Pneumonia Detection Challenge

Interim Report

Submitted By: Group 13

PRESENTED TO	Great Learning
PRESENTED BY	Group 13 – Aug C – Great Learning
DATE	24 July 2020
VERSION	1.0

Group 13 – Contacts

Name	Phone	Email
Sonal Sharma	+91 88102 64764	Sonal_1996@yahoo.com
Chandrashekar C	+91 98867 94586	chandrashekar.c@gmail.com
Susanta Mondal	+91 70440 55951	Susanta.221285@gmail.com
Gopinath Bailur	+91 99400 78996	gbailur@gmail.com

CHANGE LOG			
DATE	VERSION	AUTHOR	CHANGE DESCRIPTION
24th July 2020	1.0	Group 13	Initial Draft
27th July 2020	1.1	Group 13	Updated based on the feedback with the mentor

Table of Contents

1 Overview

Pneumonia is a form of acute respiratory infection that affects the lungs. The lungs are made up of small sacs called alveoli, which fill with air when a healthy person breathes. When an individual has pneumonia, the alveoli are filled with pus and fluid, which makes breathing painful and limits oxygen intake.

Following are some of the key facts about Pnuemonia which needs at most attention to address the problem proactively.

- Pneumonia accounts for 15% of all deaths of children under 5 years old, killing 808 694 children in 2017.
- Pneumonia can be caused by viruses, bacteria, or fungi.
- Pneumonia can be prevented by immunization, adequate nutrition, and by addressing environmental factors.
- Pneumonia caused by bacteria can be treated with antibiotics, but only one third of children with pneumonia receive the antibiotics they need.

Per WHO, Pneumonia is the single largest infectious cause of death in children worldwide. Pneumonia killed 808 694 children under the age of 5 in 2017, accounting for 15% of all deaths of children under five years old. Pneumonia affects children and families everywhere, but is most prevalent in South Asia and sub-Saharan Africa. Children can be protected from pneumonia, it can be prevented with simple interventions, and treated with low-cost, low-tech medication and care.

The WHO and UNICEF integrated Global action plan for pneumonia and diarrhoea (GAPPD) aims to accelerate pneumonia control with a combination of interventions to protect, prevent, and treat pneumonia in children with actions to:

- protect children from pneumonia including promoting exclusive breastfeeding and adequate complementary feeding.
- prevent pneumonia with vaccinations, hand washing with soap, reducing household air pollution, HIV prevention and cotrimoxazole prophylaxis for HIV-infected and exposed children.
- treat pneumonia focusing on making sure that every sick child has access to the right kind of care -- either from a community-based health worker, or in a health facility if the disease is severe -- and can get the antibiotics and oxygen they need to get well;

Based on research we have made an attempt to look at the Chest Radiography to identify the Lung Opacity and quickly help Clinical specialists to take right decisions to drive proactive measure to cure and help avoid the spread of this decease to larger extent.

2 Abstract

This project is aimed at detecting Pneumonia by locating the lung opacities on the Chest radiographs. This process can help identify the problem at an early stage as well as helps the Clinical analysis much faster and drives better decision making. This process of Pneumonia detection will be done by looking at several thousand images of Chest radiographs taken from past wherein the analysis and desired results were identified by the specialists. These past datapoints will become the indicators and these images will be processed through the Computer Vision Technology of deep learning to capture every details by which a Deep Learning Algorithm will be built.

The new patients data will be fed to this model which detects the Lung Opacity indication along with its location such that Clinical specialists will be able to confirm diagnosis quickly and can help in taking respective decisions quickly to move forward with the next steps of the treatment.

Deep neural networks models have conventionally been designed and experiments were performed upon them by human experts in a continuing trial and error method. This process demands enormous time, knowhow and resources. To overcome this problem, a novel but simple model is introduced to automatically perform optimal classification tasks with deep neural network architecture .

The Neural network architecture was specifically designed for Pneumonia image classification tasks. The proposed technique is based on the CNN algorithm, utilizing set of neurons to convolve on a given sample images to extract relevant features from them. This is demonstrated through validating the accuracy of the detection along with the objective to reduce the loss while the network is learning the details.

As part of this project, we will be demonstrating the outcome with 4 different model architecture along with their outcome in each of the model and provide the commentary for each of the models developed.

3 Project Objective

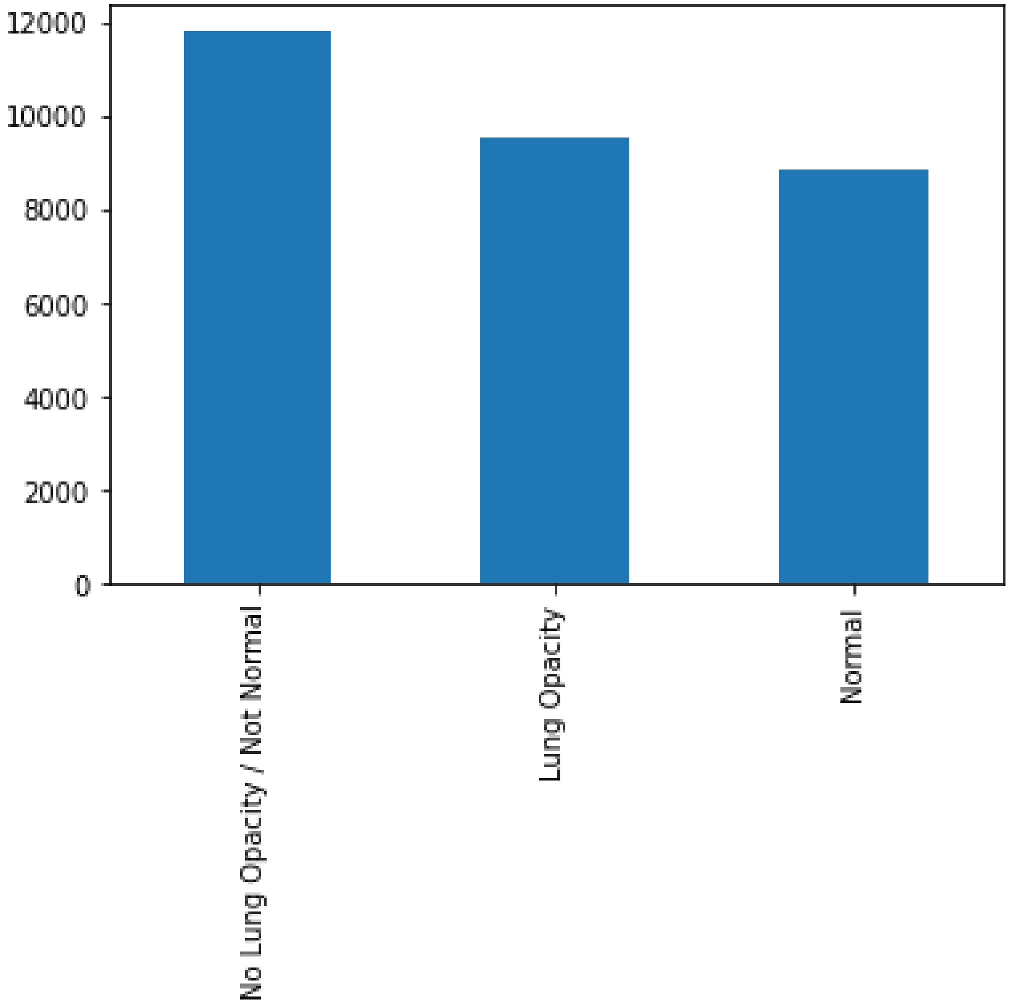
The objective of this capstone project is to build a Pneumonia detection system to locate the position of the inflammation in a Chest Radiography.

Based on the dataset provided, we will have to do the following steps to work towards building the final model and validate and results.

