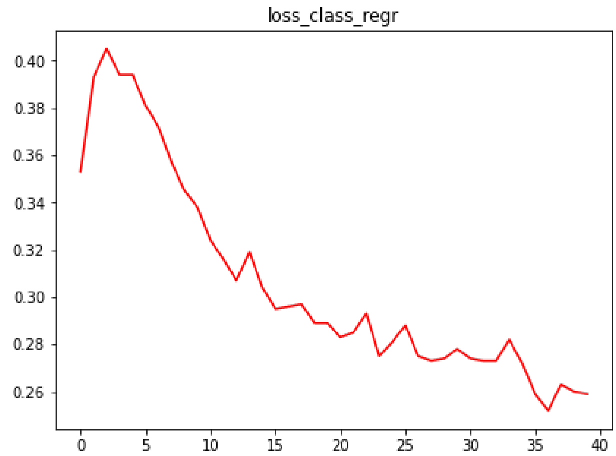


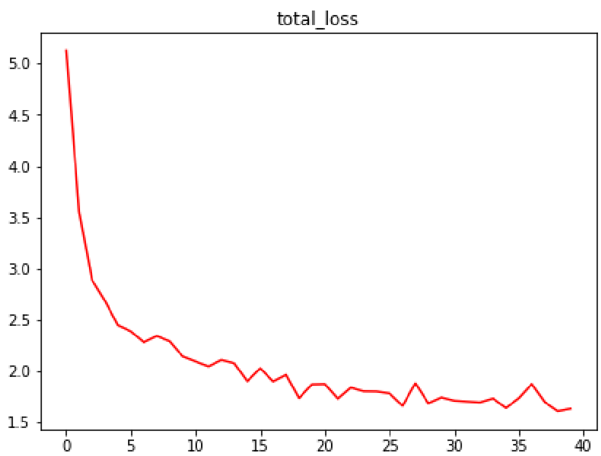


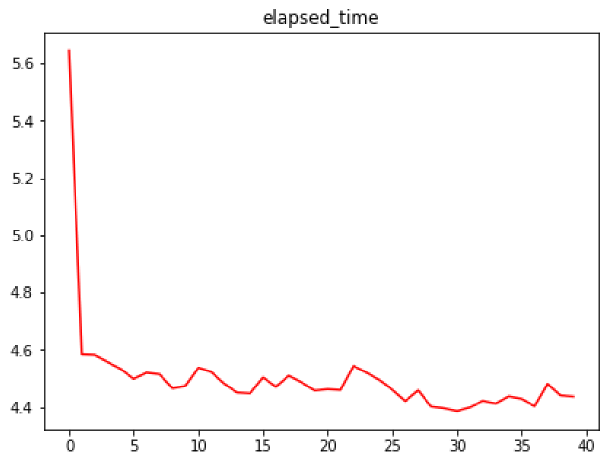
- Mean_Overlapping_bboxes and Accuracy increases as the number of Epochs increases







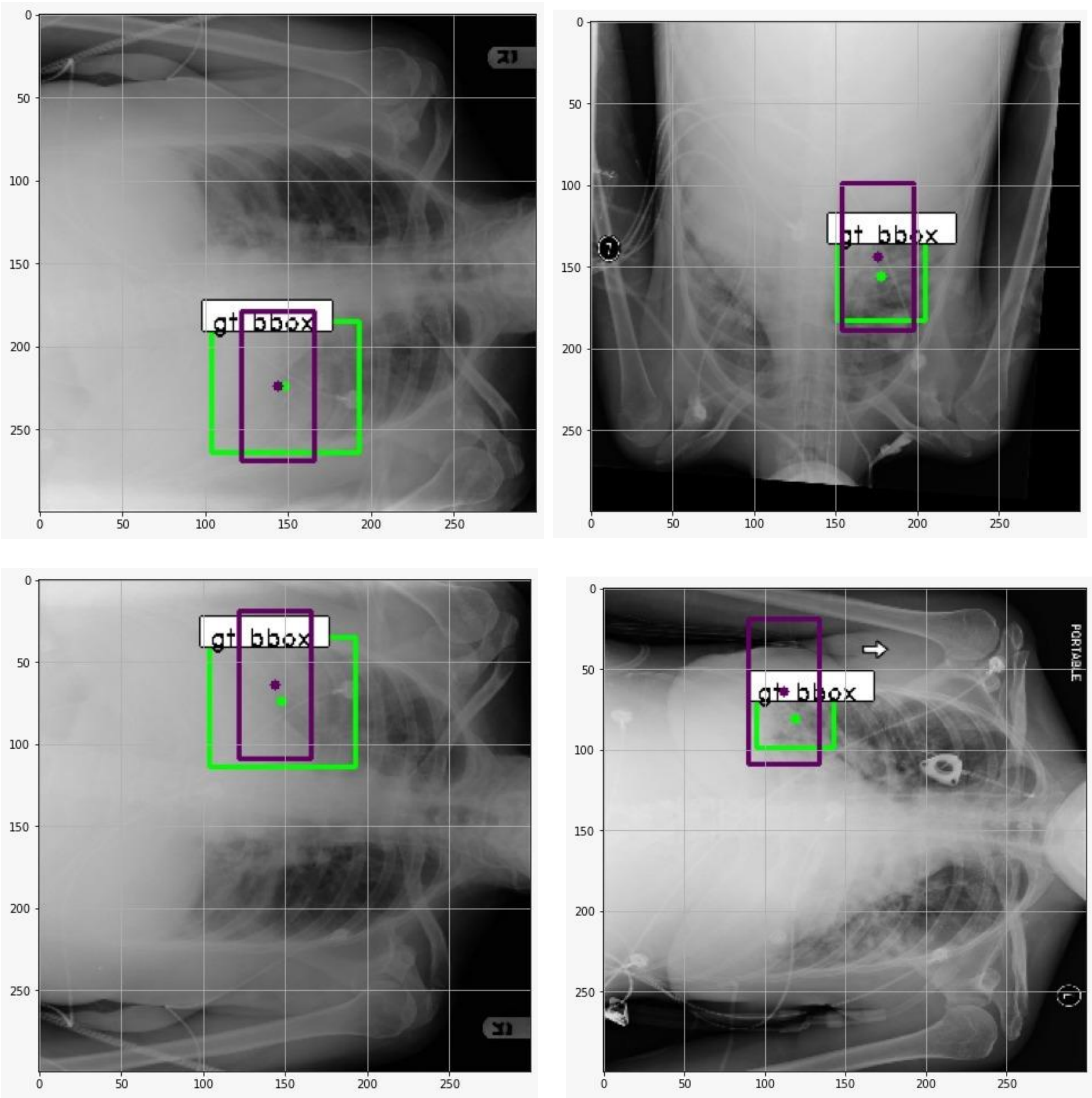




➢ Losses and Training time decreases as the number of Epochs increases

5.5 Sample Output

- The green Bounding Boxes is the ground truth
- The purple Bounding Boxes are the predicted one



- The results clearly reveal that the model performs well even the input images are flipped/ Rotated.

