



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

1 Table of Contents

1. Summary of the problem statement, Data and findings	4
1.1. Problem Statement.....	4
1.2. Project Objectives:	4
2. EDA.....	5
2.1. Approach:.....	5
2.2. Analysis:	5
2.3 Visualization:	9
2.4 Summary:	15
3. Pre-Processing	16
3.1 Pre-processing Methods	16
3.2 Pre-processing Applied	16
4. Model and Model Building.....	18
4.1 Models	18
4.1.1 DenseNet-121	19
4.1.2 CheXNet	21
4.2 Evaluation Metrics	21
4.3 Models Applied	22
4.3.1 DenseNet-121	22
4.3.2 DenseSet-121 with CheXNet weights	24
4.4 Loss/Accuracy Graphs	27
4.5 Results Summary.....	28
5. How to Improve Model Performance.....	28

1. Summary of the problem statement, Data and findings

1.1. 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 specialists in reviewing them spend much time and effort.

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.

1.2. 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).

1.3. 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.

2. 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 step is to verify the format of images as provided, and they are all DICOM images in the “dcm” file format. Read the DICOM images using the **pydicom** library for that purpose. After that, the next step would be to inspect the csv files.

2.2. Analysis:

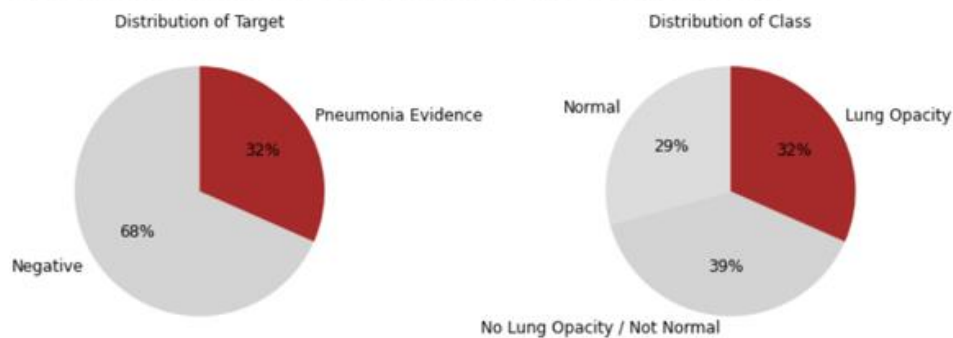
- Loading the “**stage_2_detailed_class_info.csv**” file into pandas dataframe, a quick glance reveals that it has only 2 columns:
 - **patientId** – which refers to the patientId’s corresponding image name
 - **class** – Target label of the patientId’s image

```
# Check for types of class labels
```

```
df_class["class"].value_counts()
```

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

Lets check the distribution of 'Target' and 'class' column



- 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

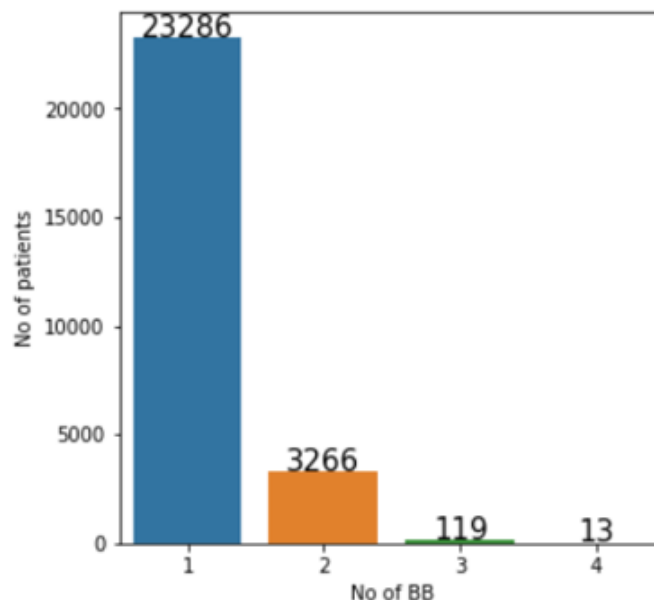
- There is total of 30,227 entries, no missing values with 9,555 images have bounding boxes. This corresponds to the data available “stage_2_detailed_class_info.csv” file.

```
train_labels_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30227 entries, 0 to 30226
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype
---  -
0   patientId   30227 non-null  object
1   x            9555 non-null   float64
2   y            9555 non-null   float64
3   width        9555 non-null   float64
4   height       9555 non-null   float64
5   Target       30227 non-null  int64
dtypes: float64(4), int64(1), object(1)
memory usage: 1.4+ MB
```

The bounding box information is present for only 9555 records out of 30227.

number_of_patientIDs_per_boxes	
number_of_boxes	
1	23286
2	3266
3	119
4	13



- 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 patients (~87% of them) provided have 1 bounding boxes while 13 patients have 4 bounding boxes. The reason is that 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.