- Validate the images and respective bounding box coordinates.
- Extract all the features from the images and build the csv file for processing further.
- Perform Exploratory Data Analysis to validate all the data points and build insights
- Based on the project objective – isolate the dataset and accordingly images as well which are fully unique by removing the duplicate records in the dataset based on target variable.
- Build the model with different architectures and showcase the accuracy and showcase the right model to approach the problem description

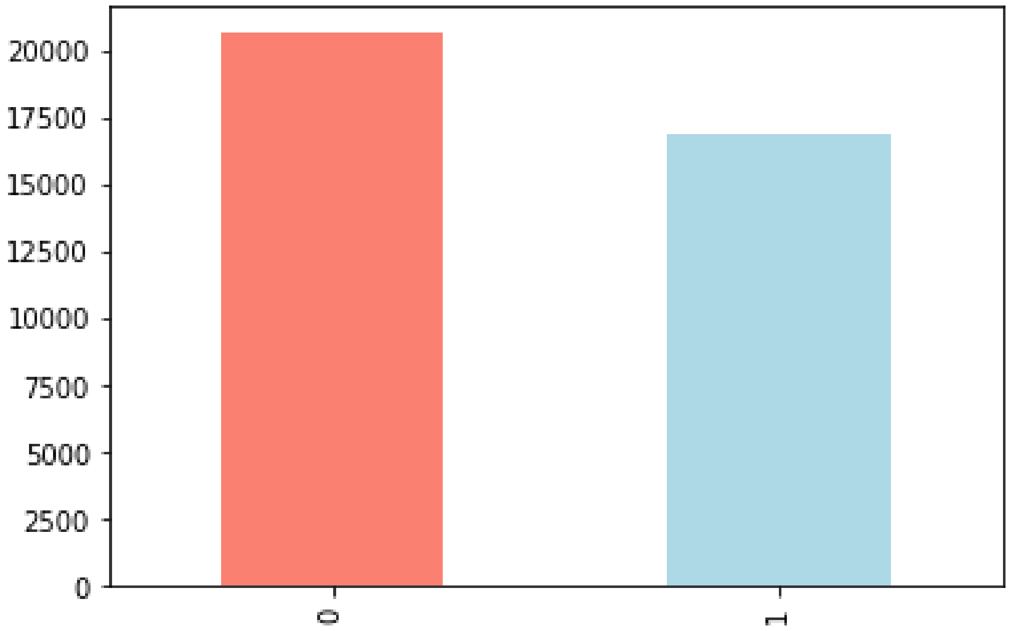
4 EDA Inference and Data pre-processing

1) Based on the sample dataset provided – we see that there are 3 class values present..



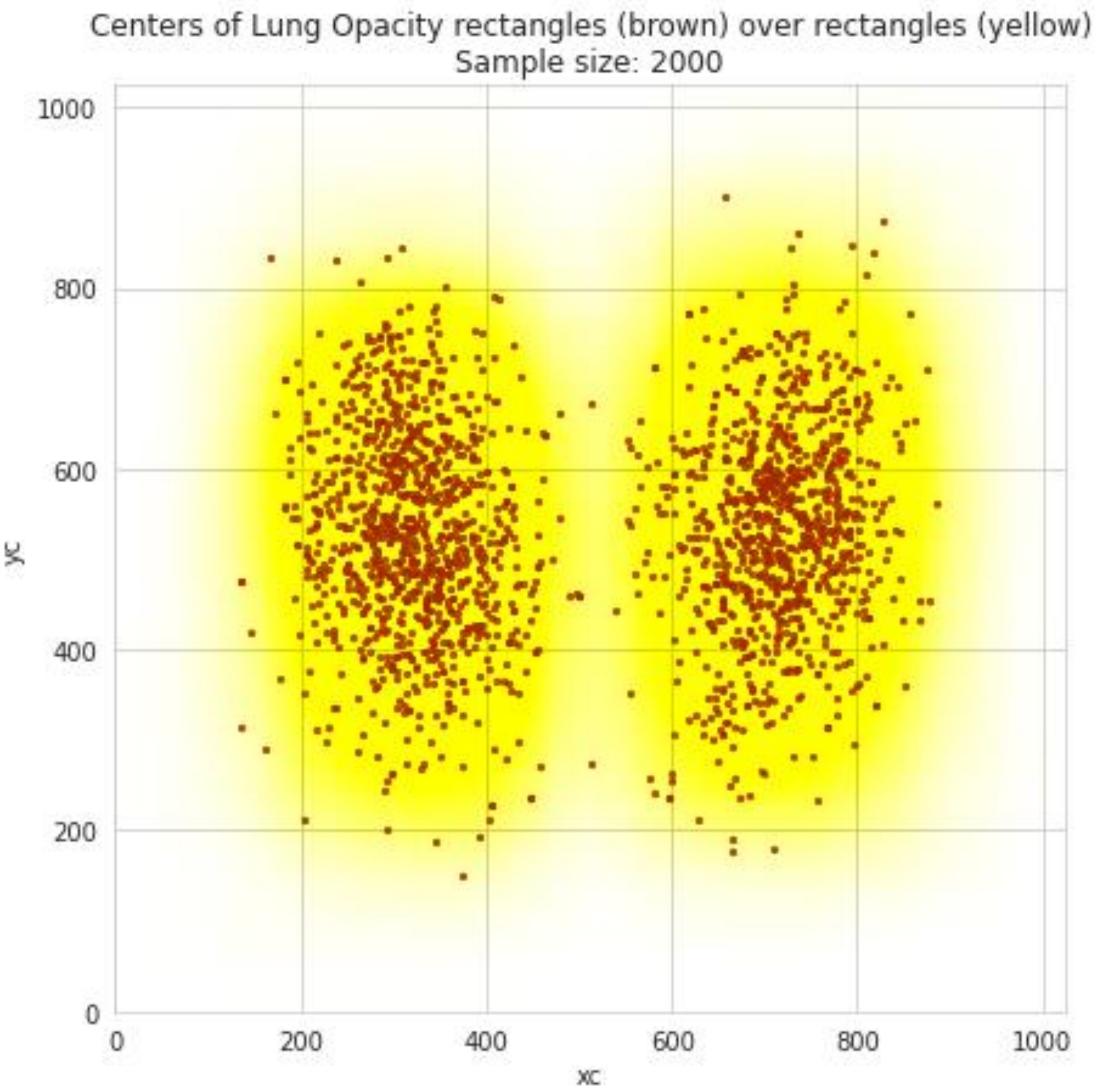
Our objective would be look at images having Lung Opacity and identify the right bounding box to locate the area of inflammation, hence we may need to club them into Lung Opacity and others as another single category

2) Post merging both the documents – class information with the labels document we see the following distribution on the "Target" variable



We observe there are more records not having lung opacity issues, which may lead to bias.

3) We observe the following heat map making the visualization where in the Target = 1



The scatter plot with brown dots shows the areas of inflammation highlighted which needs to be learnt to identify the issues.

4) Following are set of Metadata that we are able to notice from each of the chest X-ray's provided..