COMPARISON TO BENCHMARK

6.1 Classification Comparison

- The model is trained by using two different optimizers Adam and RMS Prop and Different Learning Rate.
- The best results were observed for Adam Optimizer with lr = 0.0001

Evaluation Parameters	VGG16		Resnet50	
	“Target” Attribute	“Class” Attribute	“Target” Attribute	“Class” Attribute
Training Accuracy	98.82%	96.32%	99.74%	99.46%
Validation Accuracy	72.02%	33.42%	73.47%	34.48%
Recall	49.07%	34.16 %	50 %	35.68 %
Precision	36.56 %	41.42 %	36.73 %	47.91 %
F1 Score	41.90 %	20.06 %	42.35 %	21.11 %

6.2 Lung Opacity Bounding Box Detection Comparison

- The model is trained by using two different optimizers Adam and RMS Prop and with Different Learning Rate (lr =0.001, 0.0001, 0.00001)
- The best results were observed for Adam Optimizer with lr = 0.00001
- The Tabulation below shows the best model from VGG16 and Resnet50

Evaluation Parameters	VGG16	Resnet50
Mean of Overlapping Boxes (Intersection Over Union)	5.676	5.337
Classification Accuracy	71.9 %	86.57 %
RPN Classification Loss	1.706	0.936
RPN Regressor Loss	0.154	0.134
Classifier Classification Loss	0.534	0.303
Classifier Regression Loss	0.343	0.259
Overall Average Loss	2.737	1.632
Mean Accuracy Precision (mAP)	0.0292	0.383

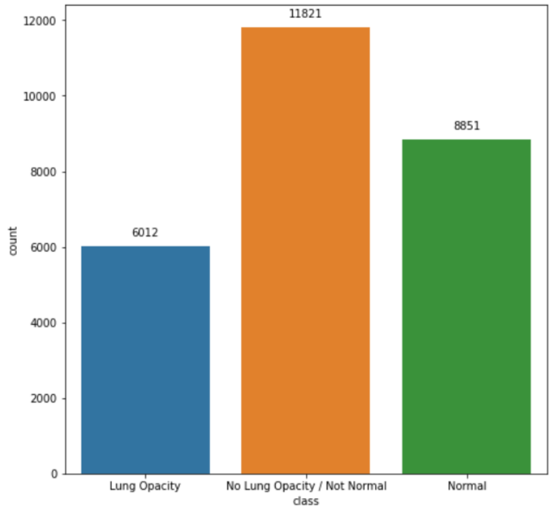
VISUALIZATION

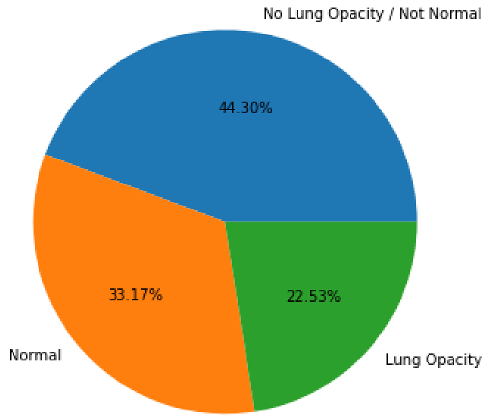
7.1 Exploratory Data Analysis

7.1.1 Exploring the attribute “class” from the metadata

- Three different classes of images were observed
    - No Lung Opacity/ Not Normal – 11821 information
    - Lung Opacity – 6012 information
    - Normal – 8851 information
- All the three classes have almost equal distributions

```
No Lung Opacity / Not Normal    11821
Normal                           8851
Lung Opacity                     6012
Name: class, dtype: int64
```

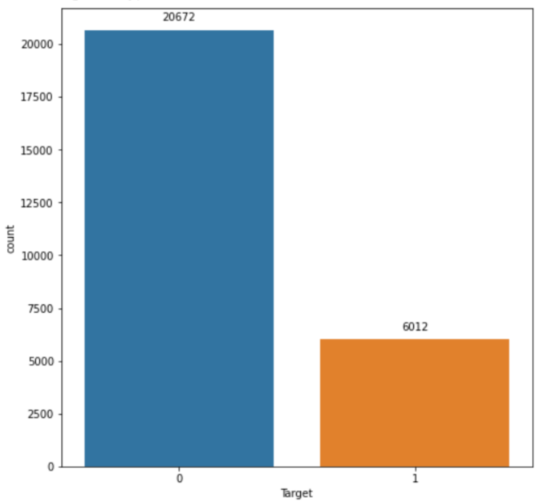


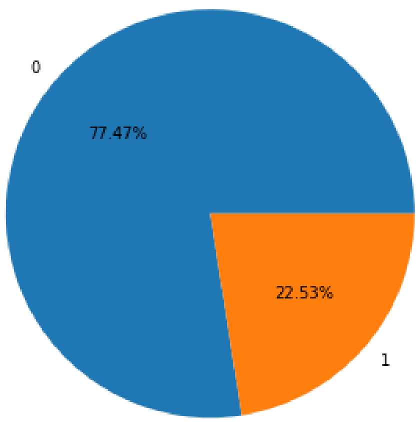


7.1.2 Exploring the attribute “Target” from the metadata

- Two different targets of images were observed
    - Target= 0 –20672 information
    - Target=1 – 6012 information
- The given information is more biased towards Target=0

```
0    20672
1     6012
Name: Target, dtype: int64
```

