

PNEUMONIA DETECTION Assignment

Interim Report



Team:

|  |  |
| --- | --- |
| **Name** | **E-mail** |
| Arup Kumar Mohanty | Arup.kumar1988@gmail.com |
| Mohit Gupta | Mohitgupta.mait@gmail.com |
| Ved Verma | Ved.verma.273306@gmail.com |
| Venkatesh Doddihal | Ds.venky1975@gmail.com |

Table of Contents

[**1.** **Summary of the problem statement, Data and findings** 4](#_Toc103337546)

[1.1. Problem Statement 4](#_Toc103337547)

[1.2. Project Objectives: 4](#_Toc103337548)

[**2.** **EDA** 5](#_Toc103337549)

[2.1. Approach: 5](#_Toc103337550)

[2.2. Analysis: 5](#_Toc103337551)

[2.3 Visualization: 8](#_Toc103337552)

[2.4 Summary: 14](#_Toc103337553)

[**3.** **Pre-Processing** 14](#_Toc103337554)

[3.1 Pre-processing Methods 14](#_Toc103337555)

[3.2 Pre-processing Applied 15](#_Toc103337556)

[**4.** **Model and Model Building** 15](#_Toc103337557)

[4.1 Models 16](#_Toc103337560)

[4.1.1 DenseNet-121 16](#_Toc103337561)

[4.1.2 CheXNet 18](#_Toc103337562)

[**4.2** **Evaluation Metrics** 18](#_Toc103337563)

[**4.3** **Models Applied** 19](#_Toc103337564)

[4.3.1 DenseNet-121 19](#_Toc103337568)

[4.3.2 DenseSet-121 with CheXNet weights 23](#_Toc103337569)

[4.4 Loss/Accuracy Graphs 27](#_Toc103337570)

[**5** **How to Improve Model Performance** 28](#_Toc103337571)

# **Summary of the problem statement, Data and findings**

## Problem Statement

Pneumonia is an infection in the lung, which requires review of a chest radiograph by highly trained specialists. Pneumonia shows up in a chest radiograph as an area of opacity. However, diagnosis of it can be complicated and much time and effort is spent by specialists in reviewing them. Chest radiograph is the most common performed diagnostic imaging study. Due to the high volume of chest radiography, it is very time consuming and intensive for the radiologists to review each image manually. As such, an automated solution is ideal to locate the position of inflammation in an image. By having such an automated pneumonia screening system, this can assist physicians to make better clinical decisions or even replace human judgement in this area.

## Project Objectives:

* To build a deep learning a pneumonia detection system, to locate the position of inflammation in an image.
* Use TensorFlow/Keras as the framework for building the model
* Read Medical images are stored in a special format called DICOM files (\*.dcm).
  1. **Data & Findings :**
* Details about the data and dataset files are given in below link,  
  <https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/data>
* The first step would be to examine the data available for this. The data is given in a zip file “rsna-pneumonia-detection-challenge.zip”, which contains the following items:
* A folder “stage\_2\_train\_images”: This folder contains all the training dataset chest radiograph DICOM images.
* A csv file “stage\_2\_train\_labels.csv”: This file contains the corresponding patientID images to the folder “stage\_2\_train\_images” and contains the bounding box of areas of pneumonia detected in each image along with a target label of 0 or 1 for pneumonia detected.
* A csv file “stage\_2\_detailed\_class\_info.csv”: This file contains the corresponding patientID images to the folder “stage\_2\_train\_images” and contains the target class labels of the images.
* A folder “stage\_2\_test\_images”: This folder contains all the test dataset chest radiograph DICOM images. We will not be using this set of images as they do not contain labels.
* A csv file “stage\_2\_sample\_submission.csv”: This file contains the corresponding patientID images to the folder “stage\_2\_test\_images”. We will not be using this set of file.

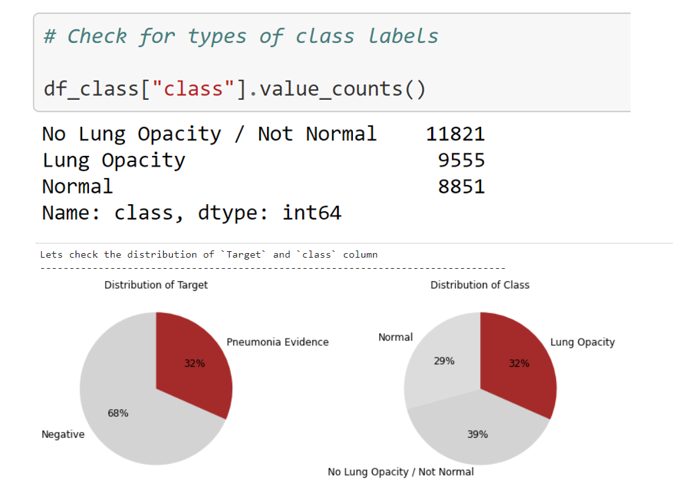
# **EDA**

## 2.1. Approach:

* To support the building of a neural network, the project will be done on **google Colab**.
* The first step is to unzip the zip file to open the above files to the google drive directory.
* Second is to verify the file format of the images provided, and they are all DICOM images in the “.dcm” file format. In order to open and read the DICOM images, we are using the **pydicom** library for that purpose. After that, the next step would be to inspect the csv files.

## 2.2. Analysis:

* Loading the “**stage\_2\_detailed\_class\_info.csv**” file into pandas dataframe, a quick glance reveals that it has only 2 columns:
* **patientId** – which refers to the patientId’s corresponding image name
* **class** – Target label of the patientId’s image



* As illustrate in above figure , there are 3 types of classes: **Normal, Lung Opacity and No Lung Opacity / Not Normal**. 11821(~39%)records belongs to No Lung Opacity / Not Normal, 32% accounts for Lung Opacity and roughly 29% marked as Normal. The primary concern of the project would be to detect images with Lung Opacity, and the others would be in the same group labelling.
* The Target distribution seems to be imbalance as 32% of the patients are having pneumonia evidences where as 68% are normal.
* Loading the “**stage\_2\_train\_labels.csv**” file into pandas dataframe, we can see that it has below fields:

**patientId** – which refers to the patientId’s corresponding image name

**x** - upper-left x coordinate of the bounding box

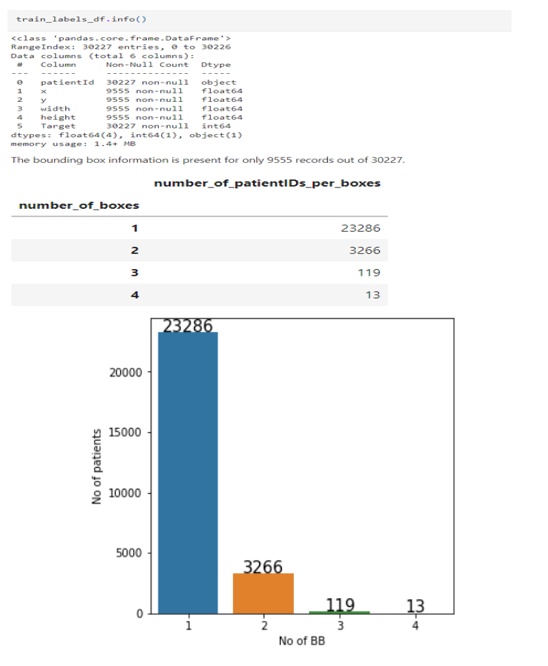
**y** - upper-left y coordinate of the bounding box

