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

THE CAPSTONE FINAL REPORT

Project: Pneumonia detection

Final - Report

Mentor: Mr. Sandeep Raghuwanshi

Team Members:

- Kishana Kumar Vempati
- Abhinav Girotra
- Sreekumar Thekkuttu Phsharam
- Gauri Dhananjay Desai
- Nalini Thiruvengadam

Date: March 27, 2022

Contents

1 Problem Statement

The problem statement provided was the detection of Pneumonia in the patients, whose X-Ray images and data set consisting of result are provided. The team followed a approach of exploring the data sets, extensive data analysis and the coming to some interesting findings like the relation between age and pneumonia, gender and pneumonia and relationship with pixels. During EDA we also explored doing image augmentation and noted its impact on results of model.

The details are explained in below sections:

1.1 Detail Problem Statement

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. 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. 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 performed diagnostic imaging study. Several 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.

1.2 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).

Guided by relevant clinical questions, powerful AI techniques can unlock clinically relevant information hidden in the massive amount of data, which in turn can assist clinical decision making.

1.3 Objective Of the Work

Now to detection Pneumonia we need to detect **Inflammation** of the lungs. In this project, we are challenged to build an algorithm to detect a visual signal for pneumonia in medical images. Specifically, our algorithm needs to automatically locate lung opacities on chest radiographs.

1.4 About The Given Dataset

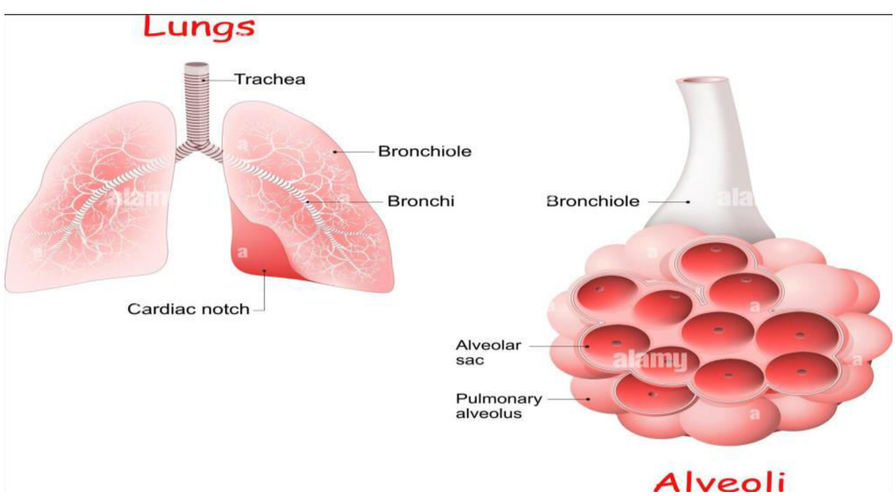
In this capstone project, the goal is to build a pneumonia detection system, to locate the position of inflammation in an image.
Tissues with sparse material, such as lungs which are full of air, do not absorb the X-rays and appear black in the image. Dense tissues such as bones absorb X-rays and appear white in the image.
While we are theoretically detecting “lung opacities”, there are lung opacities that are not pneumonia related.
In the data, some of these are labelled “Not Normal No Lung Opacity”. This extra third class indicates that while pneumonia was determined not to be present, there was nonetheless some type of abnormality on the image and oftentimes this finding may mimic the appearance of true pneumonia.

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

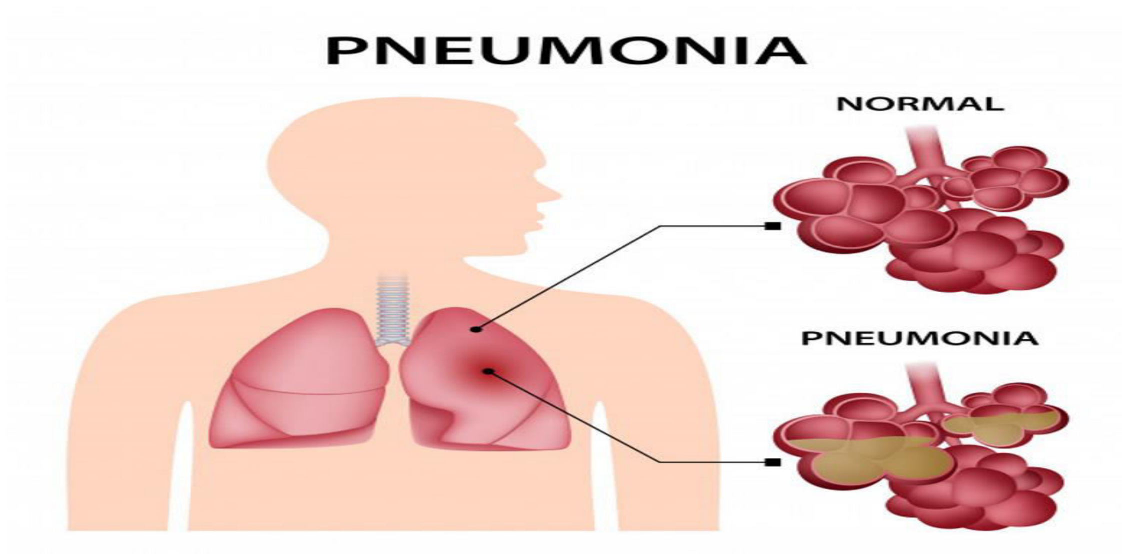
2 Introduction

2.1 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.
Alveoli are **tiny air sacs in your lungs that take up the oxygen you breathe in and keep your body going**.



When an individual has pneumonia, **the alveoli are filled with pus and fluid, which makes breathing painful and limits oxygen intake**. As shown in below figure:

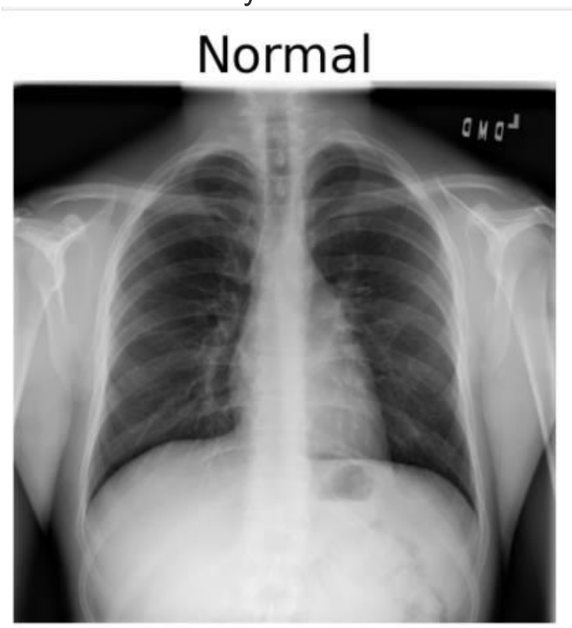


2.2 What Does a Normal Image Look Like?

In an X-ray, 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.

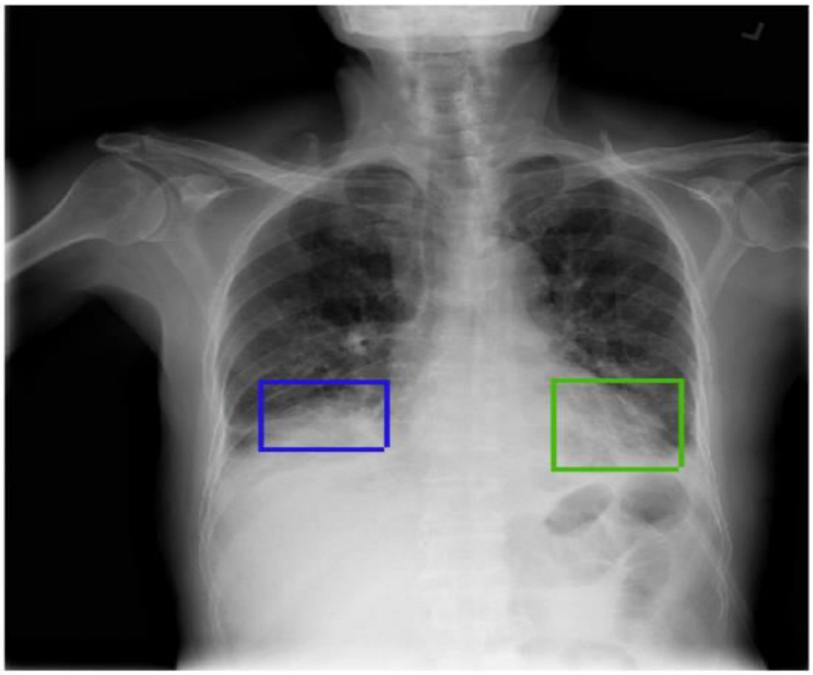
In short –

- Black = Air
- White = Bone
- Grey = Tissue or Fluid



2.3 What Does a Pneumonia Image Look Like?

Now let's look at an image from the data set with lung opacities. Opacity here is loosely defined as any area of the chest x-ray which appears whiter than it should be as shown below. These areas look like "fuzzy clouds".



Usually, the lungs are full of air. When someone has **pneumonia**, the air in the lungs is replaced by other material – fluids, bacteria, immune system cells, etc. That's why areas of opacities are areas that are grey but should be blacker. When we see them, we understand that the lung tissue in that area is probably not healthy.

2.4 Chest Radiographs Basics

The chest x-ray is the most commonly performed diagnostic x-ray examination. A chest x-ray produces images of the heart, lungs, airways, blood vessels and the bones of the spine and chest.

An x-ray exam helps doctors diagnose and treat medical conditions. It exposes you to a small dose of ionizing radiation to produce pictures of the inside of the body. X-rays are the oldest and most often used form of medical imaging.

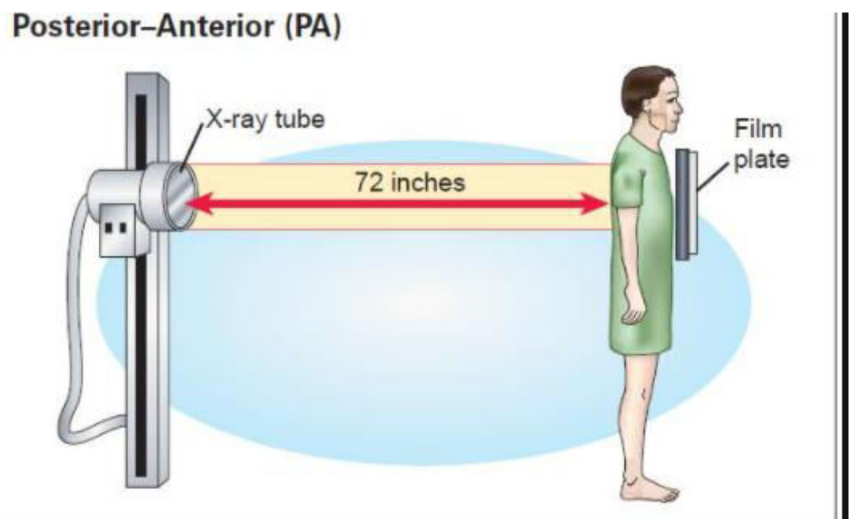
The chest x-ray is performed to evaluate the lungs, heart and chest wall.

A chest x-ray is typically the first imaging test used to help diagnose symptoms such as:

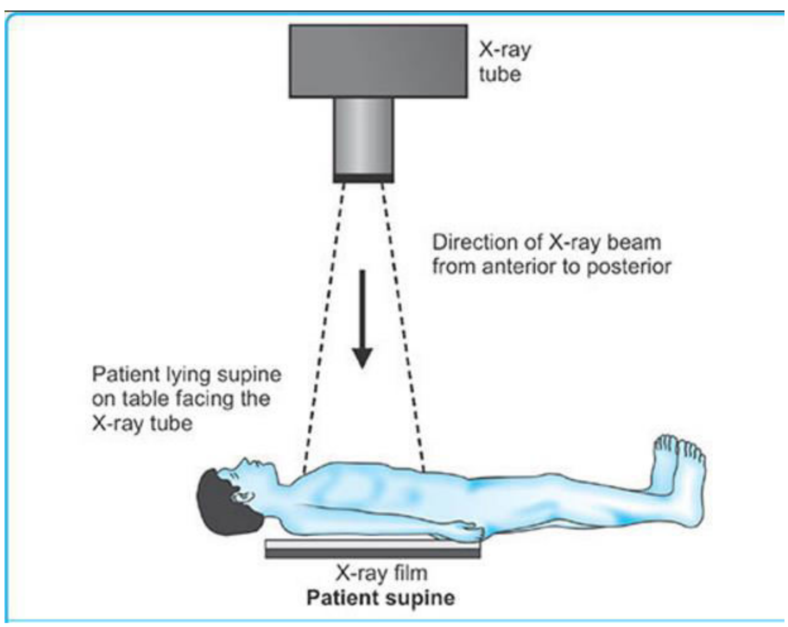
- breathing difficulties
- a bad or persistent cough
- chest pain or injury
- fever

AP (Anterior/Posterior) and PA (Posterior/Anterior):These type of X-rays are mostly used to obtain the front-view. Apart from front-view, a lateral image is usually taken to complement the front-view.AP X-rays are more than PA.

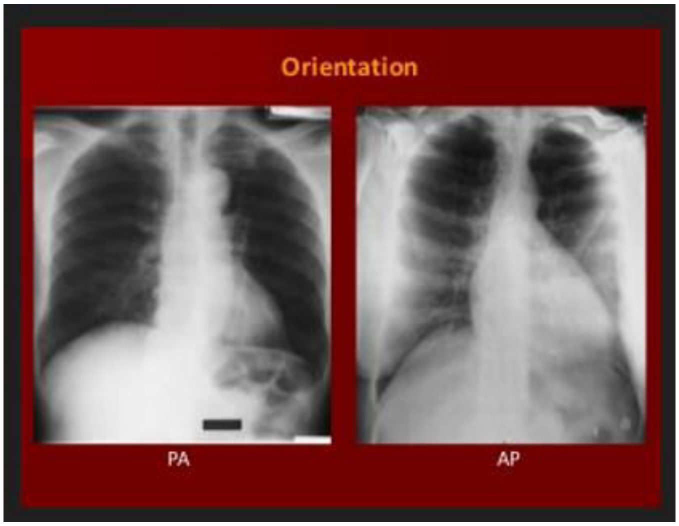
Posterior/Anterior (PA): Here the chest radiograph is acquired by passing the X-Ray beam from the patient's posterior (back) part of the chest to the anterior (front) part. While obtaining the image patient is asked to stand with their chest against the film. These are of higher quality and assess the heart size more accurately.

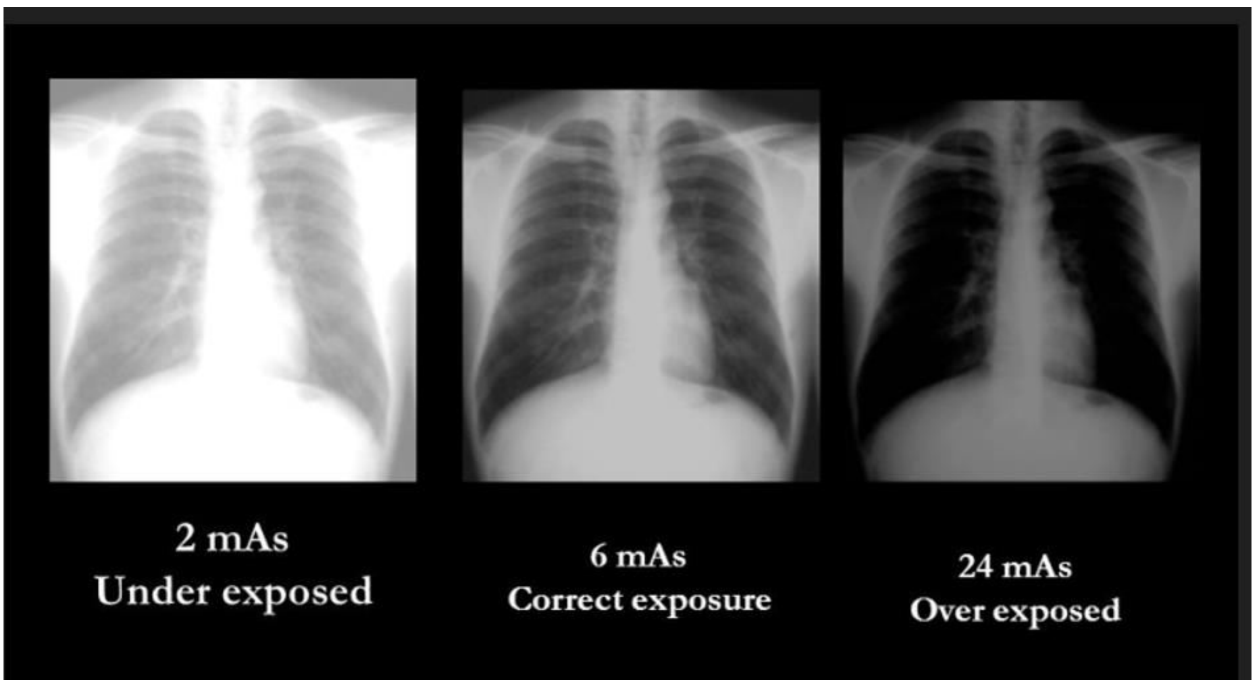


Anterior/Posterior (AP): At times it is not possible for radiographers to acquire a PA chest X-ray. This is usually because the patient is too unwell to stand. In these images the size of Heart is exaggerated.



X-ray : PA and AP position:

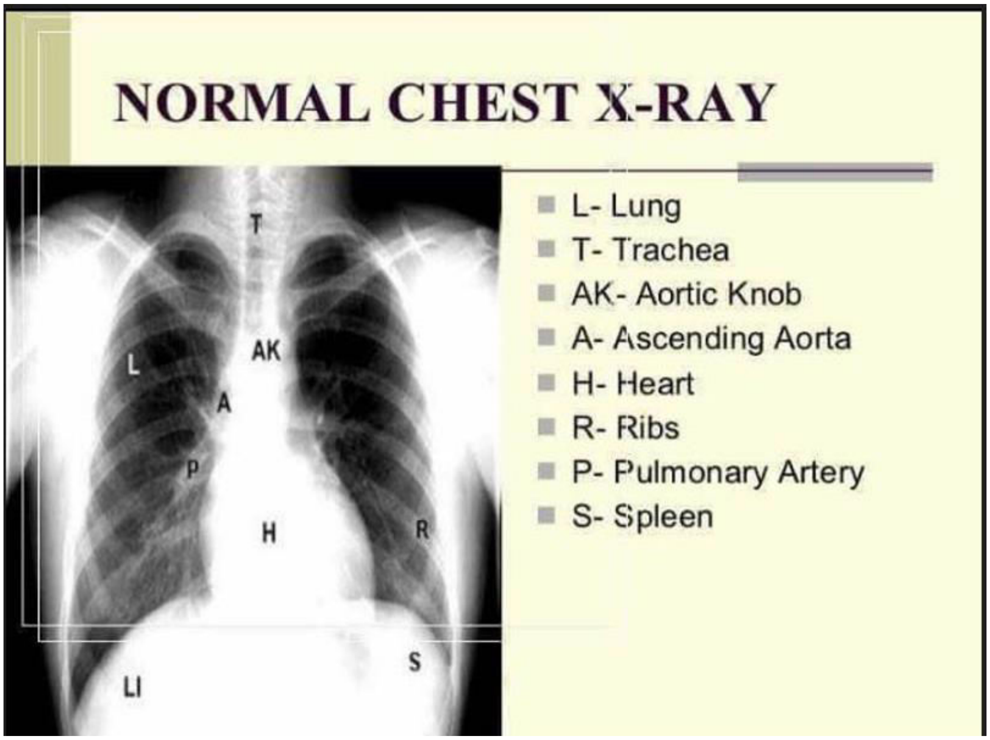


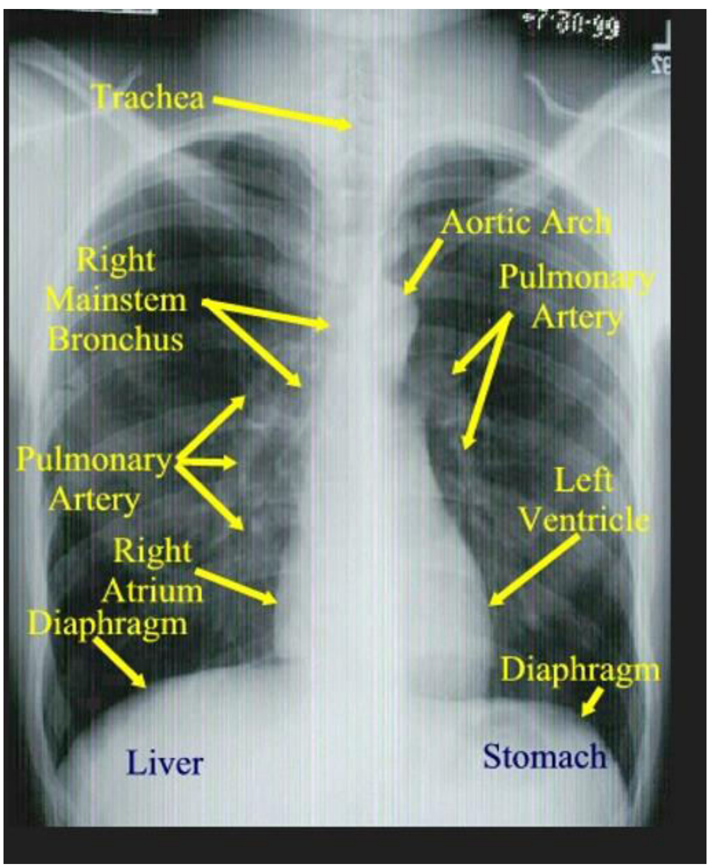


(A) **Overexposure** makes it easy to see behind the heart and the regions of the clavicles and thoracic spine, but the pulmonary vessels peripherally are impossible to see.