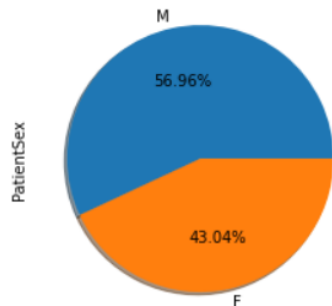
```
Dataset.file_meta -----
(0002, 0000) File Meta Information Group Length  UL: 200
(0002, 0001) File Meta Information Version       OB: b'\x00\x01'
(0002, 0002) Media Storage SOP Class UID        UI: Secondary Capture Image Storage
(0002, 0003) Media Storage SOP Instance UID     UI: 1.2.276.0.7230010.3.1.4.8323329.6379.1517874325.469569
(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: 1.2.276.0.7230010.3.1.4.8323329.6379.1517874325.469569
(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: '00436515-870c-4b36-a041-de91049b9ab4'
(0010, 0020) Patient ID                         LO: '00436515-870c-4b36-a041-de91049b9ab4'
(0010, 0030) Patient's Birth Date               DA: ''
(0010, 0040) Patient's Sex                      CS: 'F'
(0010, 1010) Patient's Age                      AS: '32'
(0018, 0015) Body Part Examined                 CS: 'CHEST'
(0018, 5101) View Position                      CS: 'AP'
(0020, 000d) Study Instance UID                 UI: 1.2.276.0.7230010.3.1.2.8323329.6379.1517874325.469568
(0020, 000e) Series Instance UID               UI: 1.2.276.0.7230010.3.1.3.8323329.6379.1517874325.469567
(0020, 0010) Study ID                           SH: ''
(0020, 0011) Series Number                      IS: '1'
(0020, 0013) Instance Number                    IS: '1'
(0020, 0020) Patient Orientation                CS: ''
```

- 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

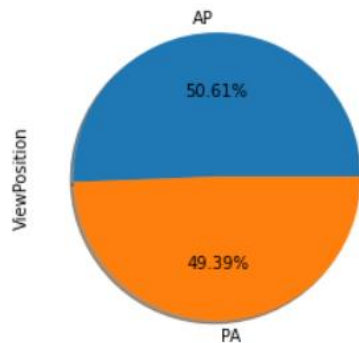
2.3 Visualization:

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

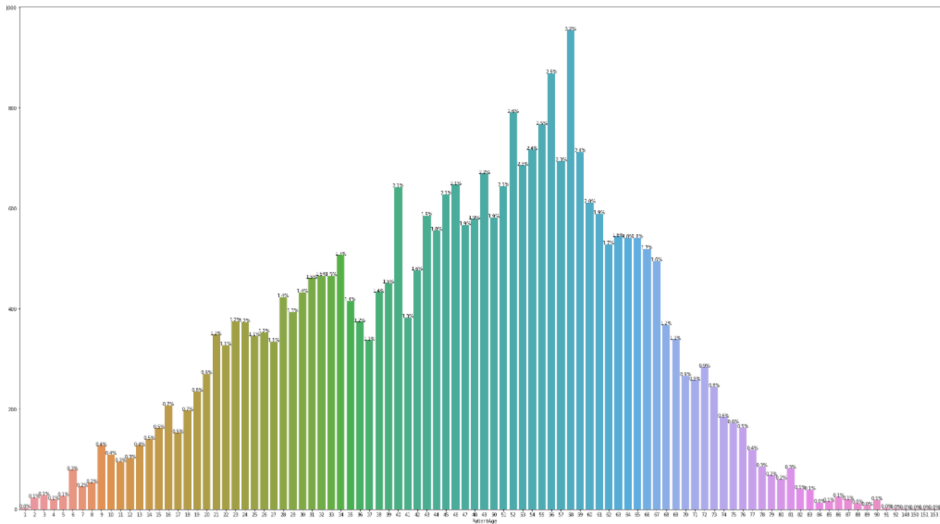
Univariate Analysis:



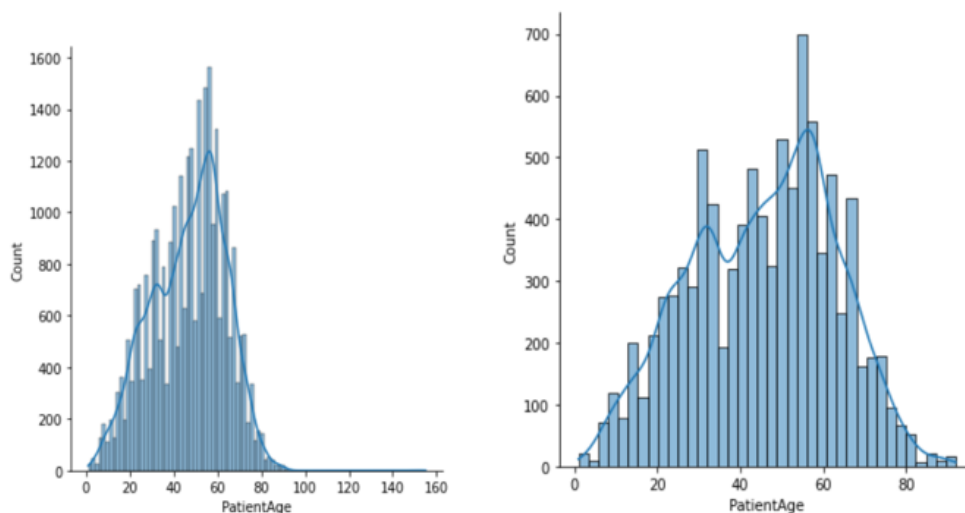
- As shown above, there are 56.96% male patients and roughly 43% female patients. There are also more no of male patients having pneumonia compared to females.



- 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 is 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 erroneous and we can safely remove them from our dataset and will keep the age



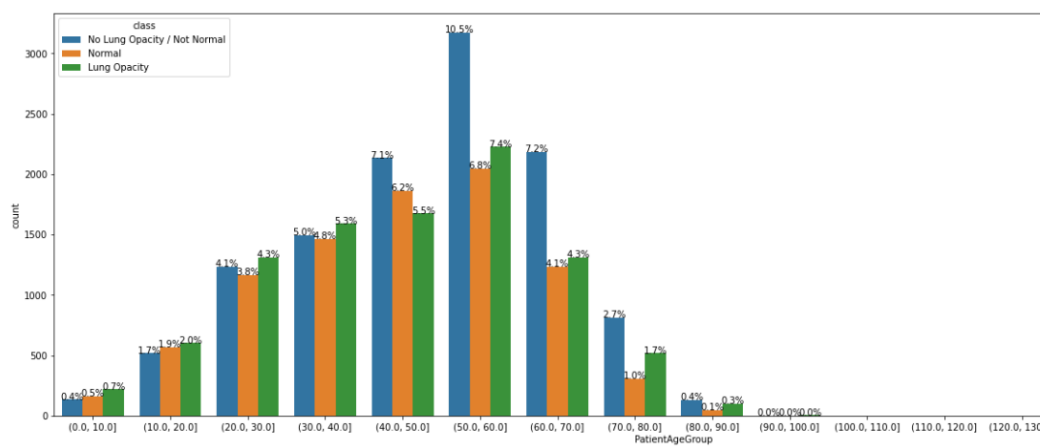
- As per the above histogram, the patient age is normally distributed with more volume of data lying between age group 40-60.
- The Age distribution for pneumonia patient is slightly left skewed with more no of patient between age group 40-60 are having pneumonia.

Bivariate Analysis:

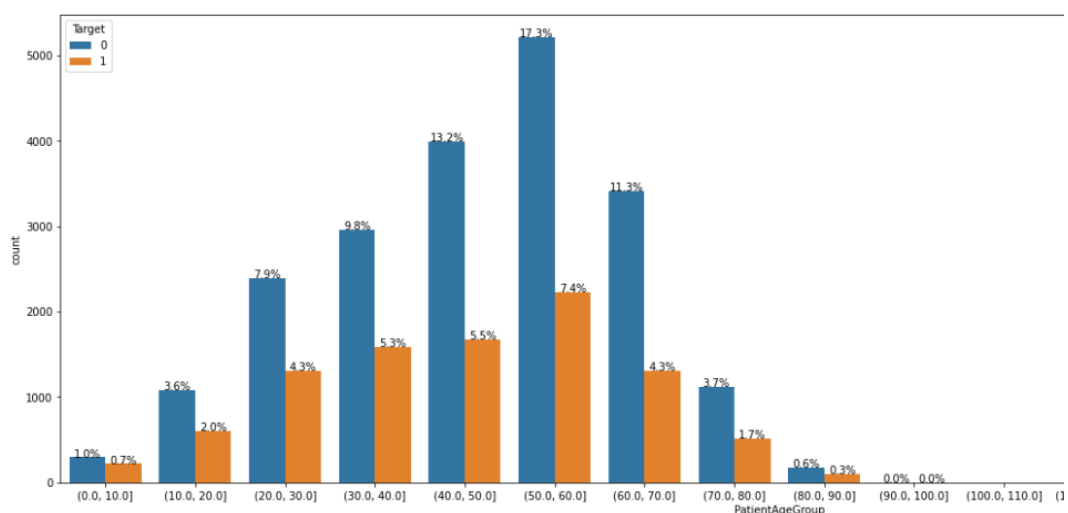
- To have a better interpretation we will make use of binning concept as shown below to group the patient age and will further visualize on the transformed data.

```
merged_df['PatientAgeGroup'] = pd.cut(merged_df['PatientAge'], patient_age_arr)
merged_df['PatientAgeGroup'].value_counts()
```

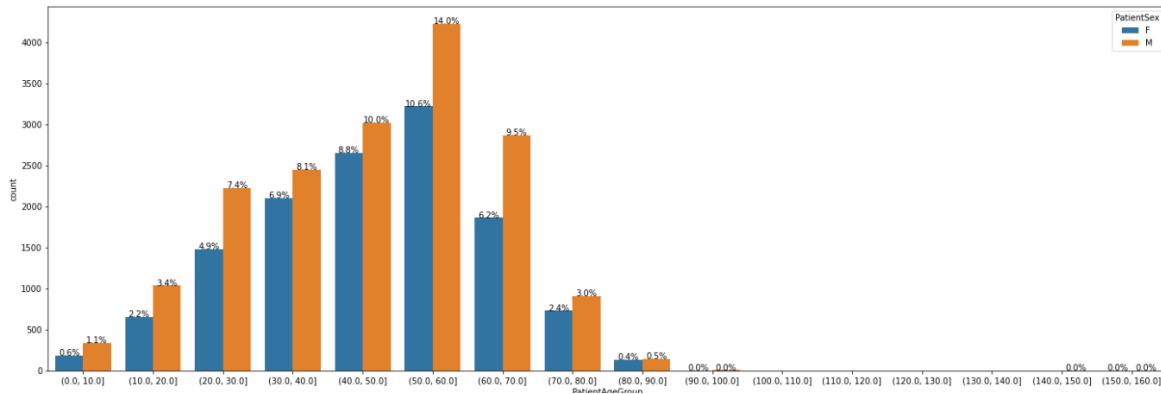
```
(50.0, 60.0]      7446
(40.0, 50.0]      5671
(60.0, 70.0]      4730
(30.0, 40.0]      4551
(20.0, 30.0]      3704
(10.0, 20.0]      1688
(70.0, 80.0]      1637
(0.0, 10.0]       515
(80.0, 90.0]       275
(90.0, 100.0]      5
(150.0, 160.0]     3
(140.0, 150.0]     2
(100.0, 110.0]     0
(110.0, 120.0]     0
(120.0, 130.0]     0
(130.0, 140.0]     0
Name: PatientAgeGroup, dtype: int64
```