**width** – the width of the bounding box

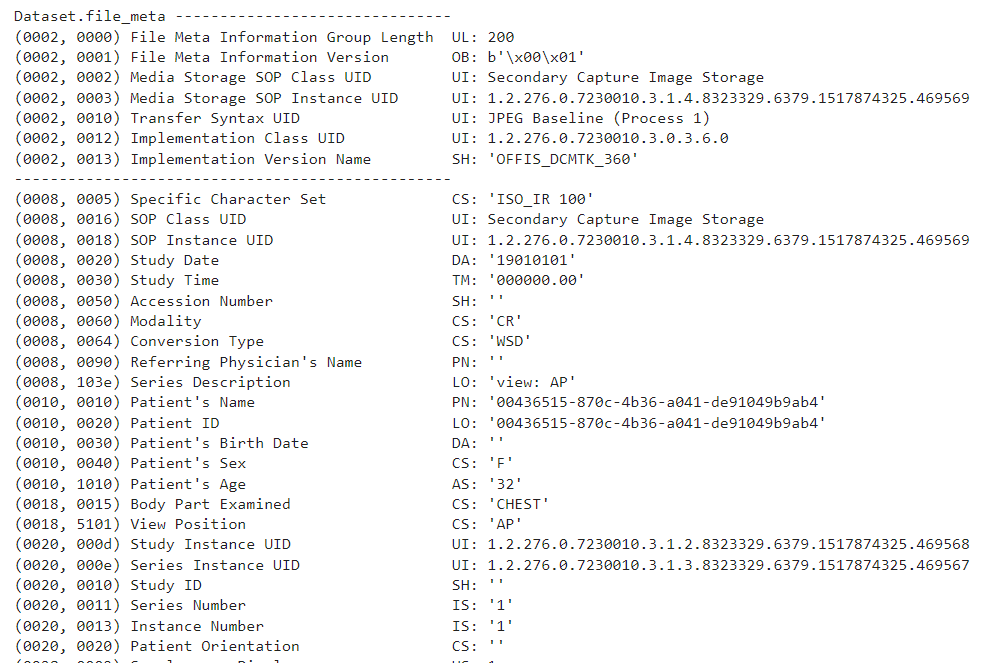
**height** – the height of the bounding box

**Target** – binary target indicating if this image has evidence of pneumonia

* There is a total of 30,227 entries, no missing values with 9,555 images have bounding boxes. This corresponds to the data available “stage\_2\_detailed\_class\_info.csv” file.



* There are multiple records for patients. Number of duplicates in patientID is 3,543.
* After merging both the csv files below are the observations.
* About 23,286 patientI
* ds (~87% of them) provided have 1 bounding boxes while 13 patients have 4 bounding boxes. The reason is because each row records a single bounding box area of pneumonia detected. However, in a patient image, it might be the case of several bounding boxes area of pneumonia detected.
* Chest examinations with Target = 1 i.e. ones with evidence of Pneumonia are associated with Lung Opacity class.
* Chest examinations with Target = 0 i.e. those with no definitive evidence of Pneumonia are either of Normal or No Lung Opacity / Not Normal class.
* The next step is to read the images in the file “**stage\_2\_train\_images**”. Images provided are stored in DICOM (.dcm) format which is an international standard to transmit, store, retrieve, print, process, and display medical imaging information. We will make use of pydicom package here to read the images.

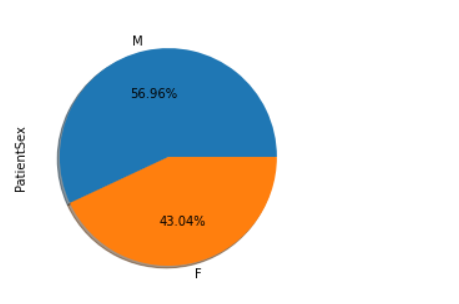


* From the above sample we can see that dicom file contains some of the information that can be used for further analysis such as **sex, age, body part examined , view position and modality**. Size of this image is 1024 x 1024 (rows x columns).
* To examine further we will merge the image features with the existing class data. This will help us understand distribution of age for those with evidence of lung opacity and those with no definite evidence of lung opacity.
* To understand distribution of male and female for those with evidence of lung opacity and those with no definite evidence of lung opacity
* Explore different view positions in the dataset
* Explore modality.
* We will pickle the file and do our analysis on the saved file.

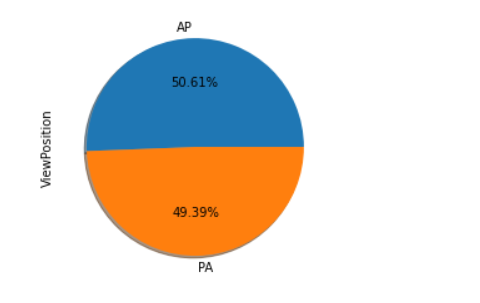
## 2.3 Visualization:

* As we proceed further we will use different visualization techniques like univariate, multivariate analysis to discover patterns and anomalies in the data.