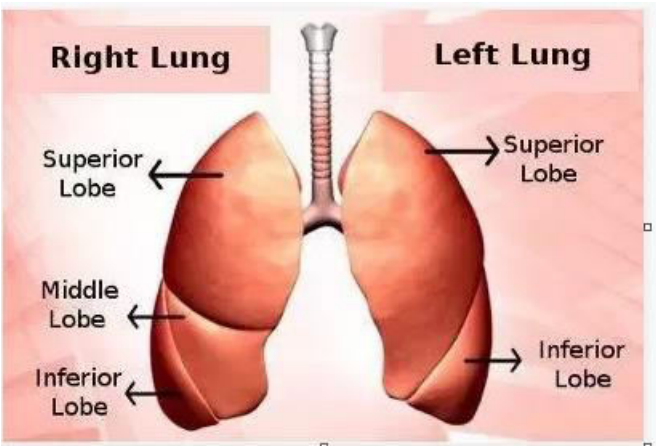
(B) **Underexposure** accentuates the pulmonary vascularity, but you cannot see behind the heart or behind the hemidiaphragms.

Different Parts Of Chest X-ray:

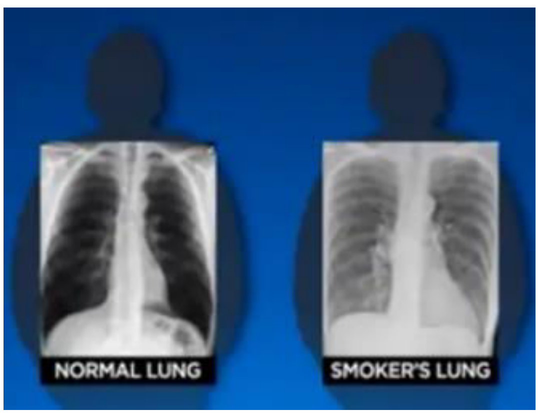




Different Lobes



Normal Lungs vs Smoker Lungs



Rib Cage

3 Data and Findings

3.1 The Initial Set Of Dataset

The dataset contains the following files and folders:

stage_2_train_labels.csv

- The training set. It contains patientIds and bounding box / target information.

stage_2_detailed_class_info.csv

- It provides detailed information about the type of positive or negative class for each image. Apart from the above-mentioned data files (in csv format), the dataset also contains the images folders

stage_2_train_images - It provides all training images.

stage_2_test_images - It provides all testing images.

The images in the above-mentioned folders are stored in a special format called DICOM files (*.dcm). They contain a combination of header metadata as well as underlying raw image arrays for pixel data.

3.2 Finding From Given Dataset

- Data set is imbalanced with more records of Normal persons than having Pneumonia.
- The training dataset (both of the csv files and the training image folder) contains information of 26684 patients (unique)
- Out of these 26684 unique patients some of these have multiple entries in the both of the csv files
- Most of the recorded patient belong to Target = 0 (i.e., they don't have Pneumonia)
- Pneumonia found in Male is more than female.
- Some of the patients have more than one bounding box. The maximum being 4 The classes "No Lung Opacity / Not Normal" and "Normal" is associated with Target = 0 whereas "Lung Opacity" belong to Target = 1
- The images are present in dicom format, from which information like PatientAge, PatientSex, ViewPosition etc are obtained.

- There are two ways from which images were obtained: AP and PA.
- There are few patients whose age is much higher. Outliers due to typo error and for the patients who does not have Pneumonia

- The centres of the bounding box are spread out over the entire region of the lungs. But there are some centres which are outliers.
- When the pixel spacing is large the number of cases of Pneumonia detection is lower.

4 Data Analysis

4.1 Shape of datasets

S No	File name	Rows	Columns
1	stage_2_detailed_class_info	30227	2
2	stage_2_train_labels	30227	6
3	stage_2_test_images	3000 Dicom format Images	
4	stage_2_train_images	26700 Dicom format Images	

4.2 Findings from the given two csv files

```
class_df=pd.read_csv('stage_2_detailed_class_info.csv')
```

```
class_df.head(5)
```

	patientId	class
0	0004cfab-14fd-4e49-80ba-63a80b6bddd6	No Lung Opacity / Not Normal
1	00313ee0-9eaa-42f4-b0ab-c148ed3241cd	No Lung Opacity / Not Normal
2	00322d4d-1c29-4943-afc9-b6754be640eb	No Lung Opacity / Not Normal
3	003d8fa0-6bf1-40ed-b54c-ac657f8495c5	Normal
4	00436515-870c-4b36-a041-de91049b9ab4	Lung Opacity

Data Information:

- Target Variable with Pneumonia is distinguished as Lung Opacity
- Target variable without Pneumonia is divided into two classes No Lung Opacity/Not Normal and Normal

```
labels_df=pd.read_csv('stage_2_train_labels.csv')
```

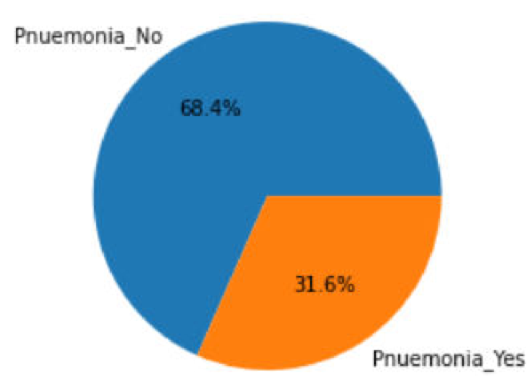
```
labels_df.head(5)
```

	patientId	x	y	width	height	Target
0	0004cfab-14fd-4e49-80ba-63a80b6bddd6	NaN	NaN	NaN	NaN	0
1	00313ee0-9eaa-42f4-b0ab-c148ed3241cd	NaN	NaN	NaN	NaN	0
2	00322d4d-1c29-4943-afc9-b6754be640eb	NaN	NaN	NaN	NaN	0
3	003d8fa0-6bf1-40ed-b54c-ac657f8495c5	NaN	NaN	NaN	NaN	0
4	00436515-870c-4b36-a041-de91049b9ab4	264.0	152.0	213.0	379.0	1

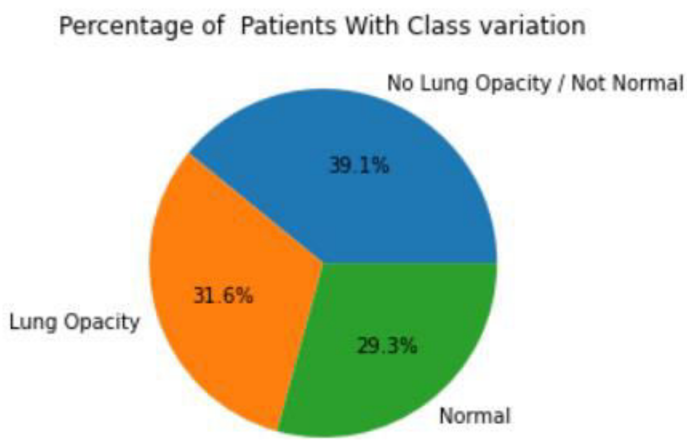
Data Information

- Labels csv consists of PatientId,
- Bounding box X,Y positions and Its Width and Height
- Target whether the Image has Pneumonia or not.
- If the patient doesn’t have Pneumonia the bounding box coordinates are NAN

Percentage of Patients With and without Pnuemonia

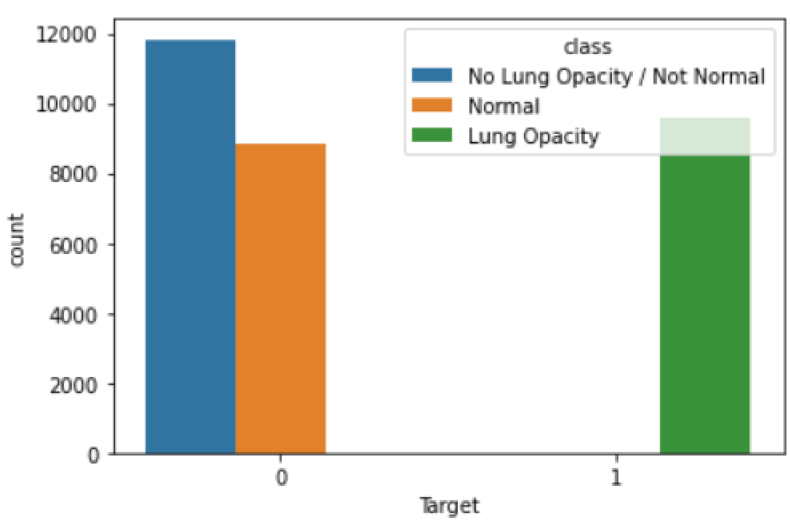


class	Lung Opacity	No Lung Opacity / Not Normal	Normal
patientId	9555	11821	8851

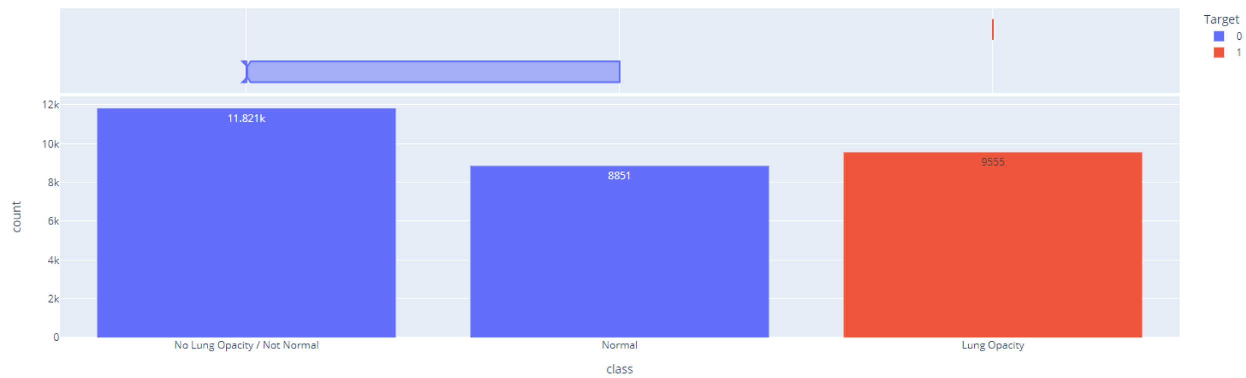


- Observation : Out of 68.4% Having no Pneumonia 39.1% has class of No lung Opacity/Not Normal and 29.3% with class as Normal.

class	Lung Opacity	No Lung Opacity / Not Normal	Normal
patientId	9555	11821	8851
x	9555	0	0
y	9555	0	0
width	9555	0	0
height	9555	0	0
Target	9555	11821	8851



“No Lung Opacity + Normal “ =0 : no Pneumonia and Lung Opacity =1 : Pneumonia



	number_of_patientId
number_of_boxes	
1	23286
2	3266
3	119
4	13

Observation: Thus, there are

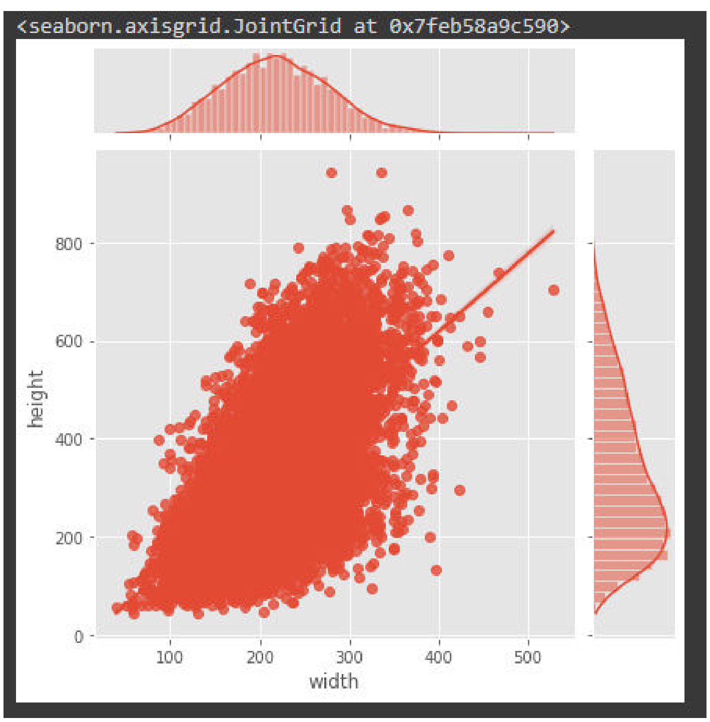
- 23286 unique patients which have only one entry in the dataset. It also has the patients bounding box
- 3266 with 2 bounding boxes
- 119 with 3 bounding boxes
- 13 with 4 bounding box coordinates.

Target	**0**	**1**	**All**
class			
Lung Opacity	0	9555	9555
No Lung Opacity / Not Normal	11821	0	11821
Normal	8851	0	8851
All	20672	9555	30227

Data observation:

- No. of Nulls for the boundary boxes matches with number of target class of No Pneumonia.(20672)
- Out of 9555 Pneumonia there might be duplicate patients who has more than one bounding box

Correlation between width and Height:



Data observation:

There is a fine correlation between width and height variables, which is obvious.

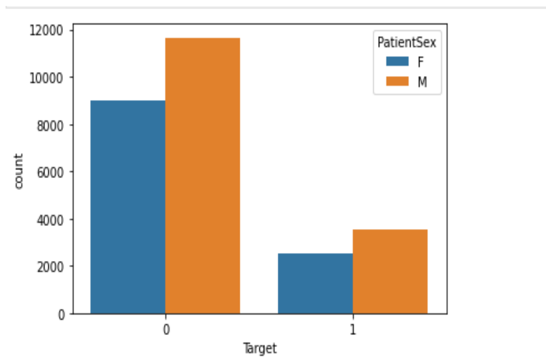
4.3 Exploring the metadata information from Image

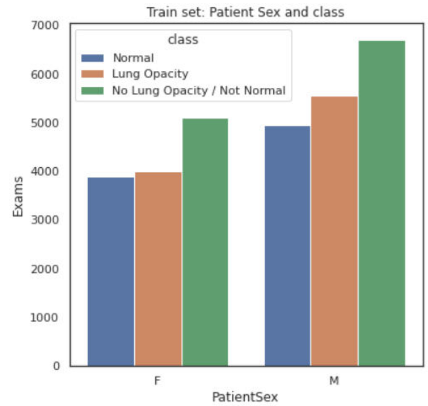
```
Metadata of the image consists of
 Dataset.file_meta -------------------------------
(0002, 0000) File Meta Information Group Length  UL: 202
(0002, 0001) File Meta Information Version       OB: b'\x00\x01'
(0002, 0002) Media Storage SOP Class UID         UI: Secondary Capture Image Storage
(0002, 0003) Media Storage SOP Instance UID      UI: 1.2.276.0.7230010.3.1.4.8323329.28530.1517874485.775526
(0002, 0010) Transfer Syntax UID                 UI: JPEG Baseline (Process 1)
(0002, 0012) Implementation Class UID            UI: 1.2.276.0.7230010.3.0.3.6.0
(0002, 0013) Implementation Version Name         SH: 'OFFIS_DCMTK_360'
-------------------------------------------------
(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: 1.2.276.0.7230010.3.1.4.8323329.28530.1517874485.775526
(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: PA'
(0010, 0010) Patient's Name                      PN: '0004cfab-14fd-4e49-80ba-63a80b6bddd6'
(0010, 0020) Patient ID                          LO: '0004cfab-14fd-4e49-80ba-63a80b6bddd6'
(0010, 0030) Patient's Birth Date                DA: ''
(0010, 0040) Patient's Sex                       CS: 'F'
(0010, 1010) Patient's Age                       AS: '51'
(0018, 0015) Body Part Examined                  CS: 'CHEST'
(0018, 5101) View Position                       CS: 'PA'
(0020, 000d) Study Instance UID                  UI: 1.2.276.0.7230010.3.1.2.8323329.28530.1517874485.775525
(0020, 000e) Series Instance UID                 UI: 1.2.276.0.7230010.3.1.3.8323329.28530.1517874485.775524
(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: [0.14300000000000002, 0.14300000000000002]
(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 142006 elements
```

Observation: Size of this image is 1024 x 1024 (rows x columns).

4.4 Exploring Data after Concatenating CSV file and Image file metadata

Female and male ratio





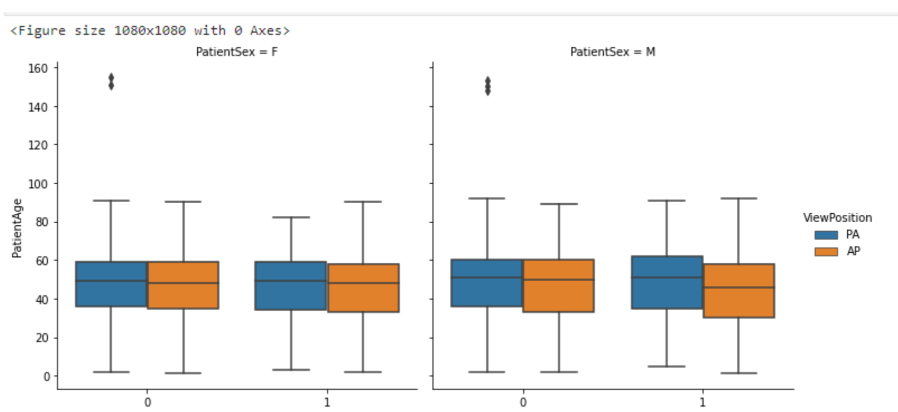
Observation: Male Patients affected with Pneumonia is high compared to Female

AP and PA details:

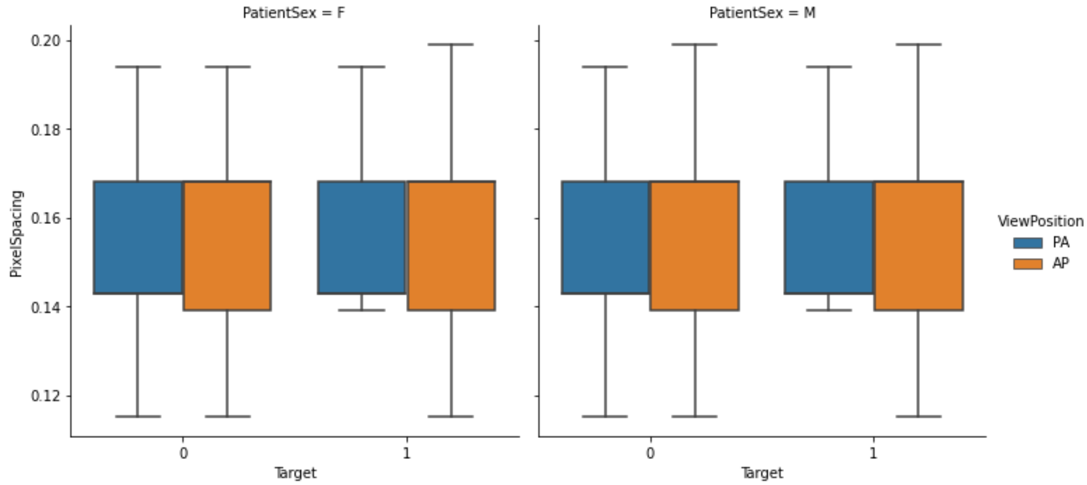
ViewPosition	AP	PA
Target		
0	0.363245	0.636755
1	0.775782	0.224218
All	0.456191	0.543809

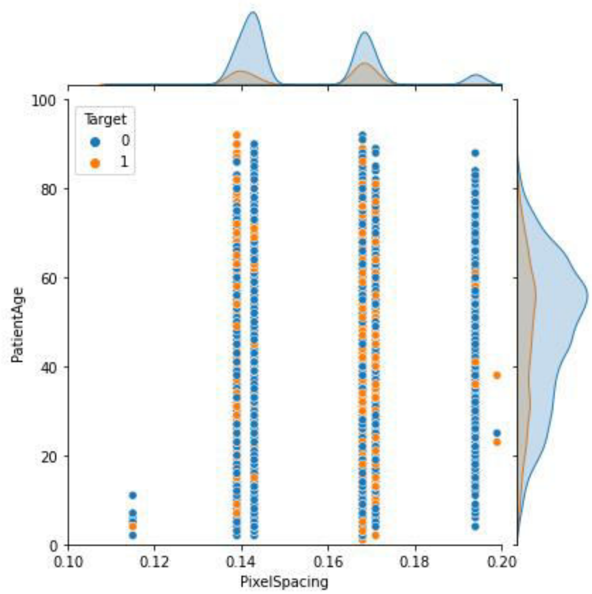
Observation:

- Patients taken with AP Posture (Normally sick already) has higher chances of Pneumonia - Close to 77% out of total Pneumonia patients
- Clustering the images into AP and PA has good chances of predictions of Pneumonia
- Percentage of people with PA Posture are higher



