FINAL REPORT SUBMISSION

AIML Capstone Project - CV1

Pneumonia Detection Challenge

-

KAARTHIG DHEENA
KAVYA PARANGI
KHYATI SHARMA
K ROHAN VARMA
SANTOSH ALGHARI
TULIKA SHEKHAR

24TH July 2022

Table of Contents

SUMMARY OF PROBLEM STATEMENT& DATA

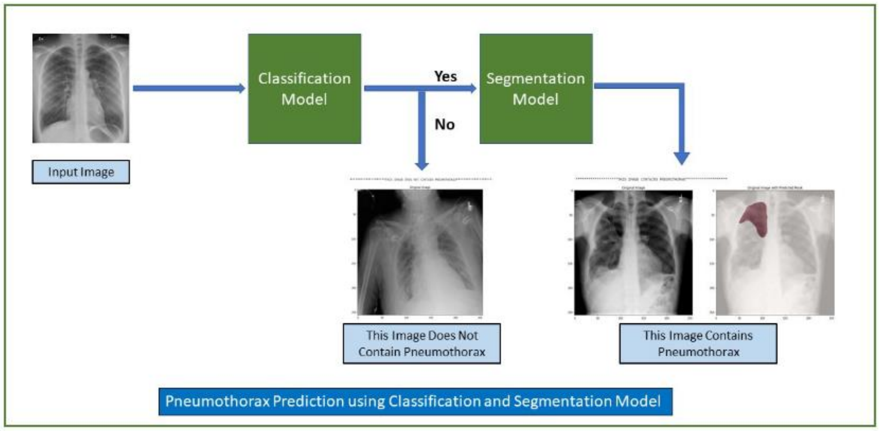
In this capstone project, the goal is to build a pneumonia detection system, to locate the position of inflammation in an image.

Talking about this in terms of AI/ML, this is an Object identification and localization problem.

The CLASSIFICATION part of the problem tells us if Pneumonia patch(es) are detected in the lungs
The LOCALIZATION part of the problem tells us the exact position of inflammation in the lungs, through bounding boxes, or using semantic segmentation.

Hence, we are building an algorithm to detect a visual signal for pneumonia in medical images. The algorithm should be able to:
Predict correctly, the presence or absence of Pneumonia
Correctly locate the position of lung opacities on chest radiographs

The below is the flow of our solution


Input Image
Classification Model
Yes
No
Segmentation Model
This Image Does Not Contain Pneumothorax
This Image Contains Pneumothorax
Pneumothorax Prediction using Classification and Segmentation Model

INTRODUCTION

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. 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 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.

What are dicom images?
DICOM (Digital Imaging and Communications in Medicine) is a standard protocol for the management and transmission of medical images and related data and is used in many healthcare facilities. It is the international standard to communicate and manage medical images and data. Its mission is to ensure the interoperability of systems used to produce, store, share, display, send, query, process, retrieve and print medical images, as well as to manage related workflows.
Imaging information systems, in compliance with DICOM(*.dcm), have largely eliminated the need for film-based images and the physical storage of these items. Instead, these days, medical images, as well as related non-image data, can be securely stored digitally, whether on premises or in the cloud.
They have a meta file of attributes which can store all the important information related to the image, making data analysis and model building much more efficient

How do dicom images detect abnormalities?
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 labeled "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.

DATA& DATA ATTRIBUTES

DICOM images: We are provided with 26684 train images, of *.dcm type
We are provided with the corresponding labels, for each image, telling us whether Pneumonia is detected in those images, and if yes, what are the bounding box co-ordinates.
We also get another level of detail for the Target class – a sub-class which tells us about the actual lung condition – if it is completely normal or if it has Pneumonia or some other issues.
We will discuss these attributes in detail in the below sections.

ATTRIBUTE DETAILS

DICOM image attributes

Pixel Data - The order of pixels encoded for each image plane is left to right, top to bottom, i.e., the upper left pixel (labeled 1,1) is encoded first. For us it is an array of 169122 elements
Photometric Interpretation - The value of Photometric Interpretation specifies the intended interpretation of the image pixel data. The permitted values are MONOCHROME1, MONOCHROME2, RGB, CMYK etc.
Samples per pixel - For monochrome (gray scale) and palette color images, the number of planes is 1. For RGB and other three vector color models, the value of this Attribute is 3. Hence for us, this value is '1'
Bits Stored — Each pixel here stores 8 bits
Pixel Representation - Data representation of the pixel samples. Each sample shall have the same pixel representation. For us, the value is '0' which stands for unsigned integer.
Lossy Image Compression - Lossy compression means that the image size is reduced while some data from the original image file is eliminated. In DICOM images, it specifies whether an Image has undergone lossy compression (at a point in its lifetime), or is derived from lossy compressed images. "0" implies that image has NOT been subjected to lossy compression while "1" implies that Image has been subjected to lossy compression. For us, it's value is "01"
Lossy Image Compression Method – What lossy image compression method has been applied is specified here. We have applied JPEG Lossy Compression ('ISO_10918_1')
Modality - Type of equipment that originally acquired the data used to create the images in this Series. For our dataset, it is "CR" – Computed Radiography
Patient ID– Patient ID
Patient Name – Patient Name
Patient Age – Age of the Patient
Patient Sex – Patient Gender
PatientBirthdate – DOB of the Patient
Body Part Examined – For us it is populated as "Chest" as these are all chest X-rays
View Position – Radiographic view of the image relative to the imaging subject's orientation. We have 2 view position -
AP - erect anteroposterior chest view is an alternative to the PA view when the patient is too unwell to tolerate standing or leaving the bed
PA - Chest Posterior Anterior

Number of rows- number of rows in the dicom image
Number of columns – number of columns in the dicom image
All images in our sample of data are square images where their size is 1024*1024

Following is a sample meta file for one of the train images:

```
Dataset.file_meta -------------------------------
(0002, 0000) File Meta Information Group Length UL: 200
(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:
1.2.276.0.7230010.3.1.4.8323329.8203.1517874336.95544
(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: 1.2.276.0.7230010.3.1.4.8323329.8203.1517874336.95544
(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: 'fffec09e-8a4a-48b1-b33e-ab4890ccd136'
(0010, 0020) Patient ID                         LO: 'fffec09e-8a4a-48b1-b33e-ab4890ccd136'
(0010, 0030) Patient's Birth Date               DA: ''
(0010, 0040) Patient's Sex                      CS: 'M'
(0010, 1010) Patient's Age                      AS: '45'
(0018, 0015) Body Part Examined                 CS: 'CHEST'
(0018, 5101) View Position                      CS: 'AP'
(0020, 000d) Study Instance UID                 UI: 1.2.276.0.7230010.3.1.2.8323329.8203.1517874336.95543
(0020, 000e) Series Instance UID                UI: 1.2.276.0.7230010.3.1.3.8323329.8203.1517874336.95542
(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: [0.139, 0.139]
(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 169122 elements
```

Target Class

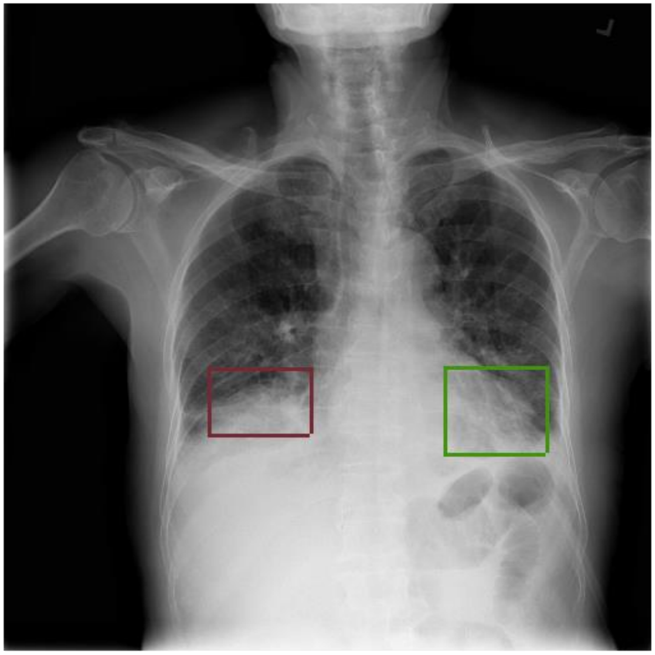
'Target' - "1" signifies Pneumonia detected while "0" signifies No Pneumonia detected – THIS IS THE CLASSIFICATION PART OF THE PROBLEM
If Pneumonia detected, then we are also provided the co-ordinates of the bounding box – THIS IS THE LOCALIZATION PART OF THE PROBLEM. We get x,y, width and height of the bounding box
If No pneumonia is detected, then the bounding box fields have blanks.

Target Class Subtypes

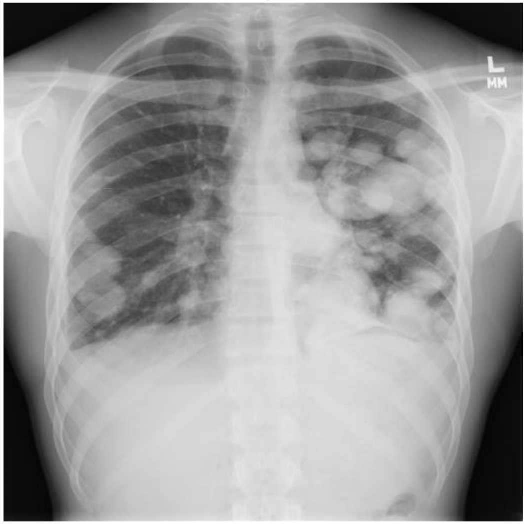
The Target sub-class can have one of the following 3 values:

Lung Opacity - Lung opacities are vague, fuzzy clouds of white in the darkness of the lungs, which makes detecting them a real challenge.
Opacity in general refers to any area that preferentially attenuates the x-ray beam and therefore, appears opaquer than the surrounding area.
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.

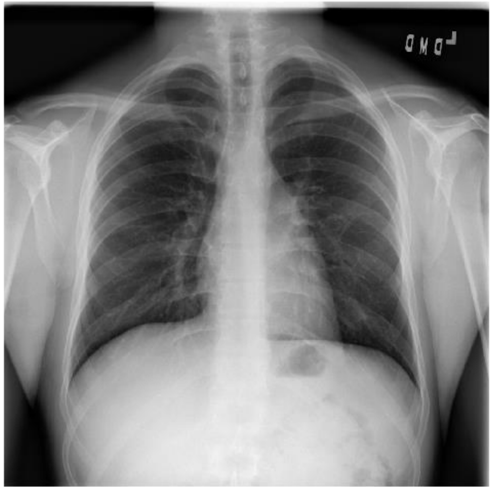
Sample Patient 2 - Lung Opacity



No lung opacity / Not Normal – No lung opacity is seen. However, there are other opacities which are there in the chest, but not related to pneumonia. Some examples here could be Nodules & Masses which could be an indication of Cancer.


Sample Patient 3 - Lung Nodules and Masses

Normal – This is when no opacities or abnormalities are found in the lungs.


Normal Image

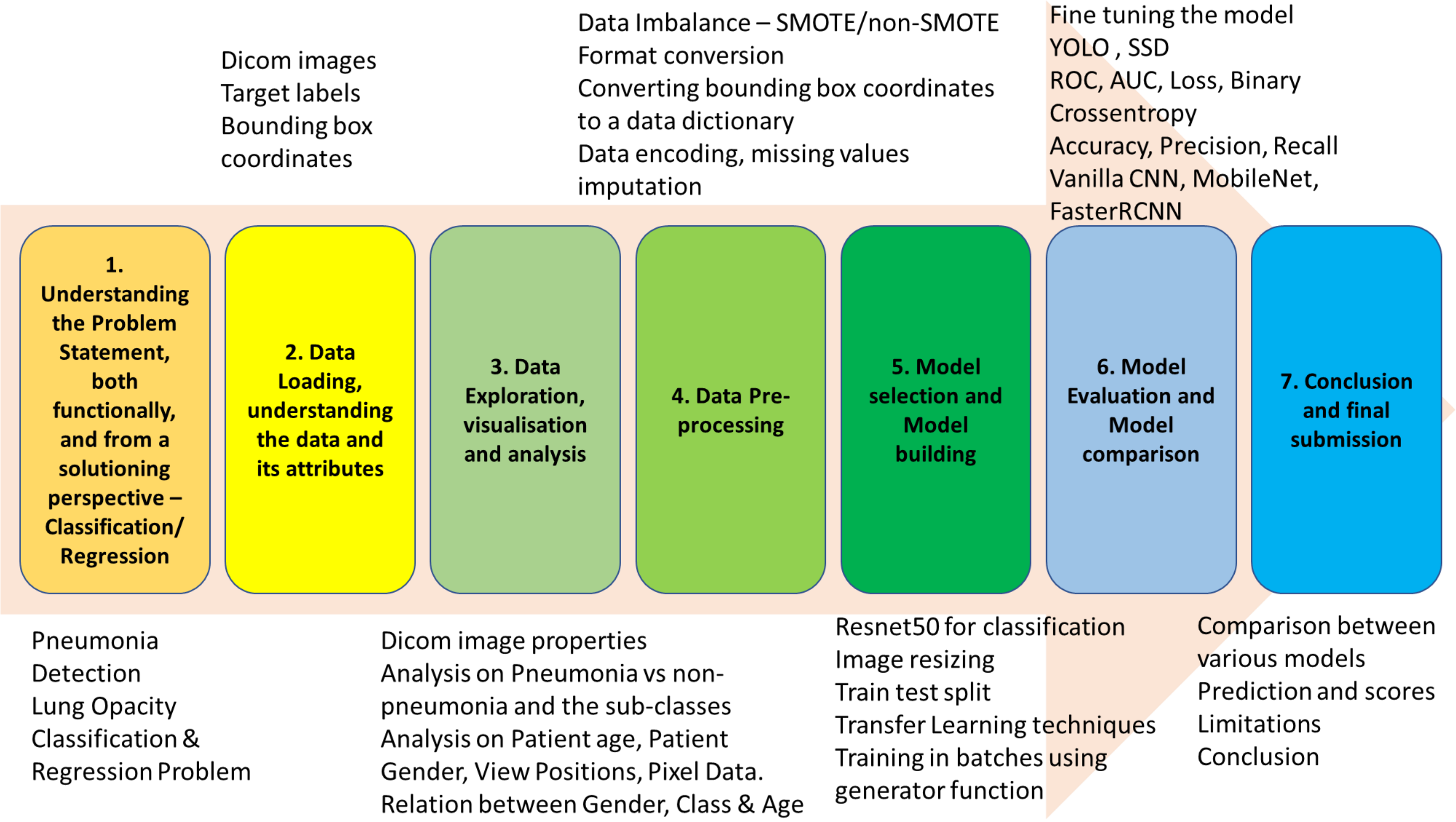
We have been provided with the following files:

sample_train_images folder – contains dicom(.dcm) images which will serve as our training input data - # of images??
stage_2_detailed_class_info.csv – contains the patient ID and information about the lung condition, apart from only detecting the presence of Pneumonia
stage_2_test_images folder – contains dicom images which we can use later to predict the lung opacities. We don’t have any target labels provided for these; hence we will not be using them during model training
stage_2_train_lab-els_csv – This file gives us the label or y values for the train images

In order to view the X-ray images, we need to install pydicom library module to access these images.

```
!pip install pydicom
```

PROCESS FLOW



1. Understanding the Problem Statement, both functionally, and from a solutioning perspective – Classification/ Regression

Pneumonia Detection
Lung Opacity
Classification &
Regression Problem

2. Data Loading, understanding the data and its attributes

Dicom images
Target labels
Bounding box coordinates

3. Data Exploration, visualisation and analysis

Dicom image properties
Analysis on Pneumonia vs non-pneumonia and the sub-classes
Analysis on Patient age, Patient Gender, View Positions, Pixel Data.
Relation between Gender, Class & Age

4. Data Pre-processing

Data Imbalance – SMOTE/non-SMOTE
Format conversion
Converting bounding box coordinates to a data dictionary
Data encoding, missing values imputation

5. Model selection and Model building

Resnet50 for classification
Image resizing
Train test split
Transfer Learning techniques
Training in batches using generator function

6. Model Evaluation and Model comparison

Fine tuning the model
YOLO , SSD
ROC, AUC, Loss, Binary Crossentropy
Accuracy, Precision, Recall
Vanilla CNN, MobileNet, FasterRCNN

7. Conclusion and final submission

Comparison between various models
Prediction and scores
Limitations
Conclusion

SUMMARY OF THE APPROACH TO EDA AND PRE-PROCESSING

EXPLORATORY DATA ANALYSIS

Following are some of the findings from the exploratory data analysis:

ANALYSIS ON DATA SIZE, TARGET AND SUB-CLASSES

In total, we have the data of **26684** unique patients (Patient IDs), with no opacity, or multiple opacities.