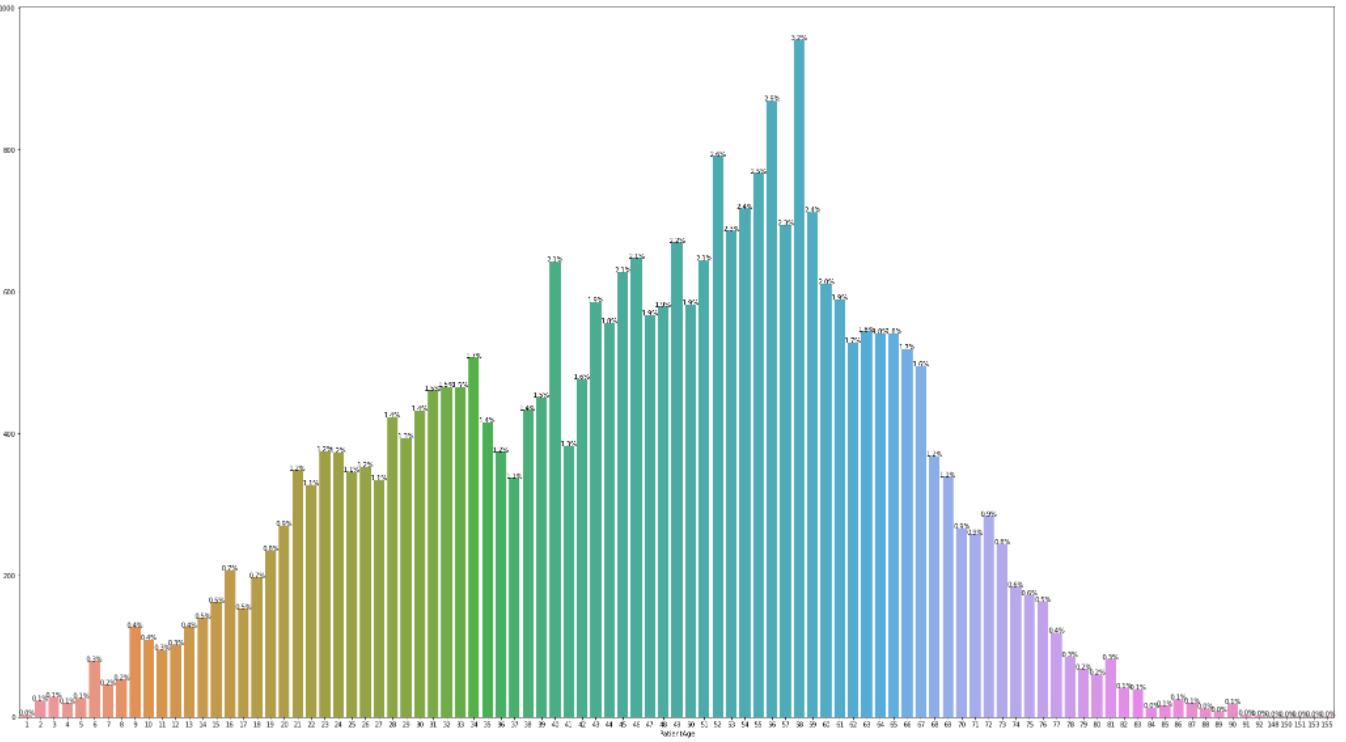
**Univariate Analysis:**



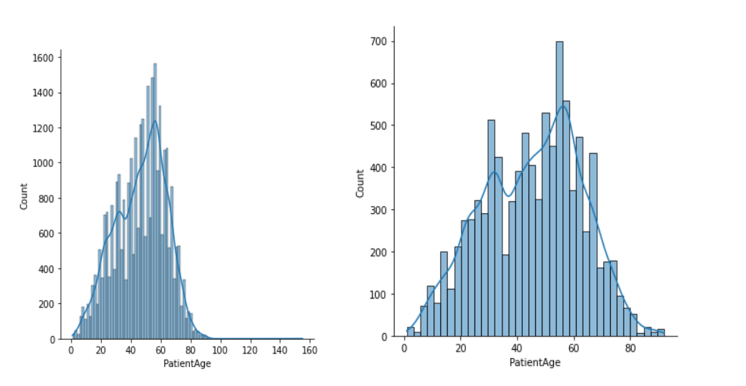
* As shown above, There are 56.96% male patients and roughly 43% female patients. There are also more no of male patients having pneumonia compared to females.



* We have two different view positions AP (Anterior/Posterior) and PA (Posterior/Anterior) in the training dataset.
* **Posterior/Anterior (PA)**: In PA, X-Ray beam hits the posterior (back) part of the chest before the anterior (front) part. While obtaining the image patient is asked to stand with their chest against the film.
* **Anterior/Posterior (AP)**: At times it's not possible for radiographers to acquire a PA chest X-ray. This is usually because the patient is too unwell to stand. AP projection images are of lower quality than PA images. Heart size is exaggerated (cardiothoracic ratio approximately 50%)
* As can be seen above in the chart , the view position attributes is almost equally distributed.



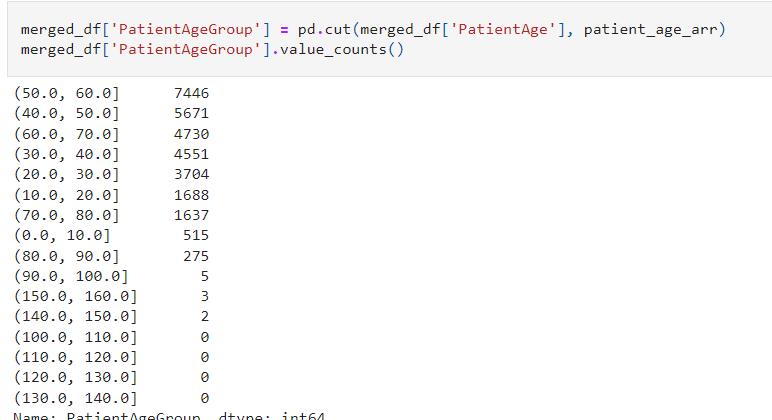
* The above graphs shows the distribution of age across all the patients. As per the data the maximum no of patients are falling within age group 40-60.
* There are also few patients where age > 100. These records seems to be erronous and we can safely remove them from our dataset and will keep the age

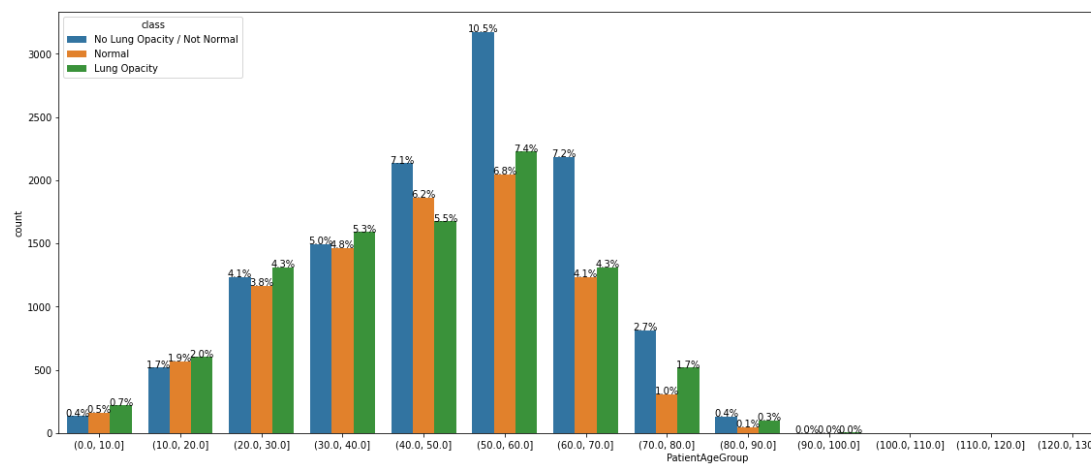


* As per the above histogram, the patient age is normally distributed with more volume of data lying between age group 40-60.
* The Age distribution for pnenumonia patient is slightly left skewed with more no of patient between age group 40-60 are having pneumonia.

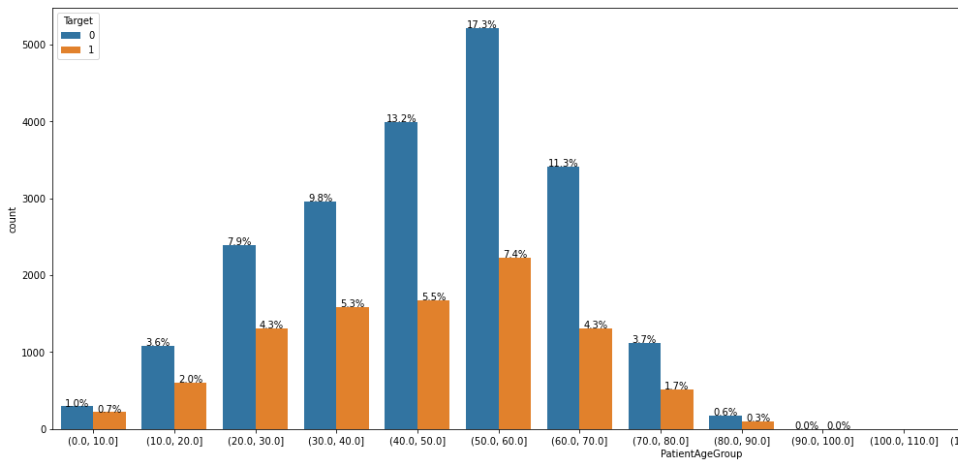
**Bivariate Analysis:**

* To have a better interpretation we will make use of binning concept as shown below to group the patient age and will further visualize on the transformed data.