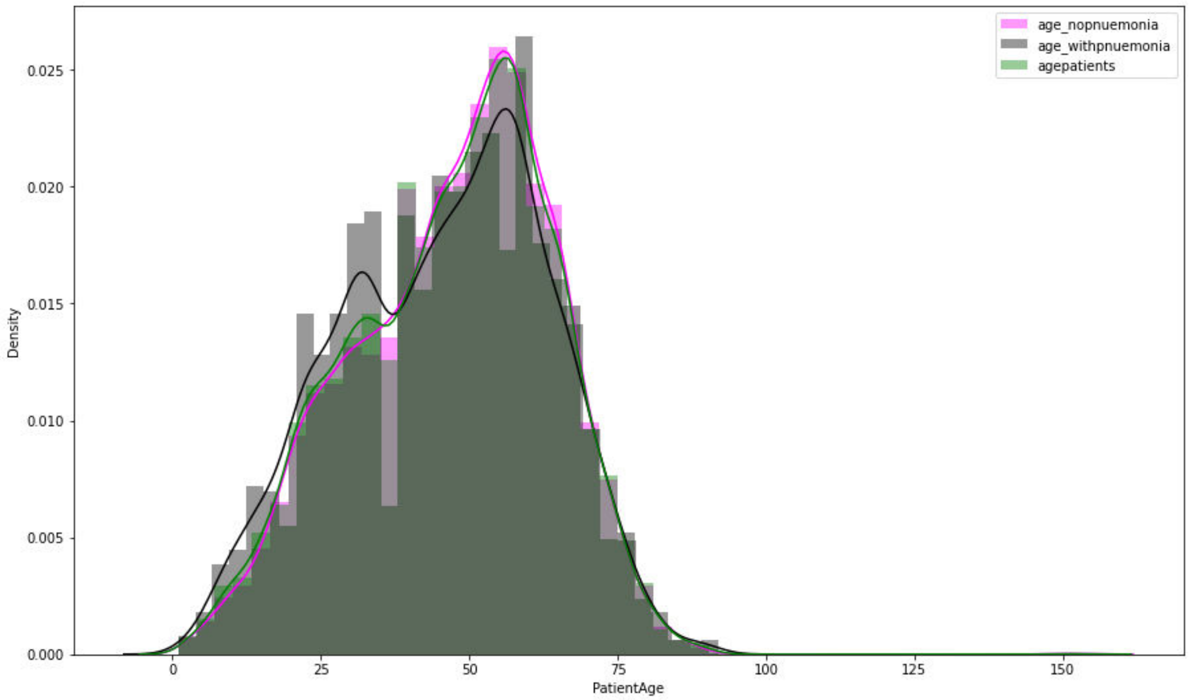
```
<Figure size 1080x1080 with 0 Axes>
```





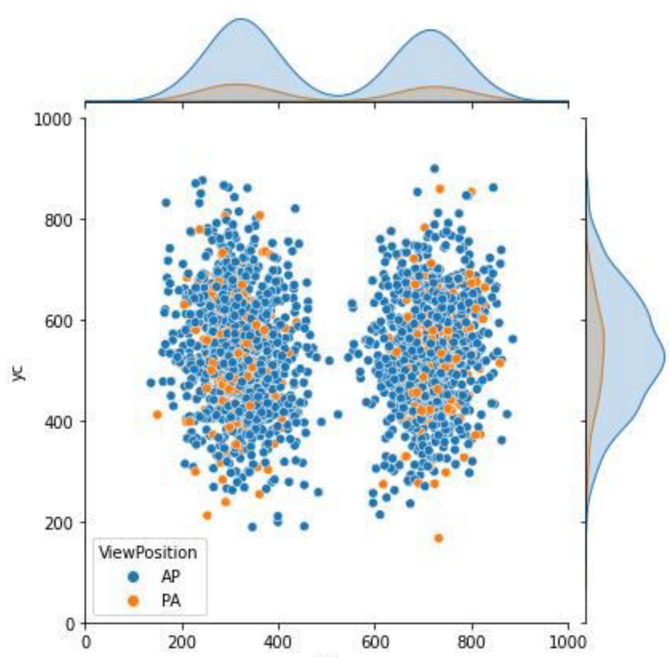
Data Observation:

- Patients with Pneumonia It shows that the median average age for AP position for Male is lower compared to Female
- There are few patients whose age is much higher. Outliers due to typo error and for the patients who does not have Pneumonia
- Patients who are affected with Pneumonia the Pixel spacing for PA has minimum starting from 0.14 whereas AP position it has 0.11
- Pixel spacing for patient age group is less than 10y for the pixel spacing around 0.12.
- Pixels pacing between 0.16 to 0.18 has larger number of patients affected with Pneumonia
- When the pixel spacing is large the number of cases of Pneumonia detection is lower



Data Observation

- We can see the mostly effected patients coming for Pneumonia check are in the age group of 50 to 60 and the greatest number of infected patients are also in the same age group
- We can see from 30 to 40y age group are affected more with Pneumonia

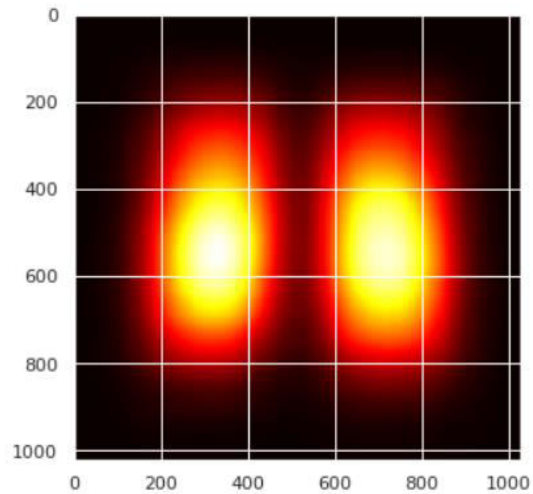


Observation

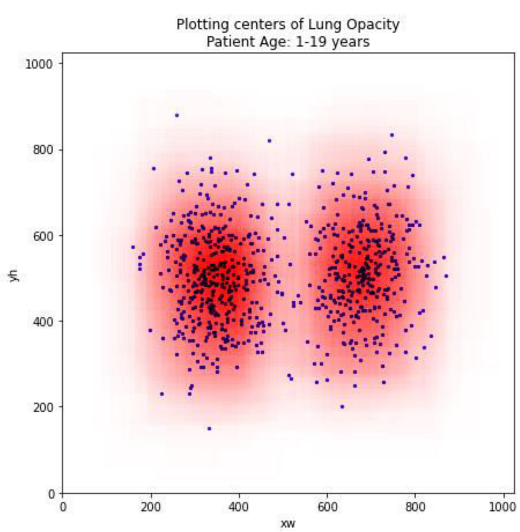
- Patients with Box centers for PA position are mostly concentrated at the centers
- AP Position has centers all over the lungs area the detection of Pneumonia
- Few Outlier boxes are visible for the left lungs with PA position

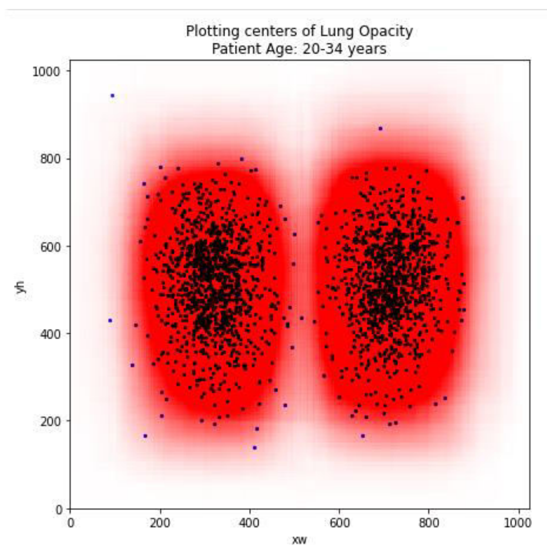
Lung Opacity

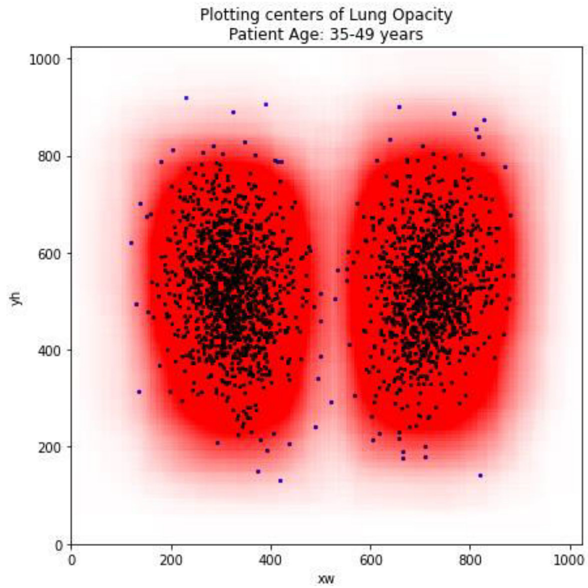
Showing bounding boxes as segmentation we can get a better probability map for where the opacity regions are most likely to occur. As we can see, the opacity is mostly concentrated in the centre.

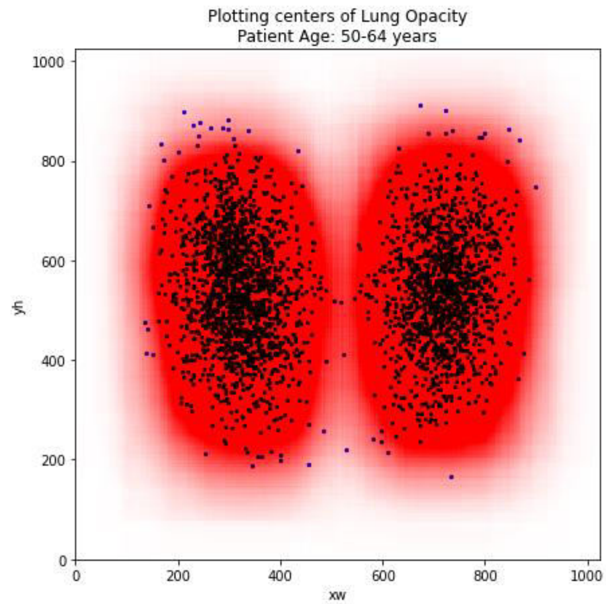


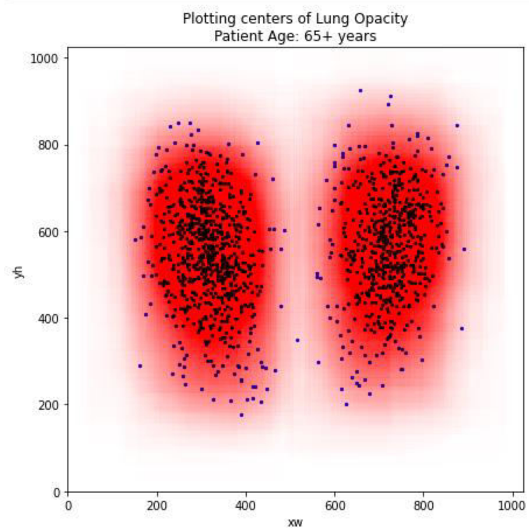
Age wise plotting canters of Lung Opacity:











Observation: Pixel spacing is low for lower age group.

4.5 Summary Of the Approach to EDA and Pre-processing

4.5.1 EDA Approach

- Exploring given data files, classes and images of different classes.
- Mostly graphical approach to data analysis
- Emphasizes uncovering underlying structure of data, extract important variables, detect outliers, maximize insight into dataset.
- Graph the data
- Analysis from the visualization of different classes.

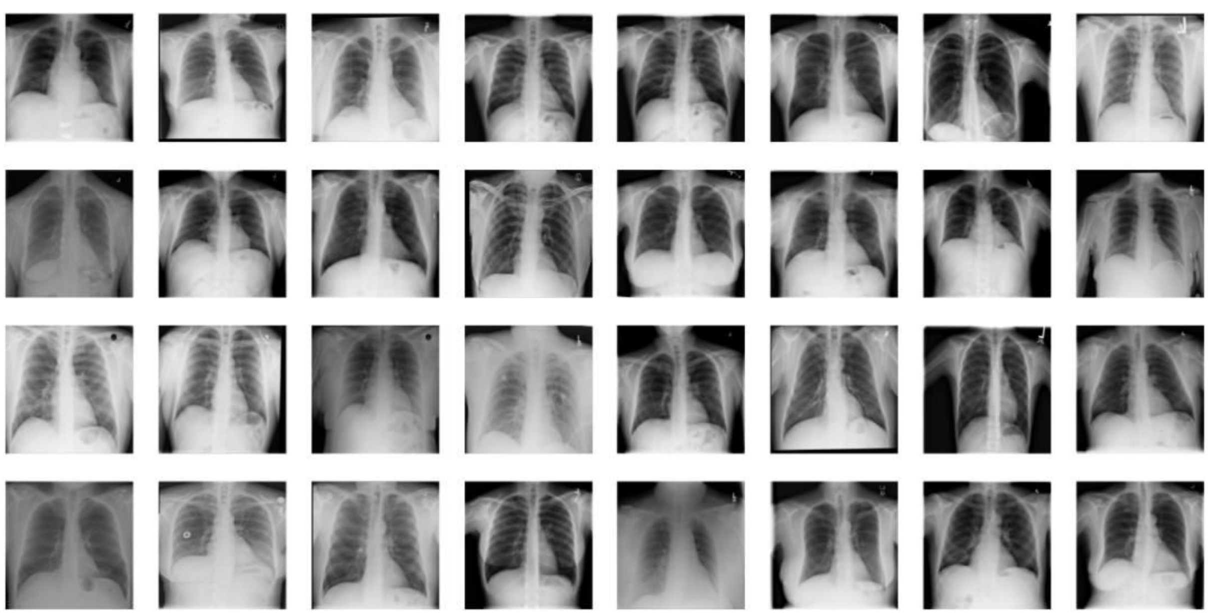
- Test underlying assumptions
- Develop basic model (using transfer learning)
- Plotting raw data using pie chart, boxplot, histogram, bar chart etc.

4.5.2 Data Pre-processing approach

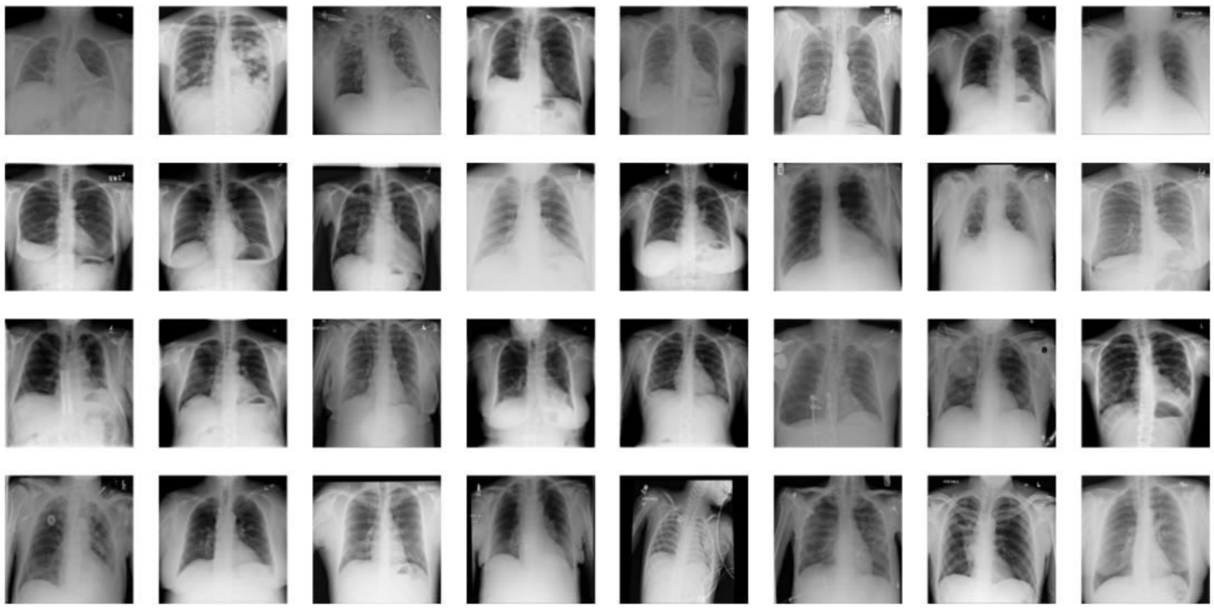
- Dimensionality Reduction
- Finding Missing Data
- Splitting data into training and test set
- Feature Scaling
- Convert 1 channel image to 3 channel images (Gray to RGB conversion)
- Used CLAHE Normalization which helps in producing sharper images
- Heatmap that shows the Image areas that contributed to the prediction decision using VGG and can use **VizGradCAM** to plot it.
- Image CLAHE with augmentation.

5 Visualizing the given images

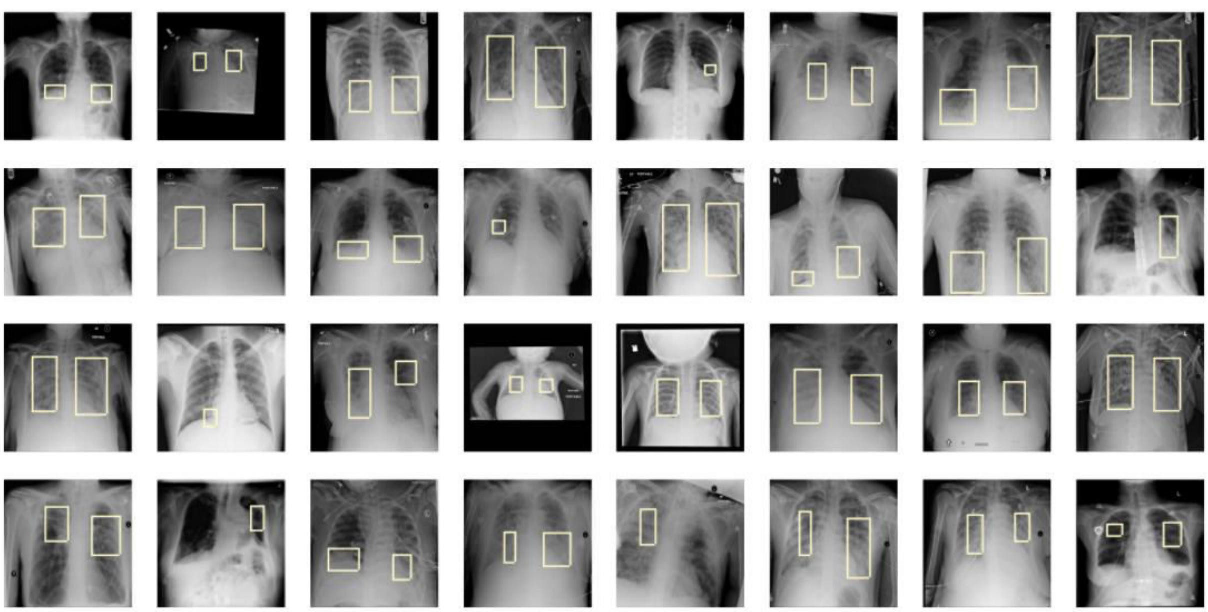
Normal Images:



No Lung Opacity / Not Normal Images

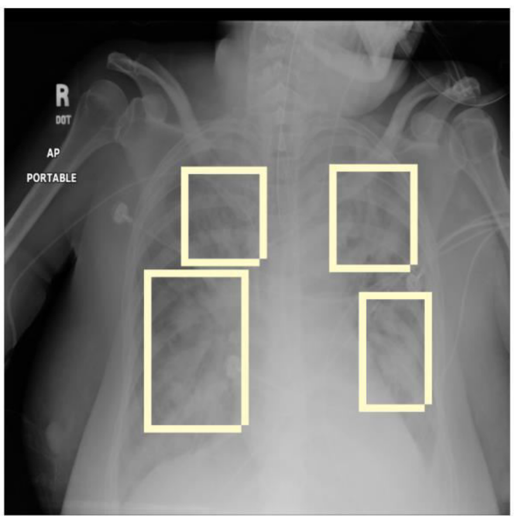


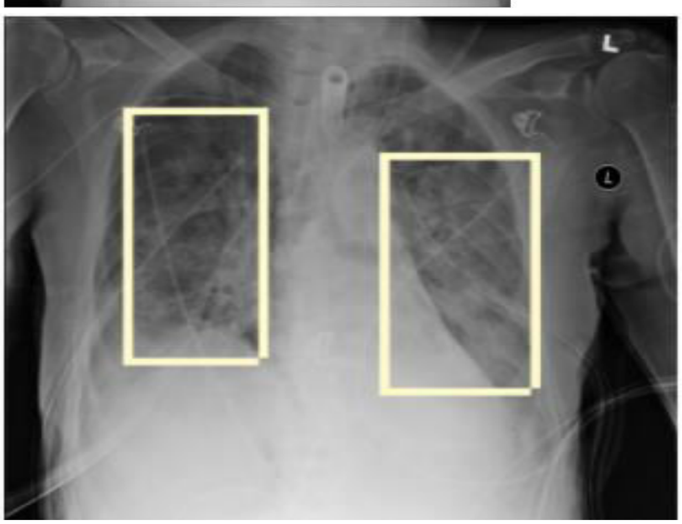
Lung Opacity Images



Side By Side Comparison:

Lung Opacity Closure Look:



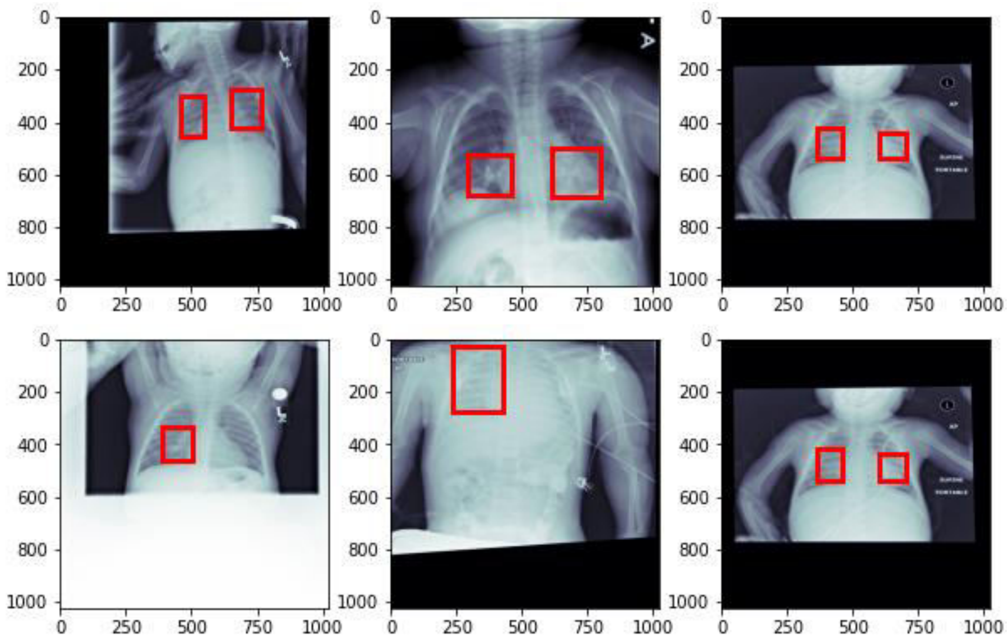


Observation: In the Lung Opacity images we can see that there is haziness were the labeled boxes are (termed ground glass opacity) and/or a loss of the usual boundaries of the lungs (termed consolidation). I will go in depth into these terms in the "A Clear and Detailed Definition of Pneumonia Associated Lung Opacities" section. You can also see that patients with pneumonia are ill and have different cables, stickers, and tubes connected to them. If you see a round white small opacity in and around the lungs it's probably an ECG sticker.