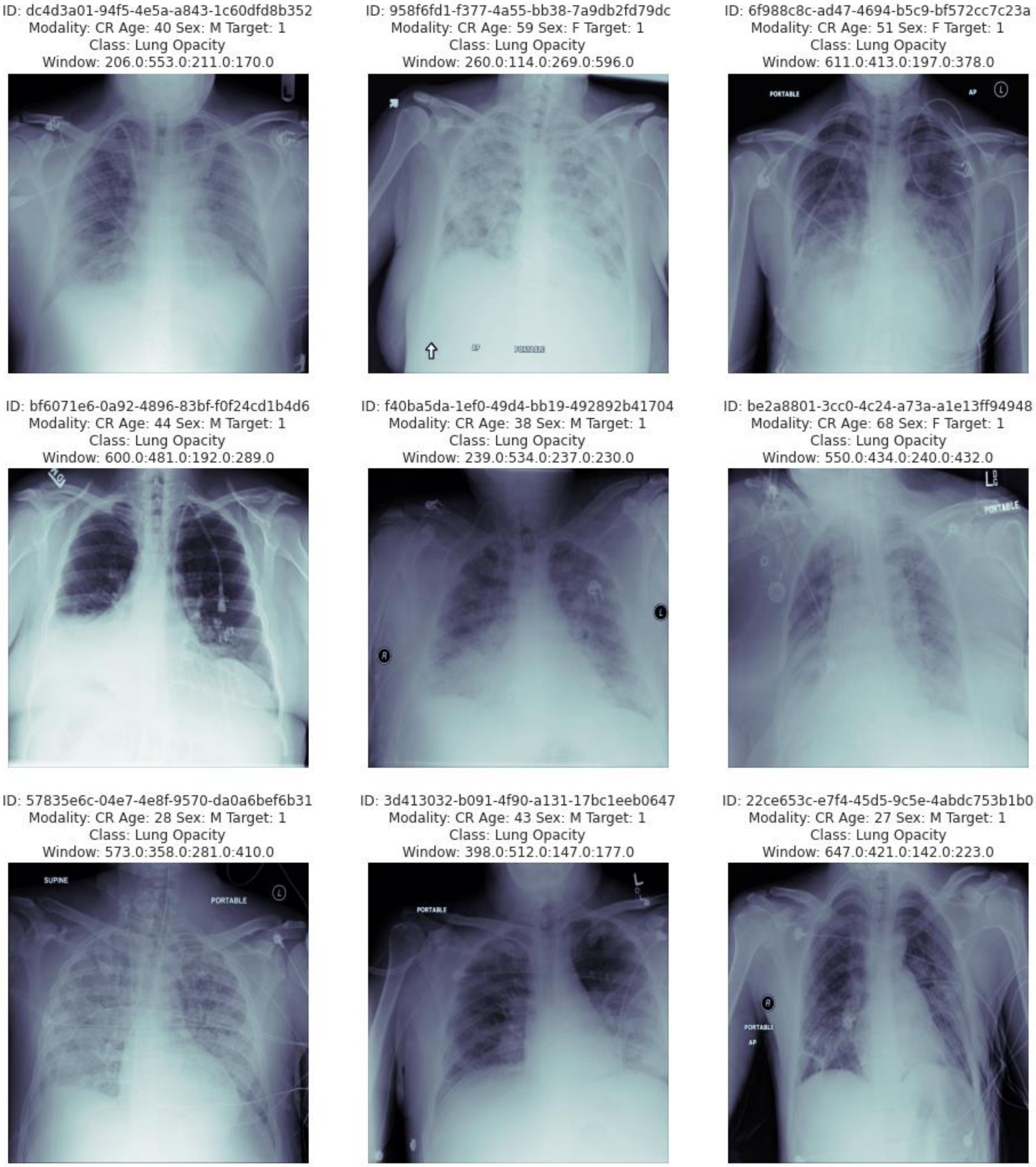
```
Dataset.file_meta -------------------------------
(0002, 0000) File Meta Information Group Length  UL: 202
(0002, 0001) File Meta Information Version       OB: b'\x00\x01'
(0002, 0002) Media Storage SOP Class UID         UI: Secondary Capture Image Storage
(0002, 0003) Media Storage SOP Instance 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
```

5) Of the above metadata – basis the observation of key fields names and its values, we decided to consider following metadata to form the CSV file for validating and processing further..

- 'patientId'
- 'Modality'
- 'PatientAge'
- 'PatientSex'
- 'BodyPartExamined'
- 'ViewPosition'
- 'ConversionType'
- 'Rows'
- 'Columns'
- 'PixelSpacing'

6) Processing the dicom images, we see the following

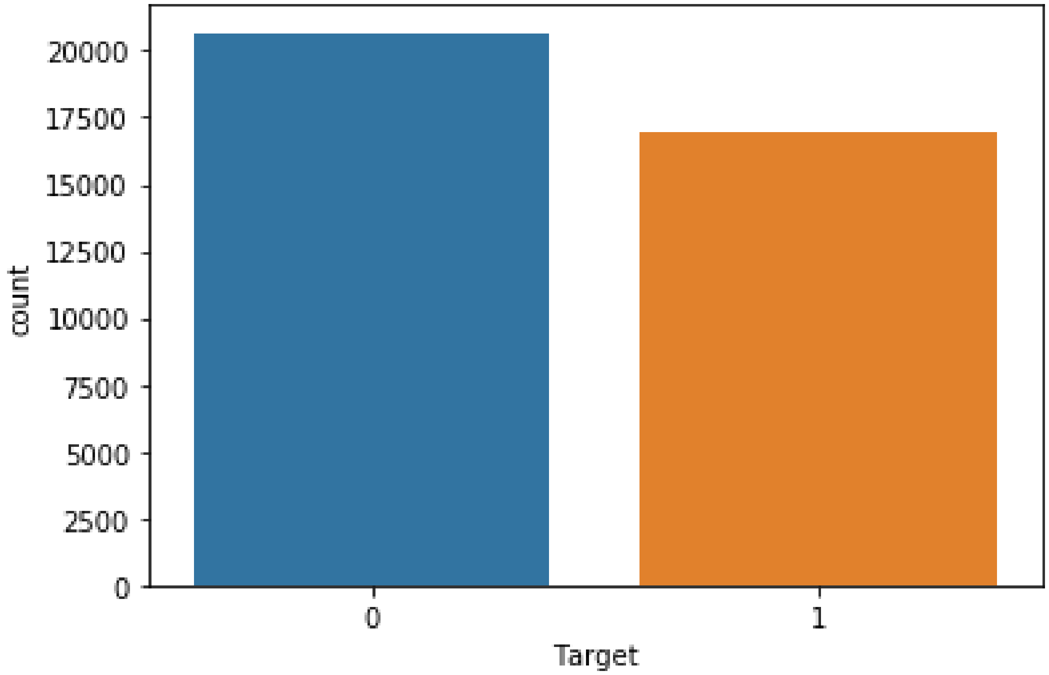

ID: dc4d3a01-94f5-4e5a-a843-1c60dfd8b352
Modality: CR Age: 40 Sex: M Target: 1
Class: Lung Opacity
Window: 206.0:553.0:211.0:170.0
ID: 958f6fd1-f377-4a55-bb38-7a9db2fd79dc
Modality: CR Age: 59 Sex: F Target: 1
Class: Lung Opacity
Window: 260.0:114.0:269.0:596.0
AP
PORTABLE
ID: 6f988c8c-ad47-4694-b5c9-bf572cc7c23a
Modality: CR Age: 51 Sex: F Target: 1
Class: Lung Opacity
Window: 611.0:413.0:197.0:378.0
PORTABLE
AP
ID: bf6071e6-0a92-4896-83bf-f0f24cd1b4d6
Modality: CR Age: 44 Sex: M Target: 1
Class: Lung Opacity
Window: 600.0:481.0:192.0:289.0
ID: f40ba5da-1ef0-49d4-bb19-492892b41704
Modality: CR Age: 38 Sex: M Target: 1
Class: Lung Opacity
Window: 239.0:534.0:237.0:230.0
ID: be2a8801-3cc0-4c24-a73a-a1e13ff94948
Modality: CR Age: 68 Sex: F Target: 1
Class: Lung Opacity
Window: 550.0:434.0:240.0:432.0
PORTABLE
ID: 57835e6c-04e7-4e8f-9570-da0a6bef6b31
Modality: CR Age: 28 Sex: M Target: 1
Class: Lung Opacity
Window: 573.0:358.0:281.0:410.0
SUPINE
PORTABLE
ID: 3d413032-b091-4f90-a131-17bc1eeb0647
Modality: CR Age: 43 Sex: M Target: 1
Class: Lung Opacity
Window: 398.0:512.0:147.0:177.0
PORTABLE
ID: 22ce653c-e7f4-45d5-9c5e-4abdc753b1b0
Modality: CR Age: 27 Sex: M Target: 1
Class: Lung Opacity
Window: 647.0:421.0:142.0:223.0
PORTABLE
AP

We observe all the images are 1024/1024 shape and it needs to be reshaped before feeding them into the model.