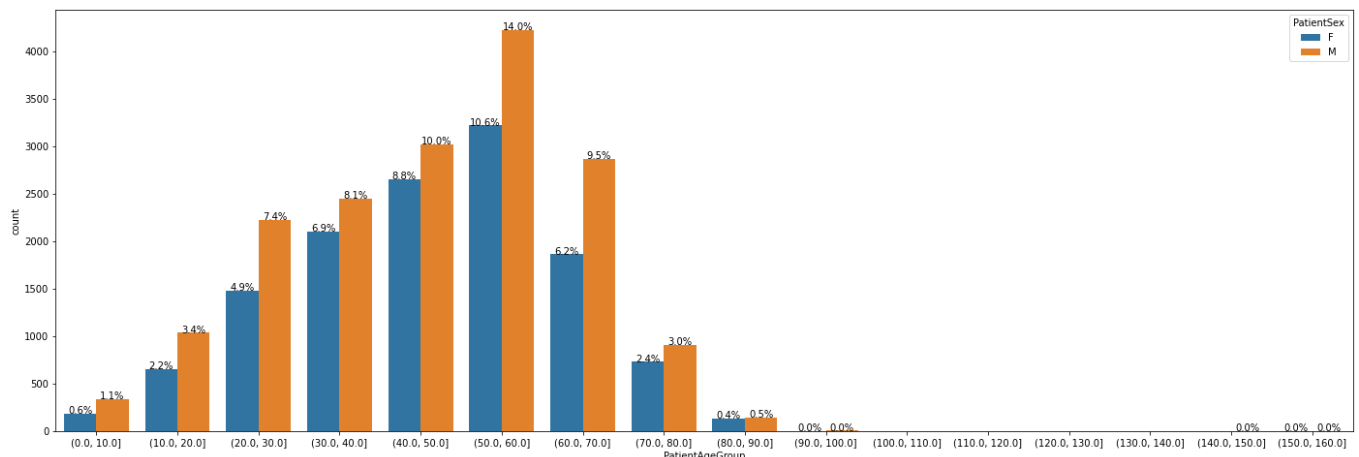




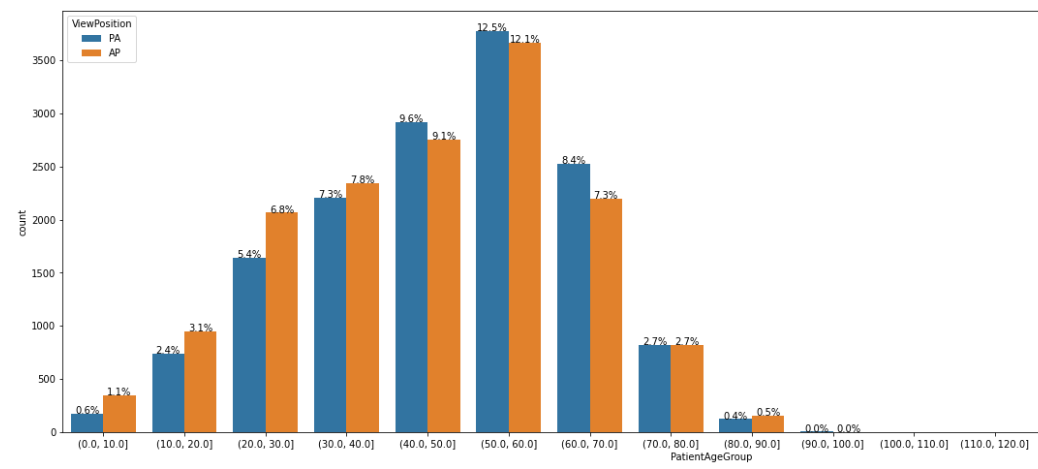
* The above picture depicts a distribution of different classes across the ‘PatientAgeGroup’.
* As can be seen patients with age group between 50-60 are having the highest probability of getting pnenumonia compared to other age groups.
* There are very less no of data points for age group>70.



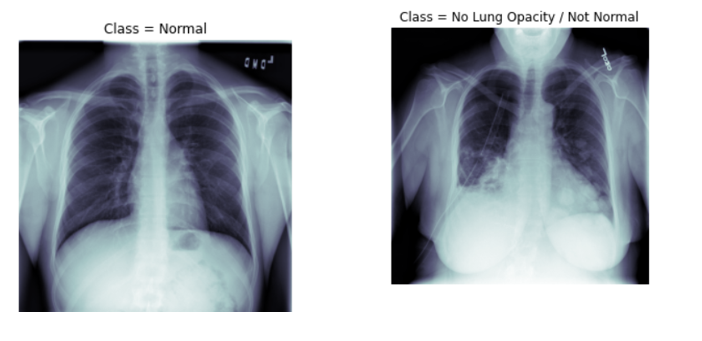
* The distribution of Age group with Target variable also reflects maximum no of postive cases between 40-60. For age group 50-60 there are 17.3% of patients who doesn’t have pneumonia compared to 7.4% who have.



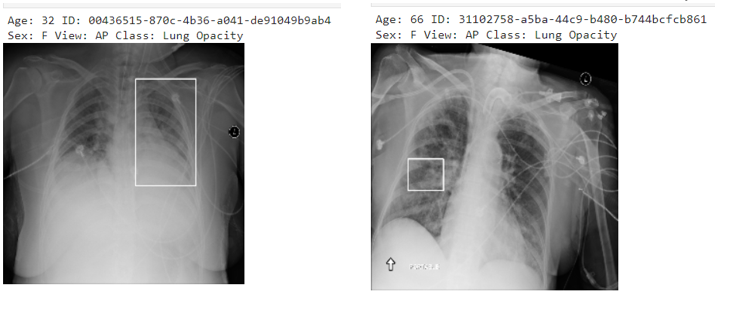
* The distribution of patient sex vs Age group shows there are maximum no of male and female present in age group 50-60 which is 14% and 10.6% respectively.
* Next majority falls between age group 40-50.
* There are very less no of records exists for age group 0-10 and 80-100.