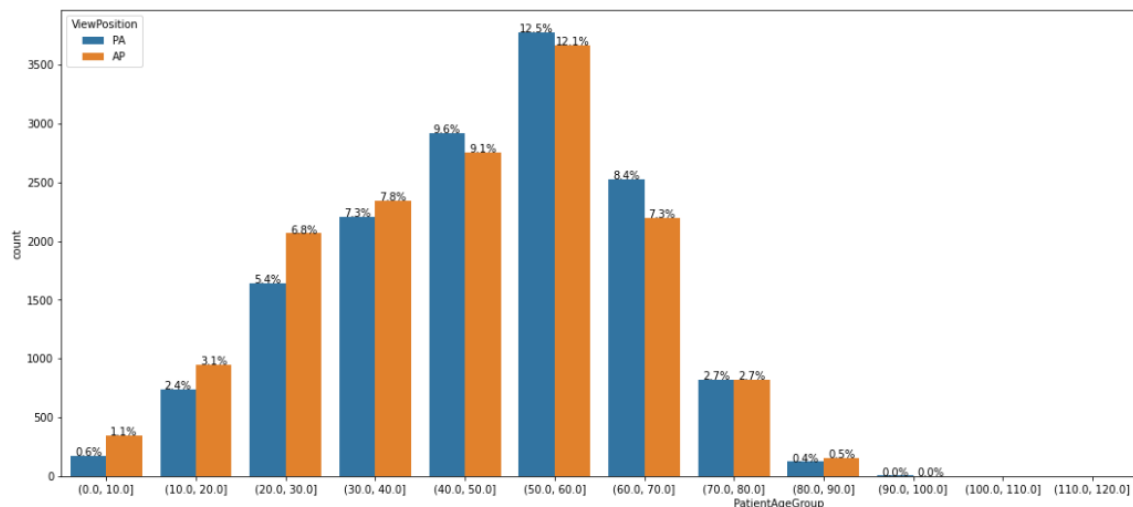
- The above picture depicts a distribution of different classes across the 'PatientAgeGroup'.
- As can be seen patients with age group between 50-60 are having the highest probability of getting pneumonia compared to other age groups.
- There are very less no of data points for age group >70.



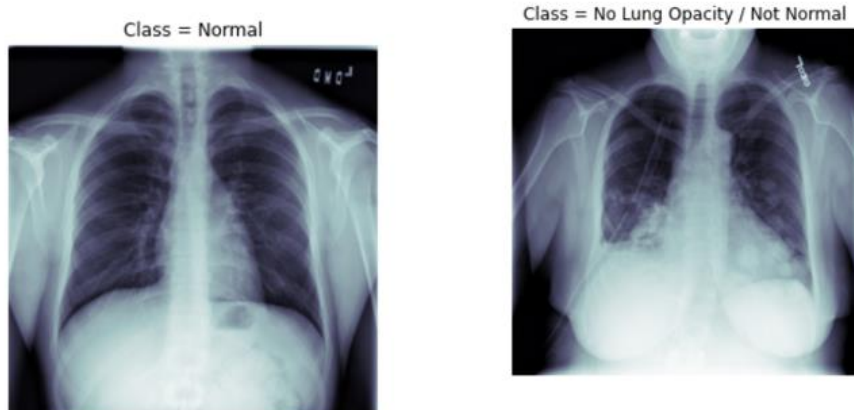
- The distribution of Age group with Target variable also reflects maximum no of postive cases between 40-60. For age group 50-60 there are 17.3% of patients who doesn't have pneumonia compared to 7.4% who have.



- 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.

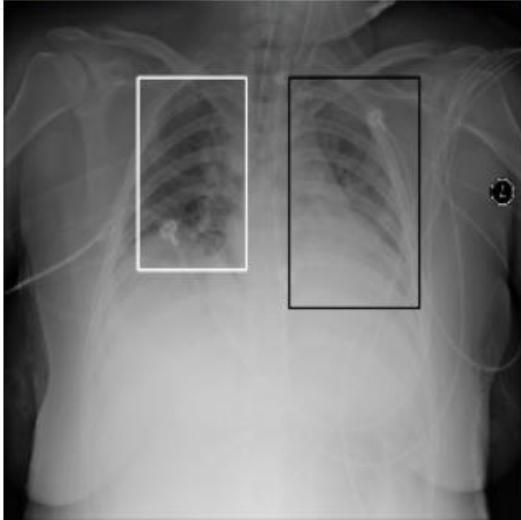


- The above two images represent classes '**Normal**' and '**No Lung Opacity / Not Normal**'. In the normal class, the image is quite prominent with no sign of any lung opacity.
- The second image signifies a patient with '**No Lung Opacity / Not Normal**'. Even though it does not have any lung opacity, still some portion of the image is blurred, giving a notion of not normal lungs. Medical practitioners will further validate these sort of cases.

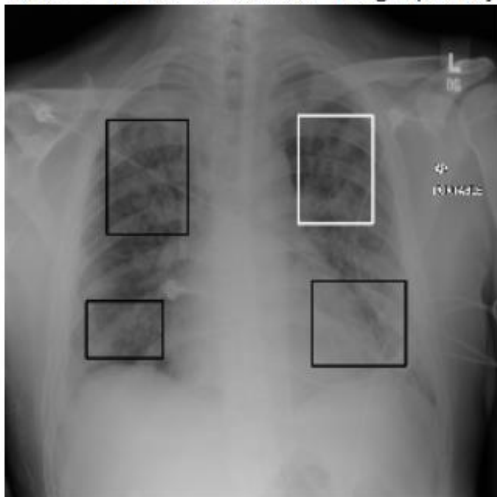


- Above two screenshots show patients having lung opacity. As can be seen, there are bounding boxes on each of the images pointing to the infections. There will be a region of infection based on which the bounding box coordinates have been defined.