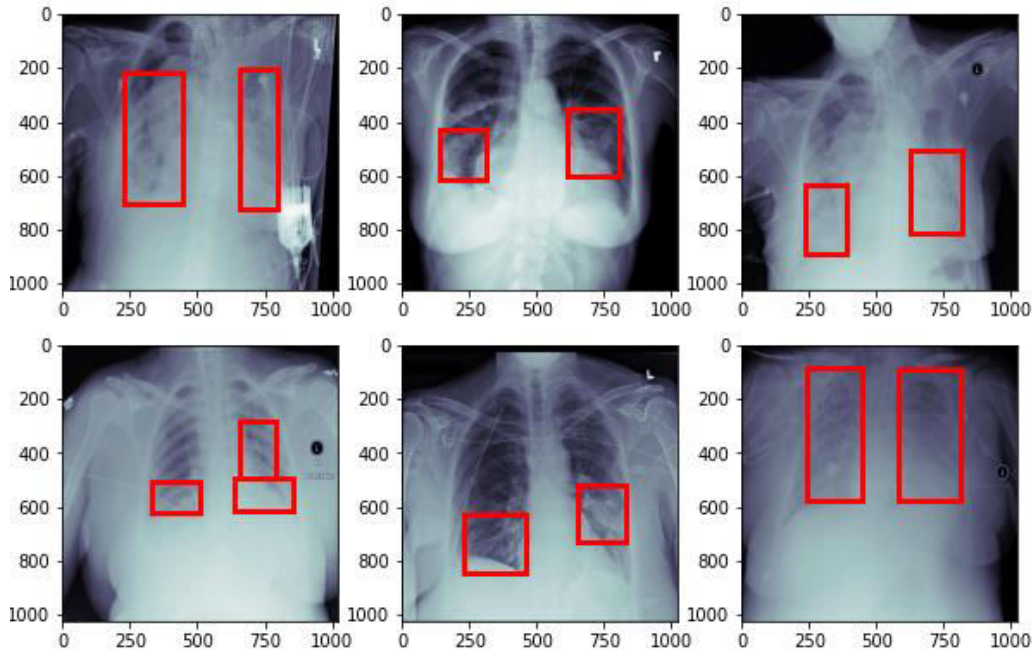
5.1 EDA on Images

Based on the EDA it is observed that the images for different age groups have different sizes and position within the given resolution .

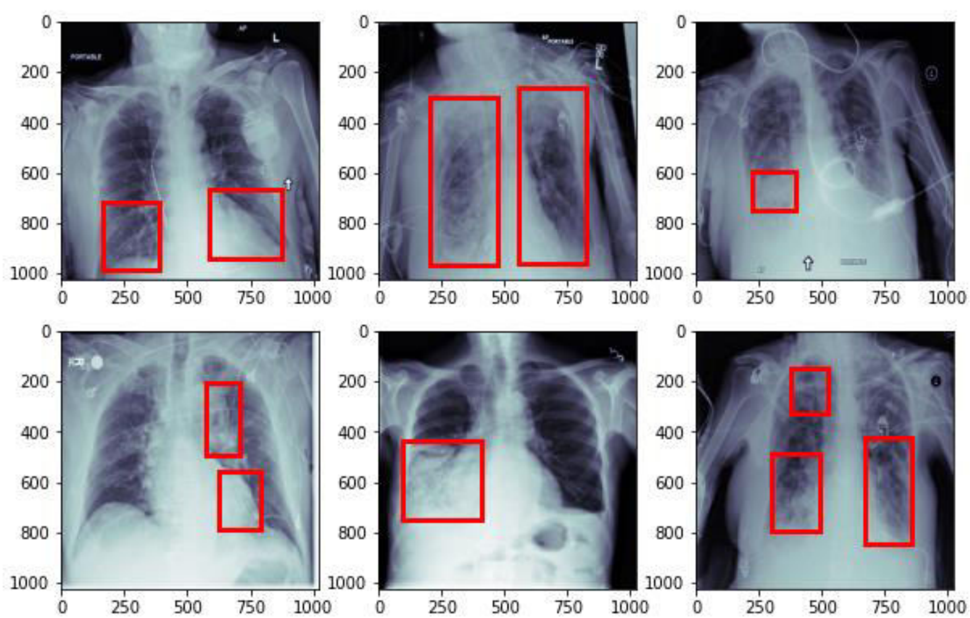
- Patients with age group less than 4y



- Patients with age group between 50 to 60



- Patients with age group greater than 80y



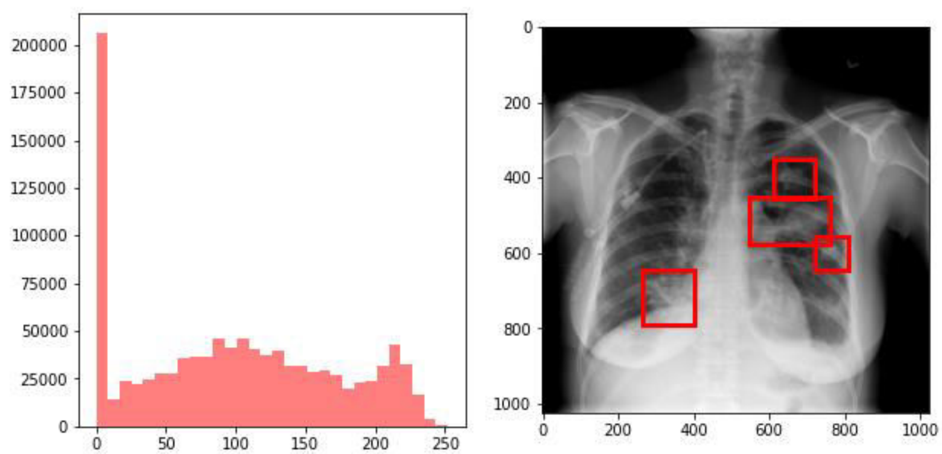
- Based on the EDA of Images we can conclude that the alignment or orientation of lungs within the resolution size of 1024x1024 is not same. Hence image augmentation would help to improve the model performance
- 4 y patients have smaller size of image

- 50 to 60y age group has increased lung height might be due to prolonged smoking and most of the bounding box sizes are higher.
- 80y and more the X-rays are rotated due to their position of taking the X rays

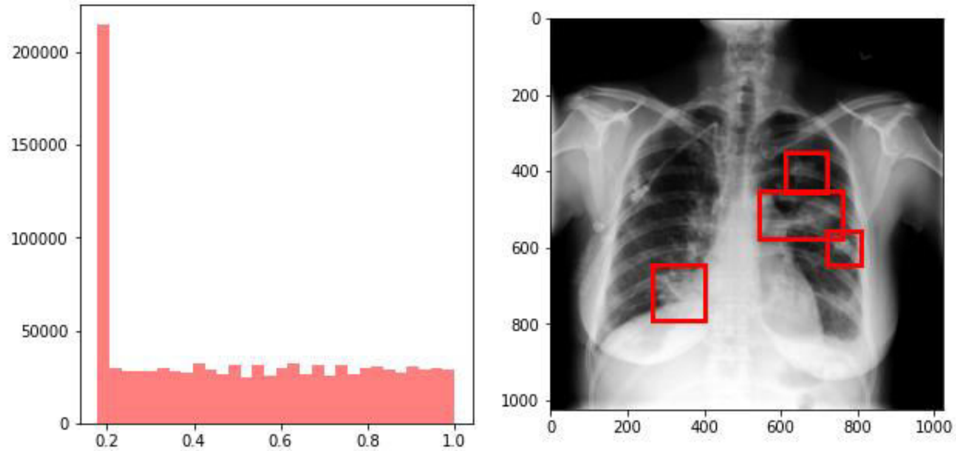
5.2 Image Normalization

- It was observed that the improvement of Image quality can be made by Normalizing the images. Based on the study and literature we had two approaches Histogram Normalization and CLAHE Normalization.
- Histogram normalization tries to make the pixels or images a bit darker.
- CLAHE Normalization helps in producing sharper images

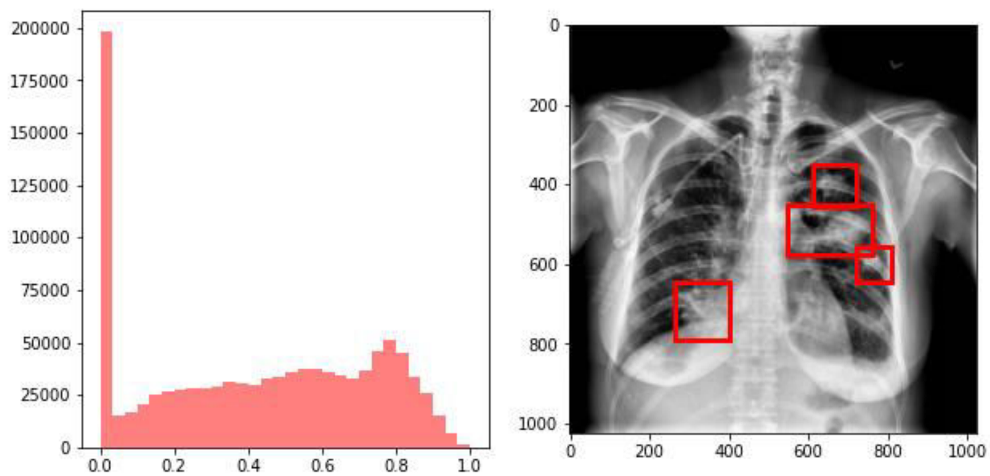
Without Normalization



Histogram Normalization



CLAHE Normalization



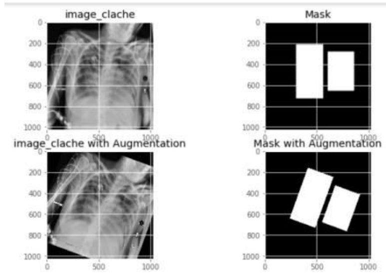
5.3 Augmentation

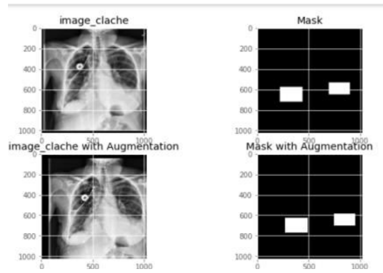
Augmentation of Images and masks is planned using Imgaug Library

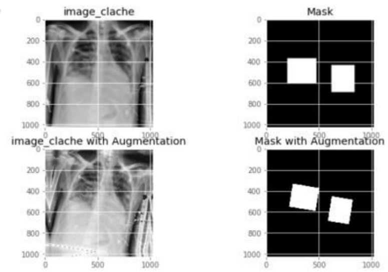
Current Augmentations were done by considering half of the images with Augmentation and rest half normal images were passed for every batch.

1. Rotation + or – 20deg
2. Translation
3. Gaussian Blur
4. Shear
5. Sharpen

Rotation **Translation** **Sharpen , Rotation and Translation**







6 Overview of final process

6.1 Algorithms Used

Models are supervised learning models. We have followed an incremental approach to model building, as the model was built in various stages:

1. Model 1 – created using SVC without using images and only of csv files data to detect Pneumonia

2. Model 2 – was built using RESnet50,InceptionV3 (transfer learning model). The model was able to detect pneumonia with 97% accuracy but did not create any bounding boxes.

3. Model 3 Classification with heatmap showing locations of interest by the model CNN and VGG net model

4. Model 4 Segmentation approach using normal Unet model and transfer model to predict the mask

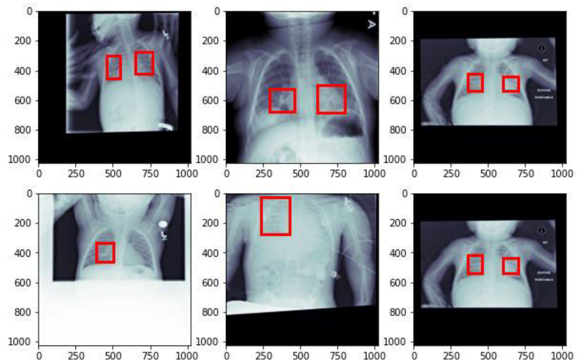
5. Model 5 Object Detection using SSD Mobilenet and Tensorflow Object Detection API

6. Model 6 RCNN to predict BB and classification of Pneumonia.

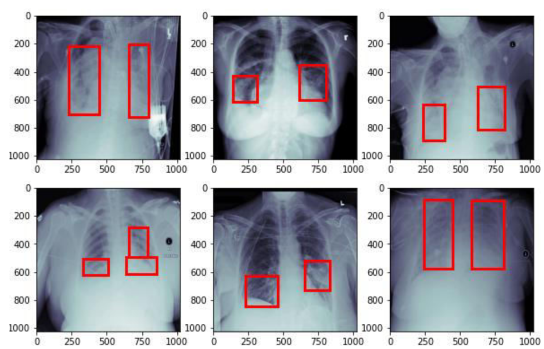
7. Model 7 Mask RCNN

6.2 Salient features of data

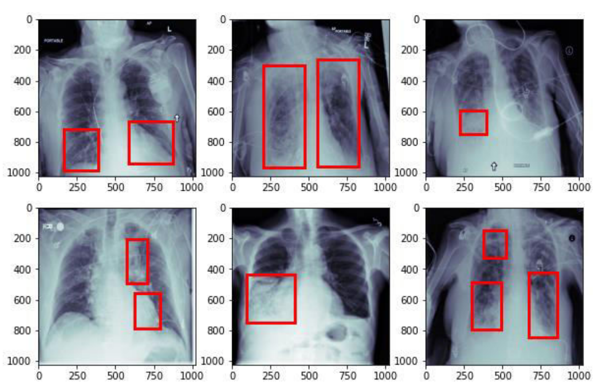
Based on the EDA it is observed that the images for different age groups have different sizes and position within the given resolution .Patients with age group less than 4y



- Patients with age group between 50 to 60

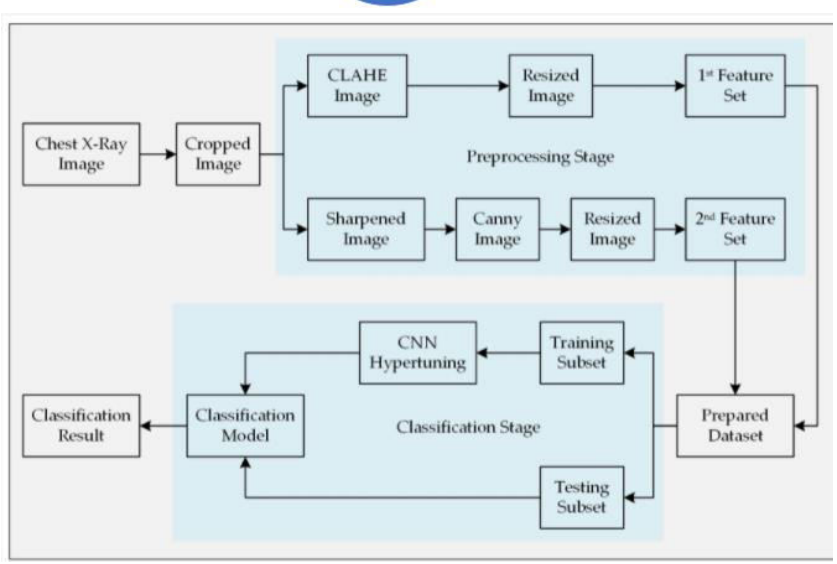


- Patients with age group greater than 80y



- Based on the EDA of Images we can conclude that the alignment or orientation of lungs within the resolution size of 1024x1024 is not same. Hence image augmentation would help to improve the model performance
- 4 y patients have smaller size of image
- 50 to 60y age group has increased lung height might be due to prolonged smoking and most of the bounding box sizes are higher.
- 80y and more the X-rays are rotated due to their position of taking the X rays

6.3 Data pre-processing steps



6.3.1 Data Cleaning

6.3.1.1 Checked Missing Values

6.3.1.2 Removing Outliers

6.3.2 Data Integration

Data Integration is a data preprocessing technique that involves combining data from multiple heterogeneous data sources

6.3.2.1 Merge the data present in multiple sources

6.3.3 Data Transformation

6.3.3.1 Normalization

- **Normalizing Image** Pixels in Keras ... In rescaling the pixel values from 0-255 range to 0-1 range, ImageDataGenerator class can be used.

- **CLAHE Normalization -** which helps in producing sharper images.

- **Histogram normalization** – This is a common technique that is used to enhance fine detail within an image.

6.3.3.2 Augmentation

Data augmentation is useful to improve performance and outcomes of machine learning models by forming new and different examples to train datasets

6.3.4 Data Reduction

Dimensionality reduction is the task of reducing the number of features in a dataset. So, reducing the amount of capacity required to store data. Thus, save computational cost or disk access cost in query processing.

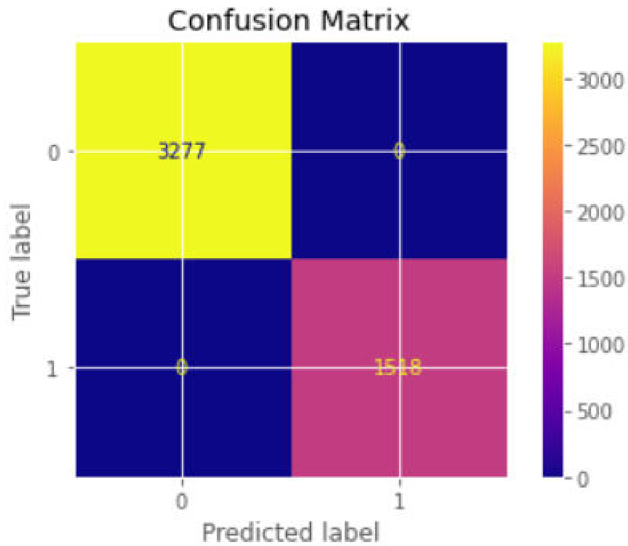
6.4 Step-by-step walk through the solution

6.4.1 Model1 : SVC

- Created using SVC without using images and only of csv files data to detect Pneumonia

- 99% accuracy

-



-

```
Classification report:

              precision    recall  f1-score   support

           0       1.00      1.00      1.00      3277
           1       1.00      1.00      1.00      1518

    accuracy                           1.00      4795
   macro avg       1.00      1.00      1.00      4795
weighted avg       1.00      1.00      1.00      4795
```

- **As this model is not using images to detect Pneumonia we decided to use model with transfer learning Model2.**

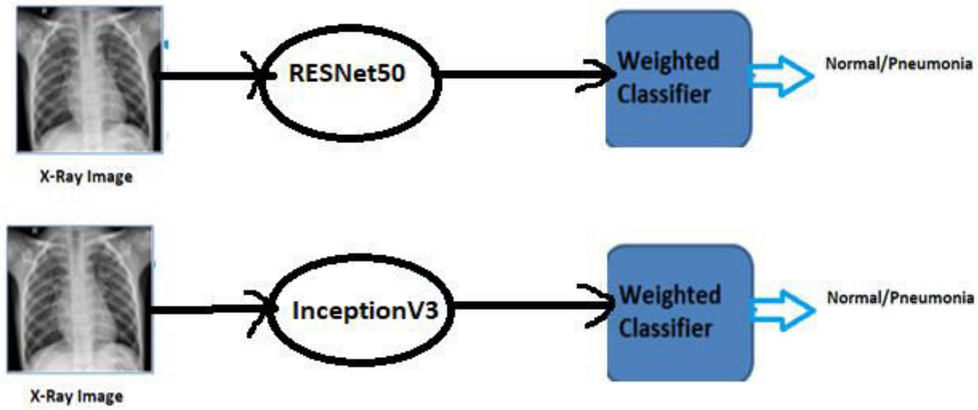
6.4.2 Model2 : Transfer learning models with CNN approach

- This model is built using RESnet50,InceptionV3 (transfer learning model). The model was able to detect pneumonia with 97% accuracy but did not create any bounding boxes.

Transfer learning model for classification of images.

- Transfer learning model ResNet50 with ImageNet weight and InceptionV3 used for predicting Pneumonia.