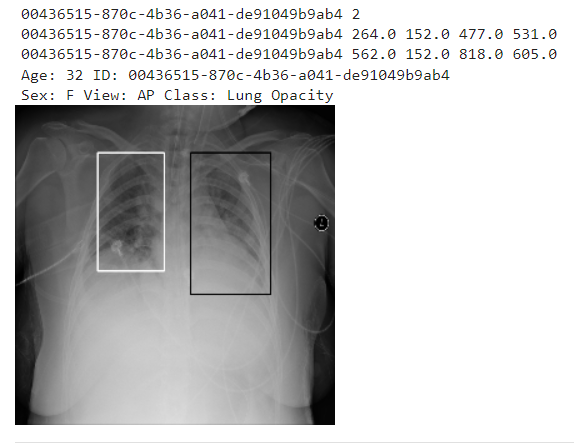
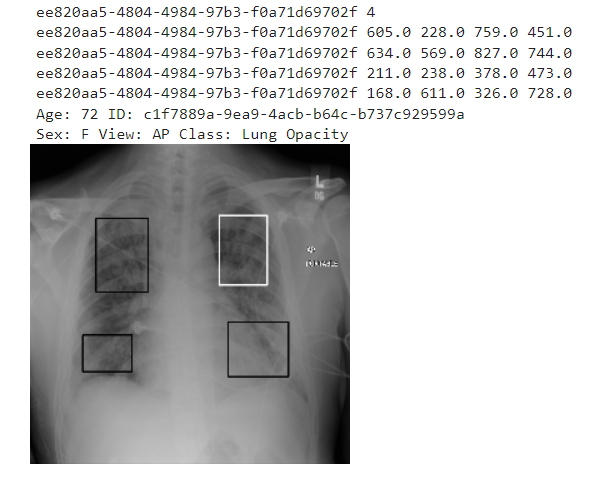
* Above picture illustrate the PA and AP view position for different Age group.
* Data shows for age group 50-60 there are 12.5% patients with AP and 12.1 % with PA position.
* Amongs all the patients between age group 30-40, 9.6% are having AP and 9.1% are having PA view positions.
* For agegroup 0-10 and 80-100 we have very less data available.
* Before proceeding, it would be good to view how the images are displayed. So loading a few DICOM images in google Colab. We will read images from training samples for both normal as well as for pneumonia patients as shown below.



* The above two images represents classes **‘Normal’** and ‘**No Lung Opacity / Not Normal’.** In the normal class the image is quite prominent with no sign of any lungs opacity.
* The second image signifies a patient with **No Lung Opacity / Not Normal’.** Even though it doesn’t have any Lung Opacity, still some portion of the image is blurred giving a notion of Not normal lungs. These sort of cases will further be validated by medical practitioners.



* Above two screenshot shows patients having lungs opacity. As can be seen there are bounding box on each of the images pointing the infections. There will be region of infection based on which the bounding box coordinates have been defined.

* There are also patient id’s which are duplicated or having multiple bounding boxes. This is valid as the patients might have multiple area with infections.
* As shown above the first image shows two bounding boxes/lungs opacity for the same patient id.
* Similarly for the second image there are 4 regions of infection for the same patient.

## 2.4 Summary:

* 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)
* 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. The age ranges from 1-155 (which were further clipped to 100)
* The centers of the bounding box are spread out over the entire region of the lungs. But there are some centres which are outliers.

# **Pre-Processing**

This section describes the pre-processing steps applied to data before modelling. The images are in dicom format which contains lot of metadata along with pixel data. The pixel data needs to be extracted and converted to either jpg or png format.

## 3.1 Pre-processing Methods

The following pre-processing methods can be applied to images.

* Conversion of image to jpg or png format.
* Find the number of channels in images and align to 1 or 3 channels.
  + Convert images to gray scale.
* Image resizing required as per base model requirements like 224\*224 for VGG16.
* Drop duplicate data.
* Set null values to 0 or drop the rows.
* Pixel normalization i.e scale the pixel values.
* Convert the pixel values to float.
* Image transformations like,
  + Rotate.
  + Flip vertically or horizontally.
  + Generate masks for the image.
  + Thresholding.
  + Erosion, Dilation.
  + Cropping.
  + Translation.
  + Noise addition. Etc.

## 3.2 Pre-processing Applied

The data generators are used to pre-process the image. The images are resized to 224\*224 and processed in batches of 32.

Duplicate rows are dropped from merged data frame “train\_feature\_engineered”.

The total number records after dropping duplicates are 26684.

The distribution of target variable and classes are given below.

Distribution of target and classes

0 20672

1 6012

Name: Target, dtype: int64

No Lung Opacity / Not Normal 11821

Normal 8851

Lung Opacity 6012

Name: class, dtype: int64

The shape of data after split into train, test and validation are as below.

Shape of the dataframes:

TRAIN:(21348, 3)

VALID:(2668, 3)

TEST:(2668, 3)

The data distribution is proper across train, test and validation data set.

Distribution of target in the training set:

0 0.78

1 0.22

Name: Target, dtype: float64

Distribution of target in the validation set:

0 0.78

1 0.22

Name: Target, dtype: float64

Distribution of target in the test set:

0 0.77

1 0.23

Name: Target, dtype: float64

# **Model and Model Building**

This section describes the approach used in modelling and the different techniques applied along with their evaluation.



## Models

The problem of lung opacity detection involves both classification and regressions. It is a classification problem as the model needs to identify if lung opacity is present or absent to aid in the detection of pneumonia. It is a regression problem as the model needs to identify the pixel areas containing the lung opacity. Some of the techniques which can be used are,

* Fast R-CNN
* Faster R-CNN
* Single Shot Detector (SSD)
* YOLO (You Look Only Once)
* SPP-net

Techniques based on R-CNN require lot of images for training and requires lot of training time with high resource consumption. SSD and YOLO perform better on time and resources comprising on accuracy. In order to overcome the above limitations, techniques of transfer learning based on existing models and their weights can be used to get better accuracy with reduced training data set in relatively less time.

Some of the pre-trained models suitable for opacity detection are,

* Densenet-121
* VGG 16
* ResNet-50
* CheXNet

### DenseNet-121

DenseNet has been designed to address the “vanishing gradient” problem with traditional CNN’s as number of layers become more. In this architecture each layer is connected directly to every layer. In case of N Layers, there are N(N+1)/2 connections.

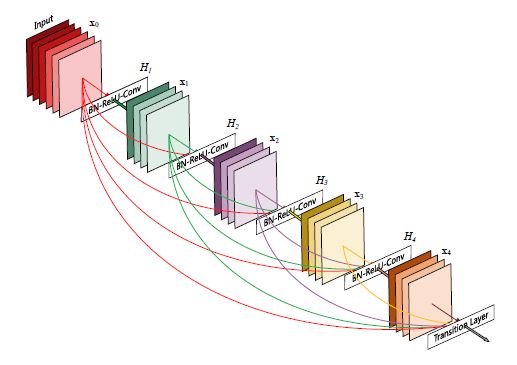


Figure : DenseNet-121 Architecture

The following layers are present in DenseNet-121,

* 1 7x7 Convolution
* 58 3x3 Convolution
* 61 1x1 Convolution
* 4 AvgPool
* 1 Fully Connected Layer

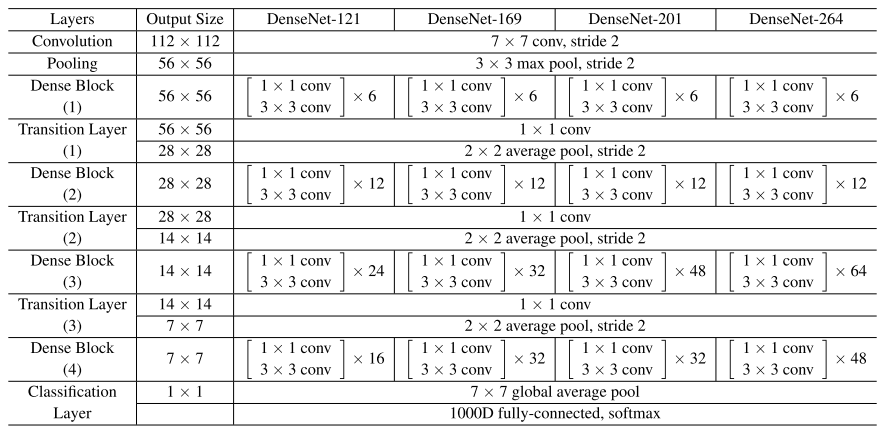


Figure : DenseNet-121 Details

The Dense blocks has varying number of layers with two convolutions. 1\*1 sized bottleneck layer and 3\*3 sized kernel for convolution. It has 120 convolutions and 4 average pools.