An important point to note here is that in the above file, 30227 rows do not denote 30227 patients, because for a single patient, multiple opacities could exist. There is one row per opacity detected.

There are 3543 duplicate records, they could have different bounding box coordinates b ut are duplicate for a patient ID

The dicom image has 5 key value pairs in the meta-file (which has its attributes)

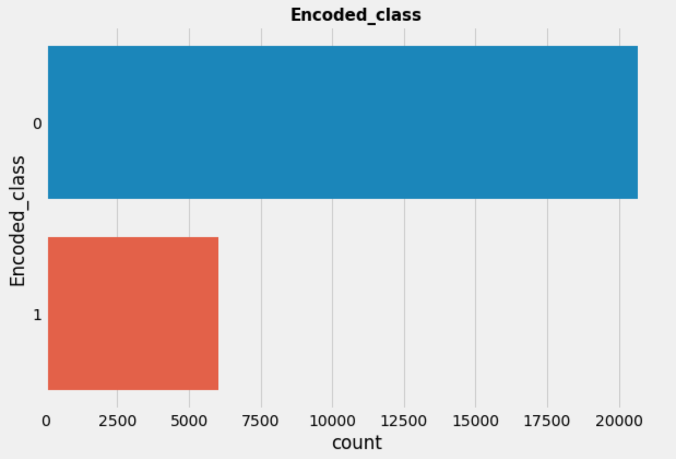
The dicom image pixel array is of shape 1024x1024

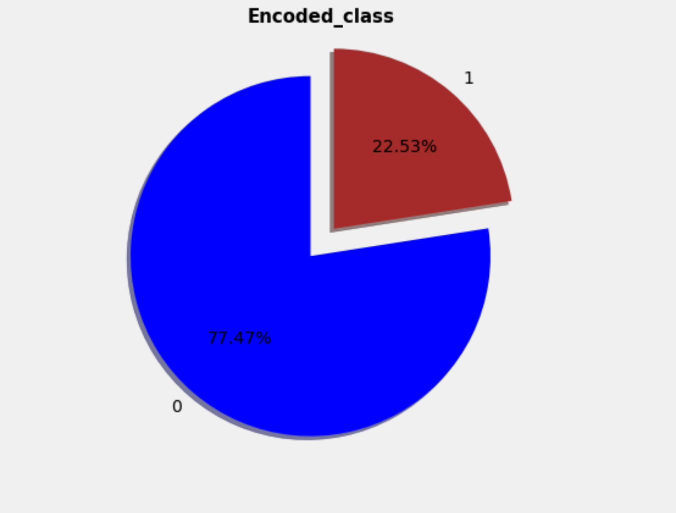
The minimum and maximum value contained in a pixel is 0 and 255 respectively

Out of the total 26684 patients:

77.47% do not have pneumonia

22.53% have pneumonia



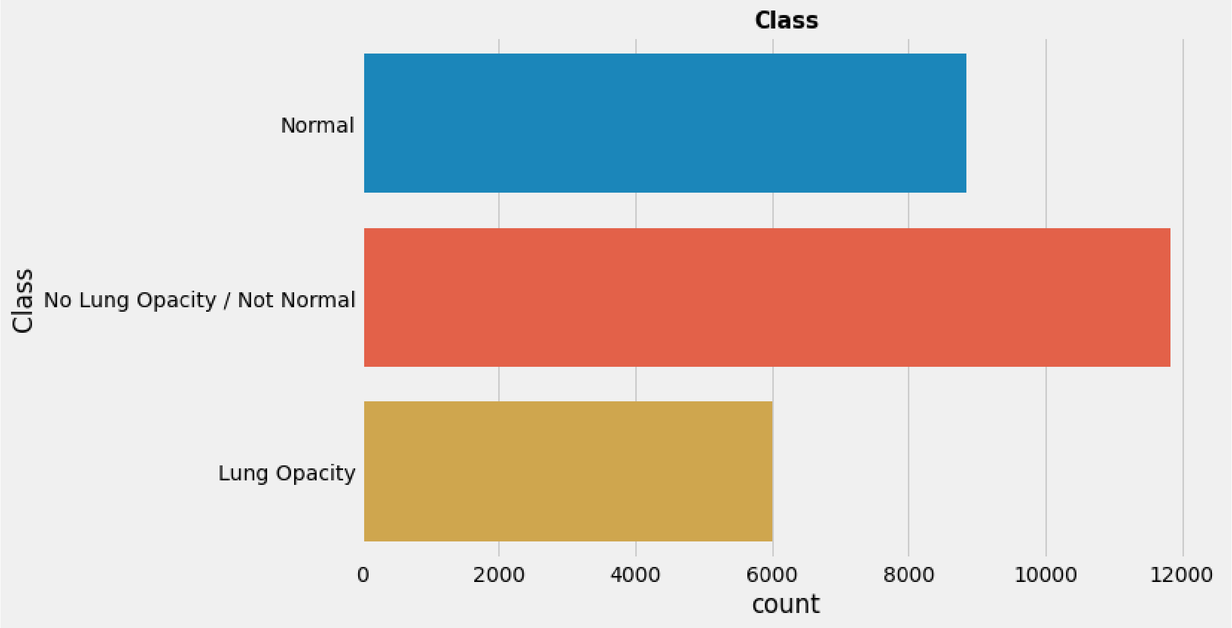


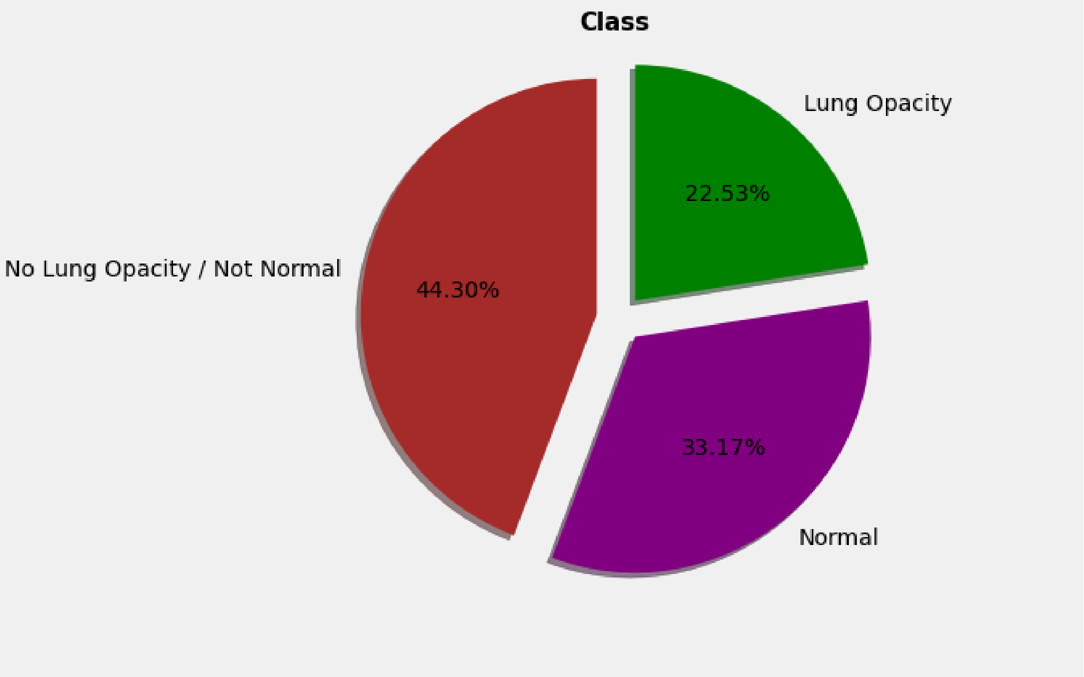
-

In terms of the Target sub-class:

33.17% belong to ‘Normal’ Category

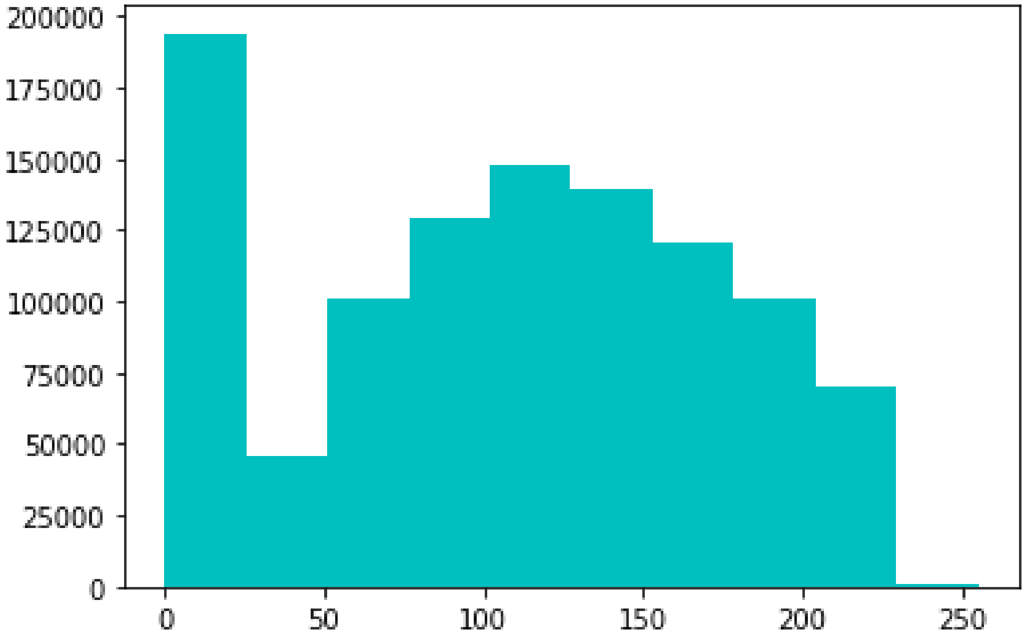
22.53% show ‘Lung Opacity’ which means there are Pneumonia patches detected
44.3%don’t show Pneumonia but show some other abnormality in the lungs





Visualizing pixel intensity distributions on a histogram:

When we plot the pixel values of a sample image on a histogram, we get the following:



What we can infer from the above graph is that these images are really dense, and the size of these images is really huge. So, it is going to be very difficult to load all of them at one time and to build a Neural Network out of it.

Hence, we might need to do 2 things:

Resize the images to make them smaller

Load the images in batches, instead of loading them all at once.

ANALYSIS ON VIEW POSITION AND VISUALIZING A FEW SAMPLE IMAGES:

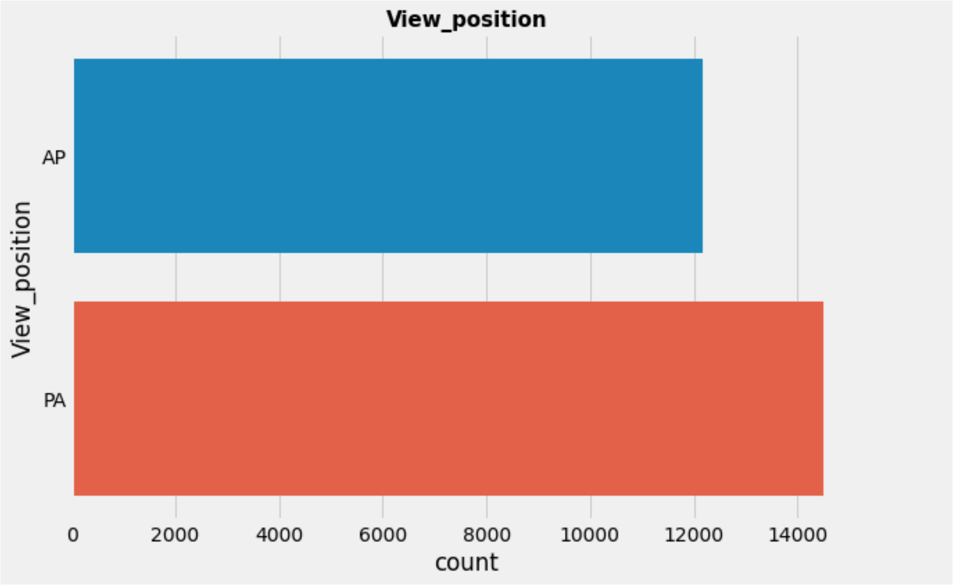
We have 2 View Positions in our sample date:

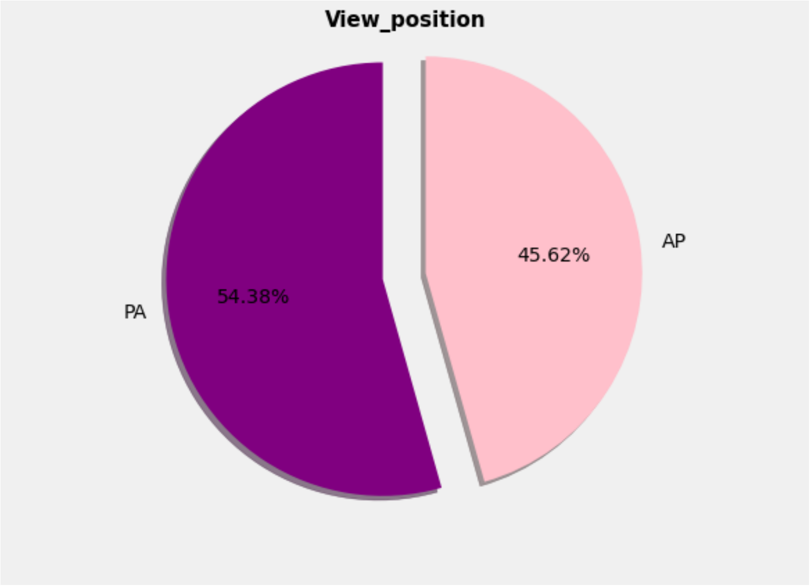
AP - erect anteroposterior chest view is an alternative to the PA view when the patient is too unwell to tolerate standing or leaving the bed

PA - Chest Posterior Anterior

If we plot the View positions against our dataset, we can see:

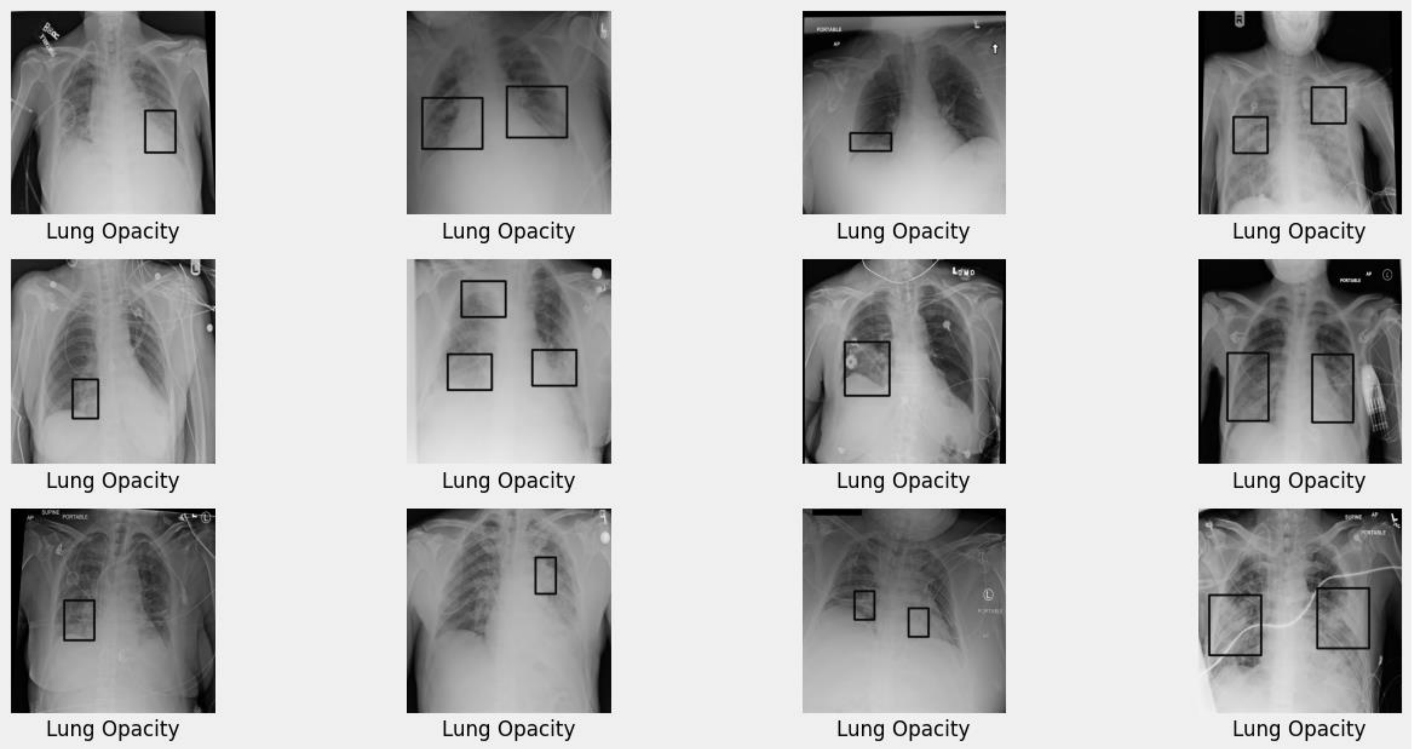
45.62% of the sample images have been taken in the anteroposterior chest view position
54.38% of the sample images have been taken in the Chest Posterior Anterior view position



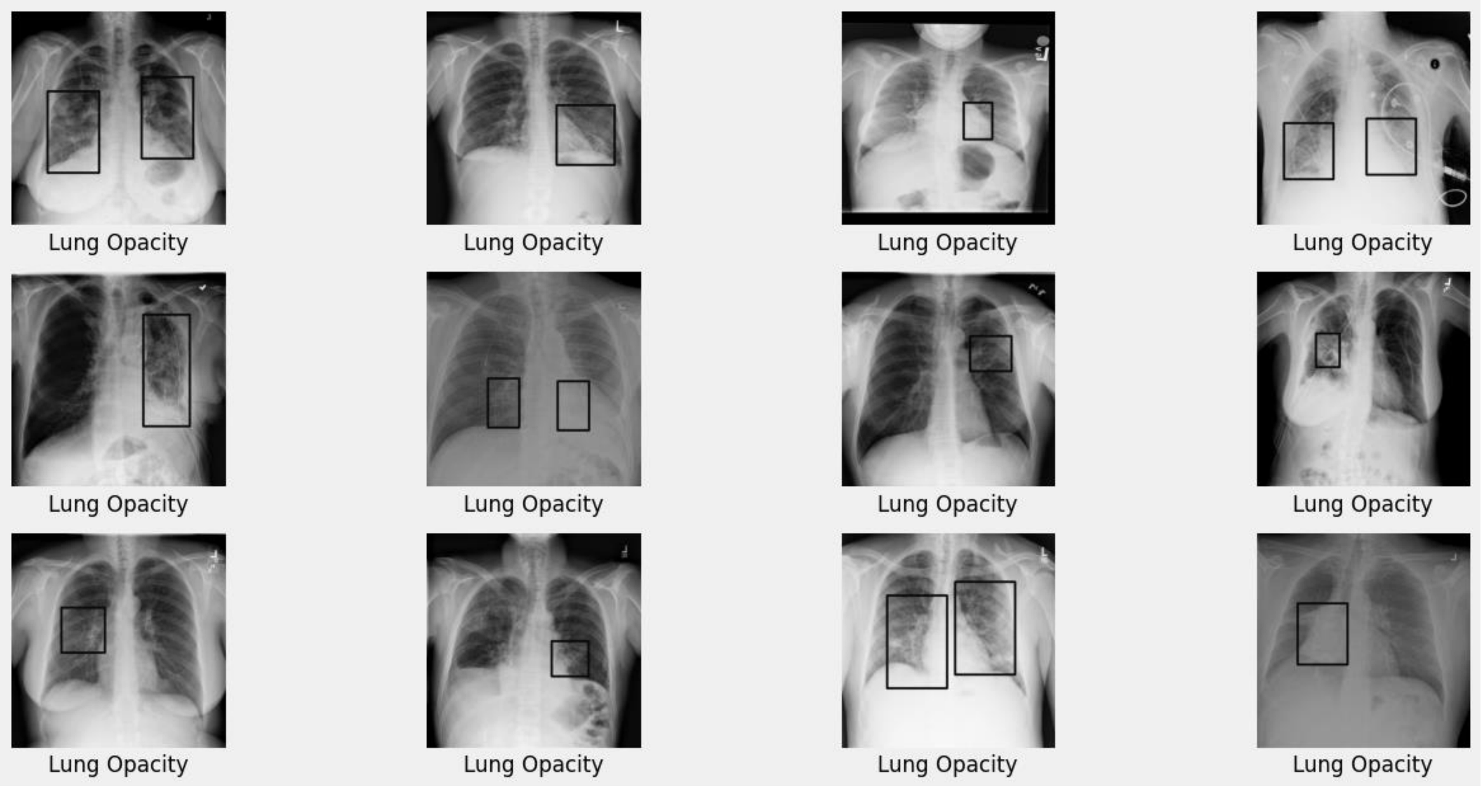


Viewing some sample images from our dataset where lung opacity is seen:

View Position = “AP”

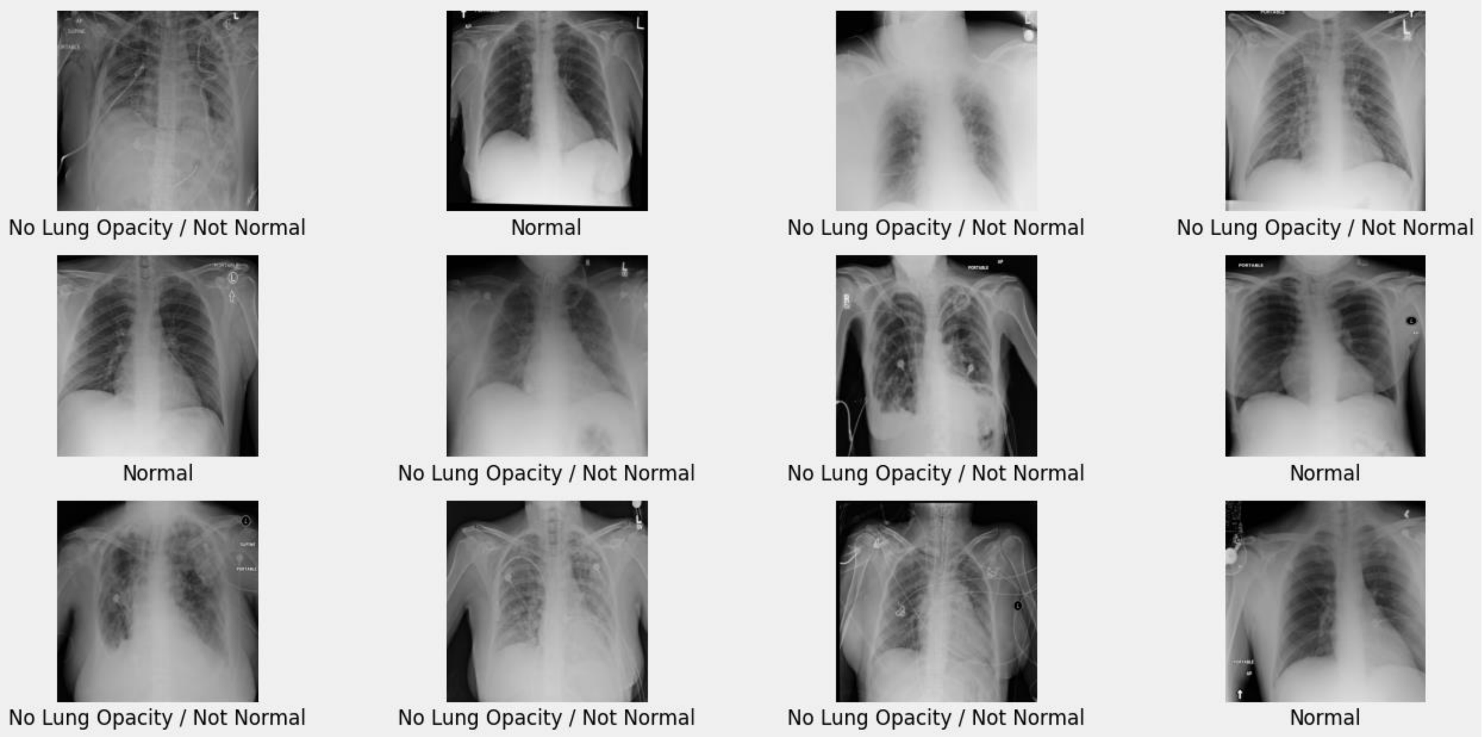


View Position = “PA”

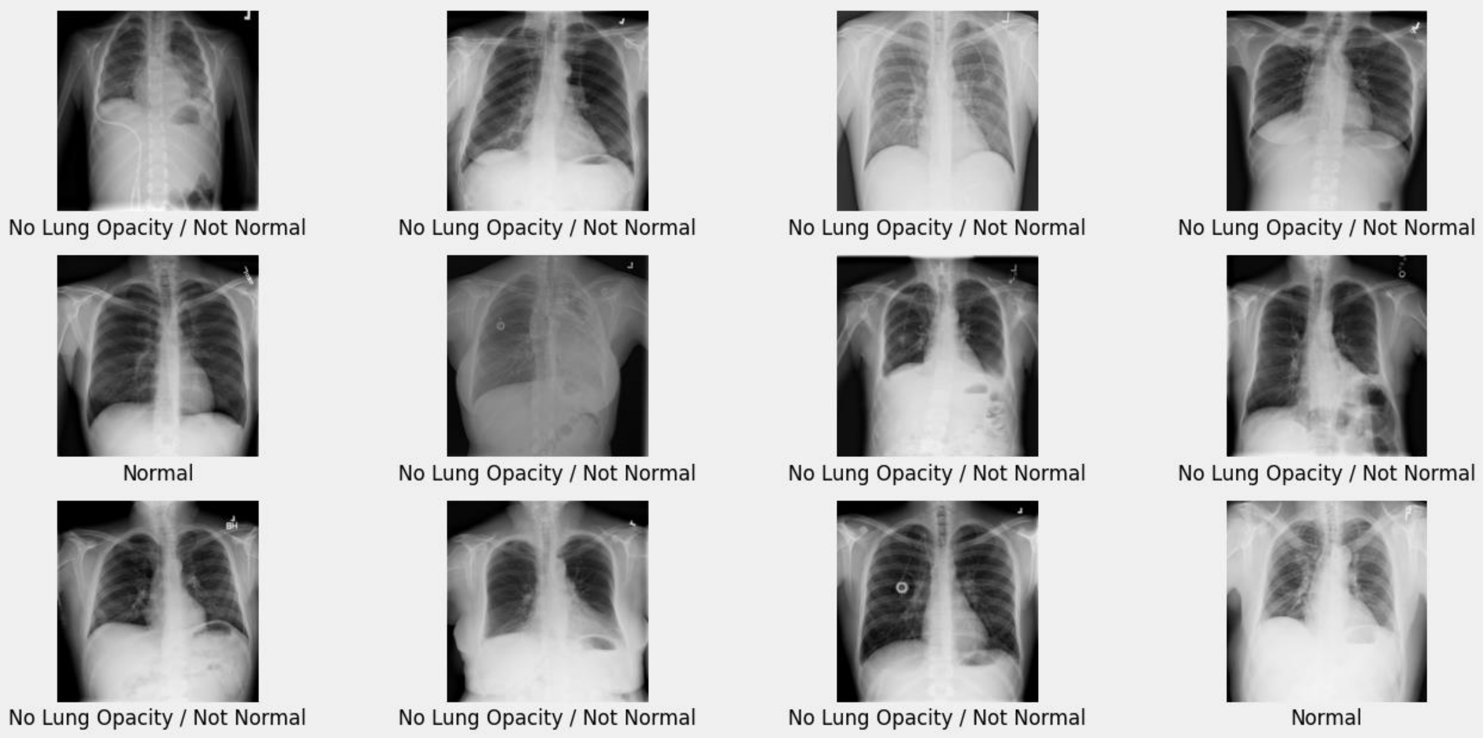


Viewing some sample images from our dataset where lung opacity is not seen:

View Position = “AP”


No Lung Opacity / Not Normal
Normal
No Lung Opacity / Not Normal
No Lung Opacity / Not Normal
Normal
No Lung Opacity / Not Normal
No Lung Opacity / Not Normal
Normal
No Lung Opacity / Not Normal
No Lung Opacity / Not Normal
No Lung Opacity / Not Normal
Normal

View Position = “PA”


No Lung Opacity / Not Normal
No Lung Opacity / Not Normal
No Lung Opacity / Not Normal
No Lung Opacity / Not Normal
Normal
No Lung Opacity / Not Normal
No Lung Opacity / Not Normal
No Lung Opacity / Not Normal
No Lung Opacity / Not Normal
No Lung Opacity / Not Normal
No Lung Opacity / Not Normal
Normal

ANALYSIS ON PATIENT GENDER:

To analyze how many patients in our input data set are Male and how many females, we will need to:

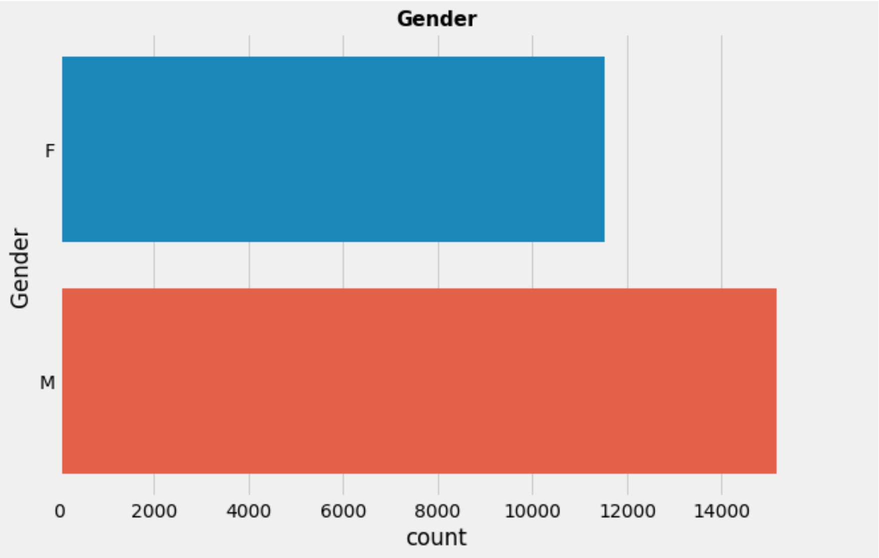
Extract the patient gender from each dicom image

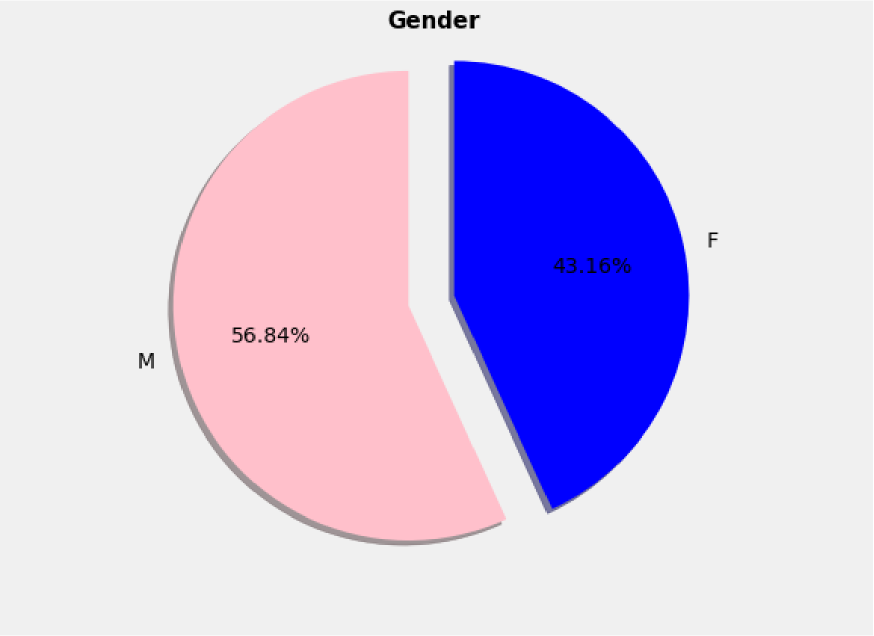
As the labels file can contain more than one entry for a single patient, we will first need to extract the unique patient IDs and then check the Gender statistics.

Upon the plotting the Patient Gender against the whole dataset (26684 unique entries), we get the following:

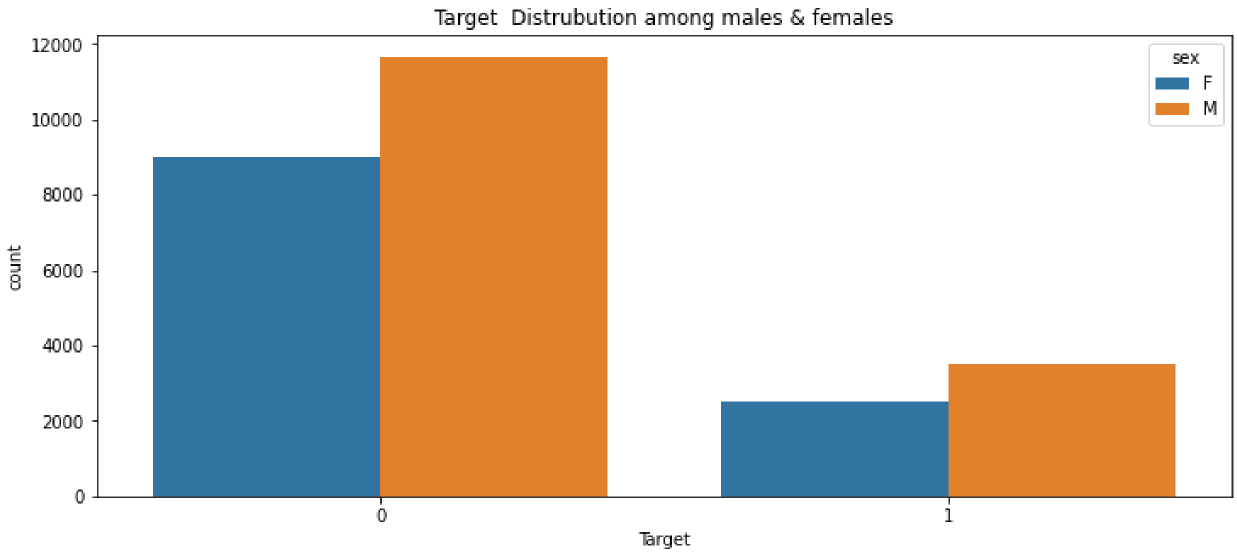
56.84% of the dataset is of the Gender Male

43.16% of the dataset is of the Gender Female



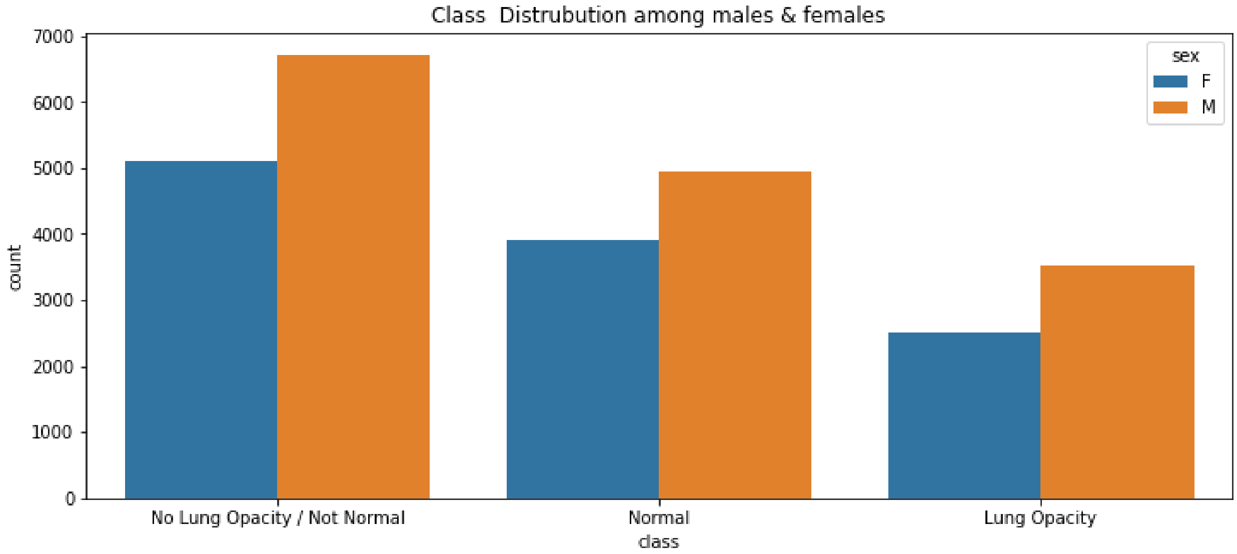


Upon plotting the Patient Gender against Target values, we get the following graph:



From the graph, it is clearly evident that number of positive Pneumonia cases have higher number of Males as compared to females.

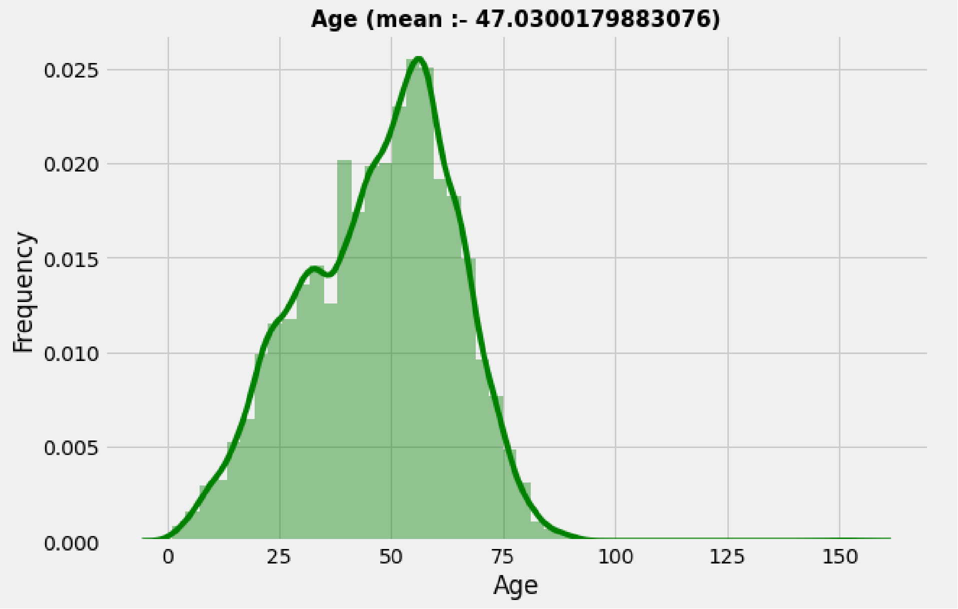
On further analysis on target classes, if we plot each class against the gender, we get the following:



We can clearly see that in all three categories of classes males are more than females

ANALYSIS ON PATIENT AGE:

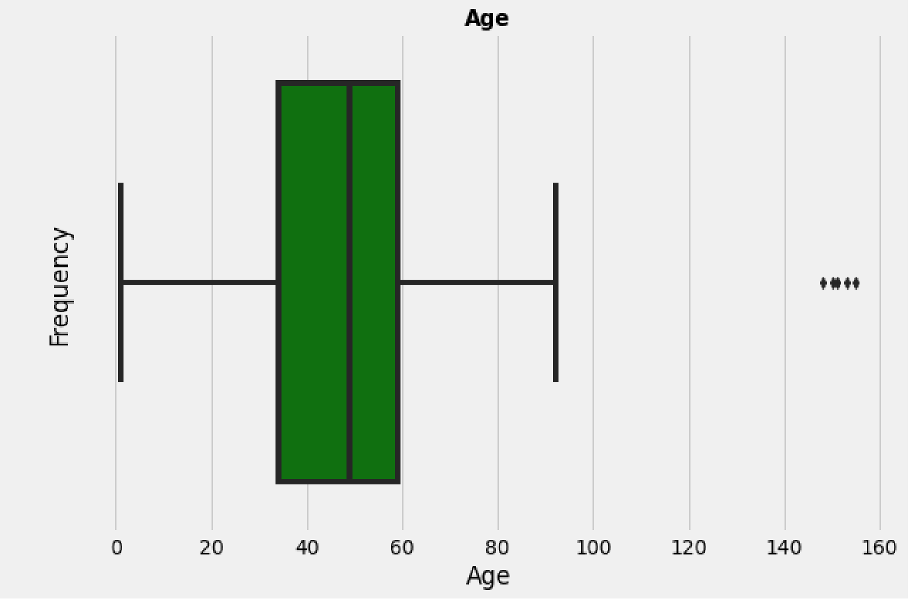
If we plot the age distribution as a distplot, we get the following graph:



Looking at the histogram, we can see that the age distribution is normal, and peaks between 58-59 years. Also, the mean age in our data set is 47 years.

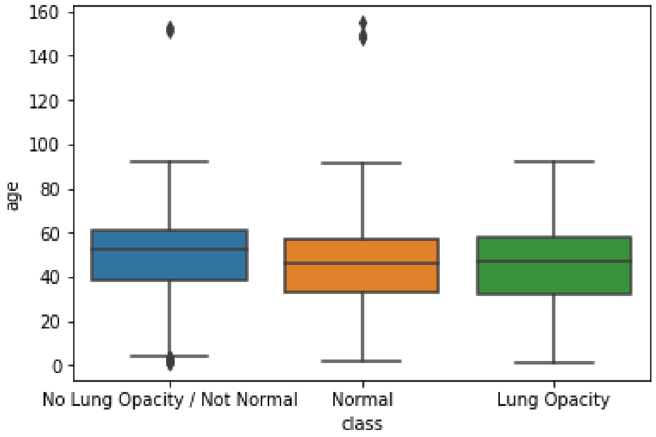
The minimum age is 1 year while the maximum is 155 (As seen in the dataframe.describe) – which means some data values could be incorrect

Plotting a boxplot for the age w.r.t the frequency, we get:



We can see some outliers here which might be due to the possible wrong entries. However, this will not affect the model training on images as model will be trained taking into consideration only the dicom image pixel data for classification purpose and detection which is the scope of this project objective.