00436515-870c-4b36-a041-de91049b9ab4 2
00436515-870c-4b36-a041-de91049b9ab4 264.0 152.0 477.0 531.0
00436515-870c-4b36-a041-de91049b9ab4 562.0 152.0 818.0 605.0
Age: 32 ID: 00436515-870c-4b36-a041-de91049b9ab4
Sex: F View: AP Class: Lung Opacity



ee820aa5-4804-4984-97b3-f0a71d69702f 4
ee820aa5-4804-4984-97b3-f0a71d69702f 605.0 228.0 759.0 451.0
ee820aa5-4804-4984-97b3-f0a71d69702f 634.0 569.0 827.0 744.0
ee820aa5-4804-4984-97b3-f0a71d69702f 211.0 238.0 378.0 473.0
ee820aa5-4804-4984-97b3-f0a71d69702f 168.0 611.0 326.0 728.0
Age: 72 ID: c1f7889a-9ea9-4acb-b64c-b737c929599a
Sex: F View: AP Class: Lung Opacity



- In addition, there are patient id's which are duplicate or have 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. However, there are some centres, which are outliers.

3. 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 grey 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. The pixel intensity values are modified to a range of values This is also known as contrast stretching.
- Image augmentation is technique used to artificially generate more variations in existing data and create additional variety in training images. This technique helps in generalizing the model better and avoids overfitting. Some of the transformations that can be applied are,
 - Rotate the images by specified angle.
 - Flip vertically or horizontally.
 - Shearing - shift one part of the image like parallelogram.
 - Generate masks for the image.
 - Thresholding – used to binarize grey scale images.
 - Erosion, Dilation – Used to either erode certain features or make them prominent.
 - Crop the images and generate many sub-sets from original images.

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 were 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

4. Model and Model Building

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

4.1 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

4.1.1 DenseNet-121

DenseNet-121 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.

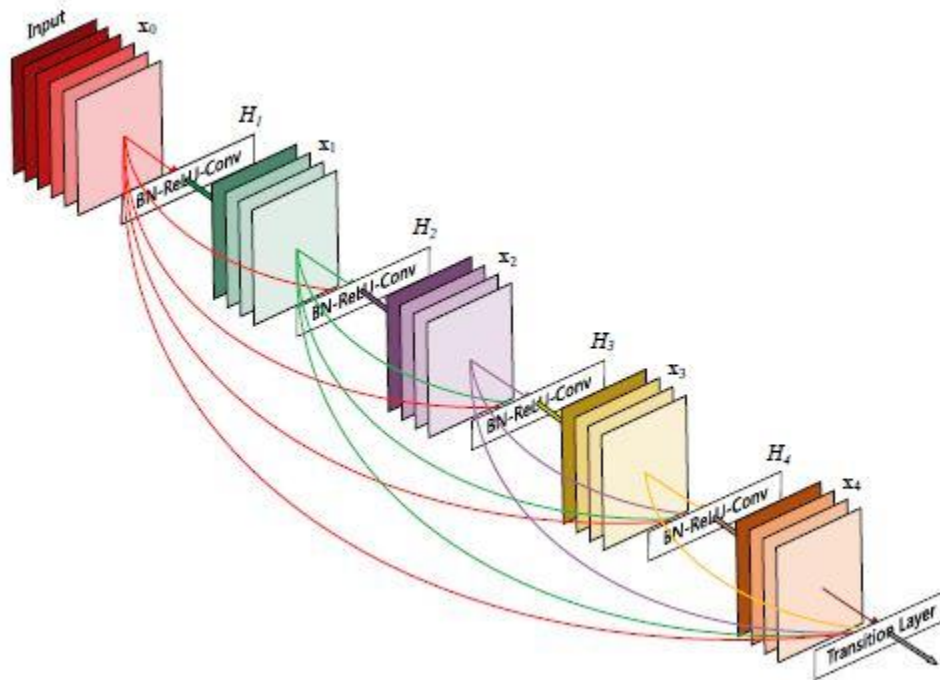


Figure 1: DenseNet-121 Architecture

The following layers are present in DenseNet-121,

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

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112×112	7×7 conv, stride 2			
Pooling	56×56	3×3 max pool, stride 2			
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56	1×1 conv			
	28×28	2×2 average pool, stride 2			
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28	1×1 conv			
	14×14	2×2 average pool, stride 2			
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14×14	1×1 conv			
	7×7	2×2 average pool, stride 2			
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1×1	7×7 global average pool			
		1000D fully-connected, softmax			

Figure 2: 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.