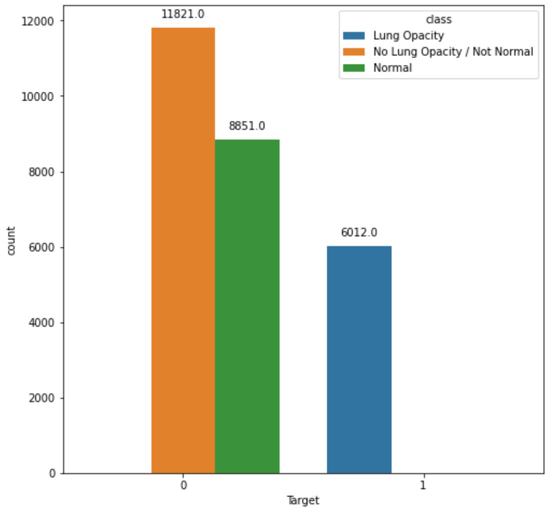




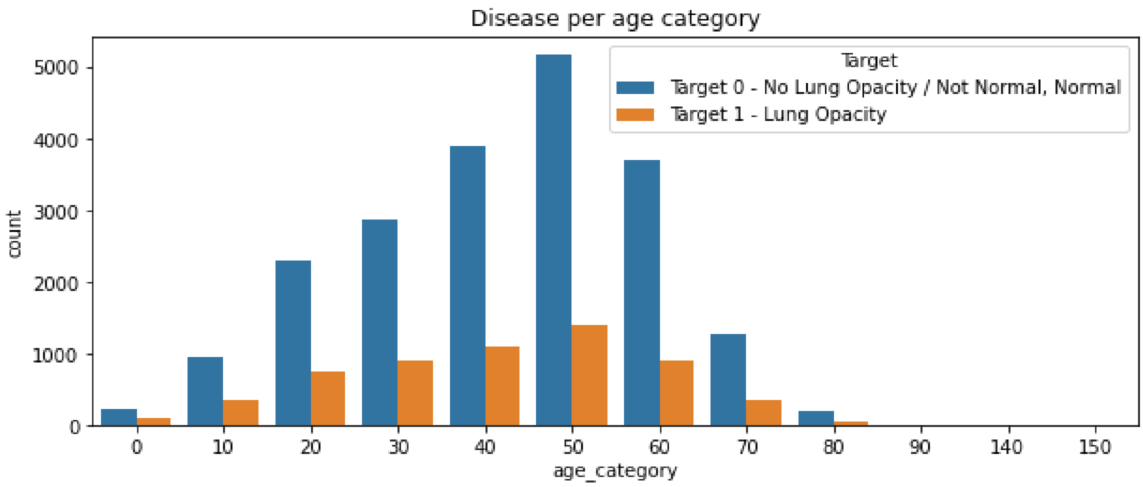
7.1.3 Relationship between “Target” and “class”

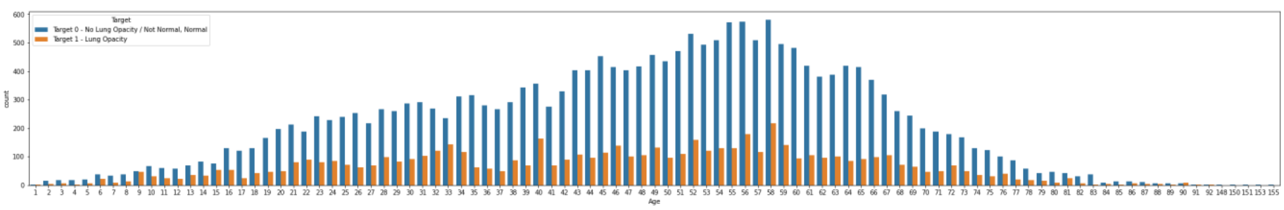
- Target = 0 includes two classes “ No Lung Opacity/ Not Normal” and “Normal”
- Class “Lung Opacity” is categorized as Target=1

Target : 0 **20672 counts**		**Target: 1** **9555 counts**
Class: **No Lung Opacity/ Not Normal** **11821 counts**	**Class:** **Normal** **8851 counts**	**Class: Lung Opacity** **6012 counts**