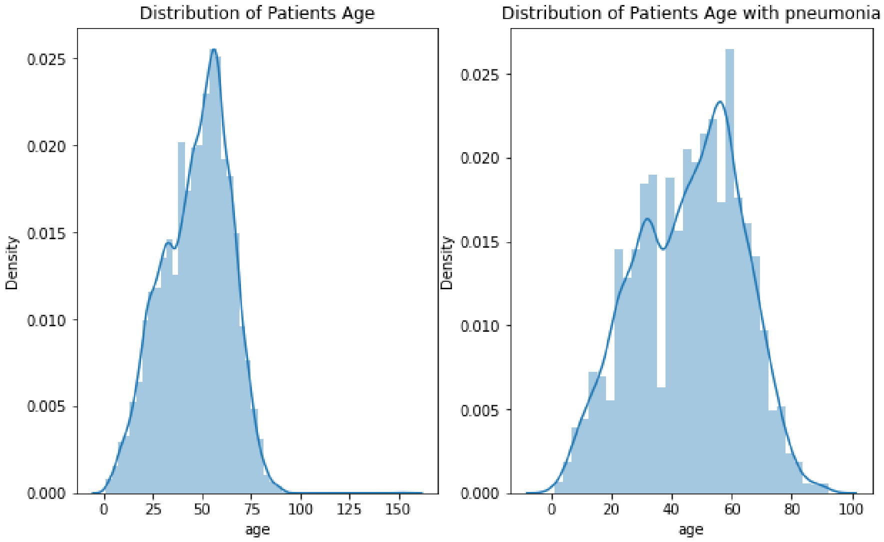
Now if we plot a boxplot for the age and class, we get the following:



Outliers in normal and not normal classes are unusual as these are in the range of 140 to 160. This might be due to the possible wrong entries. However, this will not affect the model training on images as model will be trained taking into consideration only the dicom image pixel data for classification purpose and detection which is the scope of this project objective.

Otherwise, the age for all 3 classes is very similarly distributed with a median between 40 and 60 years.

Now let's compare the age distribution in our complete sample with age distribution in patients with pneumonia. Following are the plots:

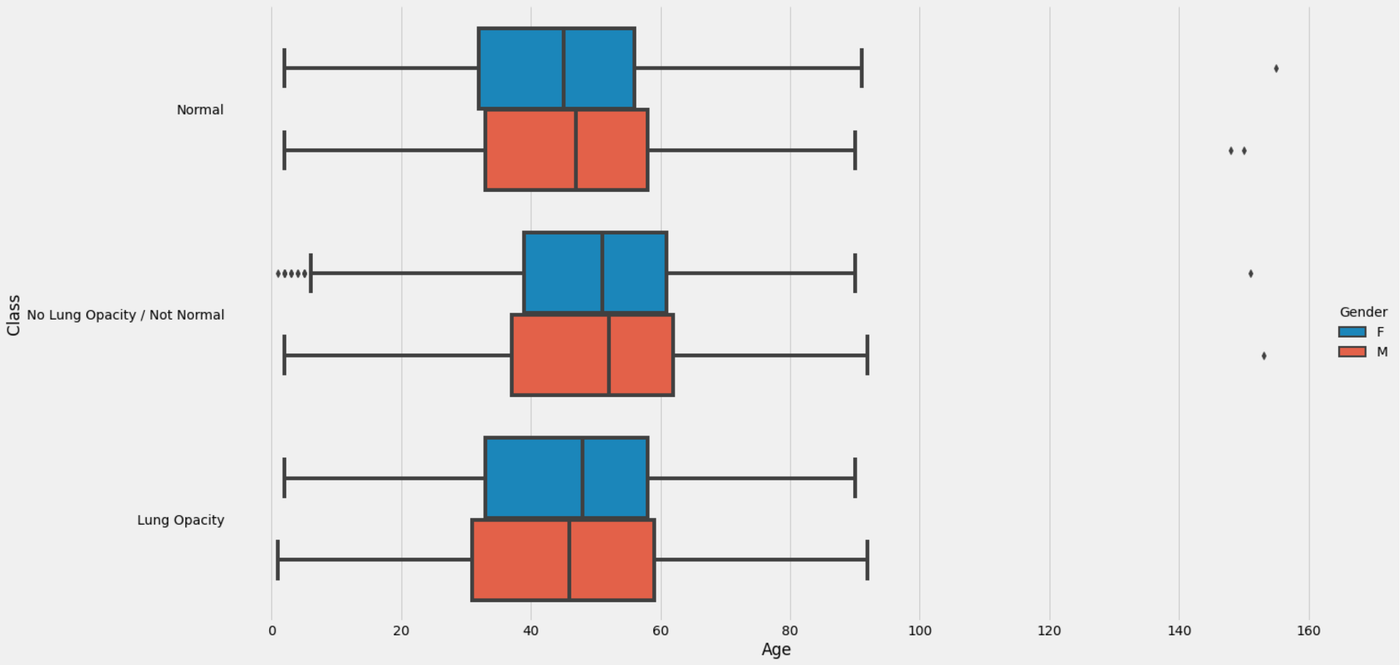


This tells us:

Pneumonia patients age distribution (Graph #2) is slightly skewed, almost normally distributed, with most of the age group around 60. The skewness could be attributed to wrong age entries for some of the rows.
Normal patients age (Graph #1) is distributed normally, there is a slight skew in the graph which is due to the wrong age entries.

ANALYSIS ON GENDER, CLASS AND AGE:

IF we plot a boxplot of Class vs Age, with hue as ‘Gender’, we get the following:

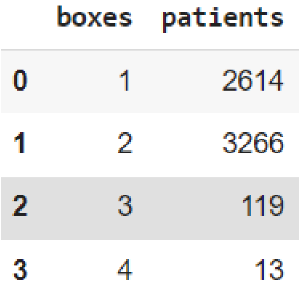


As we can see, the total number of Female cases for all the 3 classes are lesser as compared to Males. However, the age range for both the genders in each class is very close. The median age again, is very close for both male and female genders, in each class.

ANALYSIS ON THE NUMBER OF OPACITIES PER PATIENT

On further analysis on data, we can see that there are 6012 patients (unique patient IDs) who have Pneumonia and a total of 9555 pneumonia patches

Out of the 6012 patients, following is the breakup on the number of opacities(patches):

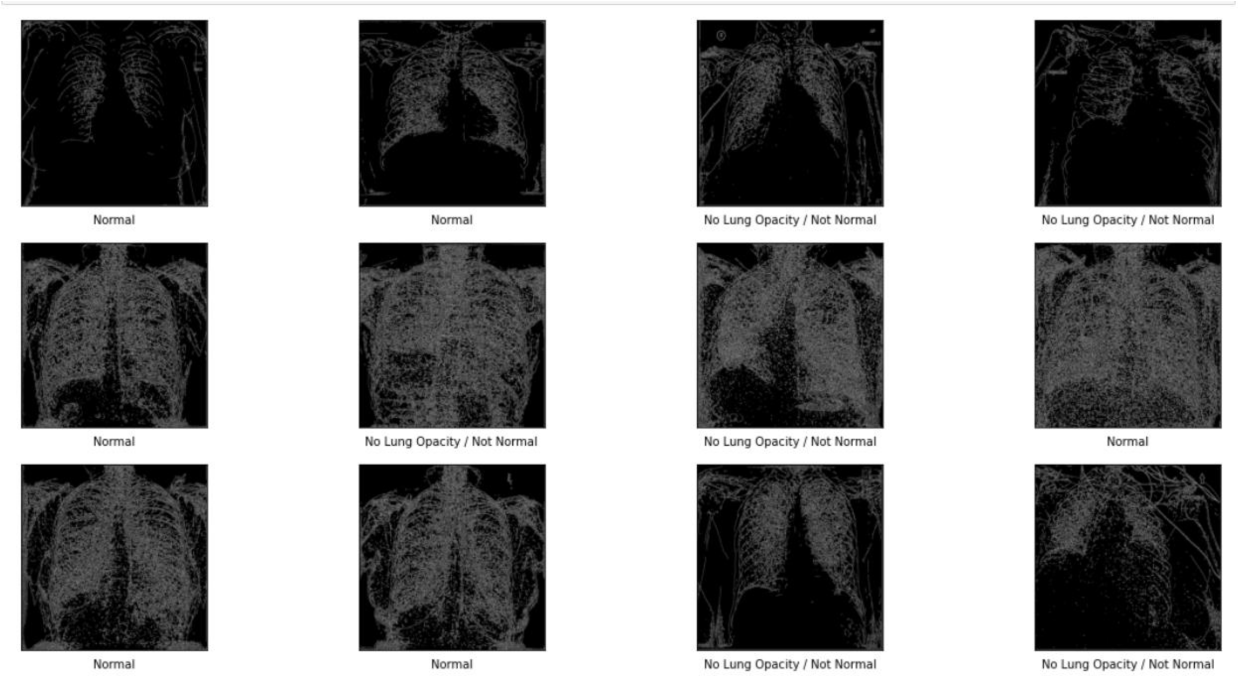


	boxes	patients
0	1	2614
1	2	3266
2	3	119
3	4	13

So, there are 2614 patients with a single pneumonia patch in the lung, 3266 patients with 2 patches, 119 with 3 patches and 13 patients with 4 pneumonia patches in the lungs.

PRE-PROCESSING

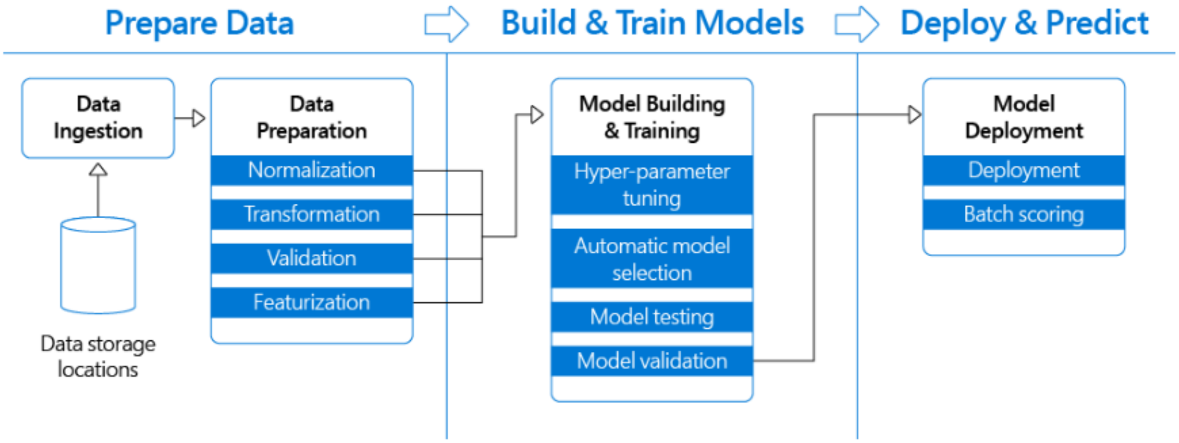
We have successfully achieved the contours detections on these images and tried to pass these contours detected images for training, But we were unsure of the quality of features what we are passing to the model, So we did not experiment by this preprocessed data. The below are the results for contours detection.


Normal
Normal
No Lung Opacity / Not Normal
No Lung Opacity / Not Normal
Normal
No Lung Opacity / Not Normal
No Lung Opacity / Not Normal
Normal
Normal
Normal
No Lung Opacity / Not Normal
No Lung Opacity / Not Normal

DECIDING MODELS AND MODEL BUILDING

Based on the nature of the problem, we had employed the state of art techniques of Computer vision and Deep learning called Object detection, where the main task of the model is to learn patterns by training sample and to predict the location of the object in the image and its labels on real world scenario. What makes this problem more challenging is in most object detection tasks, often the detection model trained and expected to give predictions of location and label of the object when find the object in the image, although there will be cases where the object might not be present, ideally on this situation these models will not give any predictions, but in our problem the model need to predict its label 'Not present' even if the object is not present on the image. So in order to handle this unique situation we have build two different modeling pipelines. Our first pipeline is of modeling classifier where the model is expected to classify labels and the second pipeline is of modeling object localization where it expected to give location of the object what in our case is Pneumonia.

The below picture depicts the basic work flow of the model building process.



Classification Pipeline

The classification pipeline has been done on various parts. These are list of steps involved in preprocessing

Sampling of data

Up sampling and data augmentation

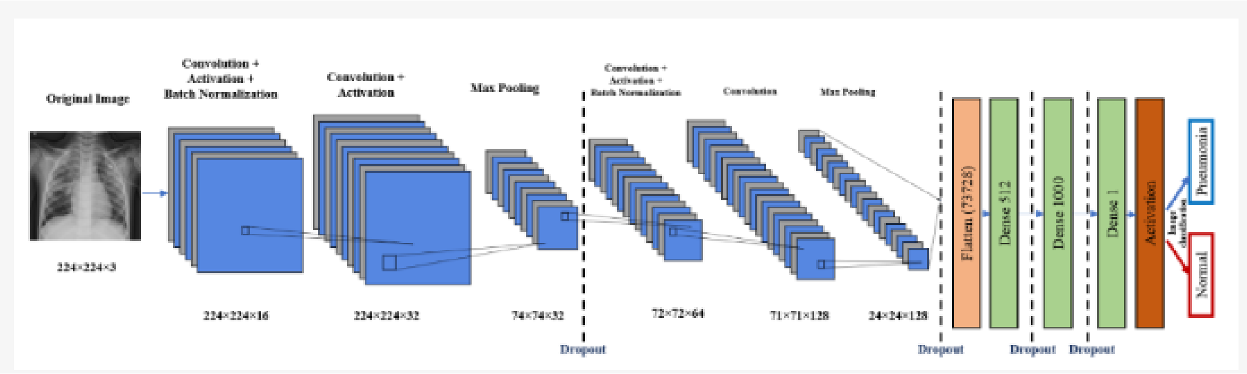
Data preprocessing

Model architecture selected.