The key advantage is it requires fewer parameters and allows reuse of features resulting in compact models providing better performance compared to Resnet or other models.

### CheXNet

CheXNet is model relies on pre-trained weights of ImageNet and trained on NIH chest X-ray images. This model is specifically trained to detect Pneumonia and is based on DenseNet-121

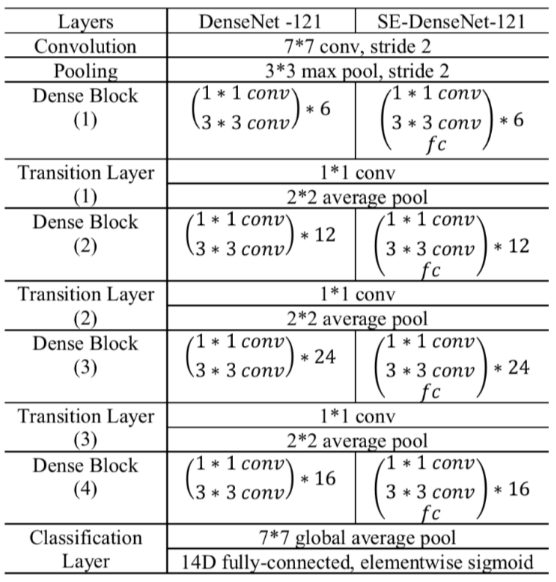


Figure : CheXNet Details

# **Evaluation Metrics**

Following metrics will be used for classification,

* Precision
* Recall
* F1-score
* Accuracy
* ROC Curve

IOU and mAP (mean Average Precision) will be used for evaluating regression part of the model.

# **Models Applied**

Data generators are used to load the data and pre-process them .The Models applied are DenseNet-121 and DenseNet-121 with CheXNet weights.



### DenseNet-121

Summary of model trainable parameters.

Create a `DenseNet121` model

----------------------------------------------------------------------

Model: "DenseNet121"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

DenseNet121 (Functional) (None, 7, 7, 1024) 7037504

global\_average\_pooling2d (G (None, 1024) 0

lobalAveragePooling2D)

dropout (Dropout) (None, 1024) 0

dense (Dense) (None, 1) 1025

=================================================================

Total params: 7,038,529

Trainable params: 6,954,881

Non-trainable params: 83,648

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

There are total of 6,954,881 trainable parameters.

The model was executed for 10 epocs with batch size of 32. The result is depicted below for each epic.

Epoch 1/10

668/668 [==============================] - ETA: 0s - loss: 0.4436 - accuracy: 0.7946 - average\_precision: 0.5853 - f1\_score: 0.4924

Epoch 1: val\_loss improved from inf to 0.57877, saving model to /content/drive/MyDrive/capstone/model\_weights/best\_densenet.h5

668/668 [==============================] - 2328s 3s/step - loss: 0.4436 - accuracy: 0.7946 - average\_precision: 0.5853 - f1\_score: 0.4924 - val\_loss: 0.5788 - val\_accuracy: 0.7789 - val\_average\_precision: 0.3563 - val\_f1\_score: 0.1586 - lr: 1.0000e-04

Epoch 2/10

668/668 [==============================] - ETA: 0s - loss: 0.3787 - accuracy: 0.8306 - average\_precision: 0.6652 - f1\_score: 0.5549

Epoch 2: val\_loss improved from 0.57877 to 0.36775, saving model to /content/drive/MyDrive/capstone/model\_weights/best\_densenet.h5

668/668 [==============================] - 1930s 3s/step - loss: 0.3787 - accuracy: 0.8306 - average\_precision: 0.6652 - f1\_score: 0.5549 - val\_loss: 0.3677 - val\_accuracy: 0.8403 - val\_average\_precision: 0.7438 - val\_f1\_score: 0.4836 - lr: 1.0000e-04

Epoch 3/10

668/668 [==============================] - ETA: 0s - loss: 0.3636 - accuracy: 0.8390 - average\_precision: 0.6910 - f1\_score: 0.5751

Epoch 3: val\_loss improved from 0.36775 to 0.35634, saving model to /content/drive/MyDrive/capstone/model\_weights/best\_densenet.h5

668/668 [==============================] - 1905s 3s/step - loss: 0.3636 - accuracy: 0.8390 - average\_precision: 0.6910 - f1\_score: 0.5751 - val\_loss: 0.3563 - val\_accuracy: 0.8407 - val\_average\_precision: 0.6648 - val\_f1\_score: 0.6103 - lr: 1.0000e-04

Epoch 4/10

668/668 [==============================] - ETA: 0s - loss: 0.3523 - accuracy: 0.8407 - average\_precision: 0.6974 - f1\_score: 0.5840

Epoch 4: val\_loss improved from 0.35634 to 0.34660, saving model to /content/drive/MyDrive/capstone/model\_weights/best\_densenet.h5

668/668 [==============================] - 1889s 3s/step - loss: 0.3523 - accuracy: 0.8407 - average\_precision: 0.6974 - f1\_score: 0.5840 - val\_loss: 0.3466 - val\_accuracy: 0.8504 - val\_average\_precision: 0.6995 - val\_f1\_score: 0.5968 - lr: 1.0000e-04

Epoch 5/10

668/668 [==============================] - ETA: 0s - loss: 0.3482 - accuracy: 0.8465 - average\_precision: 0.7091 - f1\_score: 0.5977

Epoch 5: val\_loss did not improve from 0.34660

668/668 [==============================] - 1861s 3s/step - loss: 0.3482 - accuracy: 0.8465 - average\_precision: 0.7091 - f1\_score: 0.5977 - val\_loss: 0.3646 - val\_accuracy: 0.8430 - val\_average\_precision: 0.7922 - val\_f1\_score: 0.4740 - lr: 1.0000e-04

Epoch 6/10

668/668 [==============================] - ETA: 0s - loss: 0.3435 - accuracy: 0.8479 - average\_precision: 0.7161 - f1\_score: 0.6039

Epoch 6: val\_loss did not improve from 0.34660

668/668 [==============================] - 1845s 3s/step - loss: 0.3435 - accuracy: 0.8479 - average\_precision: 0.7161 - f1\_score: 0.6039 - val\_loss: 0.3519 - val\_accuracy: 0.8531 - val\_average\_precision: 0.7541 - val\_f1\_score: 0.5993 - lr: 1.0000e-04

Epoch 7/10

668/668 [==============================] - ETA: 0s - loss: 0.3328 - accuracy: 0.8518 - average\_precision: 0.7186 - f1\_score: 0.6148

Epoch 7: val\_loss did not improve from 0.34660

668/668 [==============================] - 1873s 3s/step - loss: 0.3328 - accuracy: 0.8518 - average\_precision: 0.7186 - f1\_score: 0.6148 - val\_loss: 0.3571 - val\_accuracy: 0.8508 - val\_average\_precision: 0.7223 - val\_f1\_score: 0.6015 - lr: 1.0000e-04