7.1.4 “Age distribution” form the given dataset

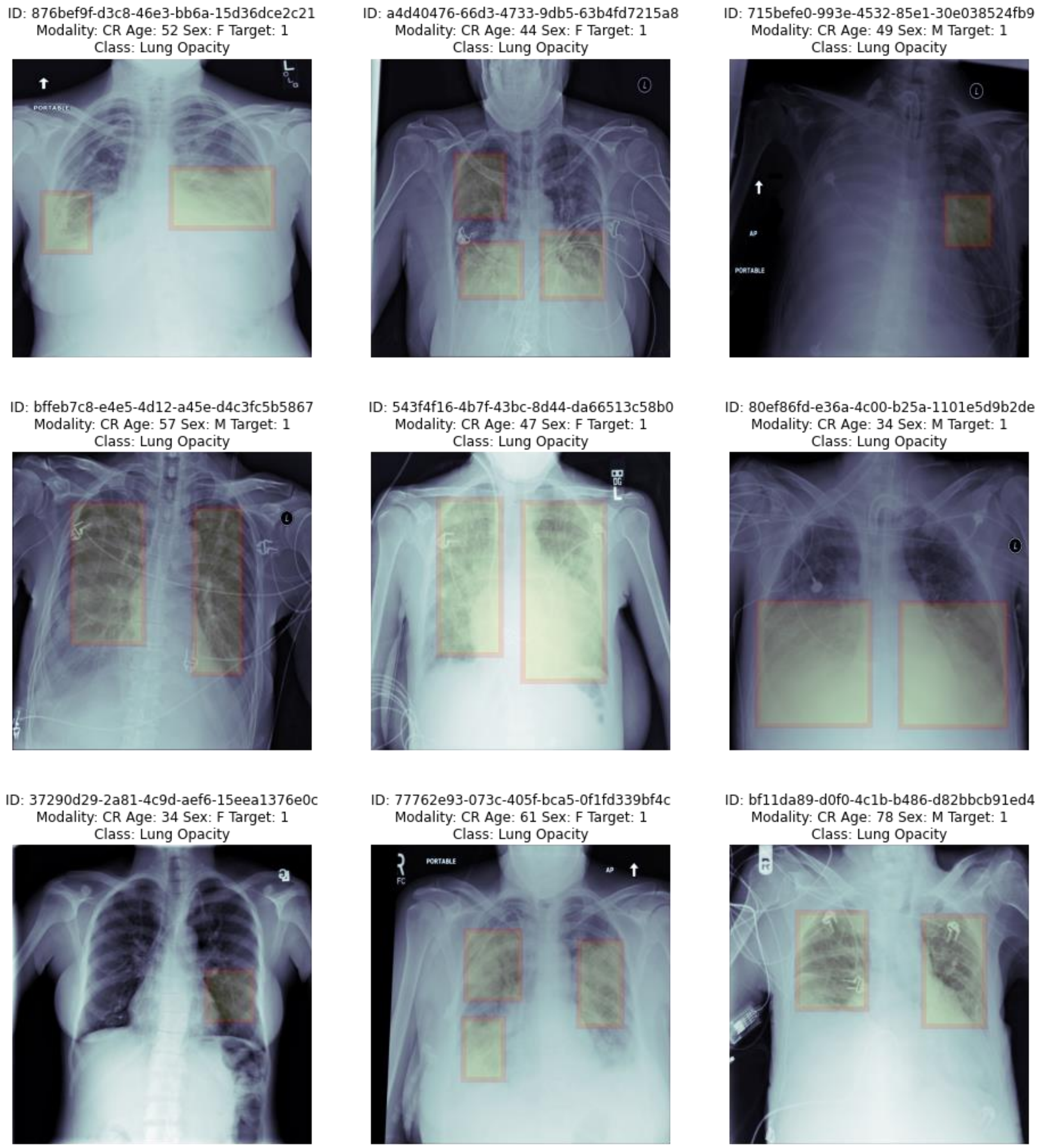




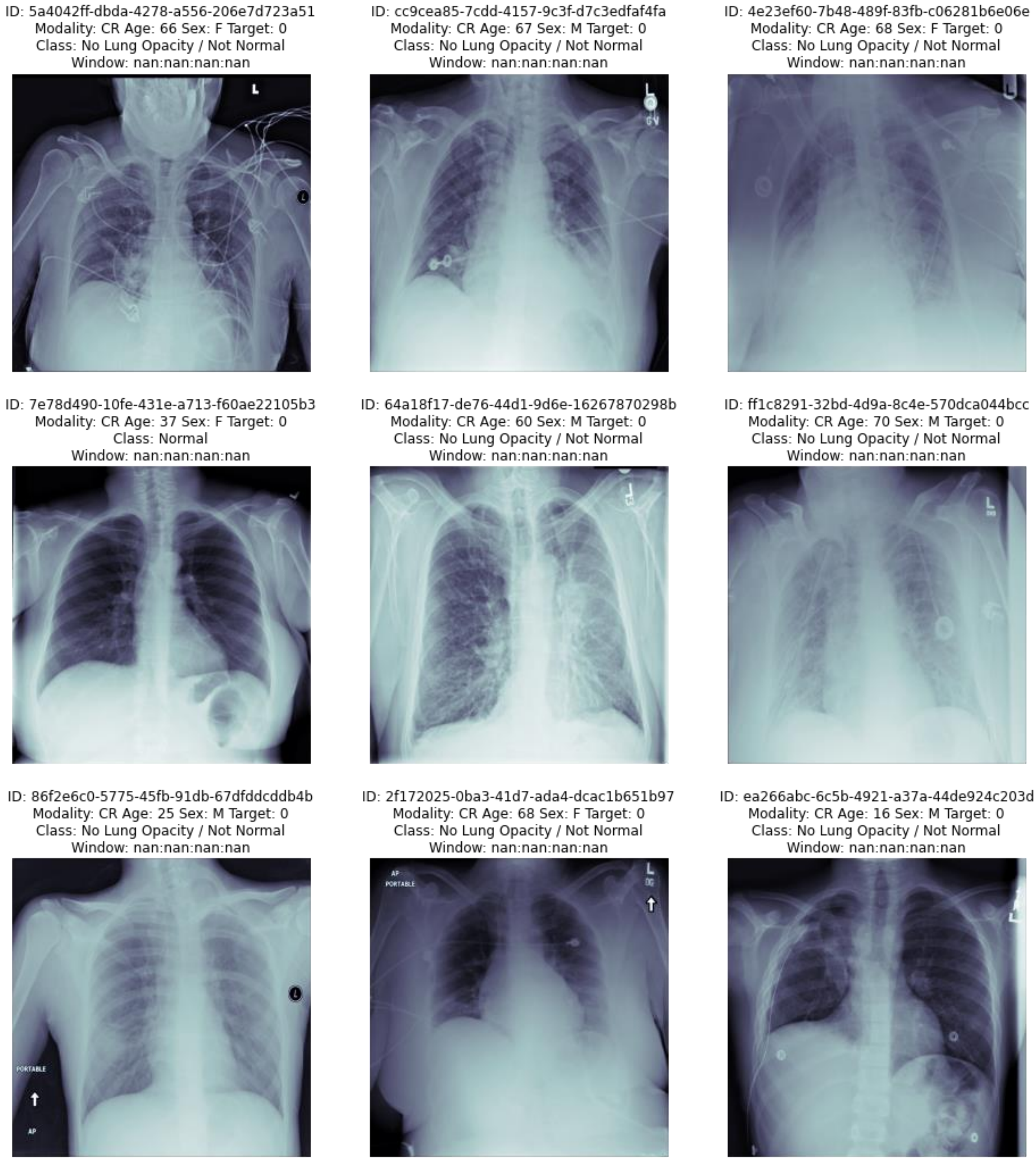
- More no. of age distribution lies within the range of 44 to 55.