4.1.2 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

Layers	DenseNet -121	SE-DenseNet-121
Convolution	7*7 conv, stride 2	
Pooling	3*3 max pool, stride 2	
Dense Block (1)	$\begin{pmatrix} 1 * 1 \text{ conv} \\ 3 * 3 \text{ conv} \end{pmatrix} * 6$	$\begin{pmatrix} 1 * 1 \text{ conv} \\ 3 * 3 \text{ conv} \\ fc \end{pmatrix} * 6$
Transition Layer (1)	1*1 conv	
	2*2 average pool	
Dense Block (2)	$\begin{pmatrix} 1 * 1 \text{ conv} \\ 3 * 3 \text{ conv} \end{pmatrix} * 12$	$\begin{pmatrix} 1 * 1 \text{ conv} \\ 3 * 3 \text{ conv} \\ fc \end{pmatrix} * 12$
Transition Layer (2)	1*1 conv	
	2*2 average pool	
Dense Block (3)	$\begin{pmatrix} 1 * 1 \text{ conv} \\ 3 * 3 \text{ conv} \end{pmatrix} * 24$	$\begin{pmatrix} 1 * 1 \text{ conv} \\ 3 * 3 \text{ conv} \\ fc \end{pmatrix} * 24$
Transition Layer (3)	1*1 conv	
	2*2 average pool	
Dense Block (4)	$\begin{pmatrix} 1 * 1 \text{ conv} \\ 3 * 3 \text{ conv} \end{pmatrix} * 16$	$\begin{pmatrix} 1 * 1 \text{ conv} \\ 3 * 3 \text{ conv} \\ fc \end{pmatrix} * 16$
Classification Layer	7*7 global average pool	
	14D fully-connected, elementwise sigmoid	

Figure 3: CheXNet Details

4.2 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.

4.3 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.

4.3.1 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 (GlobalAveragePooling2D)	(None, 1024)	0
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 epochs with batch size of 32. 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

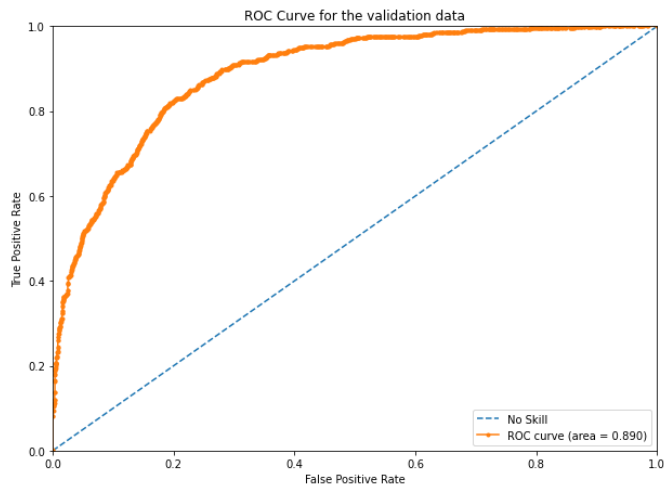


Figure 4: ROC for Validation data

ROC Curve for the test data

AUC: 0.894

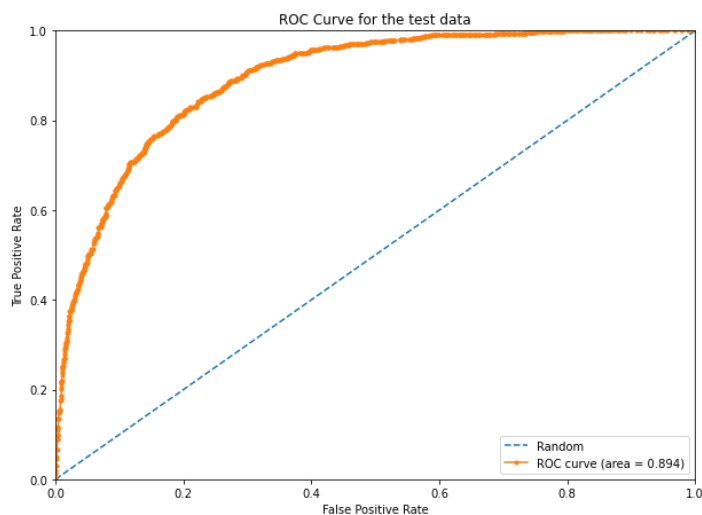


Figure 5: ROC for test data

The AUC is similar for validation and test data.