Epoch 8/10

668/668 [==============================] - ETA: 0s - loss: 0.3303 - accuracy: 0.8519 - average\_precision: 0.7292 - f1\_score: 0.6145

Epoch 8: val\_loss improved from 0.34660 to 0.33845, saving model to /content/drive/MyDrive/capstone/model\_weights/best\_densenet.h5

668/668 [==============================] - 1856s 3s/step - loss: 0.3303 - accuracy: 0.8519 - average\_precision: 0.7292 - f1\_score: 0.6145 - val\_loss: 0.3385 - val\_accuracy: 0.8430 - val\_average\_precision: 0.6651 - val\_f1\_score: 0.6124 - lr: 1.0000e-04

Epoch 9/10

668/668 [==============================] - ETA: 0s - loss: 0.3255 - accuracy: 0.8553 - average\_precision: 0.7301 - f1\_score: 0.6223

Epoch 9: val\_loss did not improve from 0.33845

668/668 [==============================] - 1841s 3s/step - loss: 0.3255 - accuracy: 0.8553 - average\_precision: 0.7301 - f1\_score: 0.6223 - val\_loss: 0.4529 - val\_accuracy: 0.8433 - val\_average\_precision: 0.7439 - val\_f1\_score: 0.4807 - lr: 1.0000e-04

Epoch 10/10

668/668 [==============================] - ETA: 0s - loss: 0.3191 - accuracy: 0.8591 - average\_precision: 0.7366 - f1\_score: 0.6374

Epoch 10: val\_loss did not improve from 0.33845

668/668 [==============================] - 1939s 3s/step - loss: 0.3191 - accuracy: 0.8591 - average\_precision: 0.7366 - f1\_score: 0.6374 - val\_loss: 0.3397 - val\_accuracy: 0.8519 - val\_average\_precision: 0.7092 - val\_f1\_score: 0.6108 - lr: 1.0000e-04

Save the final weights

--------------------------------------------------------------------------------

The binary cross entropy loss is around 0.3 and validation accuracy of 85%.

The results for model run on evaluation data are,

Evaluate the model on validation data

-----------------------------------------------------------------------

Loss: 0.338, Accuracy: 0.843, Average Precision: 0.665, F1 Score: 0.612

AUC : 0.890

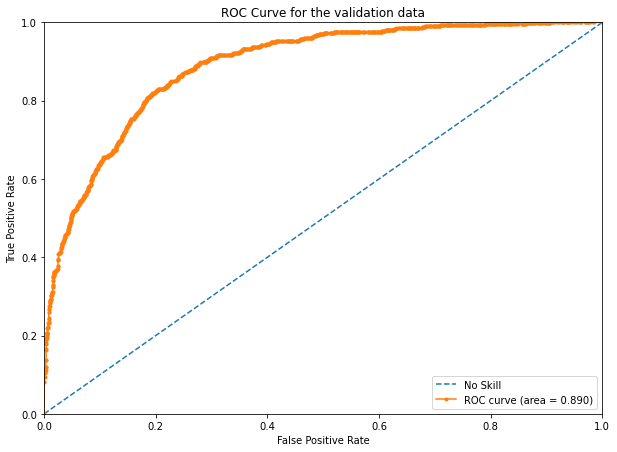


Figure : ROC for Validation data

ROC Curve for the test data

---------------------------------------------------------------------------

AUC: 0.894

A picture containing chart

Description automatically generated

Figure :ROC for test data

The AUC is similar for validation and test data.

The confusion matrix for test data indicating the predictions is shown below.

Chart, treemap chart

Description automatically generated

Figure : CM for test data

The classification report for test data :

Classification Report on the test data

------------------------------------------------------------------------------------------------------------------------

precision recall f1-score support

Normal 0.87 0.93 0.90 2042

Pneumonia 0.72 0.55 0.62 626

accuracy 0.84 2668

macro avg 0.79 0.74 0.76 2668

weighted avg 0.84 0.84 0.84 2668

The F1-score for Normal class is 0.9 indicating a high accuracy. The F1-score of 0.62 is low for Pneumonia class which is the main objective of the model.

### DenseSet-121 with CheXNet weights

The summary of model:

Create a `CheXNet-like` model using pre-trained weights

--------------------------------------------------------------------------------

Model: "CheXNet-like"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

CheXNet-like (Model) (None, 1024) 7037504

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout (Dropout) (None, 1024) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense (Dense) (None, 1) 1025

=================================================================

Total params: 7,038,529

Trainable params: 6,954,881

Non-trainable params: 83,648

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Results of model execution on training samples for 10 epochs with batch size of 32:

Train for 668 steps, validate for 84 steps

Epoch 1/10

667/668 [============================>.] - ETA: 0s - loss: 0.3733 - accuracy: 0.8300 - average\_precision: 0.6600 - f1\_score: 0.5311

Epoch 00001: val\_loss improved from inf to 0.44532, saving model to /home/pratik\_sharma\_ds/Capstone\_Pneumonia\_Detection/model\_weights/best\_chexnet.h5

668/668 [==============================] - 419s 627ms/step - loss: 0.3732 - accuracy: 0.8301 - average\_precision: 0.6605 - f1\_score: 0.5312 - val\_loss: 0.4453 - val\_accuracy: 0.8055 - val\_average\_precision: 0.6056 - val\_f1\_score: 0.3313

Epoch 2/10

667/668 [============================>.] - ETA: 0s - loss: 0.3533 - accuracy: 0.8403 - average\_precision: 0.7008 - f1\_score: 0.5837

Epoch 00002: val\_loss improved from 0.44532 to 0.37923, saving model to /home/pratik\_sharma\_ds/Capstone\_Pneumonia\_Detection/model\_weights/best\_chexnet.h5

668/668 [==============================] - 405s 606ms/step - loss: 0.3533 - accuracy: 0.8404 - average\_precision: 0.7008 - f1\_score: 0.5838 - val\_loss: 0.3792 - val\_accuracy: 0.8332 - val\_average\_precision: 0.7764 - val\_f1\_score: 0.4595

Epoch 3/10

667/668 [============================>.] - ETA: 0s - loss: 0.3511 - accuracy: 0.8427 - average\_precision: 0.7070 - f1\_score: 0.5909

Epoch 00003: val\_loss did not improve from 0.37923

668/668 [==============================] - 404s 605ms/step - loss: 0.3511 - accuracy: 0.8427 - average\_precision: 0.7075 - f1\_score: 0.5911 - val\_loss: 0.4044 - val\_accuracy: 0.8388 - val\_average\_precision: 0.7769 - val\_f1\_score: 0.4942

Epoch 4/10

667/668 [============================>.] - ETA: 0s - loss: 0.3450 - accuracy: 0.8426 - average\_precision: 0.7085 - f1\_score: 0.5871

Epoch 00004: val\_loss did not improve from 0.37923

668/668 [==============================] - 404s 604ms/step - loss: 0.3449 - accuracy: 0.8427 - average\_precision: 0.7089 - f1\_score: 0.5874 - val\_loss: 0.4144 - val\_accuracy: 0.8306 - val\_average\_precision: 0.8101 - val\_f1\_score: 0.4419

Epoch 5/10

667/668 [============================>.] - ETA: 0s - loss: 0.3449 - accuracy: 0.8457 - average\_precision: 0.7130 - f1\_score: 0.5984