Based on the processed CSV file from the metadata – EDA was performed on this and below are the findings and also pre-processing steps taken to perform the data analysis

- EDA performed on the data taken out from each of parameters extracted from the images and clubbed against their patient ID's to validate them for its accuracy, impact based on availability of information overall. Below are some of the key findings.
- There are totally 37,629 records available in the dataset where in 16 fields were extracted for validation

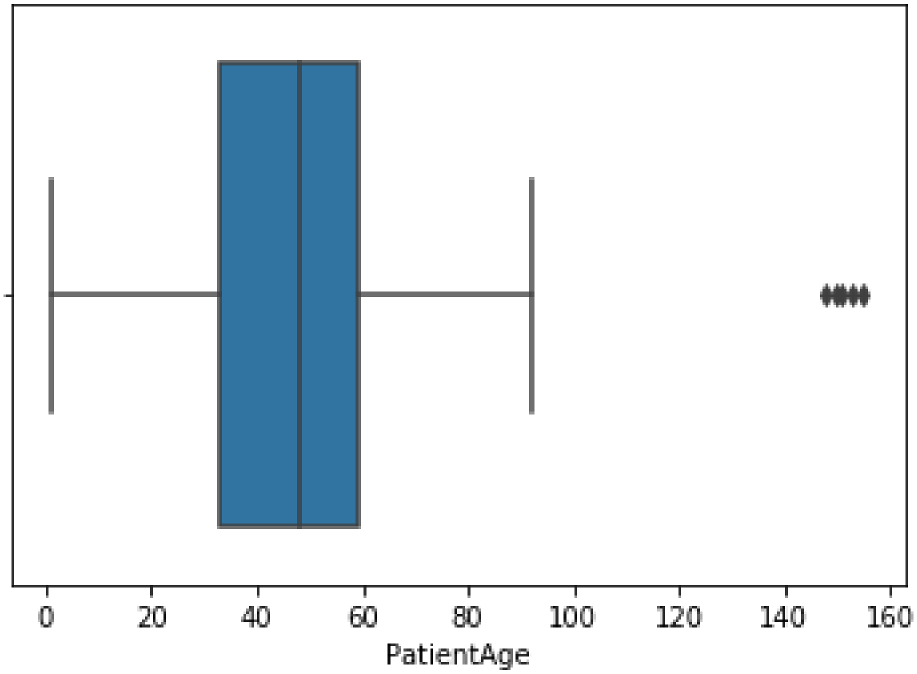
- For Pnuemonia detection - "Target" Field provides the classification. 0 : No Lung Opacity, 1 : Lung Opacity



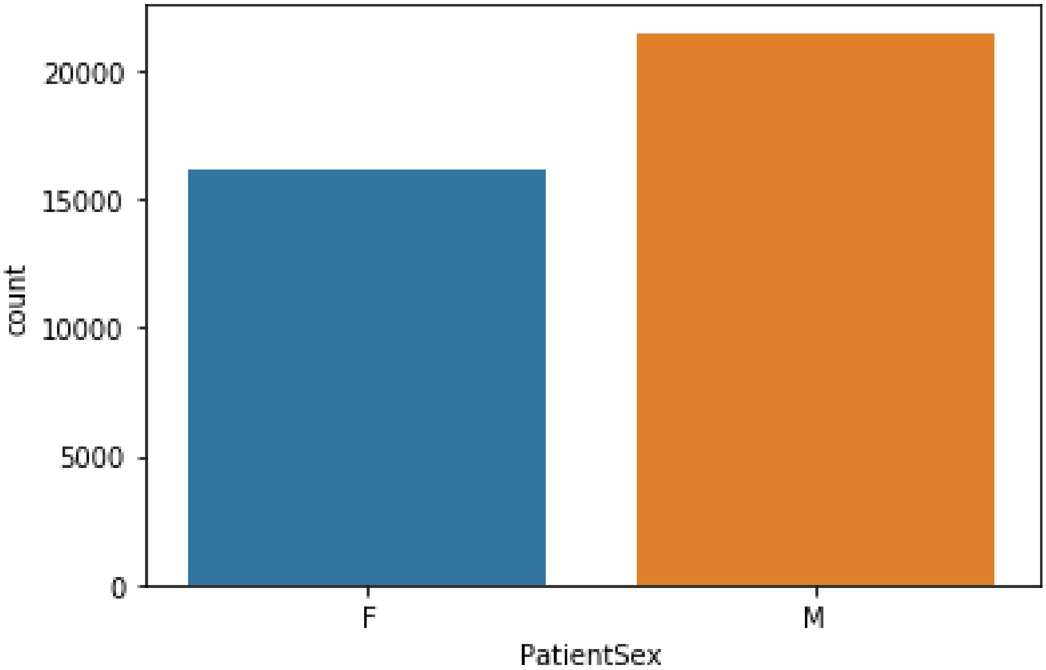
- "class" Field provides 3 groups. However when its seen against Target classification and respective coordinates availability - its doesn't seems to isolate between Normal and Not Normal cases. There are may be other problems, for the purpose of this exercise this observation will be ignored

- "Target" has 20,672 records/images which are Not having Lung Opacity and 16,957 having lung Opacity. We observe 55% of the images doesn't have lung opacity and only 45% having Lung Opacity - This may create imbalance in prediction tending towards not having Lung Opacity - Need to observe this furhter for duplicate records and plan for data augmentation

- "PatientAge" - We observe 5 records having ages above 100 - which should be dropped



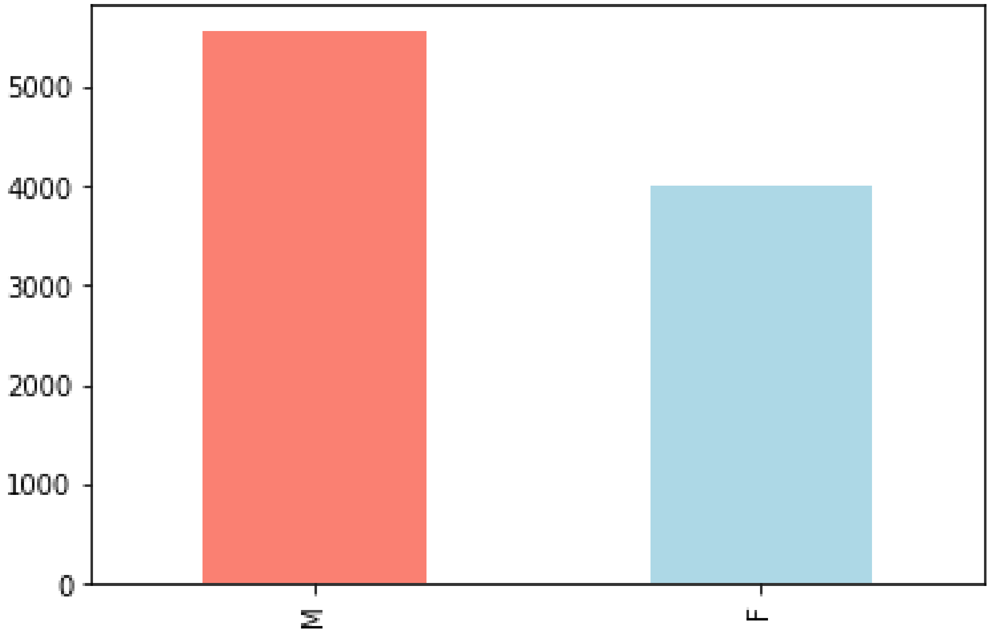
- "PatientSex" - We observe 55% of male records in the total count. This may not be a factor for recognition hence decided to continue even though there is imbalance..