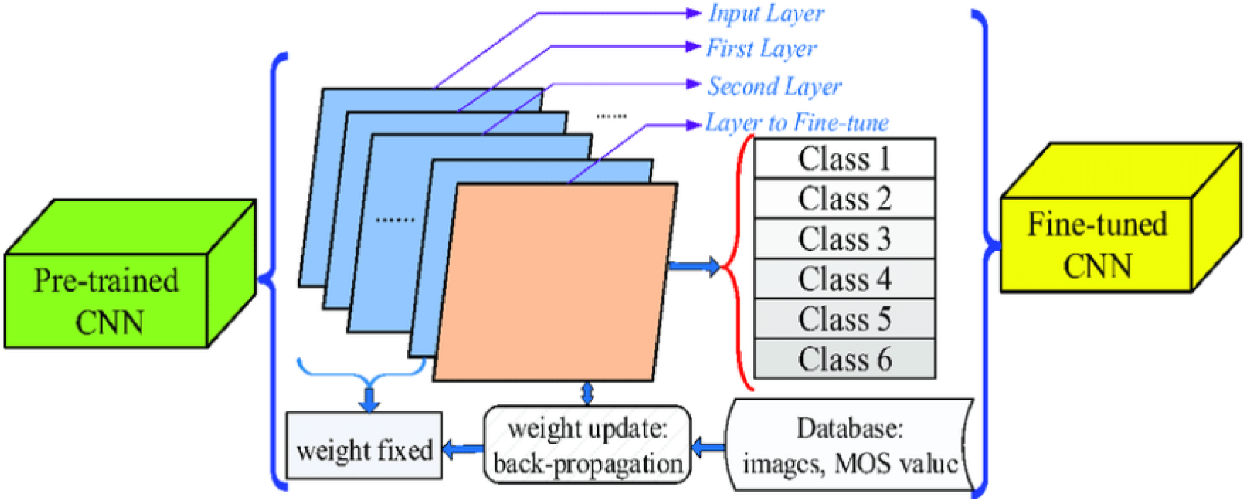
Training and hyper parameters

Validation and hyper parameter tuning.

Performance on test set



So in order to train classifier, we are going to use Fine Tuning, where we take state of art architectures of classifiers which has already been trained on Image net dataset and performed extremely well on real world data.



Sampling

The classification pipeline has been done on various parts.

To begin with we have sample of 26684 where we have only 25% of the total samples where Pneumonia is located. So as we have class imbalance problem on the sample of data we got, we have used Stratified sampling where it would split samples on similar ratio. By using Stratified sampling we have sampled our data into three different samples Training sample to train the model, Validation sample to tune hyper parameters and Test sample to estimate the performance of the model on the real world scenario. The below table shows the sampling configuration.

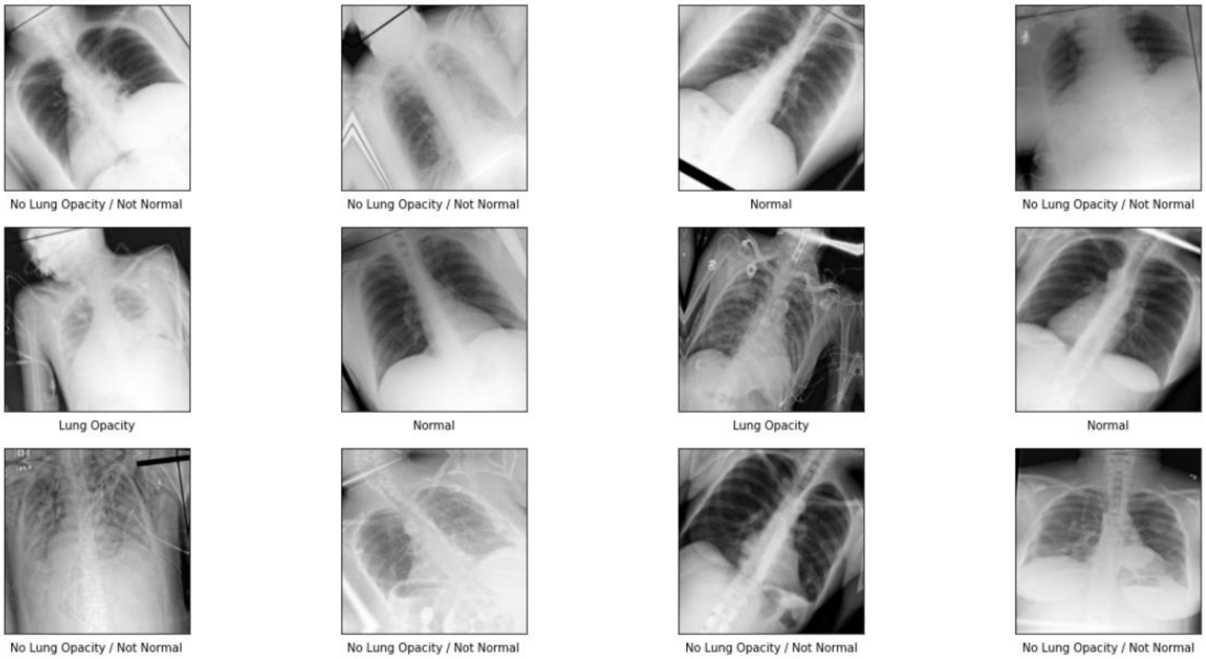
Sample name	Split percentage(%)	No of Images
Training sample	70%	20013
Validation sample	20%	5003
Test sample	10%	1668

Up sampling and Data augmentation

Secondly, as there is class imbalance problem on the original sample, in order to resolve this problem, we have added Up sampling and data augmentation techniques to handle class imbalance problem, where with the combination of Up sampling and data augmentation we have balanced the count of images on the Training sample. The data augmentation methods we used were

Random flip horizontally.
Random Rotation.
Random Zoom.
Random Brightness.

The below picture shows the sample of images after data augmentation.


No Lung Opacity / Not Normal
No Lung Opacity / Not Normal
Normal
No Lung Opacity / Not Normal
Lung Opacity
Normal
Lung Opacity
Normal
No Lung Opacity / Not Normal
No Lung Opacity / Not Normal
No Lung Opacity / Not Normal
No Lung Opacity / Not Normal

Data preprocessing

We have used four preprocessing methods before the images go for training. We had got two channel image in our sample data, so we have converted it to three channel image as all the pretrain architectures have been train on three channel images, therefore the model will train much better when we pass number of channels as three. We have also down sampled the images size from 1024*1024 to 224*224, as this will decrease the training time, less computation and in general most of the pre trained architectures have got trained on this configuration size of image. We had also standardized the pixel values in the image and divided the entire training sample into batch of 16 images which makes 1899 total batches of training sample.

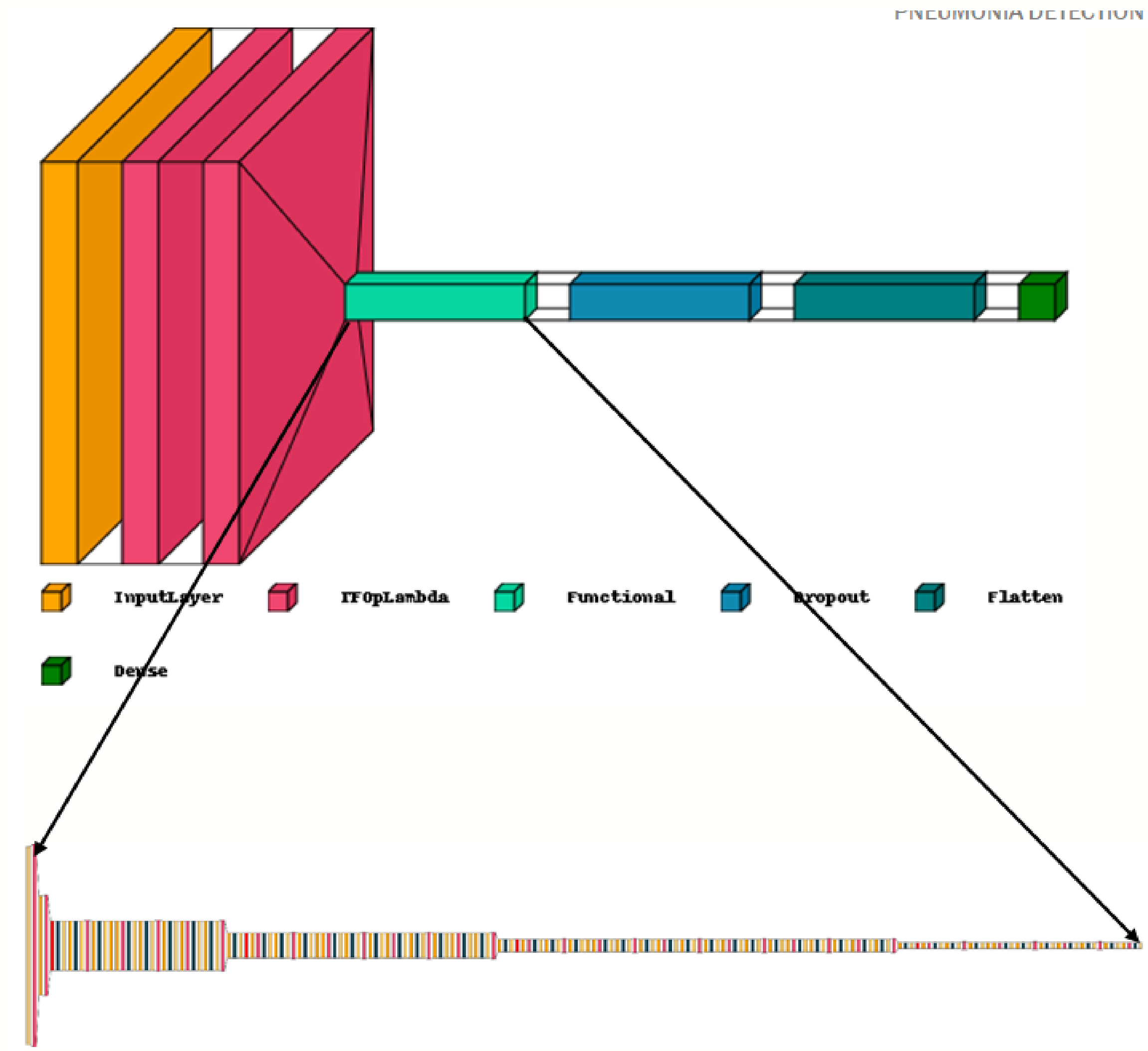
Model architecture selected

We have trained two different classifier with different backbone architectures in order to compare the performance and to have to choice to select the best one. The two are mentioned on below list

Classifier with backbone Resnet50v2 architecture.
Classifier with backbone Chexnet architecture.

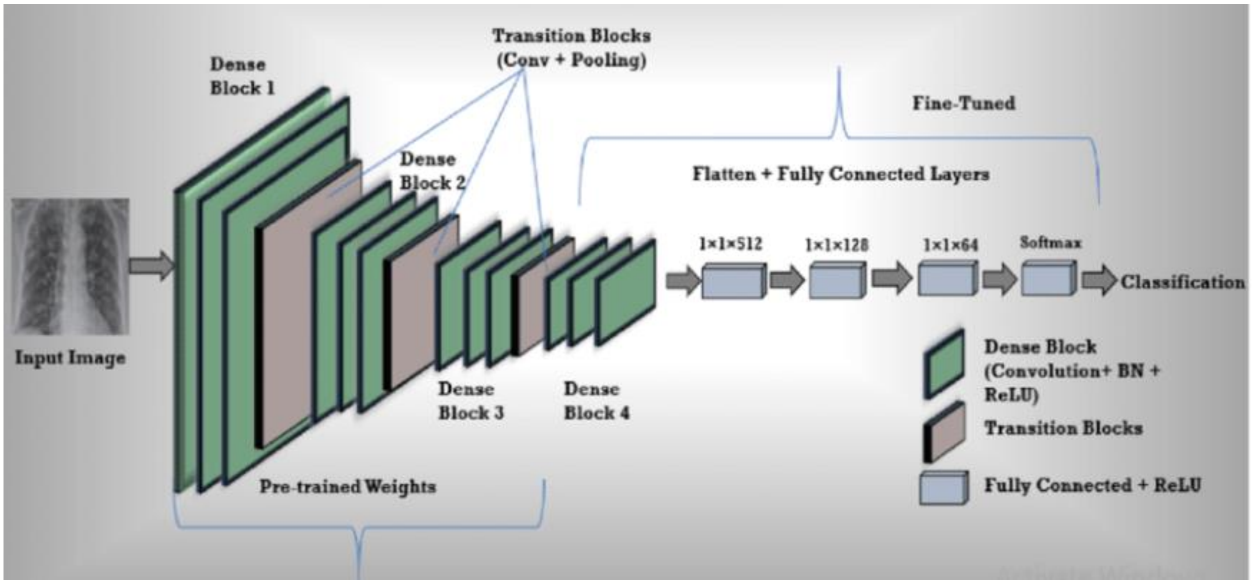
Note:- we purely did not used the above two architectures we have fine tuned the last layer and changed the activation function to Sigmoid as it's a binary classification problem where it need to predict two labels.

<u>Model architecture with Resnet50v2 backbone</u>.



For this model we have frizzed first 20 layers of the model and trained last 30 layers.

Model architecture with Chexnet as backbone.



For this model we have frizzed first 10 layers of the model and trained last 111 layers.

Training and hyper parameters Tuning

These are hyper parameters for Resnet50v2

Hyper parameters name	values
Input shape	(16,224,224,3)
Output shape	(,1)
Batch Size	16
Learning rate	0.01
Epochs	6
Optimizer	Adam (decay=0.0)
Loss function	Binary_crossentropy
Accuracy	Accuracy
Shuffle	Yes
Multiprocessing	Yes
No Workers	2*Batch size

we had applied various checkpoints while training classifier with Resnet50v2 backbone which are listed below.

- **Early stop** : we have configured early stop to monitor validation loss by patience 10
- **ReduceLRONPlateau** : Reduce learning rate when a metric has stopped improving models often benefit from reducing the learning rate by a factor of 0.2 once learning stagnates. This callback monitors a quantity and if no improvement is seen for a 4 number of epochs, the learning rate is reduced, but we had also set the min learning rate that it could reduce is 0.000000001.
- **Model check points saving automation** : We have automated the saving part of the model where after every epoch the model validate performance on validation data, if the validation accuracy is greater than previous epochs validation accuracy, it's will save the weights of the best model with max validation accuracy.

<u>These are hyper parameters for Chexnet</u>

Hyper parameters name	**values**
Input shape	(16,224,224,3)
Output shape	(,1)
Batch Size	16
Learning rate	0.0001
Epochs	6
Optimizer	Adam
Loss function	Binary_crossentropy
Accuracy	Accuracy
Shuffle	Yes
Multiprocessing	Yes
No Workers	2*Batch size

we had applied various checkpoints while training classifier with Chexnet backbone which are listed below.

- **Early stop** : we have configured early stop to monitor validation loss by patience 10
- **ReduceLRONPlateau** : Reduce learning rate when a metric has stopped improving models often benefit from reducing the learning rate by a factor of 0.2 once learning stagnates. This call back monitors a quantity and if no improvement is seen for a 4 number of epochs, the learning rate is reduced, but we had also set the min learning rate that it could reduce is 0.000000001.
- **Model check points saving automation** : We have automated the saving part of the model where after every epoch the model validate performance on validation data, if the validation recall is greater than previous epochs validation recall, it's will save the weights of the best model with max recall score.