7.2 Visualizing the given images

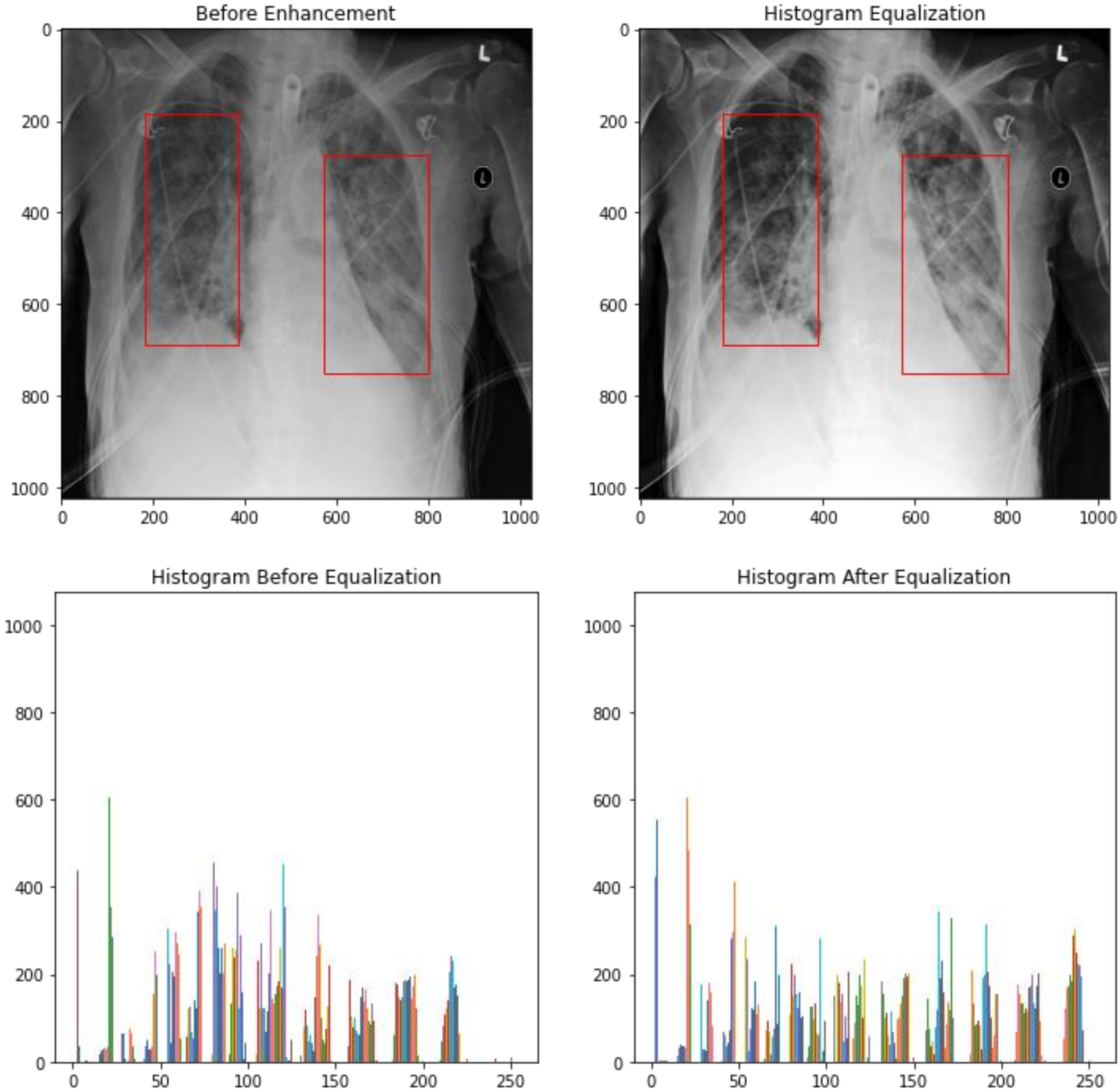
- The images with Target =1 and class = Lung Opacity
- Form the given bounding box information, the lung opacity is highlighted with a bounding box
- A patient may have lung opacity in more than one places.



➢ The images with Target =0 and class = No Lung Opacity/ Not Normal & Normal
➢ Bounding box information is not provided in this case

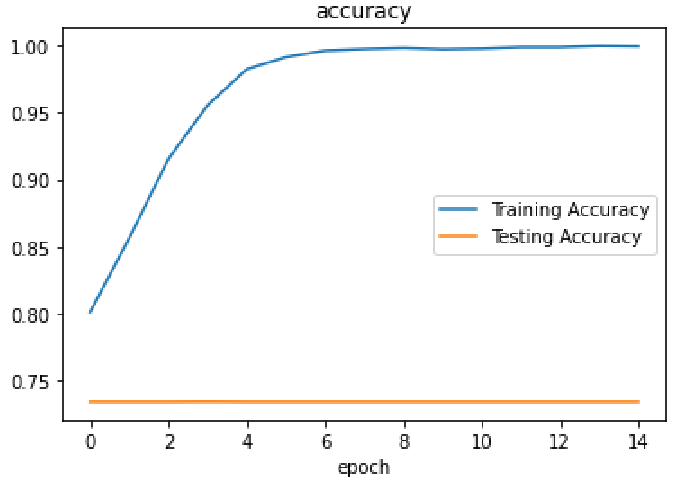
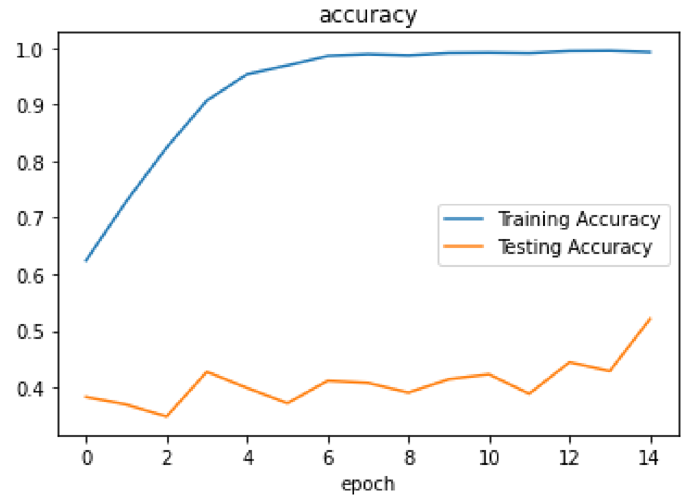