- 97% accuracy



ResNet50 with ImageNet

- **epochs**=5
- Trainable params: 2,771,281
- Non-trainable params: 23,587,712
- loss: 0.1004
- accuracy: 97%
- steps_per_epoch =10

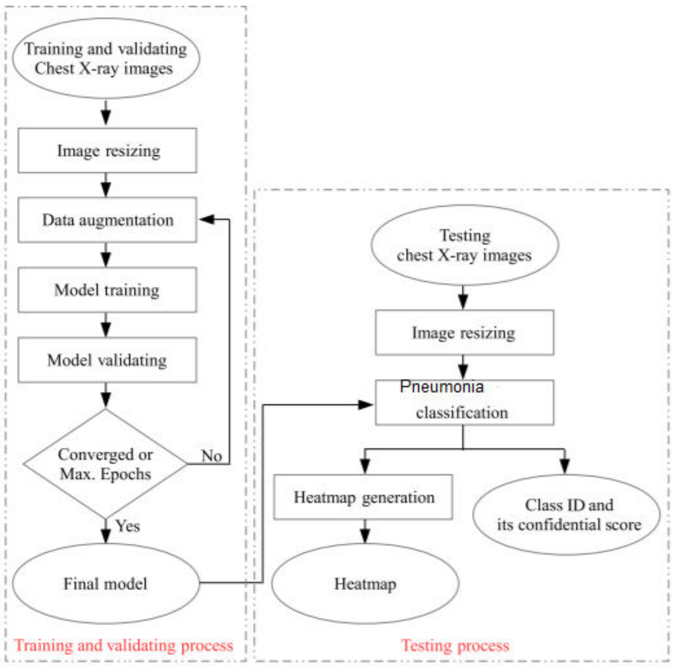
InceptionV3 with imagenet

- **epochs**=5
- Trainable params: 2,771,937
- Non-trainable params: 21,802,784
- loss: 0.6733
- accuracy: 97%
- steps_per_epoch =10

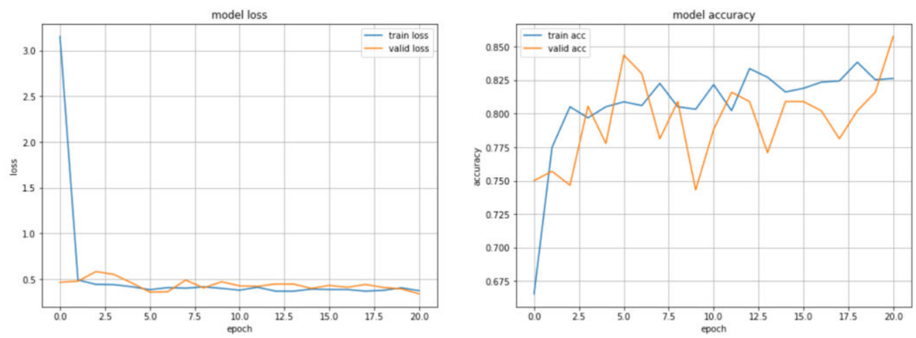
- **As this model is using images to detect Pneumonia but not area of interest in images. So we went for Model3.**

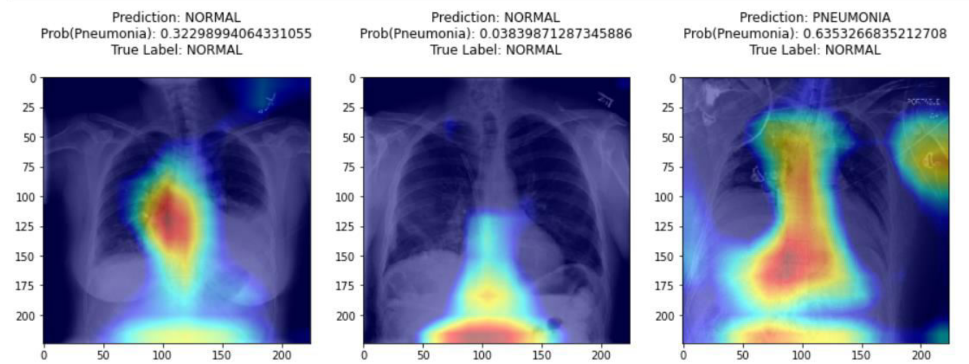
6.4.3 Model 3: Classification with heatmap

- This model is showing locations of interest by the model CNN and VGGnet model

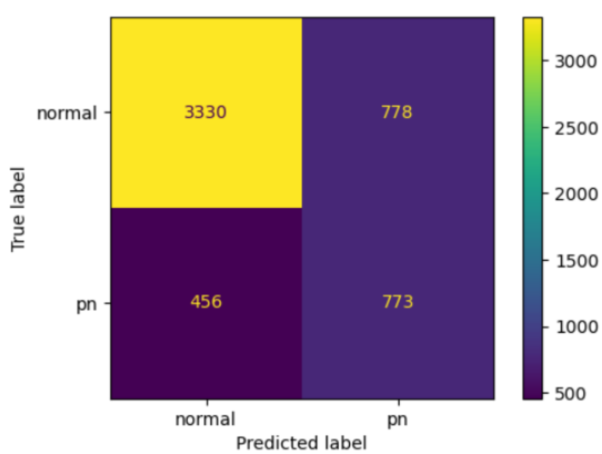


- 85% accuracy
- Used CNN and VGG net model
- Heatmap that shows the Image areas that contributed to the prediction decision using VGG and can use **VizGradCAM** to plot it.
- Trainable params: 107,157,601
- Non-trainable params: 14,714,688
- epochs=25

- 



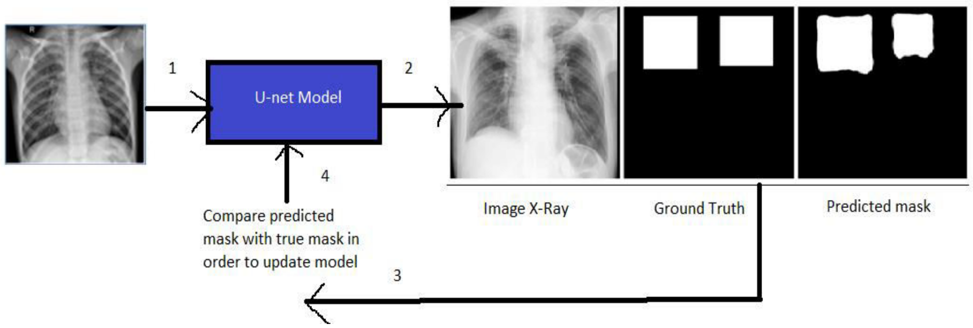

```
CLASSIFICATION REPORT
              precision    recall  f1-score   support

      normal       0.88      0.81      0.84      4108
          pn       0.50      0.63      0.56      1229

    accuracy                           0.77      5337
   macro avg       0.69      0.72      0.70      5337
weighted avg       0.79      0.77      0.78      5337

AUC is  0.8092594338538867
```



- This model is not showing boundary boxes only showing area of interest.
- **Type 1 errors** are 778 and **Type 2 errors** are 456 which are large.
- **Type1 and 2 errors are large so we went for Model4 i.e., to Unet to predict mask**. **Also, no BB detected.**

6.4.4 Model4: Autoencoder (Unet) to Predict the mask

- Unet model -predicting mask



Steps Planned to predict the mask

Model 4.1 - 2layer CNNs for Blocks with Image size 128x128,16 Filters,Dropout 30%

1. Metrics considered initially with Accuracy and IOU and F1score
2. Down sampled the train data and planned to compare different normalizations-No normalization, CLAHE and Histogram
3. For complete train data batch size arrived was 96 images per batch for better compromise on accuracy and computation effort.

Model 4.2 - 3layer CNNs for Blocks

1. Filters 24, Dropout 20% Image size 224x224

Model 4.3 - 3layer CNNs for Blocks with Image size 128x128,16 Filters,Dropout 30%

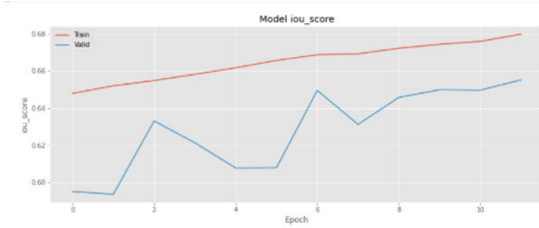
1. CLAHE and No Normalization
2. Augmentation of images

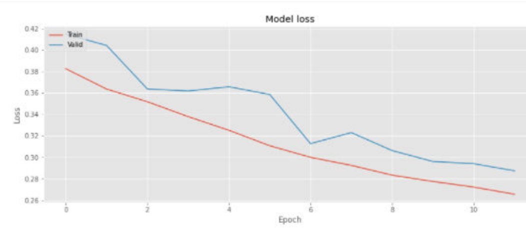
Model 4.4 – Encoder with ResNet34 Transfer model and imagenet Weights

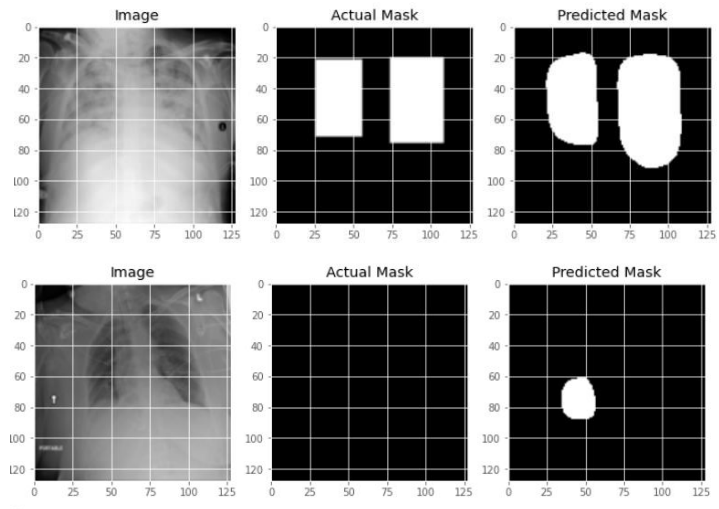
Model 4.1 – Overview of results

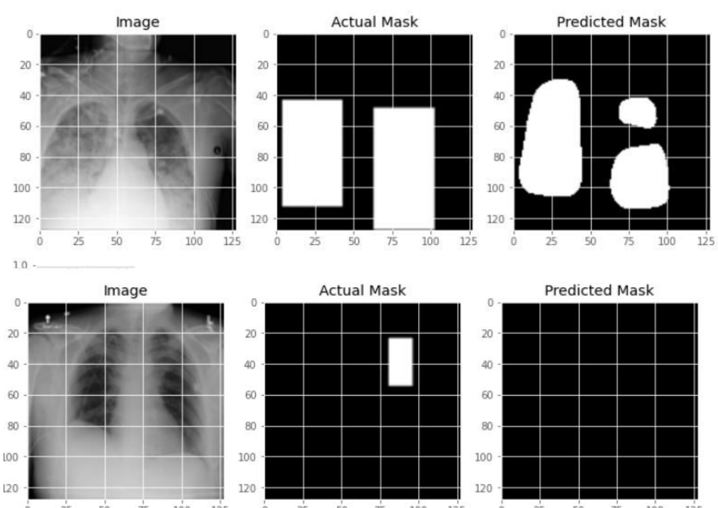
- Down Sample the train data with 6k images of no Pneumonia and 6k with Pneumonia
- Trainable Params were around 21Lakhs
- Accuracy within 8 Epochs we could reach >0.94% but as the data is highly imbalanced

- Hence the metric is changed from Accuracy to IOU with Threshold >0.5 and F1 Score









Comparison of Normalization of images

Image-No Normalization

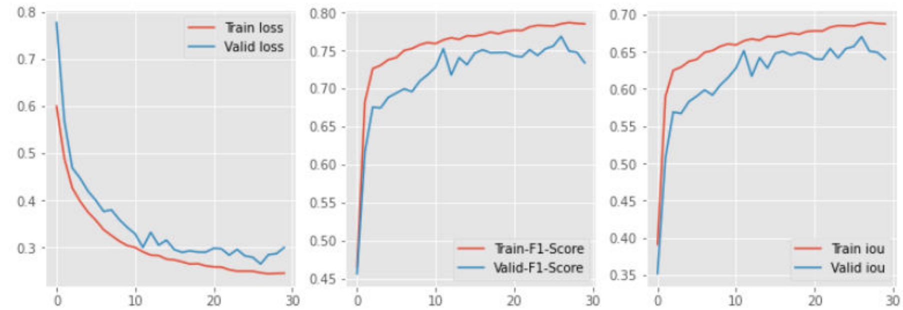


Image-Clache

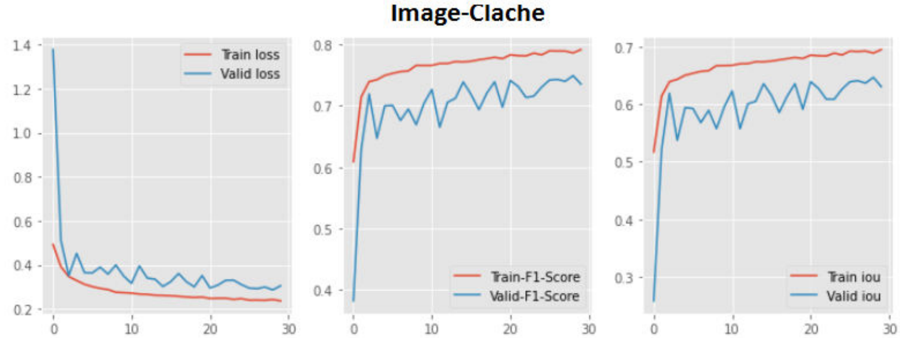
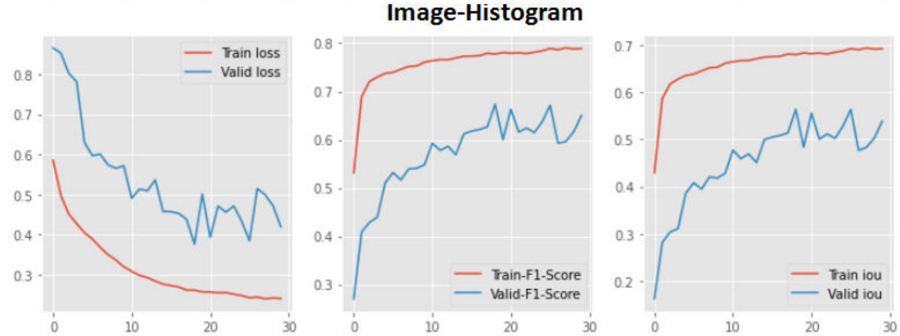


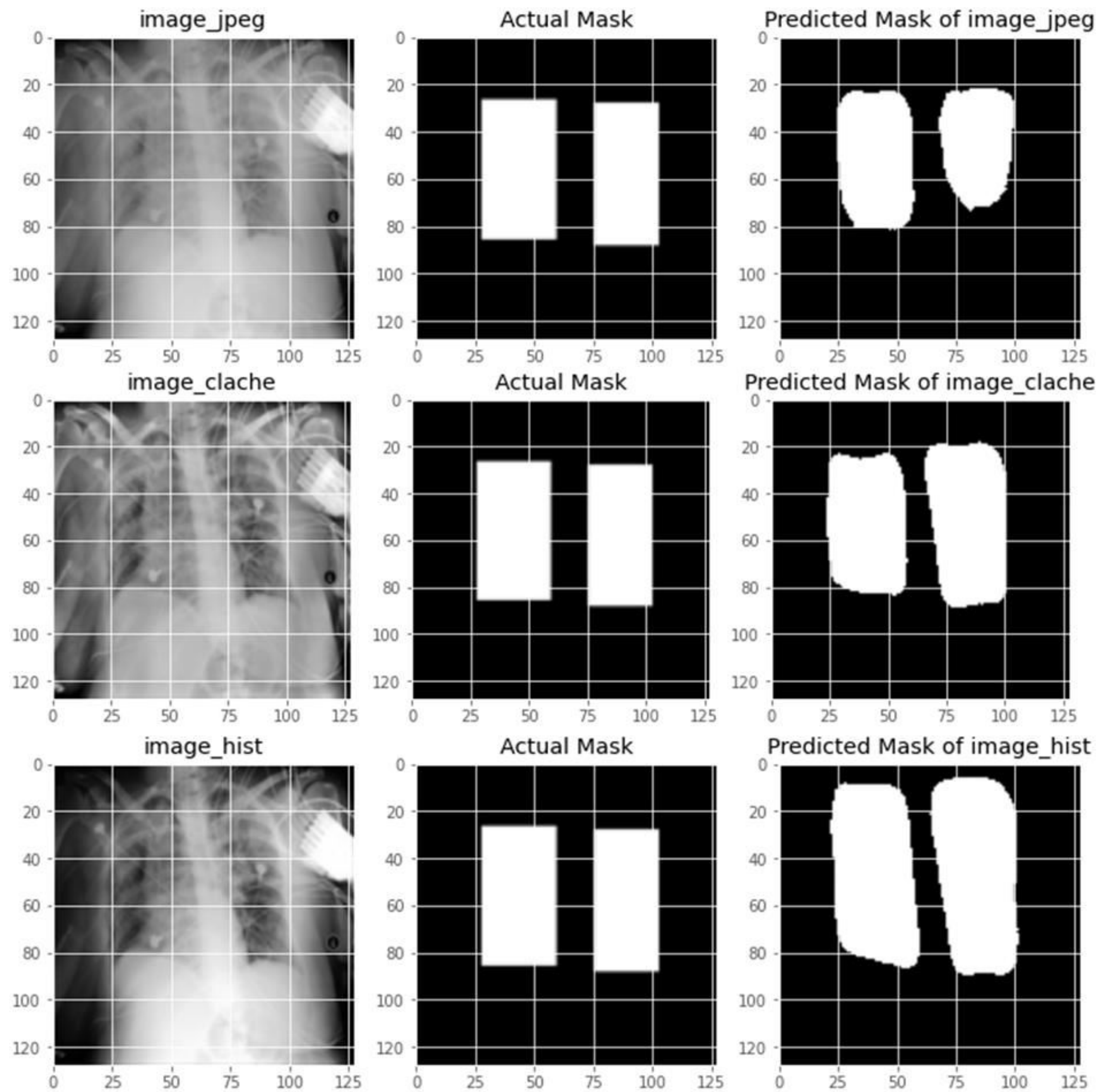
Image-Histogram



Model 4.1	Train Loss	Train IOU	Train F1	Val Loss	Val IOU	Val F1
No Normalization	0.2458	0.6871	0.7846	0.2644	0.6699	0.7681
CLAHE	0.2419	0.6888	0.7861	0.2857	0.6466	0.7490
Histogram	0.2613	0.6791	0.7773	0.3763	0.5632	0.6734

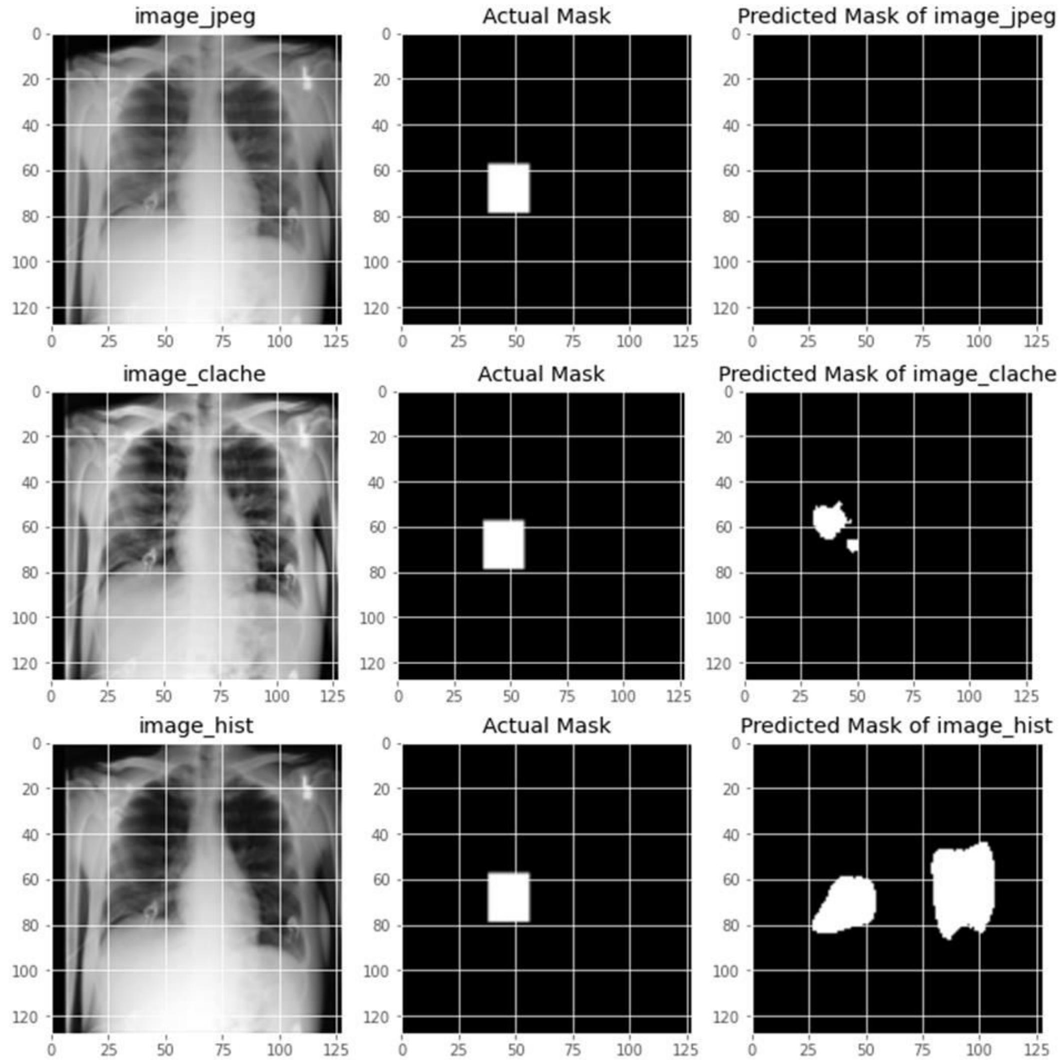
- No Over fitting observed between train and validation data for CLAHE and No Normalization
- Histogram normalization is not good which has overfitting tendency

- CLAHE normalization makes the training data loss slightly better. Need more tunning to reduce the fluctuations in Validation loss ,F1 and IOU with CLAHE Normalization



Prediction of masks is slightly better with CLAHE Normalization

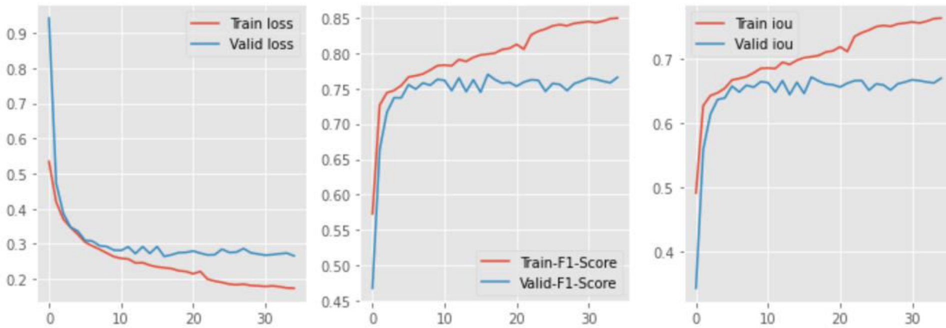
Histogram normalization has more False positives