Epoch 00005: val\_loss did not improve from 0.37923

668/668 [==============================] - 404s 604ms/step - loss: 0.3450 - accuracy: 0.8458 - average\_precision: 0.7135 - f1\_score: 0.5986 - val\_loss: 1.1231 - val\_accuracy: 0.7980 - val\_average\_precision: 0.5258 - val\_f1\_score: 0.1850

Epoch 6/10

667/668 [============================>.] - ETA: 0s - loss: 0.3409 - accuracy: 0.8464 - average\_precision: 0.7148 - f1\_score: 0.6002

Epoch 00006: val\_loss did not improve from 0.37923

668/668 [==============================] - 403s 604ms/step - loss: 0.3410 - accuracy: 0.8464 - average\_precision: 0.7143 - f1\_score: 0.5999 - val\_loss: 0.4552 - val\_accuracy: 0.8370 - val\_average\_precision: 0.7718 - val\_f1\_score: 0.5021

Epoch 7/10

667/668 [============================>.] - ETA: 0s - loss: 0.3380 - accuracy: 0.8475 - average\_precision: 0.7116 - f1\_score: 0.6055

Epoch 00007: val\_loss did not improve from 0.37923

Epoch 00007: ReduceLROnPlateau reducing learning rate to 0.00010000000474974513.

668/668 [==============================] - 403s 604ms/step - loss: 0.3379 - accuracy: 0.8475 - average\_precision: 0.7117 - f1\_score: 0.6056 - val\_loss: 1.3599 - val\_accuracy: 0.7777 - val\_average\_precision: 0.0357 - val\_f1\_score: 0.0119

Epoch 8/10

667/668 [============================>.] - ETA: 0s - loss: 0.3215 - accuracy: 0.8585 - average\_precision: 0.7409 - f1\_score: 0.6375

Epoch 00008: val\_loss improved from 0.37923 to 0.32581, saving model to /home/pratik\_sharma\_ds/Capstone\_Pneumonia\_Detection/model\_weights/best\_chexnet.h5

668/668 [==============================] - 404s 605ms/step - loss: 0.3213 - accuracy: 0.8587 - average\_precision: 0.7413 - f1\_score: 0.6379 - val\_loss: 0.3258 - val\_accuracy: 0.8538 - val\_average\_precision: 0.7211 - val\_f1\_score: 0.6140

Epoch 9/10

667/668 [============================>.] - ETA: 0s - loss: 0.3172 - accuracy: 0.8566 - average\_precision: 0.7317 - f1\_score: 0.6339

Epoch 00009: val\_loss improved from 0.32581 to 0.32542, saving model to /home/pratik\_sharma\_ds/Capstone\_Pneumonia\_Detection/model\_weights/best\_chexnet.h5

668/668 [==============================] - 408s 611ms/step - loss: 0.3170 - accuracy: 0.8568 - average\_precision: 0.7316 - f1\_score: 0.6341 - val\_loss: 0.3254 - val\_accuracy: 0.8482 - val\_average\_precision: 0.6918 - val\_f1\_score: 0.6097

Epoch 10/10

667/668 [============================>.] - ETA: 0s - loss: 0.3083 - accuracy: 0.8623 - average\_precision: 0.7464 - f1\_score: 0.6457

Epoch 00010: val\_loss did not improve from 0.32542

668/668 [==============================] - 405s 606ms/step - loss: 0.3085 - accuracy: 0.8621 - average\_precision: 0.7465 - f1\_score: 0.6455 - val\_loss: 0.3320 - val\_accuracy: 0.8482 - val\_average\_precision: 0.6931 - val\_f1\_score: 0.6104

Save the final weights

The loss is around ~0.32 and accuracy around ~85%.

Summary for evaluation on validation data:

Evaluate the model on validation data

-----------------------------------------------------------------------Loss: 0.325, Accuracy: 0.848, Average Precision: 0.692, F1 Score: 0.61

ROC Curve for validation data.

AUC: 0.898

Chart

Description automatically generated with medium confidence

Figure :ROC for validation data

ROC curve for test data:

AUC:0.895

A picture containing chart

Description automatically generated

Figure : ROC for test data

The confusion matrix for predicted labels:

Chart, treemap chart

Description automatically generated with medium confidence

Figure : CM for predicted class

The classification report on test data:

Classification Report on the test data

-----------------------------------------------------------------------

precision recall f1-score support

Normal 0.87 0.93 0.90 2042

Pneumonia 0.72 0.55 0.63 626

accuracy 0.85 2668

macro avg 0.80 0.74 0.76 2668

weighted avg 0.84 0.85 0.84 2668

The classification report shows that F1-score for Pneumonia class is still low showing a very marginal improvement.

## Loss/Accuracy Graphs

The graphs show the variation in loss and accuracy during training and validation for denseNet-121 and densenet-121 with Chexnet weights.

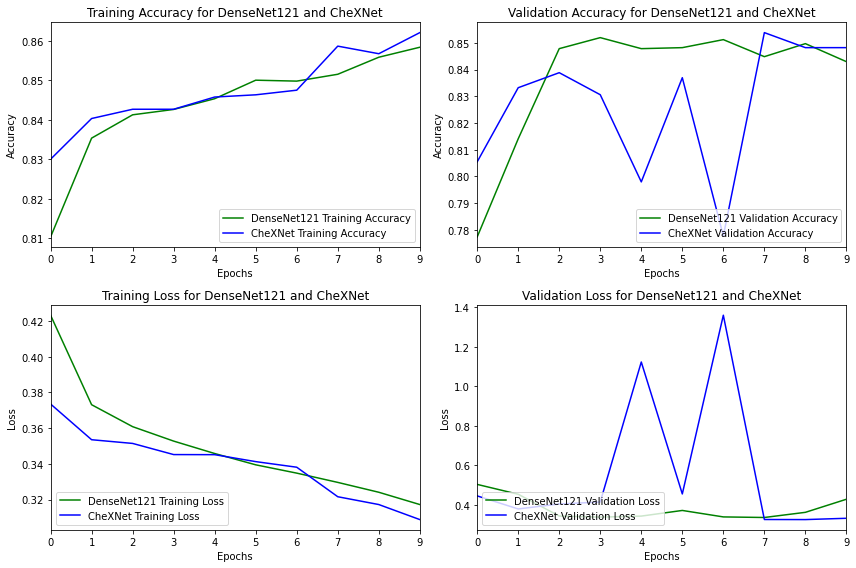


Figure :DenseNet vs Chexnet loss & accuracy

Observations:

* It is observed that training accuracy for DenseNet-121 is starting a lower value than the model with CheXnet weights. At the end of 10 epochs both models reach similar accuracy.
* The validation accuracy swings low and high during epochs where as DenseNet-121 accuracy goes high and then stays around a median.
* Training loss for DenseNet121 is high initially but in the end goes down to 0.33.Loss is slightly better when CheXNet weights are used.
* Swings are observed when ChexNet weights are used in validation loss. In the end validation loss is better when Chexnet weights are used.

# **How to Improve Model Performance**

Since the F1-score for Pneumonia class is around 0.63 and accuracy around 85% is not optimal for the given problem, following methods will be applied to improve performance,

* Data augmentation
  + Rotate
  + Flip
  + Masking
* YOLOv3 model
* Resnet-50 model