7.3 Input Image Vs Enhanced Image



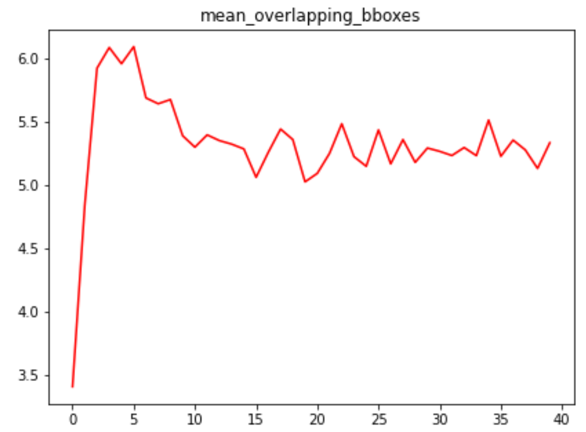
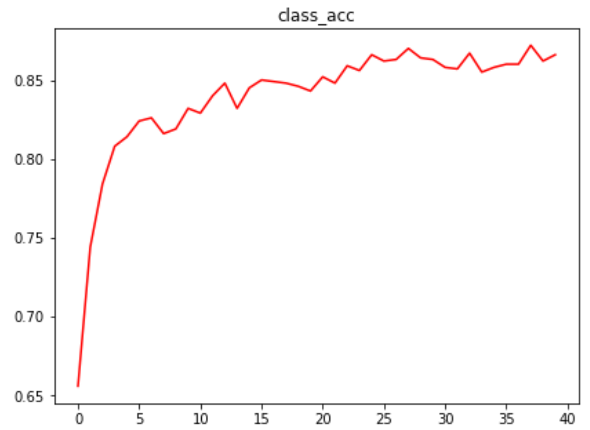
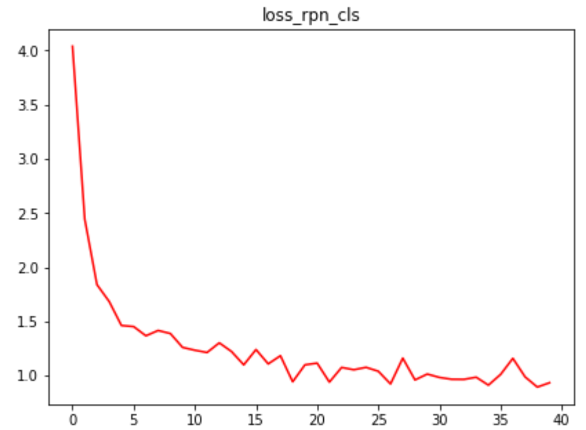
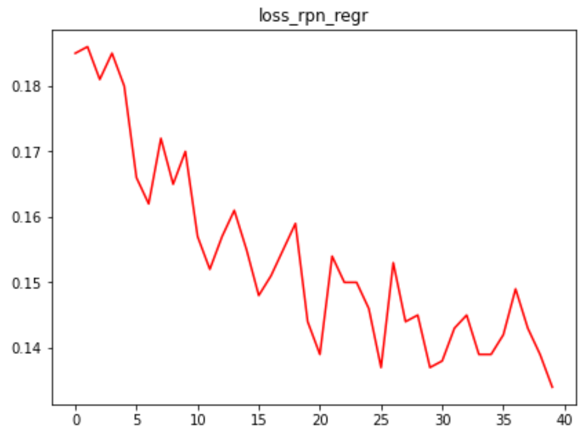
- ➢ It is observed that the enhanced images have better contrast compared to the input images

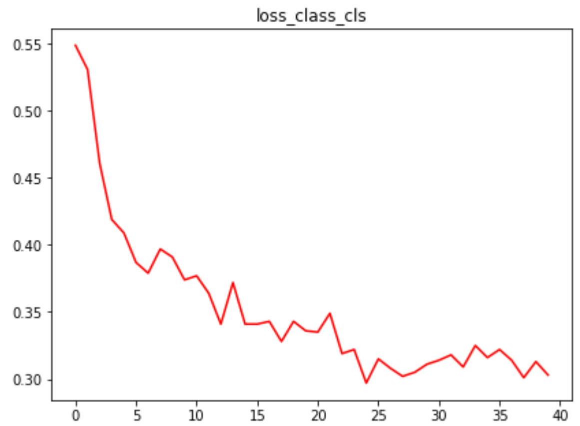
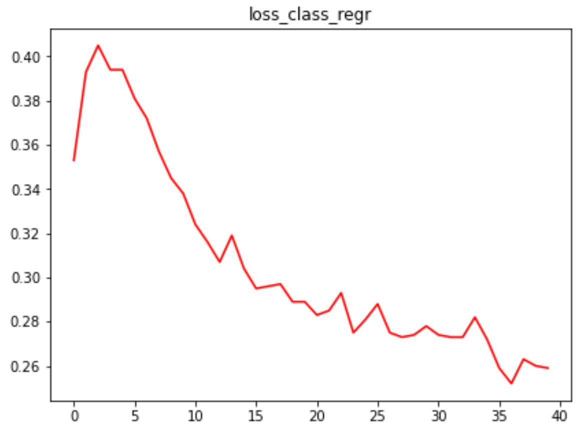
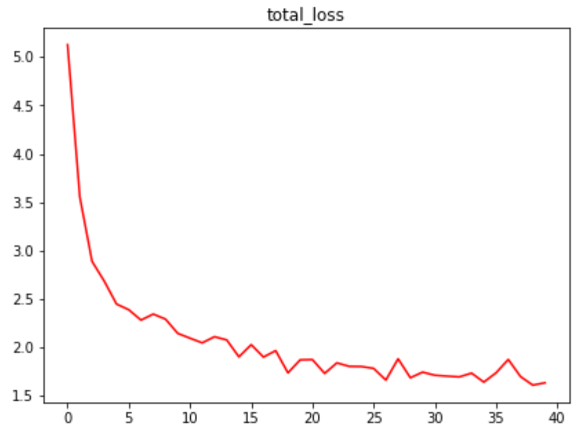
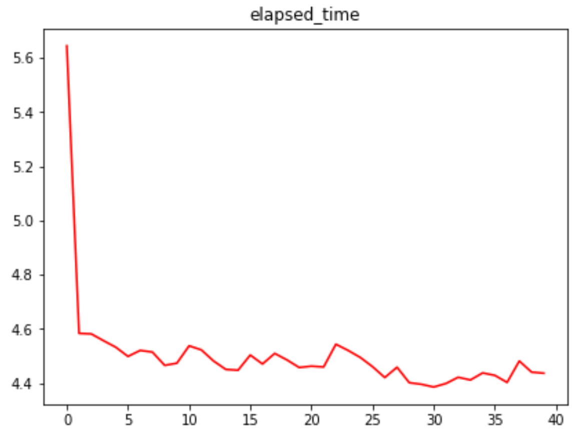
7.4 Accuracy Vs Epochs for Image Classification

Classification of Images	**Resnet50 with the attribute "Target"**	
	Resnet50 with the attribute "Class"	

- The model seems to be overfit.
- **Limitation of this model:** This model can be used for classifying the images. The location of the lung opacity cannot be predicted using this model.