Model 4.2 - 3layer CNNs for Blocks

- Filters -24, Image size – 224, Dropout -20%
- Trainable parameters – 70 lakhs
- Has tendency of overfitting

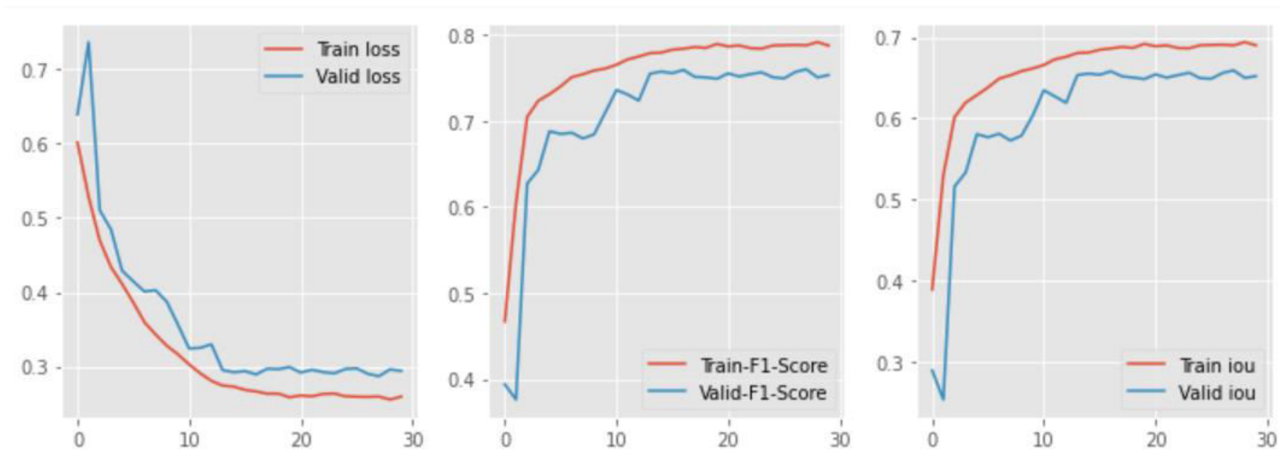
	Trainable Params	Train Loss	Train IOU	Train F1	Val Loss	Val IOU	Val F1
Model 4.1 -No Normalization	20Lakhs	0.2458	0.6871	0.7846	0.2644	0.6699	0.7681
Model4.2- No Normalization	70Lakhs	0.1739	0.7625	0.8496	0.2659	0.6692	0.7660



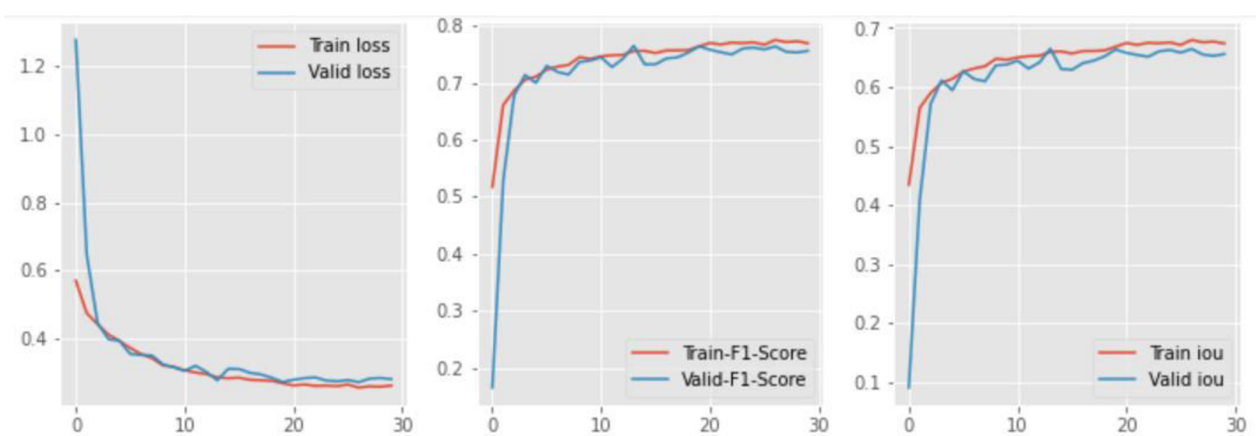
Model 4.3 - 3layer CNNs for Blocks with Image size 128x128,16 Filters,Dropout 30%

- With Image Augmentation the predictions imporved.
- No normalization and with CLAHE normalization is tested.

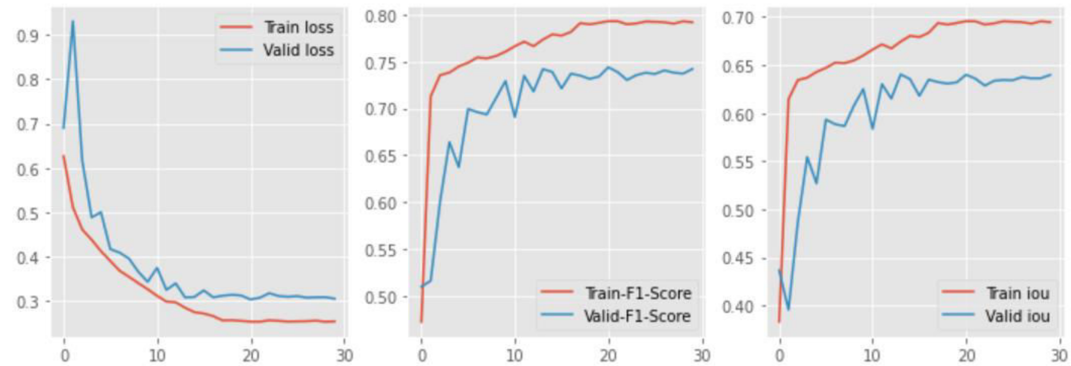
No Normalization



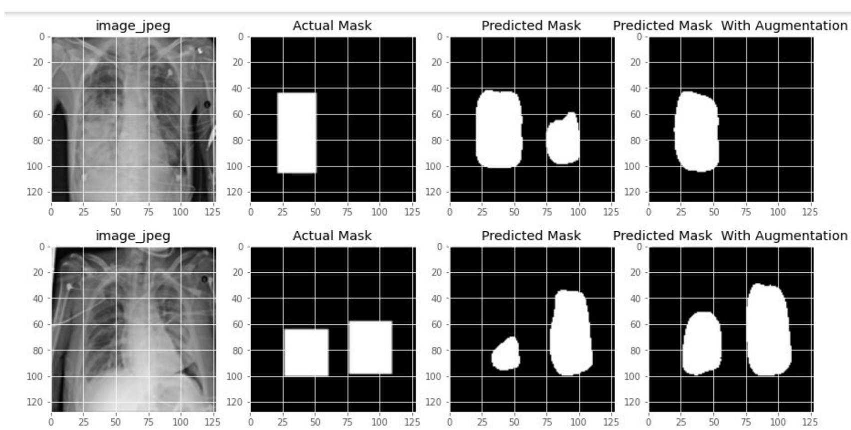
No Normalization – With Augmentation



CLAHE Normalization



	Trainable Params	Train Loss	Train IOU	Train F1	Val Loss	Val IOU	Val F1
Model 4.1 -No Normalization	20Lakhs	0.2458	0.6871	0.7846	0.2644	0.6699	0.7681
Model4.2- No Normalization	70Lakhs	0.1739	0.7625	0.8496	0.2659	0.6692	0.7660
Model4.3- No Normalization	31Lakhs	0.2593	0.6902	0.7880	0.2867	0.6592	0.7603
Model4.3- No Normalization Augmentation	31Lakhs	0.2544	0.6797	0.7749	0.2703	0.6644	0.7636
Model4.3- CLAHE	31Lakhs	0.2531	0.6955	0.7929	0.3030	0.6400	0.7439

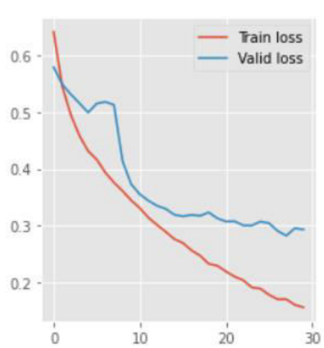


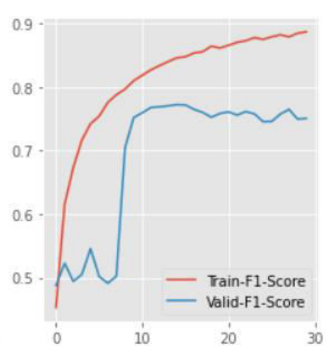
Model 4.4 – Encoder with ResNet34 Transfer model and imagenet Weights and Augmentation

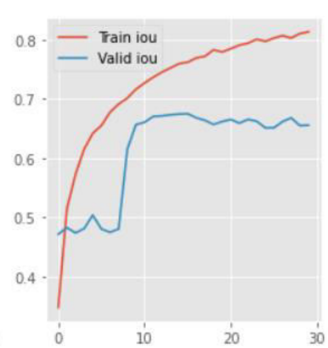
- With ResNet34 the Trainable parameters are high and causing overfitting
- Could not train with Entire train data so increased the training images from 30% to 60%
- Still with increased images for training and augmentation there is overfitting
- Need Augmentation of images study as currently alternate images were Augmented for each batch to improve the accuracy

	Trainable Params	Train Loss	Train IOU	Train F1	Val Loss	Val IOU	Val F1
Model 4.1 -No Normalization	20Lakhs	0.2458	0.6871	0.7846	0.2644	0.6699	0.7681
Model4.2- No Normalization	70Lakhs	0.1739	0.7625	0.8496	0.2659	0.6692	0.7660
Model4.3- No Normalization	31Lakhs	0.2593	0.6902	0.7880	0.2867	0.6592	0.7603
Model4.3- No Normalization Augmentation	31Lakhs	0.2544	0.6797	0.7749	0.2703	0.6644	0.7636
Model4.4-ResNet34-30% train data,Aug	240Lakhs	0.1706	0.8029	0.8793	0.2824	0.6679	0.7650
Model4.4-ResNet34-60% train data,Aug	240Lakhs	0.1822	0.7550	0.8330	0.2746	0.6616	0.7510

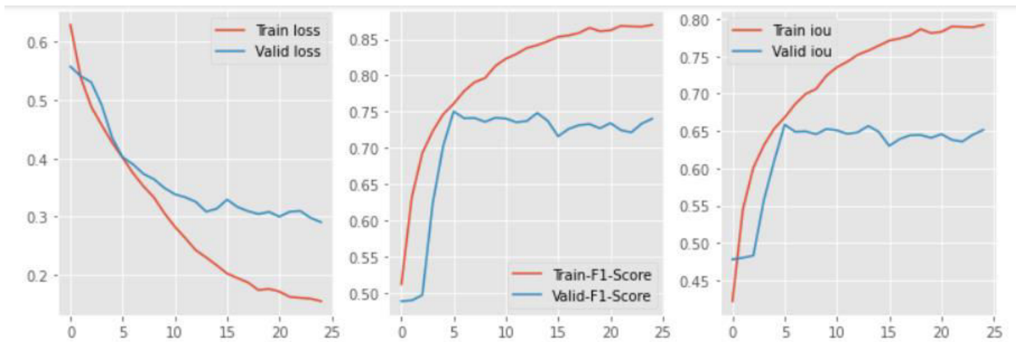
30% - Images with Non Pnuemonia – Overall Train images 6000images

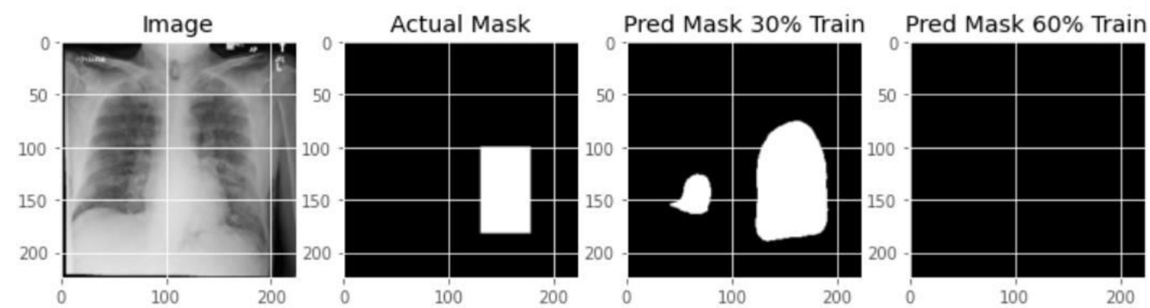






60% - Images with Non Pnuemonia – Overall Train images 12000images

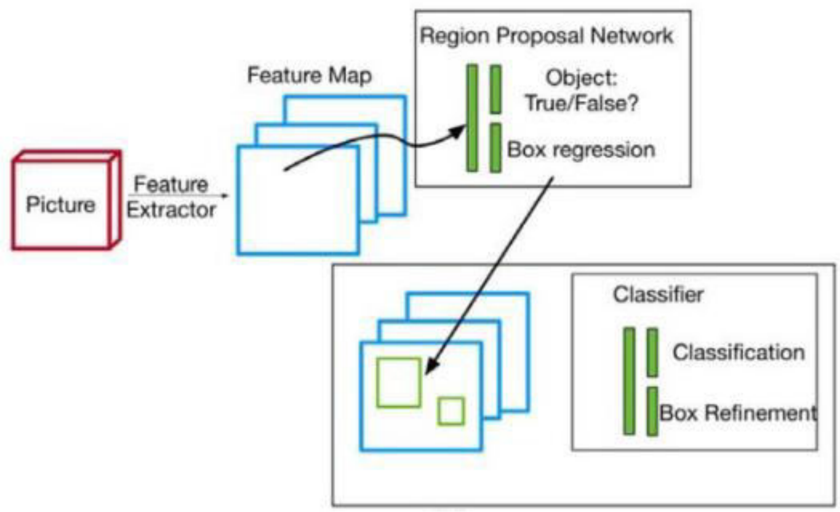




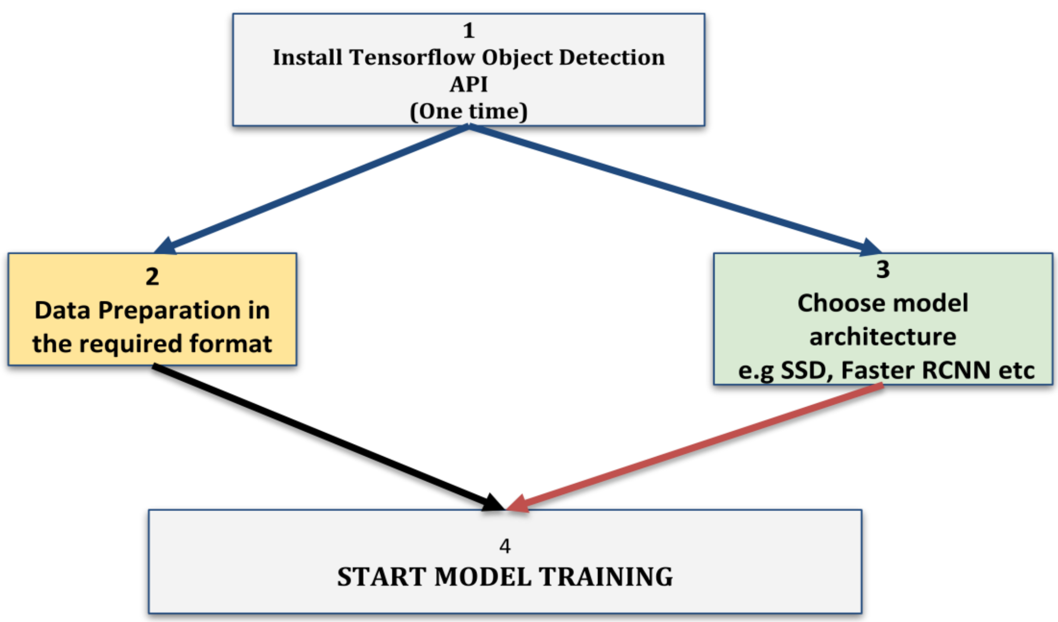
- Overall, the prediction of masks with standalone models worked well
- With Transfer learning models it would be better to release few last layers as trainable rather than taking the complete weights of imagenet for Encoder
- Current transfer model tends to overfit
- Training on the complete images was not possible due to lack of computational resources.
- CLAHE Normalization was able to predict the masks better than histogram
- Image augmentation is helpful to improve the mask prediction. Further tunning of augmentation parameters and batch size studies required to improve the accuracy
- As this model is only detect mask and not showing boundary boxes hence, we tried Object Detection using SSD Mobilenet and Tensorflow Object Detection API

6.4.5 Model5 Object Detection using SSD Mobilenet and Tensorflow Object Detection API

- SSD (Single Shot MultiBox Detector) is a popular algorithm in object detection. It's generally faster than Faster RCNN.



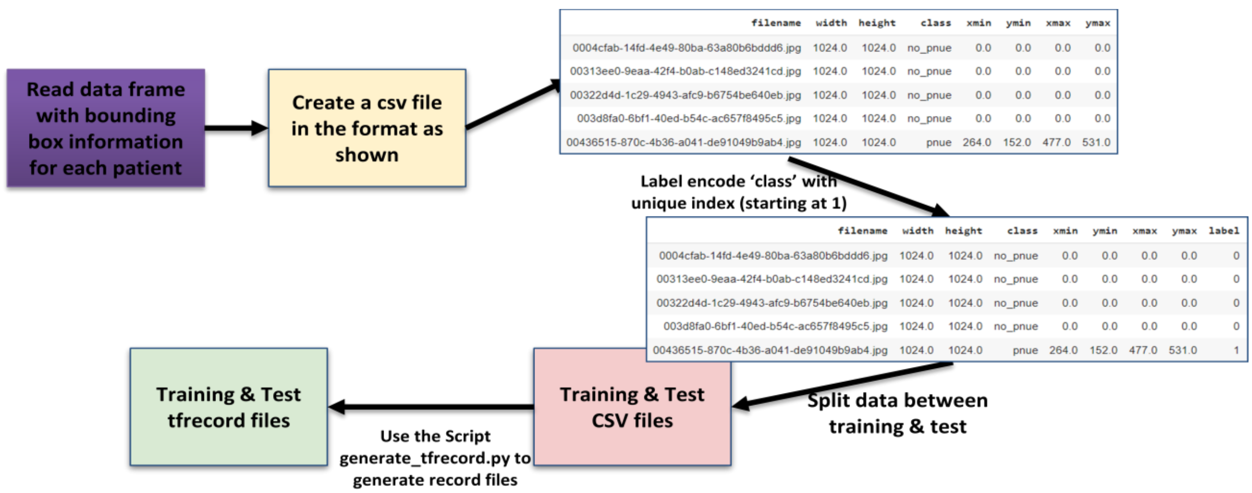
1. Tensor flow object Detection API- Procedure



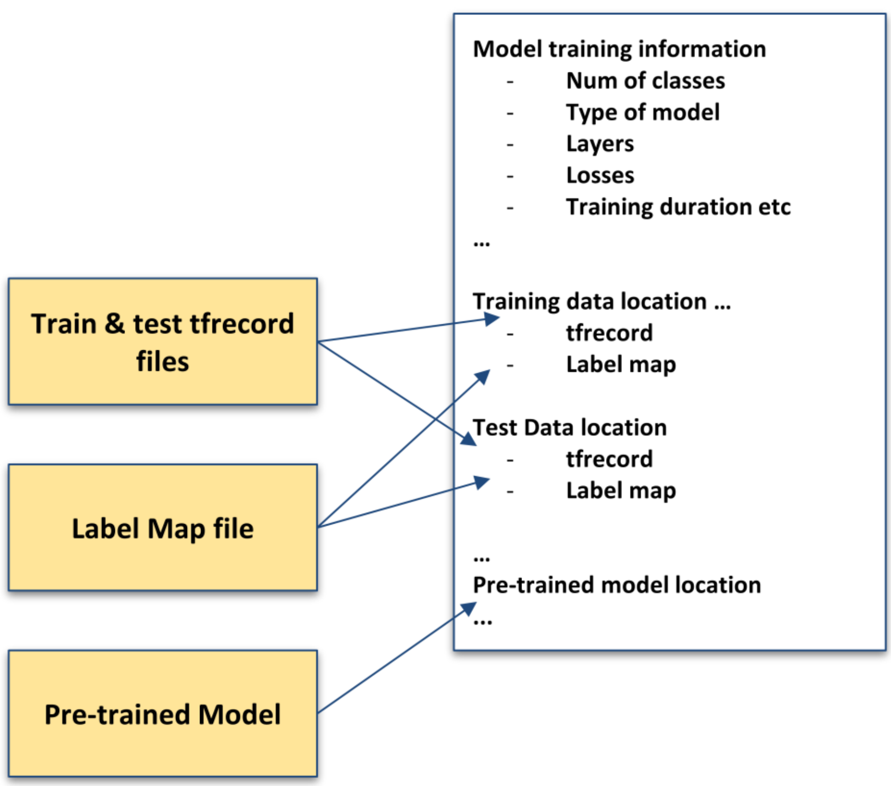
2. Data Preparation

- X ray image data to be converted to a dataframe with the 7 coulms as shown in the picture below
- Converted to csv file and each object has one record

- Image can have multiple records if the Xray image has multiple pnuemonia locations.
- Its important to change the Label for No Pnuemonia to 1 and with Pnuemonia to 2 class as 0 is reserved for background in API