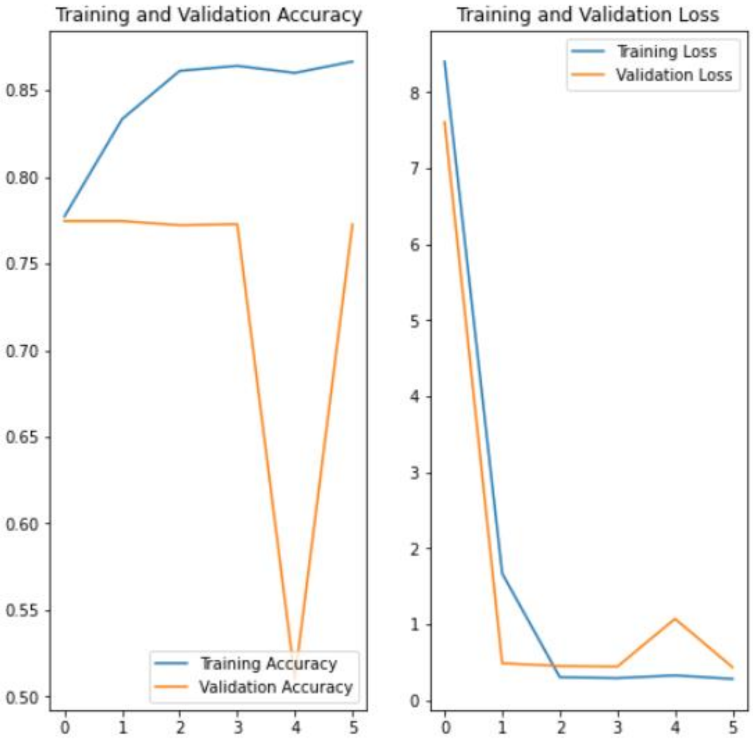
Hyper parameter tuning

All the above configuration values that we have decided based on manual hyper parameter tuning, we are aware of Keras Tuner but it’s not feasible, as we have limitation in terms of GPU access and we have massive training data where in order to train one epoch its taking around 3 to 4 hrs on an average on CPU. We did not opt for Google Colabs as, when even the colab is inactive for more than 10 to 15 minutes it get disconnected and it’s not feasible for someone to be with PC for longer hours while training.

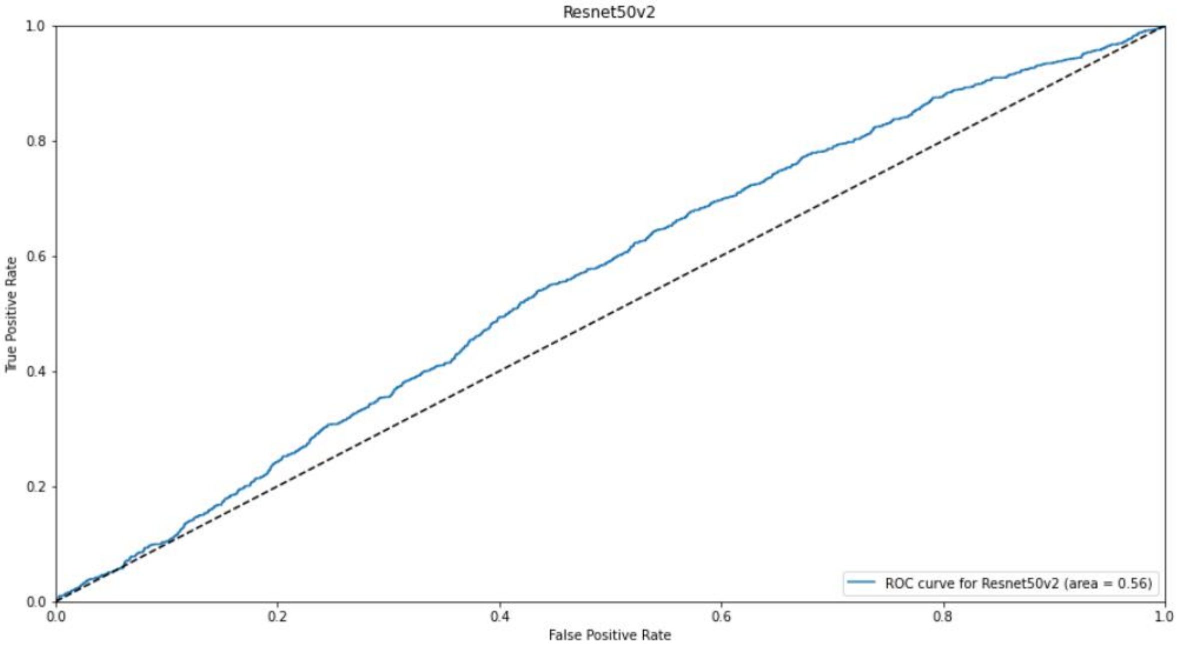
Model Evaluation

Resnet50v2 evaluation

Training logs analysis



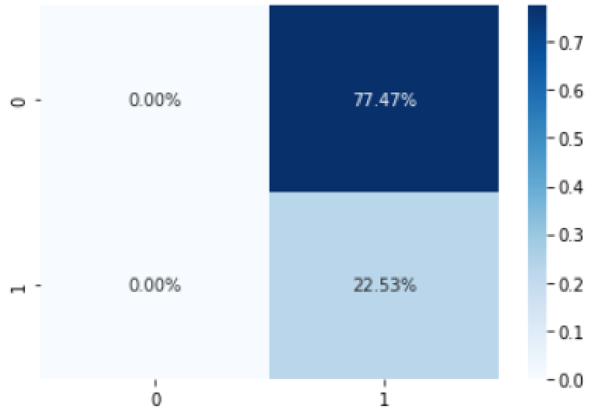
The above visualization illustrate that the model has not learned good and we can even make out that the model would poorly perform on validation set.



The ROC of the model is 0.55 which is really poor.

Training accuracy	0.86
Validation accuracy	0.77
Test accuracy	0.50
Precision in No Pneumonia case	0.0
Precision in Pneumonia case	0.22
Recall in No Pneumonia case	0.0
Recall in Pneumonia case	1.0
F1 score in No Pneumonia case	0.0
F1 score in Pneumonia case	0.36
Overall F1 score value	0.18
ROC score	0.55

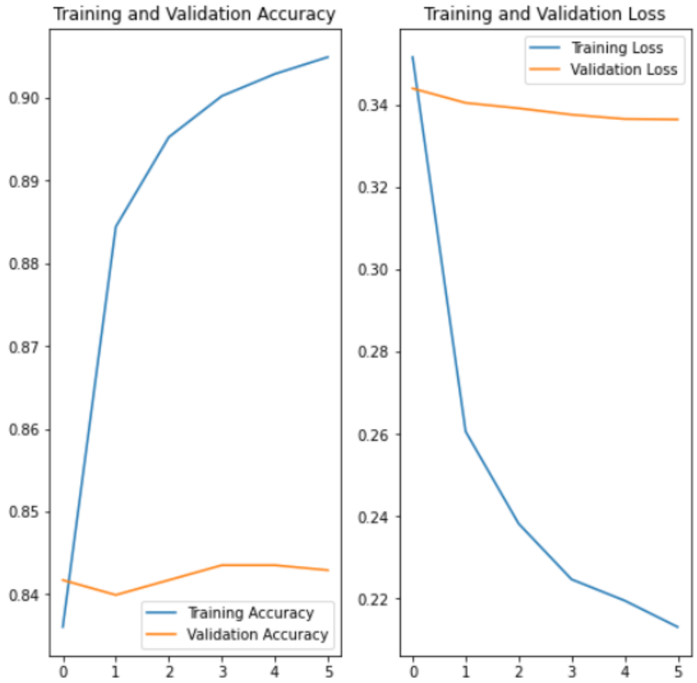
Confusion matrix



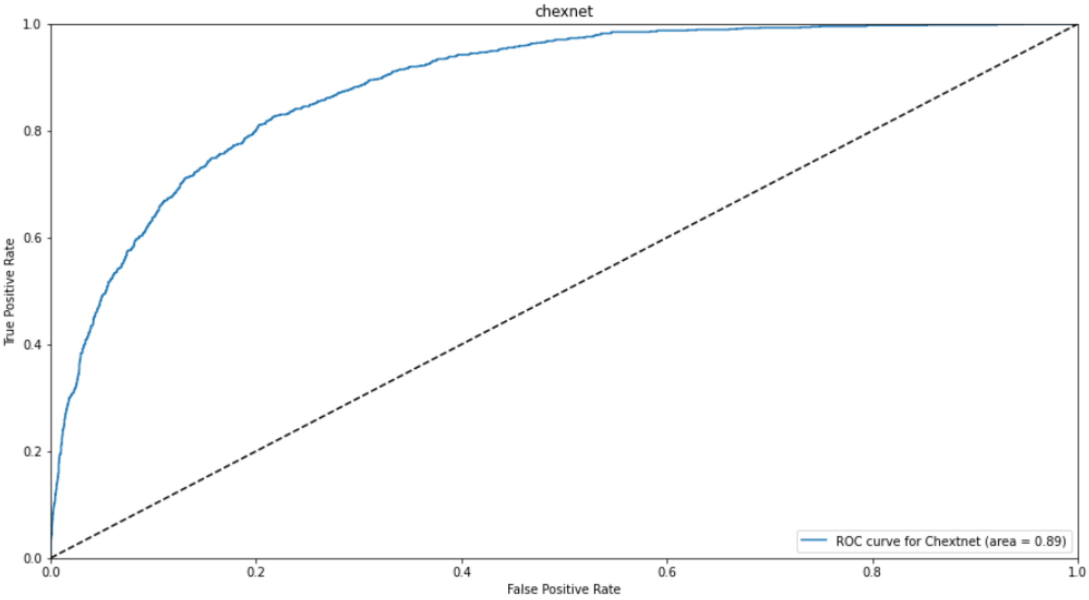
By seeing the above results we can make out that the model has learned some Pneumonia features, but its failed to learn pattern of Non Pneumonia images as we can locate on confusion matrix, the model has correctly classifier all the Pneumonia cases correctly but it failed very badly when it come to Non Pneumonia, it's basically like its saying all images which are coming are having Pneumonia.

Chexnet evaluation

Training logs analysis



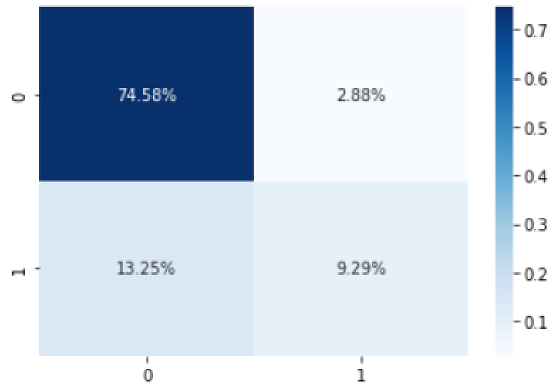
The above visualization shows, we have trained our model on very high learning rate as we see the loss curve of the validation set, where it's very flat.



The ROC of the model is 0.88 which is really good and in terms of threshold we had decided to keep as 0.6 because as the ROC is pretty high so we can increase the threshold by which the precision of classifying Pneumonia will also increase with very less decrease in the recall score.

Training accuracy	0.90
Validation accuracy	0.84
Test accuracy	0.83
Precision in No Pneumonia case	0.84
Precision in Pneumonia case	0.76
Recall in No Pneumonia case	0.96
Recall in Pneumonia case	0.41
F1 score in No Pneumonia case	0.90
F1 score in Pneumonia case	0.53
Overall F1 score value	0.72
ROC score	0.88

Confusion matrix



The above results depicts that performance of the Chexnet model is really outstanding, as it has the accuracy of 83% where it have predicted Pneumonia with the precision of 76% by maintaining the precision of predicting Non Pneumonia at 96%. Even all the other matrix are pretty much well balanced. By seeing confusion matrix we can infer that in 22% of Pneumonia cases images it has predicted 9.29% correctly and in 76% of Non Pneumonia images it has predicted 74% correctly.

Sample performance on test set

True_label :- No pneumonia,Label_predicted :- pneumonia,Confidence :- 0.6%

True_label :- No pneumonia,Label_predicted :- No pneumonia,Confidence :- 1.0%

True_label :- No pneumonia,Label_predicted :- No pneumonia,Confidence :- 1.0%

True_label :- No pneumonia,Label_predicted :- No pneumonia,Confidence :- 1.0%

True_label :- No pneumonia,Label_predicted :- No pneumonia,Confidence :- 0.7%

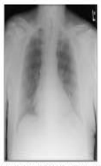
True_label :- No pneumonia,Label_predicted :- No pneumonia,Confidence :- 1.0%

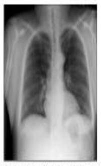
True_label :- No pneumonia,Label_predicted :- No pneumonia,Confidence :- 0.7%

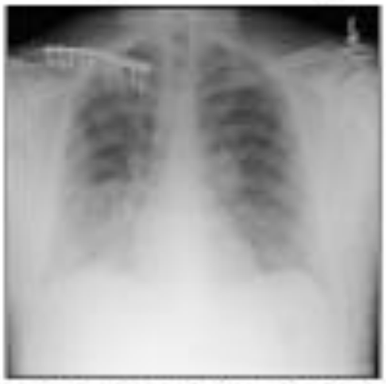
True_label :- pneumonia,Label_predicted :- pneumonia,Confidence :- 0.9%

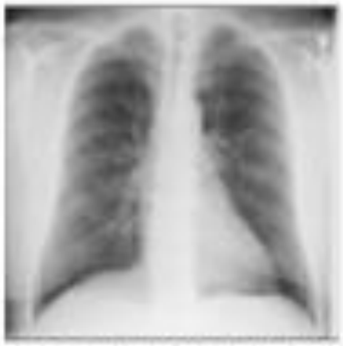
True_label :- No pneumonia,Label_predicted :- No pneumonia,Confidence :- 0.7%