- We observe 9555 records are unique - which needs to be considered for model building
- For processing further – we have taken the ROWS having Target = 1, we found that there are 16K records
- Post processing for Duplicate records – we see that the record count comes to 9555 records..

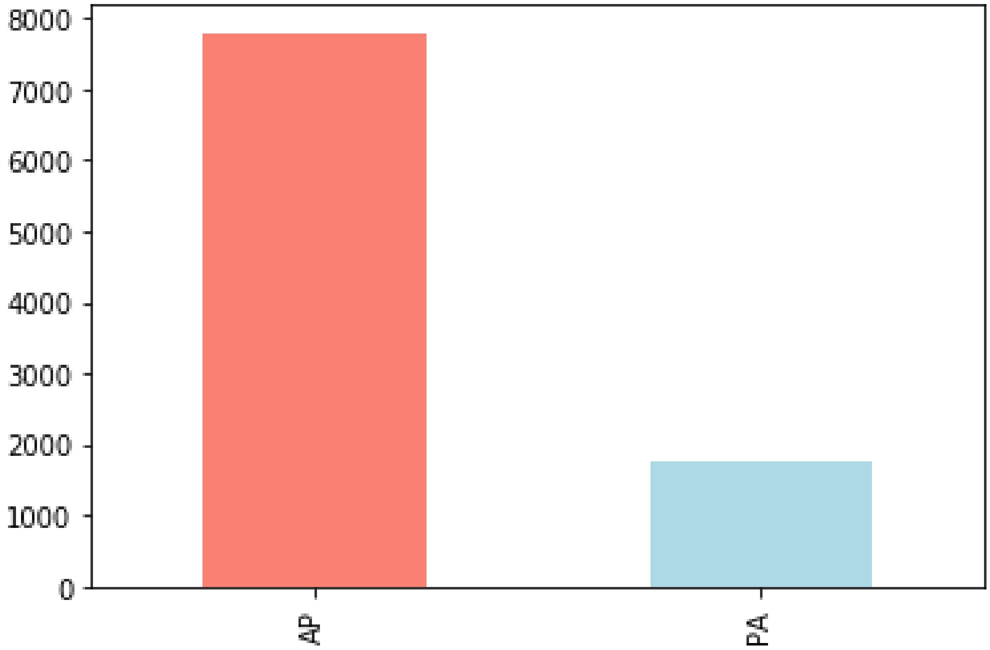
```
1    9555
Name: Target, dtype: int64
```

- Of the 9555 records – below is the distribution on M / F



Since Male records are more – this may create data imbalance even though it may not be crucial at this time this needs to be validated.

- With the column "ViewPosition" we notice the below findings..



With respect to Column "ViewPosition" correlated with the Target value of 1 - There is high imbalance that we observe with the above information having AP (Anterior-Posterior). One way to interpret this target unbalance is that patients that are imaged in an AP position are those that are more ill, and therefore more likely to have contracted pneumonia. Note that the absolute split between AP and PA images is about 50-50, so the above consideration is extremely significant

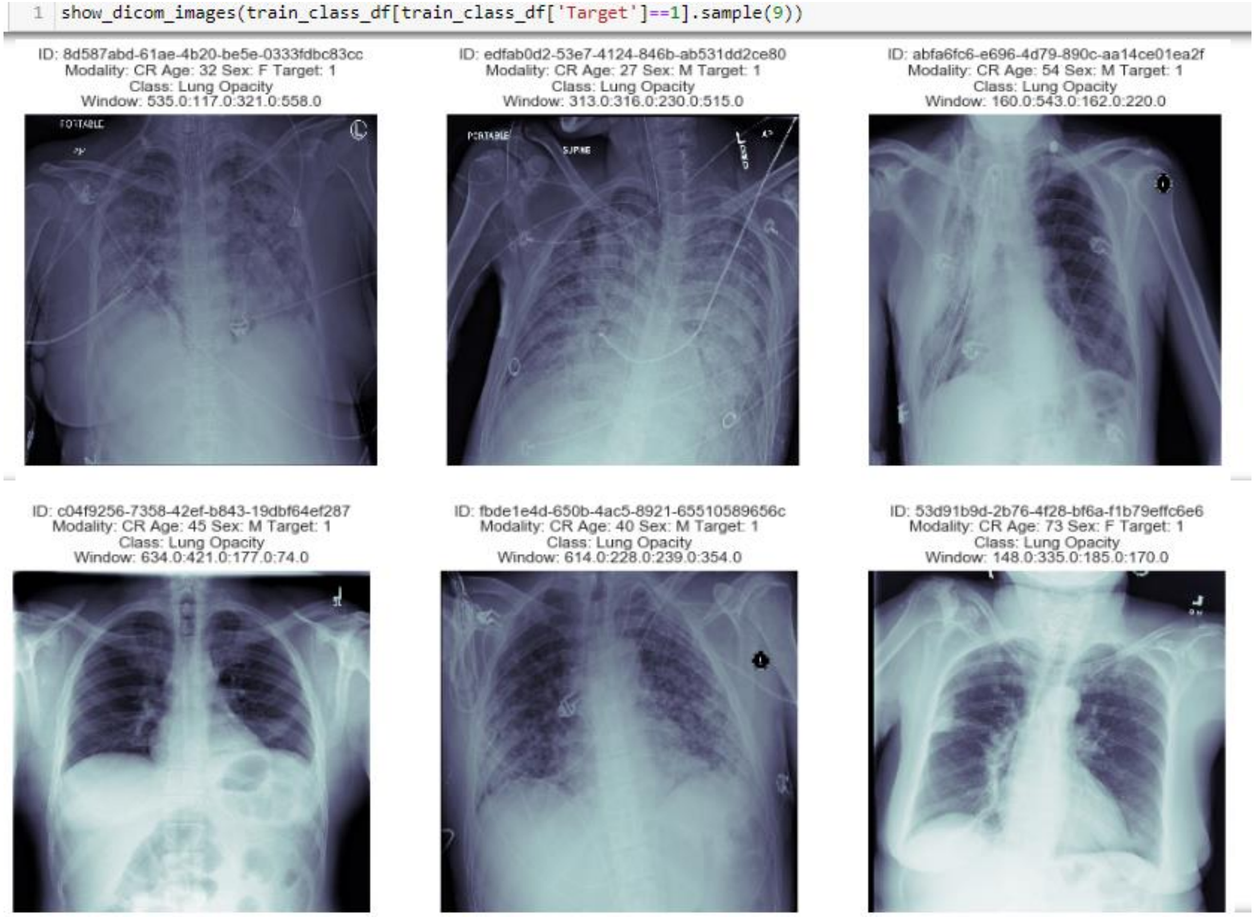
- Validating the target with age group – we observe the below pattern.

```
        PatientAge  count
0             0-10    219
1            11-20    602
2            21-30   1308
3            31-40   1591
4            41-50   1675
5            51-60   2226
6            61-70   1313
7            71-80    518
8            81-90    100
9           91-100      3
10  more than 100      0
```

Between age from 21 to 70 – The problem is highly detected and specifically 51 to 60 has more occurrence. We also observe some of th e outlier which may needs to be dropped.