7.5 Model Evaluation of Faster RCNN For Lung Opacity Detection

<table>
<tr><td rowspan="4">Lung Opacity Detection with Bounding Box</td><td>Mean of Overlapping Boxes
(Intersection Over Union)</td><td></td></tr>
<tr><td>Classification Accuracy</td><td></td></tr>
<tr><td>RPN Classification Loss</td><td></td></tr>
<tr><td>RPN Regressor Loss</td><td></td></tr>
</table>

<table>
<tr><td rowspan="4">Lung Opacity Detection with Bounding Box</td><td>Detector Classification Loss</td><td>
</td></tr>
<tr><td>Detector Regression Loss</td><td>
</td></tr>
<tr><td>Overall Average Loss</td><td>
</td></tr>
<tr><td>Elapsed Time</td><td>
</td></tr>
</table>

IMPLICATIONS

8.1 Implication of Artificial Intelligence in Medical Field

- In 2016 AI luminary Geoff Hinton provocatively declared, ***"It's quite obvious that we should stop training radiologists now."***

- Hinton's comments ruffled some feathers in the medical community, but it was hard to deny the data on which they were based. **Over the past few years, a series of studies have demonstrated that *neural networks can identify medical conditions from X-rays more accurately* than can human radiologists.**

- To give a few recent examples: in May 2019, a team of **researchers from Google, Stanford and North-western published a study in which a *deep learning model outperformed human physicians at detecting lung cancer from CT scans***.

- A few months later, a research team from NYU published a series of studies demonstrating ***AI's superior performance* detecting breast cancer from mammograms**.

- In January 2020, a research group from Google and top medical research centers released another **breast cancer detection study, with the *AI system again outperforming humans*.**

- In the January study, which received ***widespread media attention*, the AI system produced a 9.4% reduction in false negatives and a 5.7% reduction in false positives relative to human radiologists.**

- In many ways, ***radiology is an ideal use case for deep learning***. Examining images for the presence of a medical condition like pneumonia is an exercise in pattern recognition and object classification—exactly what deep learning excels at.

8.2 Difficulties of Implications in Medical Field

- Yet, several years after Hinton predicted the obsolescence of human radiologists, ***no clinic in the world has deployed AI-driven radiology tools at scale***. At best, a handful of forward-thinking health organizations have begun using it in limited settings.

- Why is this? **There is a *huge gap* between publishing academic research and building a real company**—with real patients, in real clinics, with real lives on the line—to commercialize that research.

- The world healthcare system can be byzantine for start-ups to navigate, with notoriously long sales cycles and institutional inertia. **Reimbursement regimes are complex.** Consumer education and acceptance happen only gradually.

- In addition, ***FDA approval*** **must be obtained before clinicians can use algorithms for diagnosis in real-world settings. This is a long and tedious process**; the FDA has only recently begun issuing approvals to a small handful of companies. Aidoc, a prominent AI radiology startup, just last month received FDA approval for its deep-learning-based stroke detection technology.

- **Another thorny and unresolved issue is *liability***: if a human doctor relies on an AI system to make a diagnosis that turns out to be wrong, who should be accountable?

- Finally, these **AI models are *not yet sufficiently generalizable*. Most models from the academic research community are trained using data from only one hospital.** They often falter when applied to other populations. In one example, a deep learning model trained to detect pneumonia performed at 93% accuracy when used on patients from the same hospital, but dropped as low as 73% when tested on patients from other locations. It may not work as well because the patients at the other hospitals may be different. This lack of generalizability can have serious implications for minority groups who are underrepresented in historical datasets

- The net result of these challenges is that **AI-based radiology tools are still in only the *earliest stages of commercial deployment***. For instance, in the Google breast cancer study discussed above, the researchers directly acknowledged that their technology was not yet ready for real-world use.

This is not to say that no start-ups are working to commercialize this technology. Last year, CureMetrix became the first company to **received FDA approval** for its AI-based breast cancer technology; the company plans to deploy in several clinical settings this year. Other start-ups angling to commercialize and scale AI-based radiology in the near term include Arterys, Aidoc, Zebra Technologies and DeepHealth.

8. 3 Methodologies to Overcome the Implication Problem

- Due to the above factors, ***the AI radiology market today remains surprisingly undeveloped.*** But until the field matures and becomes stable there is always a scope to run the AI systems as a **production parallel deployment.**

- In this case , ***there would be both radiologists and AI model both predicting the nature of the x-rays.*** **There may be a duplication in the effort but there is always data being generated out of this with which we can improve the model in comparison to the radiologist until it matures and gets perfect** and ready to be deployed into the real world as a whole.

- So, AI Models should be generalized. This can be achieved ***by training a model by collecting larger and more diverse datasets*** with which to train AI models. **But such an effort will be expensive, time-consuming and operationally intensive.**

- ***"We will continue to explore and build upon our model, working with additional partners across the world, before considering bringing it into clinical practice,"*** said Shravya Shetty, **a Google researcher** who co-authored the paper.

LIMITATIONS

9.1 Limitations

Due to lack of computation resources we were using **google colab for its GPU utilization**. During the training of retinanet model, colab has caused lots of problems some of them including

1) Executors getting killed : Which on researching we understood that it was due to instability of the mounted google drive. So we tried to copy the jpg files to colab local drive but constantly got pestered by another error owing to drive timeout. This has caused quite inconvenience and had to pause the training there owing to lack of computation resources like GPU.
2) But can use the retina net model which will definitely excel upon the faster RCNN owing to its FPN unlike the RPN used by faster RCNN.

9.2 What extra tuning and improvements can be done on this model

Due to limitation in time and computation resources we had to compromise and go with the basic parameters for optimization, augmentation and evaluation metrics.