True_label :- pneumonia,Label_predicted :- pneumonia,Confidence :- 0.9%


True_label :- No pneumonia,Label_predicted :- No pneumonia,Confidence :- 0.9%


True_label :- No pneumonia,Label_predicted :- No pneumonia,Confidence :- 1.0%

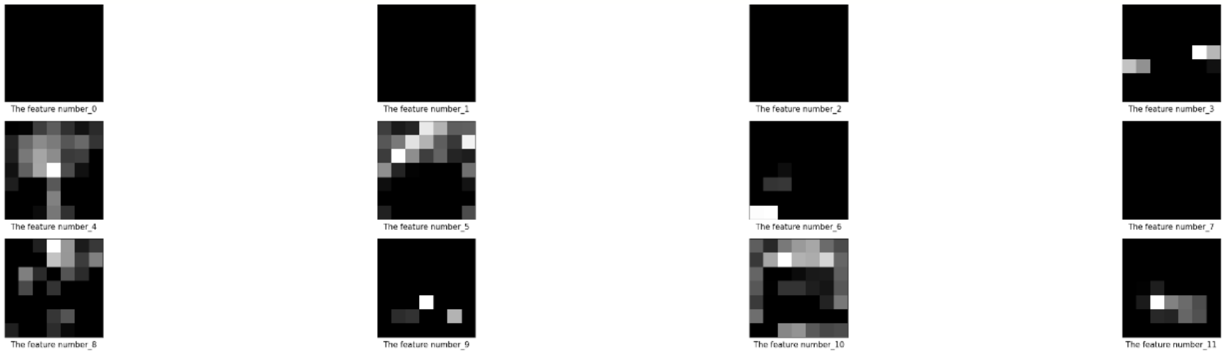

True_label :- pneumonia,Label_predicted :- pneumonia,Confidence :- 0.9%


True_label :- No pneumonia,Label_predicted :- No pneumonia,Confidence :- 1.0%

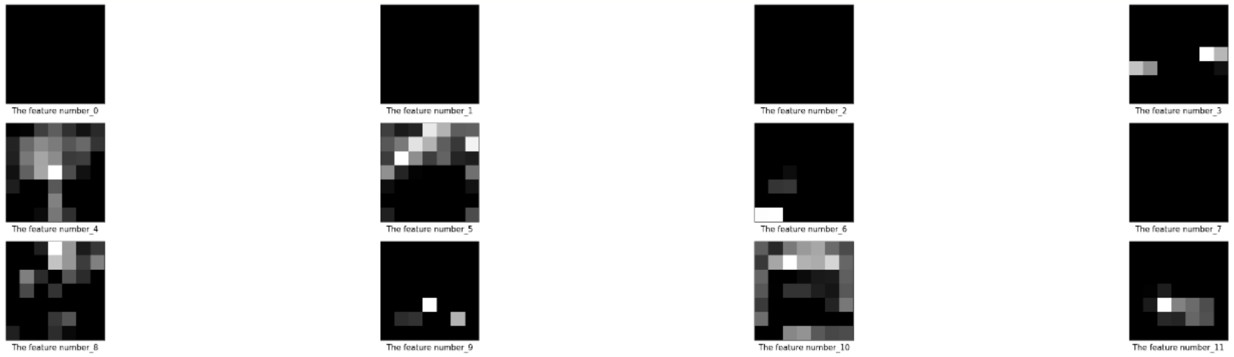
Important features on which decisions have been made by Chexnet

To understand our model predictions better, we have extracted the features map from the last second layer which gives 1024 features with 7*7 dimensions. As the feature maps are being normalized, we cannot view the clear features .The below sample images from 1024 features are few examples of what features are getting into classifier layer based on which its learning to classify the labels.

The below sample of 12 features are for images which does not contain Pneumonia.



The below sample of 12 features are for images which does contain Pneumonia.



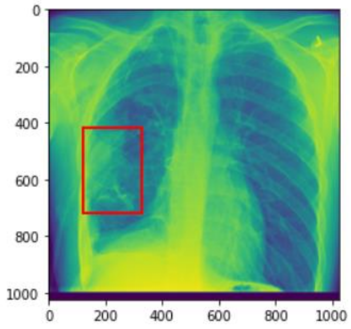
We initially though by doing this, we can get better inference on our model 's prediction, but after experimenting we learned that by the above way of visualization we could not infer much about the important features of the image.

We found one way of finding the important features of image, the technique is **Integrated Gradient.** But due to time constraint we did not implement this method.

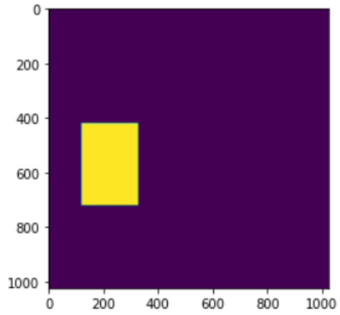
This concludes our findings on the classification part of the problem!!

Object localization Pipeline

The Object localization is the task of finding location where the Pneumonia is located by predicting coordinates of its bounding boxes. Below are the steps for this pipeline. But to obtain this objective we have selected Instance Segmentation model name **UNET.** So the main idea is at first we are going to convert bounding boxes as masks, you can see the example on below diagram.

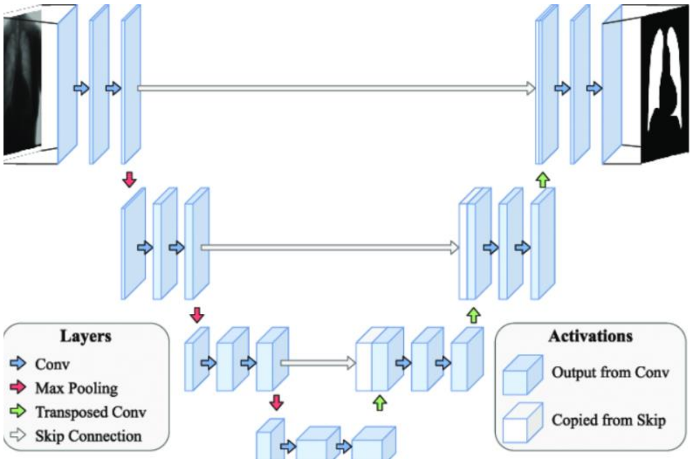


With coordinates



Masks

Then we are passing origin image and masks as ground truth, we are expecting model to learn patterns where there is patch on the image and to predict the masks for given test image. The below image depicts the flow of UNET architecture. So for our use case, we have considered VGG16 as backbone for the UNET architecture.



The below are steps for the pipeline.

- Sampling the data on to different samples
- Data preprocessing and augmentation
- Model Architecture
- Training and hyper parameter tuning.
- Model eval.
- Results.

Sampling the data

So we are considering only those images which contain Pneumonia, So after filtering we got 6012 images as of whole sample. So below table shows the sampling configuration.

Sample name	Split percentage(%)	No of Images
Training sample	70%	4208
Validation sample	20%	1623
Test sample	10%	181

Data preprocessing

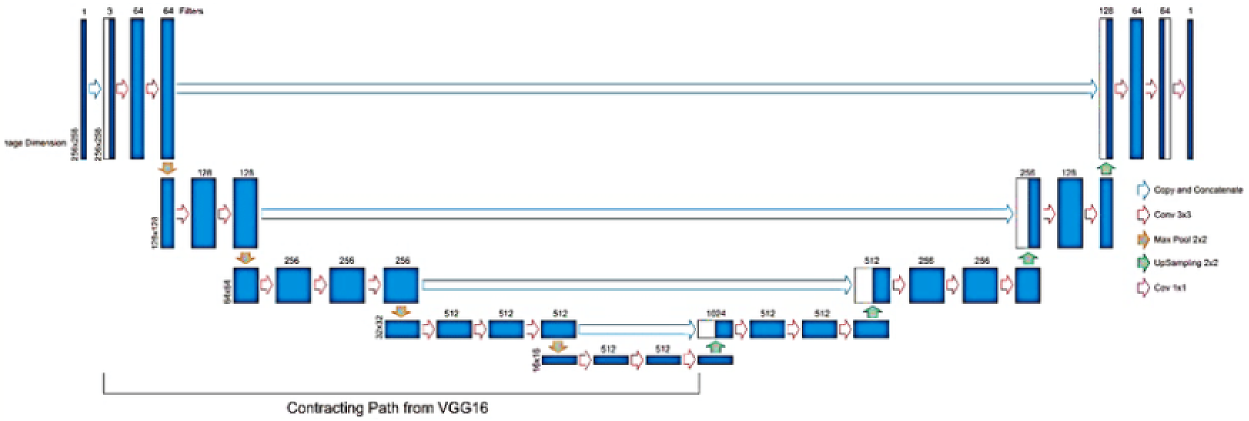
We have used four preprocessing methods before the images go for training. We had got two channel image in our sample data, so we have converted it to three channel image as all the pretrain architectures have been train on three channel images, therefore the model will train much better when we pass number of channels as three. We have also down sampled the images size from 1024*1024 to 224*224, as this will decrease the training time, less computation and in general most of the pre trained architectures have got trained on this configuration size of image. As we resized the image, we should scale the coordinates to the new size of the image. The scaling of the coordinates have been done by below formula.

New x = old x * (resized image width/old image width)
New y = old y* (resized image height/old image height)

Then we have standardized the pixel values of the image. In data augmentation we have done Random horizontal flip.

Model Architecture

UNET VGG16 backbone



Training and hyper parameter tuning

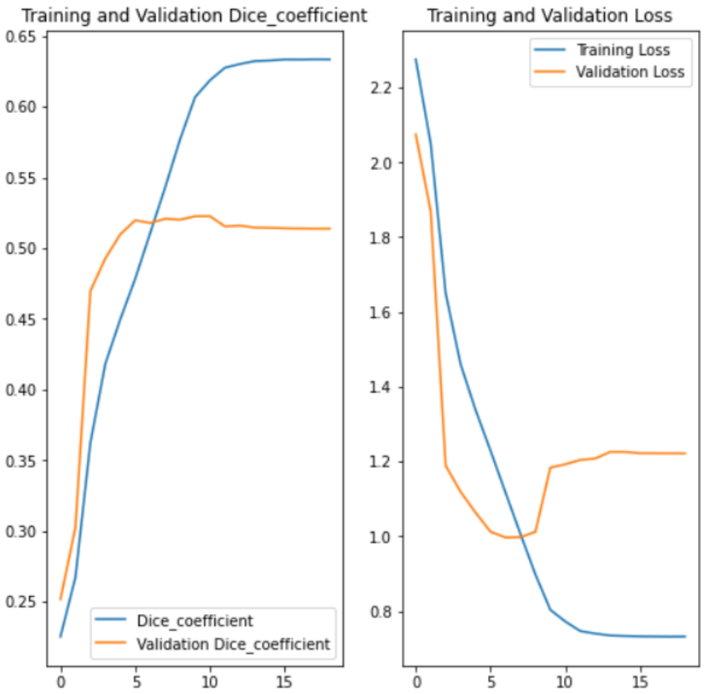
VGG16 UNET

Hyper parameters name	values
Input shape	(16,224,224,3)
Output shape	(16,224,224)
Batch Size	16
Learning rate	0.00001
Epochs	20
Optimizer	Adam (decay=0.0)
Loss function	binary_crossentropy – (Dice_coefficent+epsilone)
Accuracy	Dice_coefficient
Shuffle	Yes
Multiprocessing	Yes
No Workers	2*Batch size

we had applied various checkpoints while training Unet with VGG16 backbone which are listed below.

- **Early stop** : we have configured early stop to monitor validation loss by patience 10
- **ReduceLRONPlateau** : Reduce learning rate when a metric has stopped improving models often benefit from reducing the learning rate by a factor of 0.2 once learning stagnates. This callback monitors a quantity and if no improvement is seen for a 4 number of epochs, the learning rate is reduced, but we had also set the min learning rate that it could reduce is 0.000000001.
- **Model check points saving automation** : We have automated the saving part of the model where after every epoch the model validate performance on validation data, if the validation Dice coefficient is greater than previous epochs validation accuracy, it's will save the weights of the best model with max validation Dice coefficient.

Model evaluation



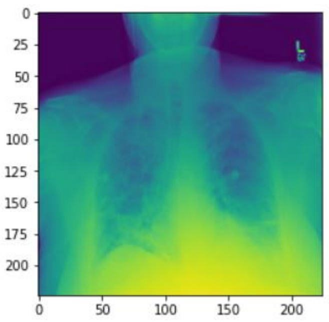
By above visualization depicts the learning rate what we have selected is looking good and even got some decent Dice coefficient scores.

Performance report

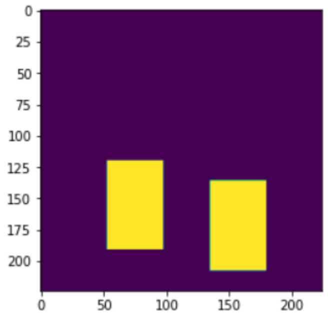
Training Dice Score	0.62
Validation Dice Score	0.52
Testing Dice Score	0.44
Training loss	0.77
Validation loss	1.19

Sample Result on Test Image

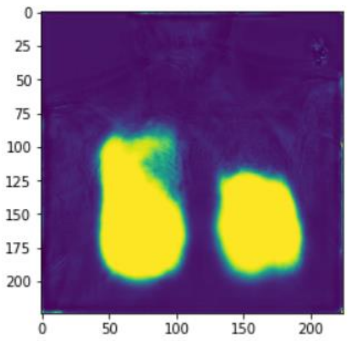
The Dice coefficient is 0.59



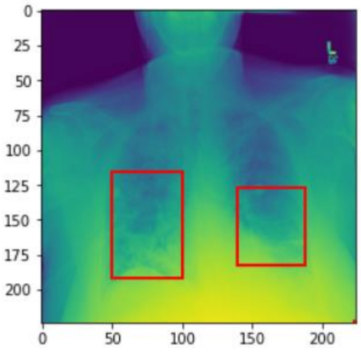
Original Image



Ground Truth



Predicted Masks



Bounding Box extracted

Conclusion

If user try to pass any vertical flipped image or image with some rotations or with distorted image , the classification model would work, but the localization model would fail in terms of getting right bounding boxes. So In order to handle this problem we can add more data augmentation like vertical flips , rotation or distortion to the training images while training the localization Model.

Another limitation that we faced was in terms of infrastructure availability because of which we could not do extensive hyper parameter tuning or try new models. Access to better infrastructure would help us to improve the model performance.

Implications

We can convert this solution to some application which would automate the detection of Pneumonia regions by passing DICOM images. We can use this application for medical education purpose as well as we can integrate this solution to the Radiologist work flow where this can reduce the time of diagnosis. Our model predicts with 83% of confidence on an average, but the confidence score is very much dependent on type of image we pass. Our model gives the confidence score for each image which has been passed for prediction.

Closing reflections

Given the resource and time we can experiment with more image preprocessing methods like image contouring. We can also experiment with other state-of-the-art models.

We have learned that there is always scope of improvement on the model performance based on time and resources we have. Also Computer vision methods have been very effective and optimal to solve this unique problem in this health care domain.

REFERENCES

These are the reference used for getting images of the document

https://medium.com/microsoftazure/how-to-accelerate-devops-with-machine-learning-lifecycle-management-2ca4c86387a0
https://www.researchgate.net/figure/The-sketch-map-of-the-fine-tuning-strategy-To-transform-a-pre-trained_fig3_334060618
https://www.mdpi.com/2075-4418/10/9/649/htm
https://www.researchgate.net/figure/a-Architectural-design-of-CheXNet-model-b-Proposed-fine-tuned-CheXNet-model-with_fig3_351940168
https://www.researchgate.net/figure/Simplified-U-Net-architecture-Adapted-from-Ronneberger-et-al-9_fig2_341098419

These are the reference of the actual project implementations.

https://arxiv.org/abs/1711.05225?context=cs.LG
https://medium.com/analytics-vidhya/detection-and-semantic-segmentation-of-pneumothorax-disease-from-x-ray-images-using-deep-learning-890bbfcb5bd6
https://arxiv.org/pdf/1512.03385v1.pdf