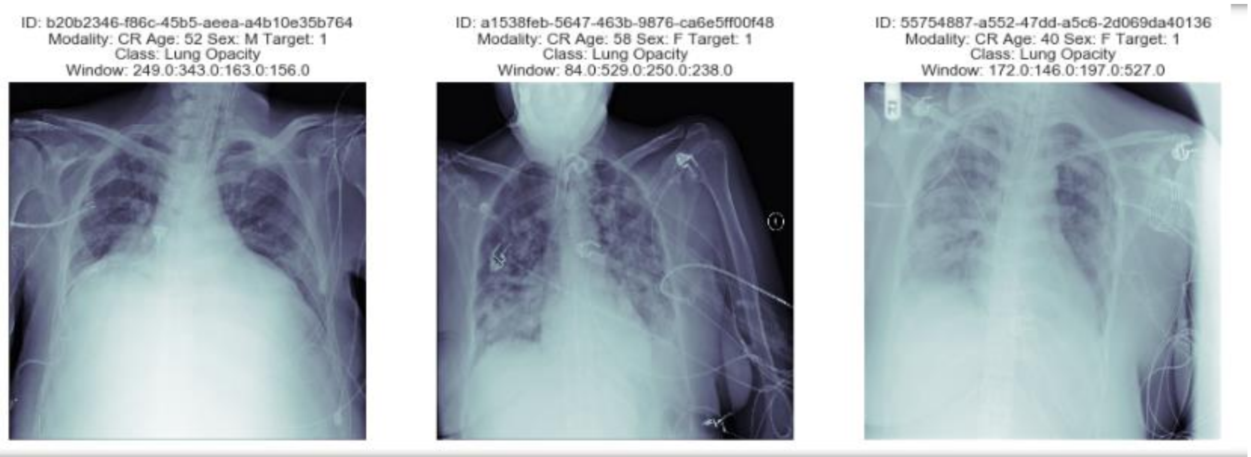
- We were able to extract and visualize the diacom images provided and visual representation are detailed below.

RAW Images:



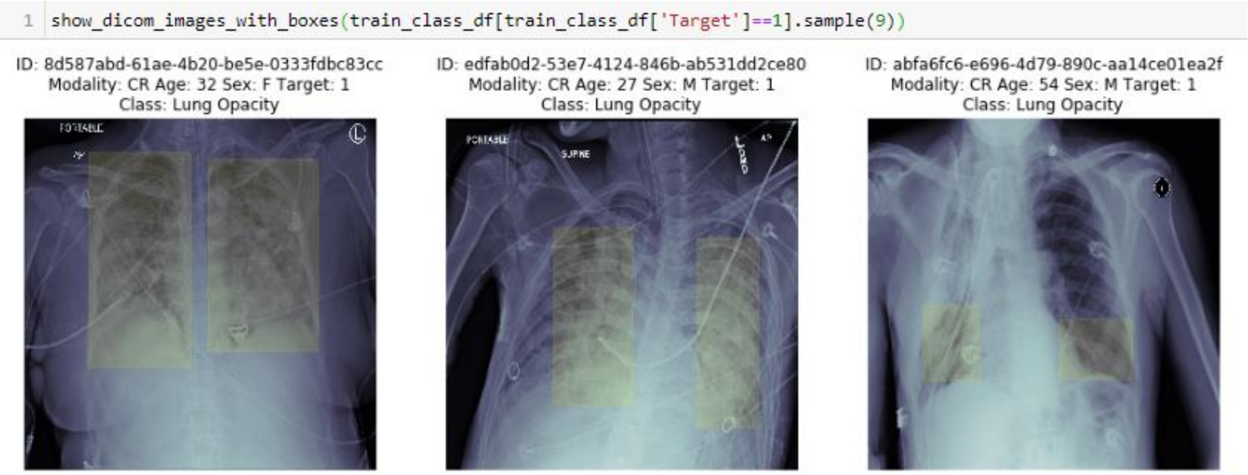
```
show_dicom_images(train_class_df[train_class_df['Target']==1].sample(9))
```

```
ID: 8d587abd-61ae-4b20-be5e-0333fdbc83cc
Modality: CR Age: 32 Sex: F Target: 1
Class: Lung Opacity
Window: 535.0:117.0:321.0:558.0

ID: edfab0d2-53e7-4124-846b-ab531dd2ce80
Modality: CR Age: 27 Sex: M Target: 1
Class: Lung Opacity
Window: 313.0:316.0:230.0:515.0

ID: abfa6fc6-e696-4d79-890c-aa14ce01ea2f
Modality: CR Age: 54 Sex: M Target: 1
Class: Lung Opacity
Window: 160.0:543.0:162.0:220.0

ID: c04f9256-7358-42ef-b843-19dbf64ef287
Modality: CR Age: 45 Sex: M Target: 1
Class: Lung Opacity
Window: 634.0:421.0:177.0:74.0

ID: fbde1e4d-650b-4ac5-8921-65510589656c
Modality: CR Age: 40 Sex: M Target: 1
Class: Lung Opacity
Window: 614.0:228.0:239.0:354.0

ID: 53d91b9d-2b76-4f28-bf6a-f1b79effc6e6
Modality: CR Age: 73 Sex: F Target: 1
Class: Lung Opacity
Window: 148.0:335.0:185.0:170.0
```



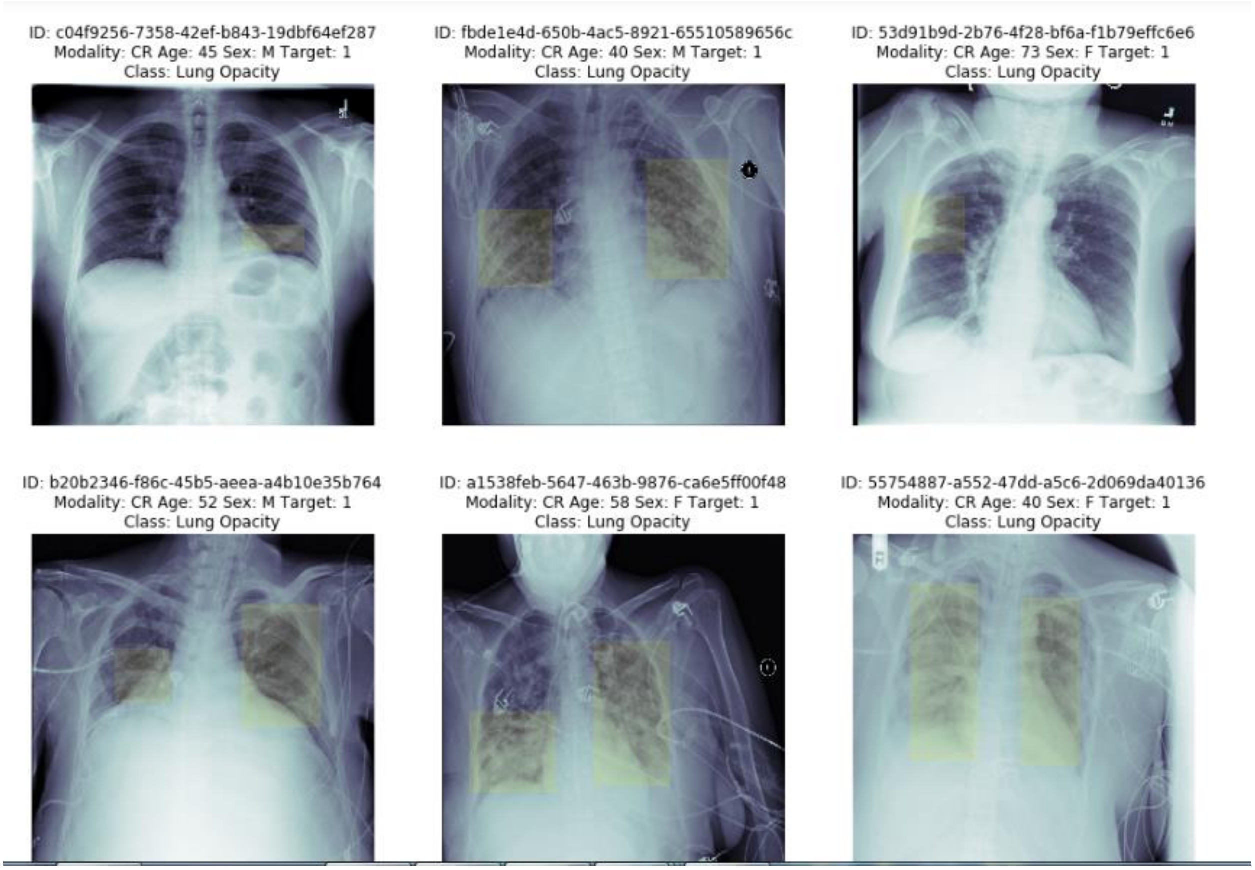
```
ID: b20b2346-f86c-45b5-aeea-a4b10e35b764
Modality: CR Age: 52 Sex: M Target: 1
Class: Lung Opacity
Window: 249.0:343.0:163.0:156.0

ID: a1538feb-5647-463b-9876-ca6e5ff00f48
Modality: CR Age: 58 Sex: F Target: 1
Class: Lung Opacity
Window: 84.0:529.0:250.0:238.0

ID: 55754887-a552-47dd-a5c6-2d069da40136
Modality: CR Age: 40 Sex: F Target: 1
Class: Lung Opacity
Window: 172.0:146.0:197.0:527.0
```