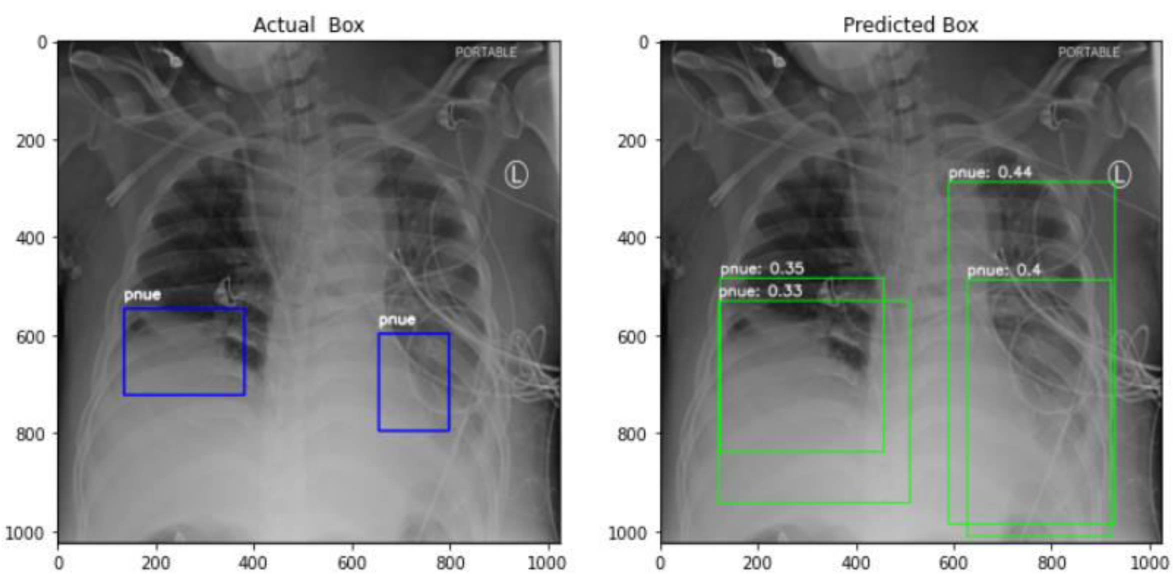
3. Minimum Changes to be done for config file

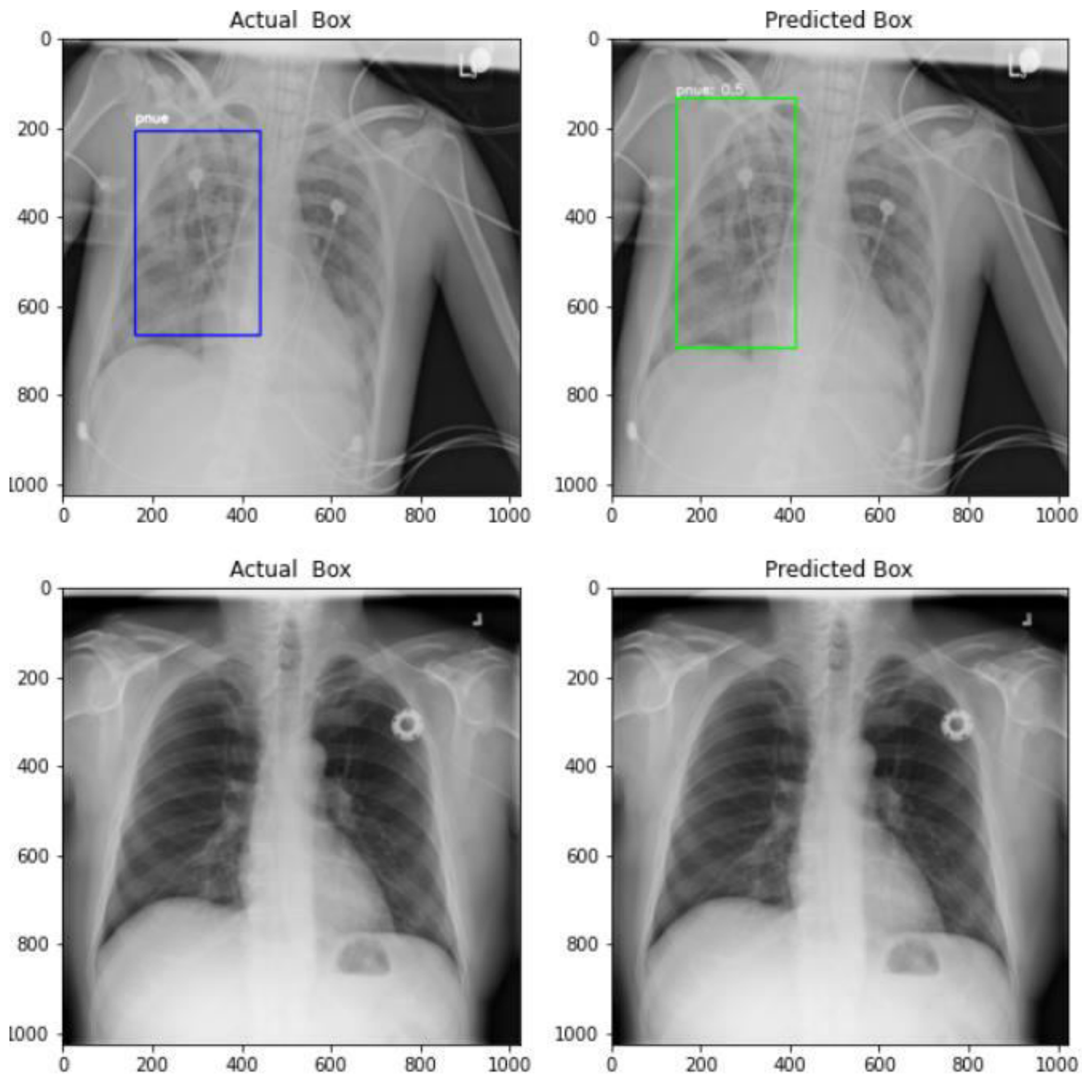


Minimum changes that are required for the config file are

1. Change num_classes parameter to number of classes in your dataset (2 for the current Xray image data)
2. For 'train_input_reader' change 'input_path' to filepath of train.record file.
3. For 'train_input_reader' change 'label_map_path' to filepath of pascal_voc.pbtxt file.
4. Repeat above two steps for 'eval_input_reader'.
5. Change fine_tune_checkpoint to filepath where pre-trained model.ckpt file is available e.g ssd_mobilenet_v1_coco_2018_01_28/model.ckpt
6. Change 'batch_size' accordingly to available memory.
7. Change 'num_steps' to indicate how long the training will done e.g. 200000. Removing this parameter means that you can train indefinitely.

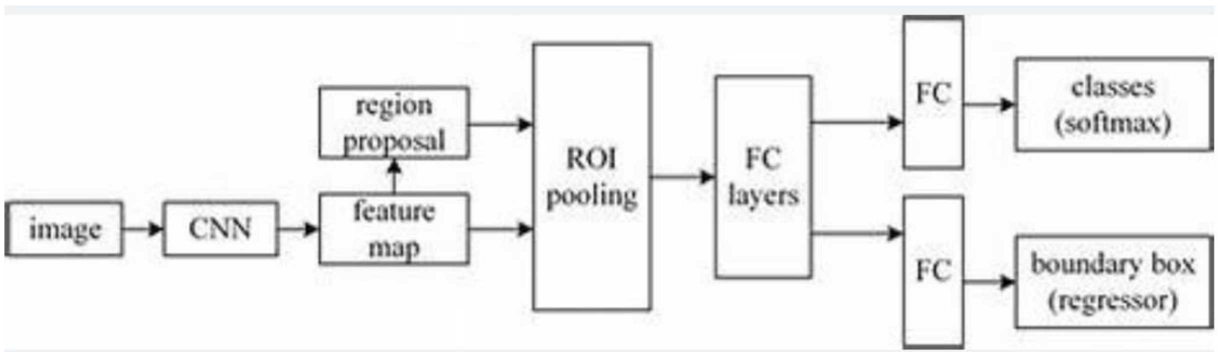
Checking Actual and Predicted Boxes for different Test data





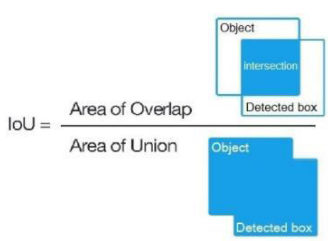
- To get better prediction tried Model 6 RCNN to predict BB and classification of Pneumonia.

6.4.6 Model6: RCNN model to predict BB and classification of Pneumonia




```
Epoch 1/2
100/100 [==============================] - ETA: 0s - loss: 0.5657 - accuracy: 0.9306 - mean_iou: 0.5904
Epoch 1: val_loss improved from inf to 0.52416, saving model to /content/drive/MyDrive/AIML/capstone/kaggle/cnn_best_model.h5
100/100 [==============================] - 5772s 58s/step - loss: 0.5657 - accuracy: 0.9306 - mean_iou: 0.5904 - val_loss: 0.52
42 - val_accuracy: 0.9246 - val_mean_iou: 0.4833 - lr: 0.0010
Epoch 2/2
100/100 [==============================] - ETA: 0s - loss: 0.5064 - accuracy: 0.9577 - mean_iou: 0.6173
Epoch 2: val_loss did not improve from 0.52416
100/100 [==============================] - 5759s 57s/step - loss: 0.5064 - accuracy: 0.9577 - mean_iou: 0.6173 - val_loss: 0.53
27 - val_accuracy: 0.9716 - val_mean_iou: 0.7278 - lr: 9.9606e-04
```

- RCNN Model(Used: Semantic segmentation)
- Training accuracy: 95%
- Validation accuracy 97%
- mean_iou: 0.6173
- val_mean_iou: 0.7278
- val_loss: 0.5327
- epochs:2

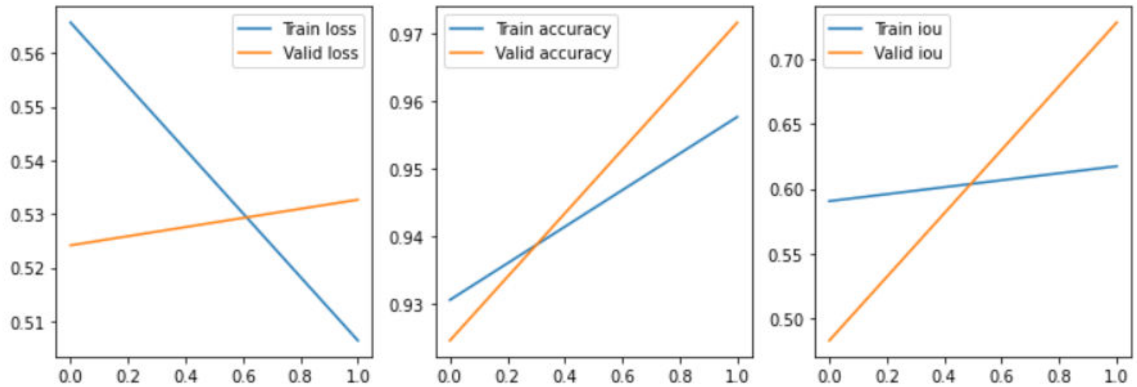


Evaluation function : mean IoU

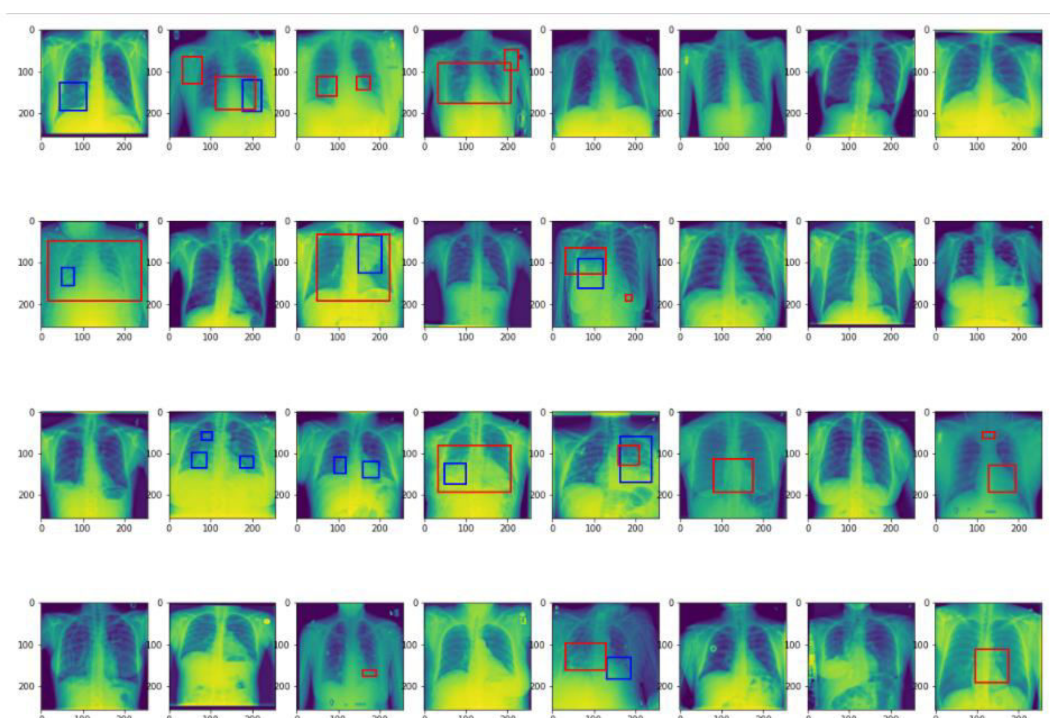
An Intersection over Union score > 0.5 is normally considered a “good” prediction which is in this case

training : 61%

testing: 72%



The blue bounding box represents the ground truth and the red bounding box is our model's estimated pneumonia prediction.



Submission.csv – first 5 rows.

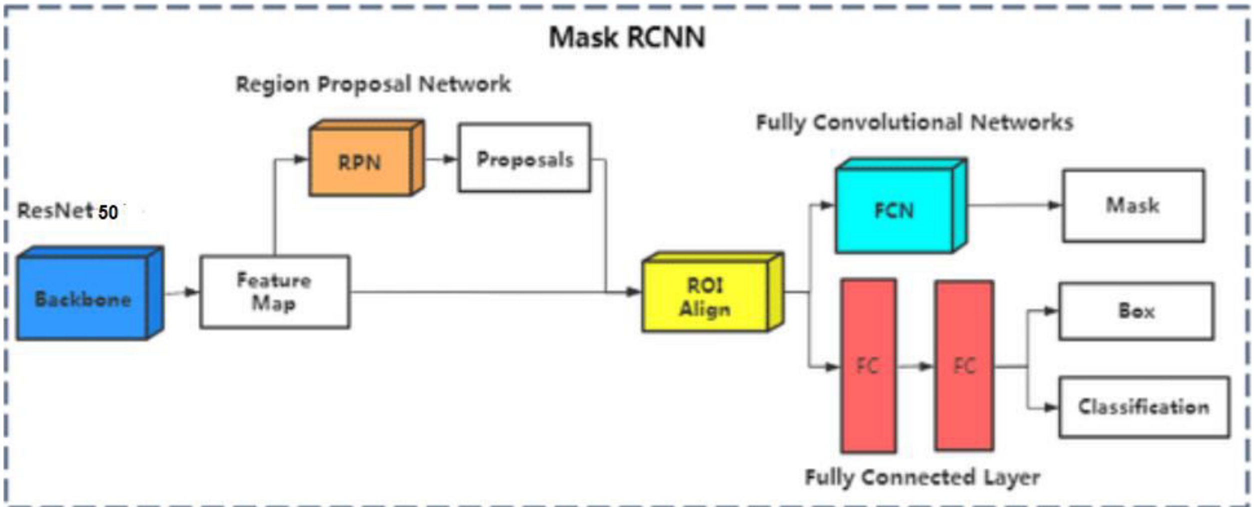
	patientId	PredictionString
0	2616069f-24c4-4d3d-8b76-c5564686eea0	NaN
1	260b87ff-df13-4177-90bb-53fb0618b1db	NaN
2	2616499c-8097-43cd-8864-575ba5d4d932	NaN
3	261c2f87-c4e3-461e-b6eb-e9a01dd3d4c1	0.5162151 191 193 640 510
4	261dcd8b-8a6e-4f8f-8ecb-a36460733b1f	0.6289014 127 321 832 448

Problems with R-CNN are:

- It still takes a huge amount of time to train the network as you would have to classify 2000 region proposals per image.
- It cannot be implemented real time as it takes around 47 seconds for each test image.
- The selective search algorithm is a fixed algorithm. Therefore, no learning is happening at that stage. This could lead to the generation of bad candidate region proposals.
- As these problems are there we tried mask RCNN.

6.4.7 Model7: Mask RCNN Model to Predict Mask and Boundary Boxes

Mask R-CNN was developed on top of Faster R-CNN, a Region-Based Convolutional Neural Network.



- **Mask RCNN:** Used Semantic segmentation.
- **Simplicity:** Mask R-CNN is simple to train.
- **Performance:** Mask R-CNN outperforms all existing, single-model entries on every task.
- **Efficiency:** The method is very efficient and adds only a small overhead to Faster R-CNN.
- **Flexibility:** Mask R-CNN is easy to generalize to other tasks.

- Applied Image Augmentation
- **epochs**:16

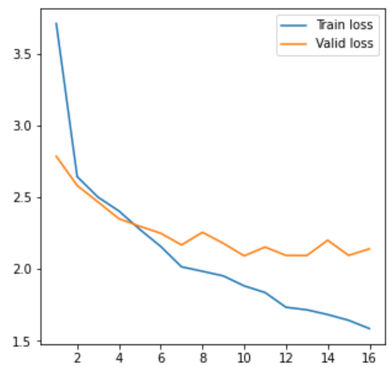
- **BACKBONE** = 'resnet50'
- **IMAGE_MIN_DIM** = 64
- **IMAGE_MAX_DIM** = 64
- **RPN_ANCHOR_SCALES** = (32, 64)
- **TRAIN_ROIS_PER_IMAGE** = 16
- **GPU_COUNT** = 1
- **IMAGES_PER_GPU** = 8
- **STEPS_PER_EPOCH** = 100
- **LEARNING_RATE** = 0.001

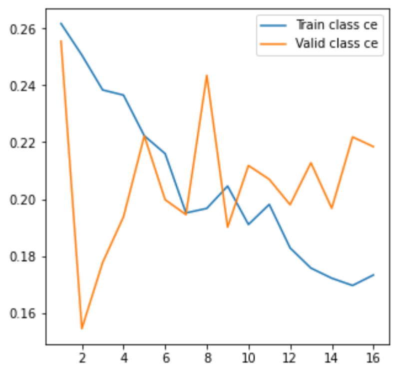
	val_loss	val_rpn_class_loss	val_rpn_bbox_loss	val_mrcnn_class_loss	val_mrcnn_bbox_loss	val_mrcnn_mask_loss	loss	rpn_class_loss	rpn_bbox_loss	mrcnn_class_loss	mrcnn_bbox_loss	mrcnn_mask_loss
1	2.860117	0.357921	1.214252	0.207142	0.507276	0.573508	3.587467	0.707568	1.245019	0.278966	0.700228	0.655669
2	2.544181	0.308102	1.050937	0.184823	0.486642	0.513659	2.646472	0.336230	0.958180	0.236162	0.547185	0.568696
3	2.552455	0.292944	1.030456	0.204827	0.484418	0.539792	2.409849	0.279096	0.855846	0.236043	0.495829	0.543018
4	2.429339	0.285549	0.930285	0.205433	0.498503	0.509550	2.302363	0.266309	0.808523	0.230176	0.470676	0.526661
5	2.472690	0.284762	1.033599	0.178531	0.474470	0.501310	2.211700	0.257779	0.759754	0.225847	0.459485	0.508817
6	2.358279	0.260244	0.926958	0.191243	0.490595	0.489221	2.162490	0.263890	0.732981	0.229128	0.446337	0.490137
7	2.392042	0.295519	0.949779	0.189822	0.508921	0.447984	2.039846	0.232759	0.687044	0.216500	0.427167	0.476358
8	2.347421	0.285171	0.964096	0.181759	0.480170	0.436208	2.006848	0.236777	0.689974	0.203599	0.411667	0.464814
9	2.262837	0.269756	0.891382	0.187312	0.466465	0.447904	1.951588	0.231023	0.666597	0.197916	0.398830	0.457205
10	2.184583	0.251970	0.877776	0.170768	0.458526	0.425526	1.899435	0.225155	0.649289	0.194447	0.380029	0.450498
11	2.209282	0.236848	0.888702	0.199611	0.453890	0.430213	1.857985	0.213817	0.627810	0.195000	0.375094	0.446246
12	2.206719	0.219807	0.863354	0.211238	0.506849	0.405454	1.785815	0.205720	0.589146	0.192801	0.357741	0.440390
13	2.444141	0.277956	0.951826	0.246281	0.516766	0.451293	1.762051	0.201825	0.591368	0.185090	0.347408	0.436342
14	2.268199	0.255068	0.934136	0.163034	0.489720	0.426224	1.731342	0.193466	0.576705	0.187373	0.340222	0.433558
15	2.212224	0.233249	0.890507	0.192421	0.486820	0.409209	1.675955	0.189281	0.550041	0.182634	0.327454	0.426527
16	2.288083	0.254146	0.960973	0.186357	0.466706	0.419883	1.684653	0.197899	0.548582	0.186817	0.321354	0.429984

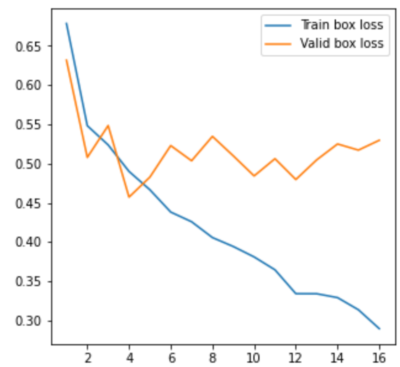
- **rpn_class_loss** = RPN anchor classifier loss
- **rpn_bbox_loss =** RPN bounding box loss graph
- **mrcnn_class_loss =** loss for the classifier head of Mask R-CNN
- **mrcnn_bbox_loss** = loss for Mask R-CNN bounding box refinement
- **mrcnn_mask_loss =** mask binary cross-entropy loss for the masks head
- Each of these loss metrics is the sum of all the loss values calculated individually for each of the regions of interest. The general **loss** metric given in the log is the sum of the other five losses (you can check it by summing them up) as defined by the Mask R-CNN's authors.
- The classification loss values are basically dependent on the confidence score of the true class; hence the **classification losses reflect** how confident the model is when predicting the class labels, or in other words, **how close the model is to predicting the correct class**. In the case of mrcnn_class_loss, all the object classes are covered, whereas in the case of rpn_class_loss the only classification that is done is labelling the anchor boxes as foreground or background

- **mrcnn_class_loss** is the classification loss, which tells how close the predictions are to the true class.
- **mrcnn_bbox_loss** bounding box loss, which tells how good the model is at localization.
- **mrcnn_mask_loss** loss for mask prediction, which is calculated by taking the binary cross-entropy between the predicted mask and the ground truth

We can improve performance by running more epochs.