- **Augmentation :**

  We can use Image generator class from keras to do pre-processing (augmentation) and use its powerful inbuilt features. We can play around with the impacting features like width_shift_range, height_shift_range, brightness_range, shear_range,zoom_range, shuffling and flips. This would require a lot of re-iteration process as train_generator would create augmented input image arrays in every iteration which is a time consuming task but will definitely help in improving model.

  Especially tweaking with brightness, contrast, zoom will be helpful in this case as we are kind of recognising a pattern which is lung opacity.

  Flips can be used but will require a separate custom method to recompute the bounding box values accounting to horizontal or vertical flip and the range.

- **Optimizations :**

  We have used ADAM and RMS Prop optimizers in our model and found ADAM better. We also wanted to use other optimizers like ADAMAX, ADAGRAD and ADADELTA to compare the performance which will again take back to the problem of increase in the number of iterations during training which consume lot of time.

- **Evaluation metrics :**

  We have used custom methods to evaluate metrics like mAP, loss and accuracy. We also want to implement precision, recall basically the confusion matrix which can show the percentage of false positives and false negatives but could not implement owing to time factor.

- **A better model**

  The same time when we were working on the Faster RCNN model , we also tried working on the state of art Keras-Retinanet model with resnet 101 as backbone.

9.3 Real world scenario

1) For application in real world , model accuracy should definitely be near 100 like 99.7-99.9. This is because pneumonia being a disease which can take life there should very minimal error in prediction. So for now this model is not yet ready to go into real world.
2) Even though there is little overfit, model would definitely fail when tested on chest xrays of people belonging to different demographics and different hospitals. This is because, every individual is different in his/her own way as well as every race, ethnic group is different in its own way. Due to different demographic clusters among the people in world the characteristics of lungs like size,shape etcetera maybe different in individuals belonging to different clusters but maybe similar if they belong to same cluster. This has not been considered in this project as there was no data belonging to this in the given dataset.
3) As explained during the introduction, lung opacities maybe caused due to various factors excluding pneumonia. But in this model we have limited to the hypothesis of opacity directly related to pneumonia only. So there maybe a situation where an opacity maybe due to another condition like the Covid-19 or Dyspnea. In this case the model will classify it as pneumonia which results in the failure.
4) The detection of pneumonia can not be done solely by judging the lung opacity as said in the previous point. It also needs to account for various symptoms like cough with bloody mucus,

Fever, shallow breathing, sharp chest pain, fatigue etc in the patients. We should also consider the repetition and tenure of these symptoms and monitor these regularly in a certain time frame.

So, there should also be an NLP model or maybe another classification model to take account of such symptoms also repeatedly in the time frame where the patient is monitored and only after this, we should be able to say that the patient is affected by pneumonia. Simply by recognizing the lung opacity classifying a patient as affected by pneumonia or not does not make sense.

CLOSING REFLECTIONS

10.1 Reflections and learnings

This project has actually given us abundant experience and knowledge.

- Firstly, patience is the key. Handling real life datasets with such a huge quantity and variance and processing them needs high levels of patience and very high meticulous care.

- EDA is mostly not taken as an important step by students or new learners. We found out that this is one of the major and important steps which actually does 30% of the work for us. We can understand the data in depth by EDA and this actually makes our pre processing, model building and choosing the model pretty easy.

- The process of choosing a model was a very insightful and research oriented one. We brainstormed a lot, researched intensively over different articles, went through lots of insightful research papers , got to learn different models implemented and running in the production used by industry giants. This has exposed us to the actual deep learning scenario existing in the present world.

- When comparing the faster RCNN with retina-net we got to learn the state of art networks like RPN and FPN.

- We also experimented Faster RCNN models with different backbones like VGG and Resnet and deep dived into the structure and characteristics of the above in detail.

- We learnt different features of keras like Image generators, train and validation generators which come with built in pre-processing, Dicom files and their processing.

- Evaluation metrics like mAP is something new which we have learnt during this project.

- We also started to think with respect to the problem statement and its effects in the real world rather than just thinking about solving the problem like a test.

- Most importantly, the team work and distribution of tasks along with managing other works along with completion of the capstone, this was a great and tough journey.

10.2 What can be done better?

We could choose and experiment on various other models which actually proved to give better results on similar problem statements like detecting lung cancers and other lung related conditions. But owing to time and computation resource restrictions we could not enter that lane. But if there was time we wanted to implement the same with such models and also as mentioned earlier ,

1) Collecting different data based on demographics, hospitals.
2) Collected data related to symptoms
3) Model can be tuned to account for the collected symptoms to predict Pneumonia
4) Then based on the model built we could scale up the model to detect other conditions in lungs.
5) Once the model is well and working well , we can also scale up to different problems in the radiology domain.

10.3 Take-away from the Capstone Project

What we learned till now was just 1% but what we learnt and saw was that there was still 99% remaining under the water which we need to explore. Overall the capstone project revealed to us that what we have learnt was actually the tip of the iceberg, and what needs to be still explored and learned is actually pretty huge.

REFERENCES

[1] https://www.kaggle.com/zahaviguy/what-are-lung-opacities

[2] https://www.healthline.com/health/pneumonia#in-kids

[3] https://en.wikipedia.org/wiki/Histogram_equalization

[4] https://towardsdatascience.com/histogram-equalization-5d1013626e64

[5] https://arxiv.org/pdf/1905.08545.pdf

[6] Redmon, J.; Farhadi, A. Yolo9000: Better, faster, stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.

[7] LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. Nature 2015, 521, 436–444. [CrossRef] [PubMed]

[8] Lin, T.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.

[9] https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b

[10] https://towardsdatascience.com/faster-r-cnn-object-detection-implemented-by-keras-for-custom-data-from-googles-open-images-125f62b9141a

[11] https://tryolabs.com/blog/2018/01/18/faster-r-cnn-down-the-rabbit-hole-of-modern-object-detection/

[11] https://pjreddie.com/darknet/yolo/

[12] https://arxiv.org/abs/1708.02002

[13] https://www.researchgate.net/publication/330330962_Deep_RetinaNet-Based_ Detection_ and_Classification_of_Road_Markings_by_Visible_Light_Camera_Sensors

[14] https://blog.zenggyu.com/en/post/2018-12-05/retinanet-explained-and-demystified/