RAW Images along with its bounding boxes (for the same samples above):



```
show_dicom_images_with_boxes(train_class_df[train_class_df['Target']==1].sample(9))
```

```
ID: 8d587abd-61ae-4b20-be5e-0333fdbc83cc
Modality: CR Age: 32 Sex: F Target: 1
Class: Lung Opacity

ID: edfab0d2-53e7-4124-846b-ab531dd2ce80
Modality: CR Age: 27 Sex: M Target: 1
Class: Lung Opacity

ID: abfa6fc6-e696-4d79-890c-aa14ce01ea2f
Modality: CR Age: 54 Sex: M Target: 1
Class: Lung Opacity
```


ID: c04f9256-7358-42ef-b843-19dbf64ef287
Modality: CR Age: 45 Sex: M Target: 1
Class: Lung Opacity
ID: fbde1e4d-650b-4ac5-8921-65510589656c
Modality: CR Age: 40 Sex: M Target: 1
Class: Lung Opacity
ID: 53d91b9d-2b76-4f28-bf6a-f1b79effc6e6
Modality: CR Age: 73 Sex: F Target: 1
Class: Lung Opacity
ID: b20b2346-f86c-45b5-aeea-a4b10e35b764
Modality: CR Age: 52 Sex: M Target: 1
Class: Lung Opacity
ID: a1538feb-5647-463b-9876-ca6e5ff00f48
Modality: CR Age: 58 Sex: F Target: 1
Class: Lung Opacity
ID: 55754887-a552-47dd-a5c6-2d069da40136
Modality: CR Age: 40 Sex: F Target: 1
Class: Lung Opacity

- Post this process as part of pre-processing we also associated the proper labelling for the respective images identified for processing further, we also validated this against the total records of 9555 to make sure that images along with associated labels are properly matched.

5 Current Approach on Model building

Deep neural networks models have conventionally been designed and experiments were performed upon them by human experts in a continuing trial and error method. This process demands enormous time, knowhow and resources. To overcome this problem, a novel but simple model is introduced to automatically through CNN models.

We also noted that constructing and training a complex deep learning model from scratch is mostly infeasible due to the lack of hardware infrastructure. Therefore, we decided to exploits the idea of transfer learning which is the improvement of learning in a new prediction task through the transfer of knowledge from a related prediction task that has already been learned. This will improve the current computer vision methods based on the use of deep learning to diagnose X-rays images more effectively. By utilizing convolutional neural networks re-trained with our obtained data, we would like to experiment them to achieve the greater classification accuracy.

We decided to develop the following models as part of our experiment and present our findings with details.

- MobileNet
- YOLO
- SSD
- Mask R-CNN

Following section details the MobileNet Model implementation details:

As we begin our MobileNet implementation by adopting to the Transfer learning operation, we selected the model architecture through Tensor Hub which can be referenced in this URL https://tfhub.dev/google/imagenet/mobilenet_v2_140_224/feature_vector/4

TensorFlow Hub 2.0 allows the easier way to select the appropriate model based on our requirement and provides us the directional inputs on using that further in our model building exercise. We wanted to attempt this as a new feature to explore during this project cycle and selected Version 4 for our development purposes.

Some of the key steps / decisions taken as part of model development is detailed as below.

Pre-processing data:

1) Pre-process the image by converting them from Dicom to JPG images
2) Convert the images into Tensors
3) Resize the image from 1024/1024 to 224 / 224 as expected by MobileNet architecture

The above steps are defined as functions so that it returns the appropriate images.

Turning our data into batches:

To make the model development efficient and faster to visualise we set the batch size to 32. Say, we are process 1000+ images they all may not fit into the memory, hence decided to set this to 32 per batch and this can be modified based on trials.

Also, in order to use TensorFlow effectively, we need our data in the form for Tensor Tuples which looks like (image, lable). For which we have defined a function “get_image_label” which does all the steps of image conversion, processing, reshaping and returns the images and corresponding labels.

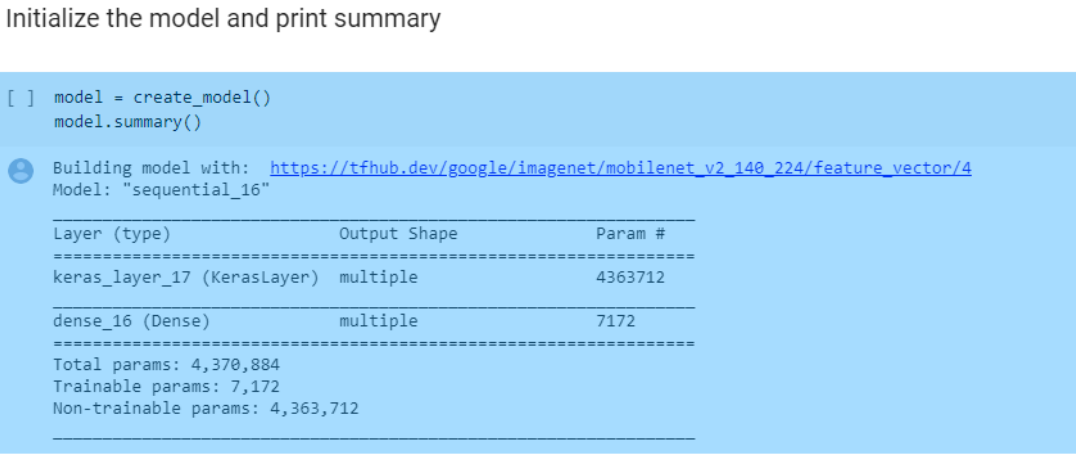
Building the model:

Below steps were followed to build the model..

```
# Setup input shape to the model
INPUT_SHAPE = [None, IMG_SIZE, IMG_SIZE, 3] # batch, height, widhth, color channelsl

# Setup output shape of our model
OUTPUT_SHAPE = 4

# Setup model URL from TensorFlow Hub
MODEL_URL = "https://tfhub.dev/google/imagenet/mobilenet_v2_140_224/feature_vector/4"# @param
["https://tfhub.dev/google/imagenet/mobilenet_v2_140_224/feature_vector/4"]
```

NOTE: Model URL was provided to direct to the mobilenet architecture that we decided to use going forward..

The model summary is defined below:



Initialize the model and print summary

```
[ ] model = create_model()
    model.summary()

    Building model with:  https://tfhub.dev/google/imagenet/mobilenet_v2_140_224/feature_vector/4
    Model: "sequential_16"
    _________________________________________________________________
    Layer (type)                 Output Shape              Param #   
    =================================================================
    keras_layer_17 (KerasLayer)  multiple                  4363712   
    _________________________________________________________________
    dense_16 (Dense)             multiple                  7172      
    =================================================================
    Total params: 4,370,884
    Trainable params: 7,172
    Non-trainable params: 4,363,712
    _________________________________________________________________
```

Defining callback:

We have setup the Tensorboard feature to define the call back and below steps are taken to address them.