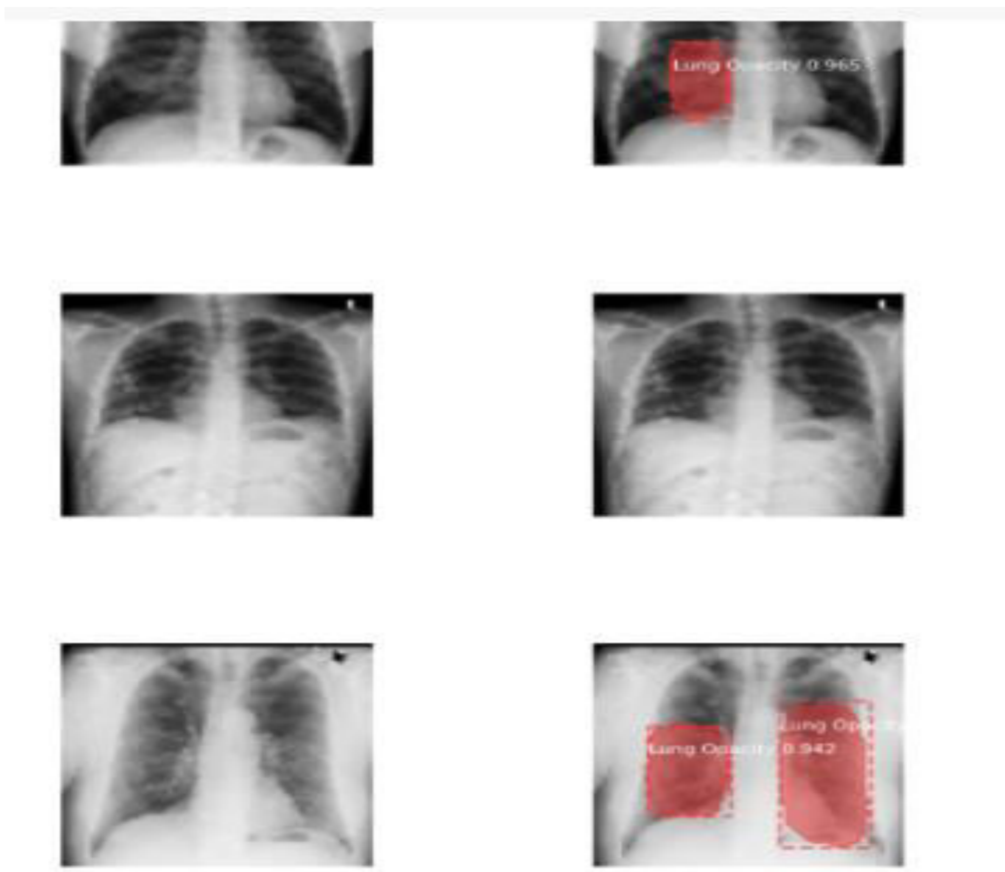


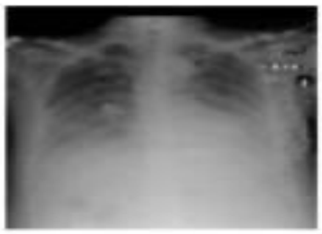




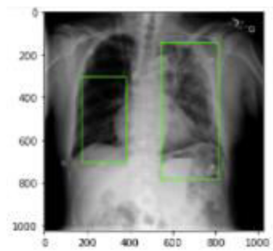
Observation

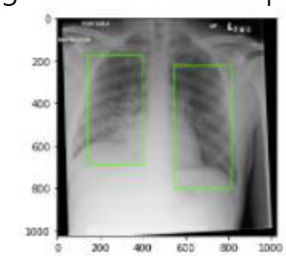
Difference between Total Training loss and Total Validation loss is very less.





Some of the Test images detection example





Out[40]:

	id	pred_string
0	15212951-d7a3-4332-8fc9-51ef843a6ade	NaN
1	1225df58-5bdb-4dfb-bf61-9eeae31ac7c0	NaN
2	03f24805-e3c3-4943-ba67-a9bcb43a2533	NaN
3	28e03bc2-1c2e-4c4d-ad70-1f0d6b74bc3d	NaN
4	2e4fecfe-a70e-4316-b964-db2750c9cb5d	NaN
5	23f6c434-cd8b-44a1-b8ce-621f6bb55e9a	NaN
6	1fec0554-a0e1-4328-99f7-28146ad2577a	NaN
7	26aff776-ff9d-40f2-b899-6b88f1355506	NaN
8	2de21def-1d19-4204-8189-6c4ce33a5aec	0.99 200 320 199 252
9	2139df25-6c09-4136-b987-4c2d7e71e200	0.99 161 355 186 168
10	c1507764-540b-4036-ae74-8271effd56c5	NaN
11	24845f34-a0db-4bd7-939b-417364bd0285	NaN
12	20563db8-4509-4d84-93e0-72d19c2ad6f8	NaN
13	11a2a4b6-0b05-4d40-a34b-5afbb2ca53d2	NaN
14	003206b4-bd4a-4684-8d49-76f4cb713a30	NaN
15	28e124e0-e71c-4444-ab34-3db1a5f7d1fc	NaN
16	0110a566-f774-4554-bdda-a1883ebd2f5c	NaN
17	05b1c573-7f35-41b1-887d-bb6e8cedf29d	NaN
18	0e863460-f754-4139-8ffe-8742cb4ba145	NaN
19	24b01fee-8258-4952-9b06-75a5e245f8e7	0.98 233 414 167 352
20	1018a3fa-be7c-4839-8cb4-a604a335f237	NaN
21	0428a988-106d-4ce8-a67b-2ee1827a825a	NaN
22	0e46f58b-4caa-4b3b-9306-3a68054bf193	NaN
23	1d3af918-bbac-4859-a6e0-4497599334dc	NaN
24	1e1ce192-3106-4922-a0d3-3d81b8eececf	NaN
25	2ea0e3dd-d671-42d0-b5db-0d73c79b4b82	NaN
26	3140ad27-ac31-4bf5-b74a-e2742ba630ef	0.98 267 449 149 326
27	274c940b-9c97-4aaa-a557-f8b97de6c268	NaN
28	2173269e-6264-44fb-a0f0-9278e7e42c8f	NaN
29	c1d9b4c9-39cc-4b37-b95b-82d26a1b68df	NaN
30	2c2502ab-8c4c-4142-b7b4-a1b36bf685a0	0.98 246 341 162 275

6.5 Model Evaluation

Model evaluation based on metrics.

ID	Model Name	Epoch	Accuracy	Recall	F1-Score	IOU_mean	Limitation
1	SVC	5	99%	1.0	1.0		Only possible on csv files. Can't Classify images.
2	Transfer Model Resnet50 and Inception with Imagenet weight	5	97%				This model is using images to detect Pneumonia but not area of interest in images.
3	Classification with heatmap	25	85%	0.81	0.84		Type I and II errors are large and also BB is not detected by this model.
4	Autoencoder Unet for predicting mask	20	94%		0.73	0.63	This model only predicting mask but not BB
5	Object Detection using SSD Mobilenet	10k Steps		0.518 (Classification loss)	1.1591 (Localization loss)	1.818 (Total loss)	To detect boundary boxes tried this model.
6	RCNN	2	97%			0.72	This model can do classification as well as detect BB, however it's taken more time to train the images
7	**Mask RCNN**	16		**0.9**	**0.66**		**This model can be used for classification as well as detection BB. Training and Testing loss is also very less.** **Fast to train** **Results are also good.** **Recall value is also 0.9**

Mask RCNN model can be used for classification as well as detection BB. Training and Testing loss is also very less. Fast to train. Results are also good.

6.6 Comparison to bench mark

How does your final solution compare to the benchmark you laid out at the outset? Did you improve on the benchmark? Why or why not?

We improved the F1 score of Bounding for the Pneumonia predicted class using Mask RCNN compared to the bench mark model of RCNN interms of detecting the Class and Bounding box.

7 Visualization

7.1 Quantifying your model and the solution

Model parameters

Learning Rate	0.001
Number of iterations	16
Batch size	8
Error functions	Loss_mask, loss_rpn_cls, loss_box_reg, loss_rpn_loc
Final Loss_mask	0.4

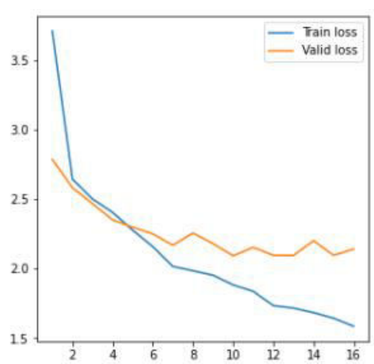
Final Box Loss	0.2
Final Class Loss	0.1
BACKBONE	resnet50

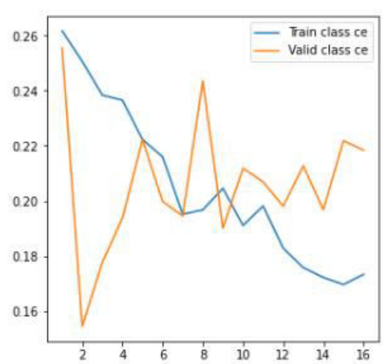
	id	pred_string
0	003d17f0-bd8a-485c-bc8b-daec33f53efa	0.99 223 336 197 228
1	24e408ee-ffab-4bc2-b82e-356d87e2b20d	NaN
2	2badcdd6-4733-47ae-bdfc-412bb853e9a1	NaN
3	0000a175-0e68-4ca4-b1af-167204a7e0bc	0.99 210 339 218 382
4	30e2f3c5-3315-46cf-ac57-b1ad8bb32f9a	NaN
5	19dbd4e3-db2f-4b33-b3db-96a3d993e5f9	NaN
6	2a731e8d-4b28-46ec-b22d-01cbcbf67d12	0.99 188 523 213 207
7	252841f2-7c5e-4554-8181-0c651a885d41	NaN
8	1d5d0cf0-d3e3-4919-af0d-2ee56f31e0e0	NaN
9	0d5b9d41-4d7b-4539-8f63-0bcbb56d3d59	NaN
10	2bd3afa7-414c-431f-ad98-0d1b03f175b8	NaN
11	2d76b5ea-17e8-49a9-a342-9daf21a912bb	NaN
12	043d020e-c61b-40ac-8704-be5ed7ded293	NaN
13	2e1220d7-1c72-4afa-881e-5380d356bca8	0.99 284 224 326 560
14	14f7d431-f66f-4098-a9e3-b94d84e9a5cf	NaN
15	04ba9dc3-9cf1-41f5-9e4d-939bd69c97f6	NaN
16	2b47029a-b0e9-4d5a-807a-62c8e2d36855	NaN
17	2634cbd2-0a3f-4503-bc07-b8f4f2da80c4	NaN
18	1d3e1e8b-f6d0-4289-b5e1-c9e992ee1225	0.98 219 186 223 548
19	2f612e6c-474d-474a-875d-f227a78911cc	NaN
20	2abcf934-facd-40fc-a75e-0062441fd206	1.0 188 331 221 426
21	012469b7-1b4b-4407-8c1b-7236861d49a3	NaN
22	03ae3399-3ca3-4e4e-bf4e-5e5d57dc20e3	NaN
23	26586d01-b992-41c4-a7a4-c9f1f8d102ad	0.99 156 329 260 444
24	037f16ff-46b6-45bf-b737-74072c46fab7	NaN
25	23ea49ef-6f5e-4842-83b5-8514e2179fa7	0.98 246 485 189 220
26	0d2e6cd3-a646-4cad-942b-59fcf226fc27	0.99 570 339 226 575
27	2215b907-f15e-4b26-ba95-e9eebcaef2b6	NaN
28	1d10afa7-26c2-43ed-8a9e-931c2fc79522	NaN
29	2b2d704b-cb66-4bcc-afb0-4eff225d620d	NaN
30	28dc9ac8-e269-4834-8483-aeddd22cb878	NaN

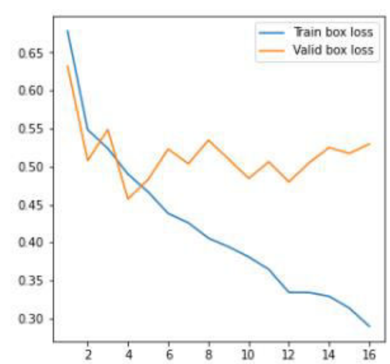
Results based on training and testing datsets.

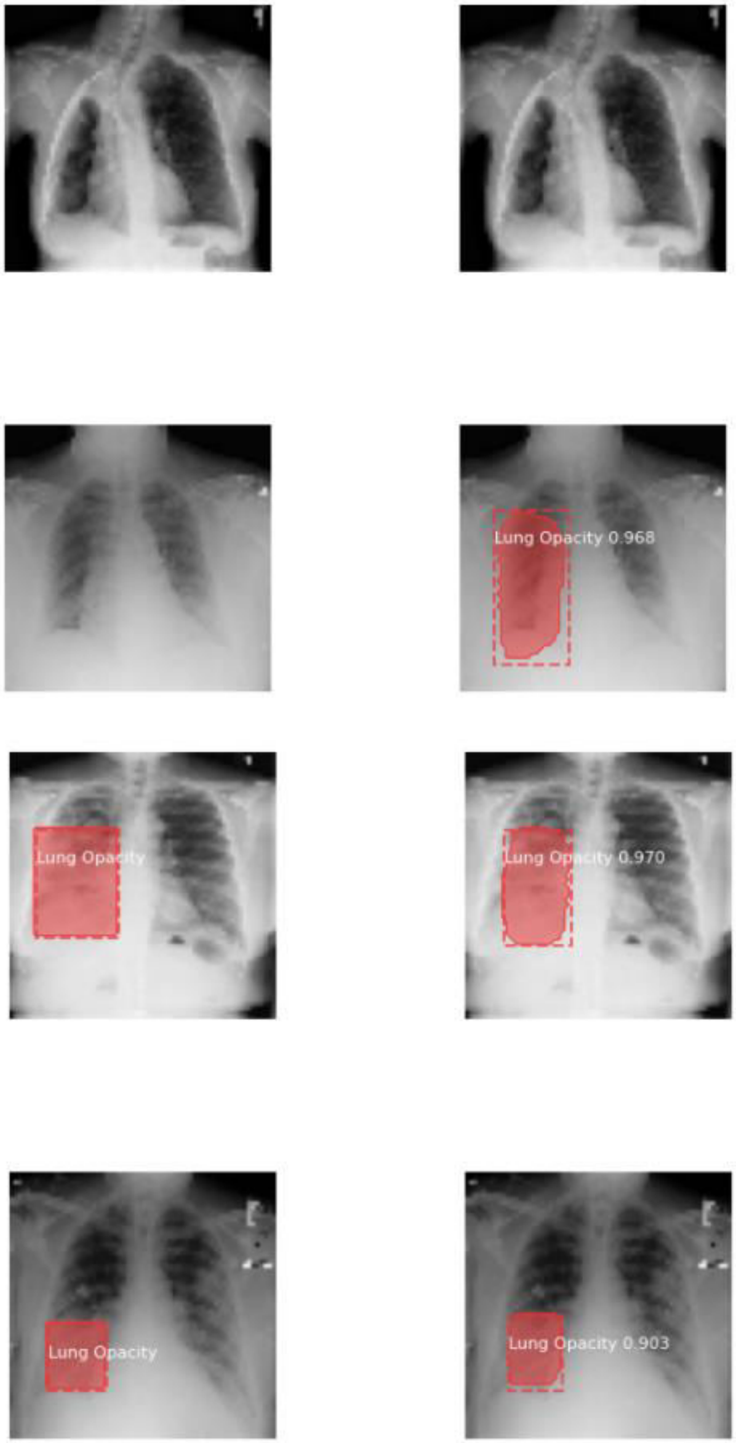
	val_loss	val_rpn_class_loss	val_rpn_bbox_loss	val_mrcnn_class_loss	val_mrcnn_bbox_loss	val_mrcnn_mask_loss	loss	rpn_class_loss	rpn_bbox_loss	mrcnn_class_loss	mrcnn_bbox_loss	mrcnn_mask_loss
1	2.785107	0.309759	1.027207	0.255521	0.631850	0.560753	3.709759	0.835436	1.303811	0.261722	0.678563	0.630210
2	2.581876	0.329296	1.037763	0.154440	0.507830	0.552529	2.643545	0.343727	0.941584	0.250639	0.548251	0.559327
3	2.467058	0.291885	0.898004	0.177690	0.548468	0.550995	2.500657	0.318917	0.881542	0.238371	0.523755	0.538054
4	2.350571	0.300205	0.893559	0.193741	0.457209	0.505839	2.404286	0.292260	0.866617	0.236552	0.489966	0.518873
5	2.297774	0.281064	0.817162	0.222201	0.482674	0.494655	2.276189	0.285134	0.798577	0.222249	0.466606	0.503606
6	2.249526	0.273171	0.782419	0.199778	0.522918	0.471223	2.157872	0.272745	0.741663	0.215961	0.438140	0.489346
7	2.167954	0.237506	0.763439	0.194524	0.503565	0.468903	2.015646	0.235040	0.688119	0.195120	0.425827	0.471522
8	2.255899	0.267769	0.747862	0.243509	0.534641	0.462100	1.984579	0.243069	0.673377	0.196718	0.405626	0.465771
9	2.179340	0.272185	0.757293	0.190113	0.509845	0.449888	1.952339	0.242052	0.651920	0.204560	0.394231	0.459559
10	2.091784	0.244334	0.719994	0.211797	0.484233	0.431409	1.882829	0.219453	0.641652	0.191056	0.380926	0.449725
11	2.153412	0.262608	0.733881	0.206932	0.506200	0.443775	1.836536	0.218905	0.611662	0.198122	0.364486	0.443344
12	2.095072	0.257813	0.734037	0.198021	0.479653	0.425531	1.733941	0.208539	0.569554	0.182822	0.334076	0.438932
13	2.094219	0.211695	0.735021	0.212742	0.504718	0.430027	1.716114	0.196738	0.577601	0.175726	0.333995	0.432036
14	2.200802	0.256697	0.769378	0.196754	0.524974	0.452982	1.683120	0.194090	0.561787	0.172175	0.329044	0.426007
15	2.095247	0.219014	0.708367	0.221833	0.517148	0.428868	1.643274	0.191378	0.545271	0.169611	0.313565	0.423431
16	2.140937	0.234504	0.736303	0.218435	0.529689	0.421989	1.585219	0.186310	0.515450	0.173298	0.289506	0.420638

7.2 Visualizations that support the ideas/insights that we gleaned from the data








Lung Opacity 0.968
Lung Opacity
Lung Opacity 0.970
Lung Opacity
Lung Opacity 0.903

8 Implications

How does your solution affect the problem in the domain or business? What recommendations would you make, and with what level of confidence?

Mask RCNN model can be used for classification as well as detection BB. Training and Testing loss is also very less. Fast to train. Results are also good. Confidence level is 0.96.

9 Limitations

What are the limitations of your solution? Where does your model fall short in the real world? What can you do to enhance the solution?

- While running models for more Epochs faced RAM memory issues or colab disconnection issues.
- Required more memory while running models. So, we will get better results.
- To enhance solution, we will train and evaluate Mask R-CNN with different backbone structures (ResNet101 and ResNet152) along with hyperparameter tuning. Run model for more Epochs.

10 Closing Reflection

What have you learned from the process? What you do differently next time?

- When to use Supervised learning in real time scenarios.
- How to use Convolution Neural Network
- EDA
- How to detect Pneumonia using Deep learning
- What all evaluation metrics used in while using different models.
- Different types of Normalization and importance of it while doing Pneumonia detection.
- Data pre-processing steps
- How to use Tensorboard as a visualization tool. Can be used to visualize the graphs.
- Hyperparameter turning

11 References

- http://www.southsudanmedicaljournal.com/archive/2008-05/how-to-read-a-chest-x-ray-a-step-by-step-approach.html
- https://www.medicinenet.com/chest_x-ray/article.htm
- https://youtu.be/rEm2HkI5UDs
- https://youtu.be/HBS-DJ9ez-c
- https://analyticsindiamag.com/my-experiment-with-unet-building-an-image-segmentation-model/
- https://neptune.ai/blog/how-to-train-your-own-object-detector-using-tensorflow-object-detection-api
- https://github.com/sauravmishra1710/U-Net---Biomedical-Image-Segmentation/blob/main/UNet%20-%20Biomedical_Segmentation.ipynb
- https://github.com/AarohiSingla/Faster-R-CNN/blob/main/classifier.ipynb
- https://www.youtube.com/watch?v=4fNNQuHKh1o
- https://www.youtube.com/watch?v=1u-dm5JMH1Q
- https://www.youtube.com/watch?v=iHf2xHQ2VYo
- https://www.youtube.com/watch?v=cReOzRvlLVA
- https://www.youtube.com/watch?v=if1tzf1p0gA
- https://github.com/AarohiSingla/Faster-R-CNN-on-Custom-Dataset
- https://github.com/AarohiSingla/Faster-R-CNN/blob/main/rpn_layer.ipynb