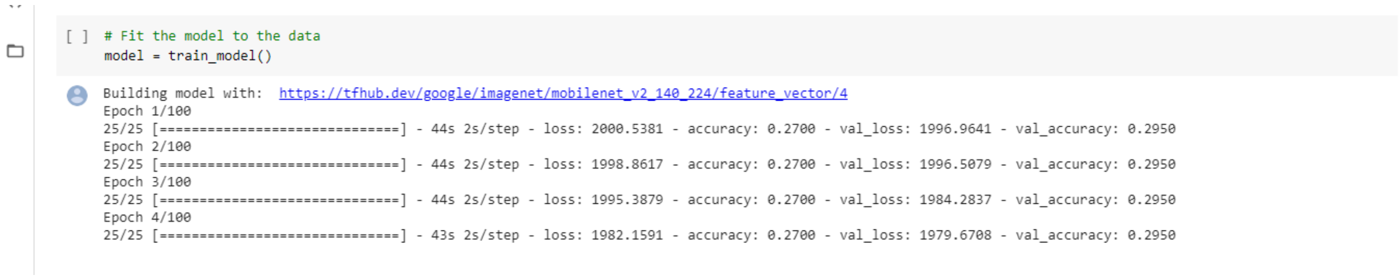
- Load the tensorboard notebook extension
- Create tensortboard callback – by which we will able to save logs to a directory and pass it to our model during fit
- To visualize our training logs we will use %tensorboard magic function

Defining Early stopping:

We have also defined early stopping to ensure we stop the process if there is a overfit scenario and/or loss value is not improving much. This way we will able to validate and re-run the model by changing right hyper parameter for better accuracy outcome.

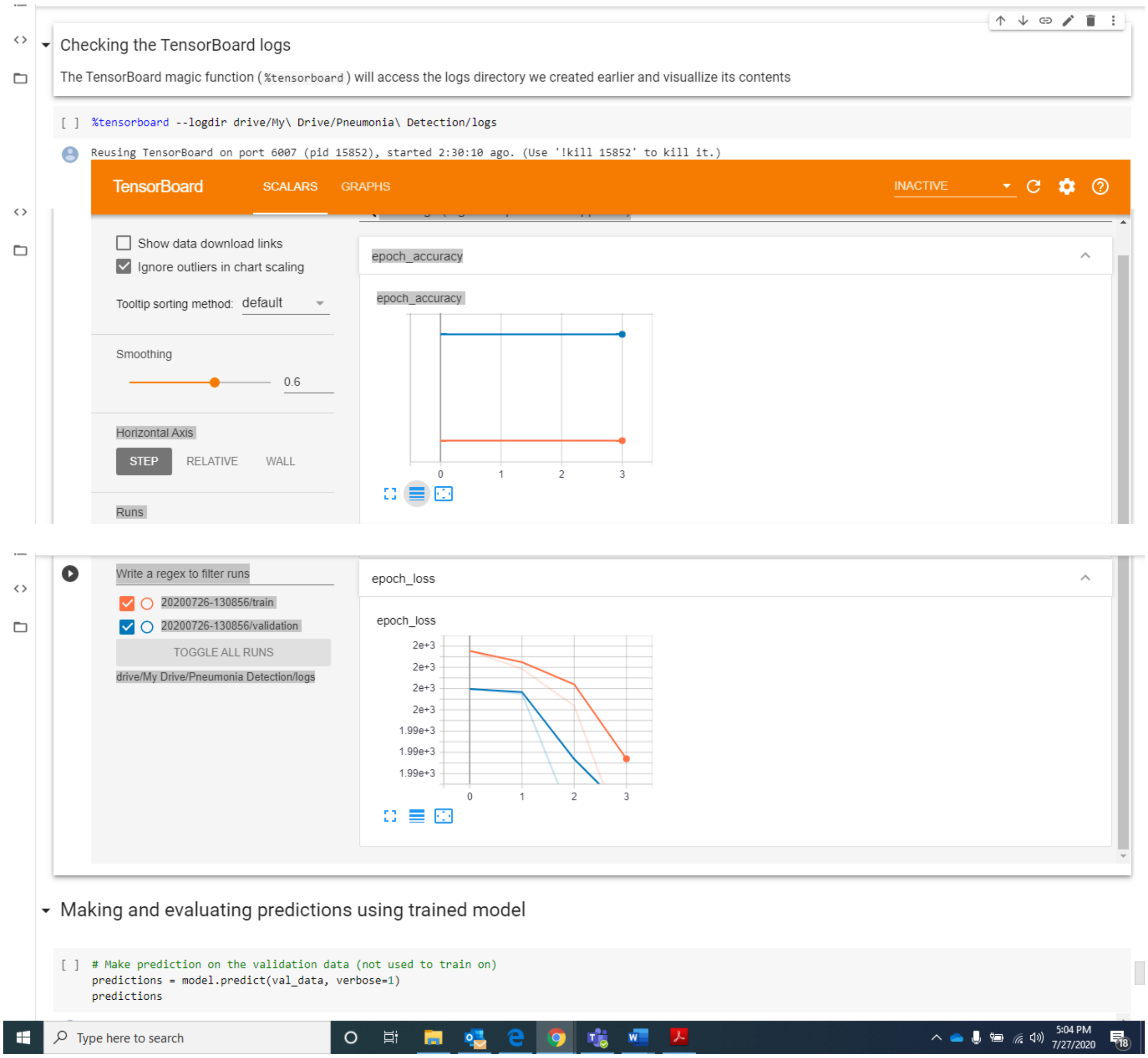
Running the model:

We have set the epochs to 100 to start with and called the fit function to execute the model.. Below is our INTERIM Observation..



```
[ ] # Fit the model to the data
    model = train_model()

    Building model with:  https://tfhub.dev/google/imagenet/mobilenet_v2_140_224/feature_vector/4
    Epoch 1/100
    25/25 [==============================] - 44s 2s/step - loss: 2000.5381 - accuracy: 0.2700 - val_loss: 1996.9641 - val_accuracy: 0.2950
    Epoch 2/100
    25/25 [==============================] - 44s 2s/step - loss: 1998.8617 - accuracy: 0.2700 - val_loss: 1996.5079 - val_accuracy: 0.2950
    Epoch 3/100
    25/25 [==============================] - 44s 2s/step - loss: 1995.3879 - accuracy: 0.2700 - val_loss: 1984.2837 - val_accuracy: 0.2950
    Epoch 4/100
    25/25 [==============================] - 43s 2s/step - loss: 1982.1591 - accuracy: 0.2700 - val_loss: 1979.6708 - val_accuracy: 0.2950
```

Validating the logs:



Checking the TensorBoard logs

The TensorBoard magic function (`%tensorboard`) will access the logs directory we created earlier and visuallize its contents

```
[ ] %tensorboard --logdir drive/My\ Drive/Pneumonia\ Detection/logs
```

```
Reusing TensorBoard on port 6007 (pid 15852), started 2:30:10 ago. (Use '!kill 15852' to kill it.)
```

□ Show data download links

☑ Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing 0.6

Horizontal Axis: STEP

Runs

☑ 20200726-130856/train

☑ 20200726-130856/validation

drive/My Drive/Pneumonia Detection/logs

Making and evaluating predictions using trained model

```
[ ] # Make prediction on the validation data (not used to train on)
    predictions = model.predict(val_data, verbose=1)
    predictions
```

We did get into some of the challenges in progressing with the image translation and hence we observe high loss and very low accuracy to start with. We will continue further to address the challenges with the above approach to drive a very high accuracy when we finally conclude on this.

6 Next steps and improvements.

As we continue further, our objective is to build the model with 4 different architectures by adopting the transfer learning techniques and we will continue to try on YOLO, SSD and Mask R-CNN along with Mobilenet..

Based on accuracy level and understanding the model behaviour further for each of them, In order to tune these models further – we will follow the below steps.

- Adding a layer in addition to what the model provides
- Validating the hyperparameters based on the results to drive the improvement in accuracy
- Trying different loss function and optimizers to see where things can be improved.