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

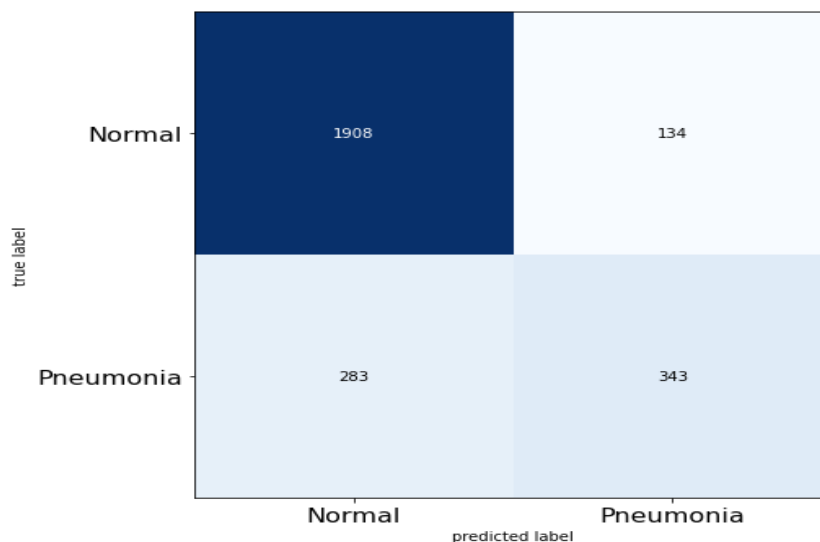


Figure 6: CM for test data

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.

4.3.2 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
<hr/>		
dropout (Dropout)	(None, 1024)	0
<hr/>		
dense (Dense)	(None, 1)	1025

=====

Total params: 7,038,529
Trainable params: 6,954,881
Non-trainable params: 83,648

=====

The loss is around ~ 0.32 and accuracy is 85% after model execution on training samples for 10 epochs with batch size of 32.

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

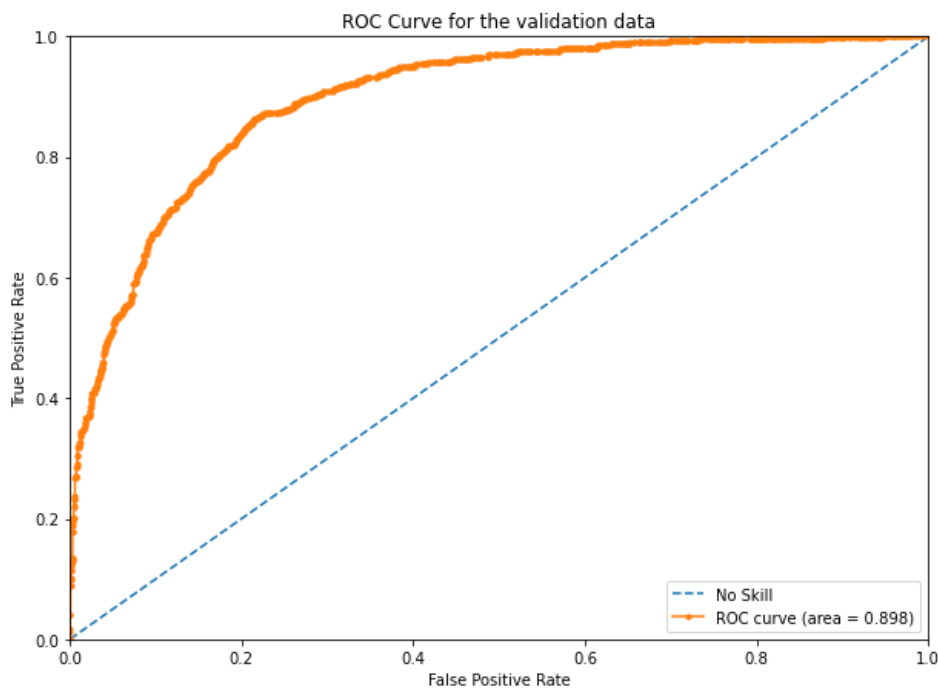


Figure 7: ROC for validation data

ROC curve for test data:

AUC: 0.895

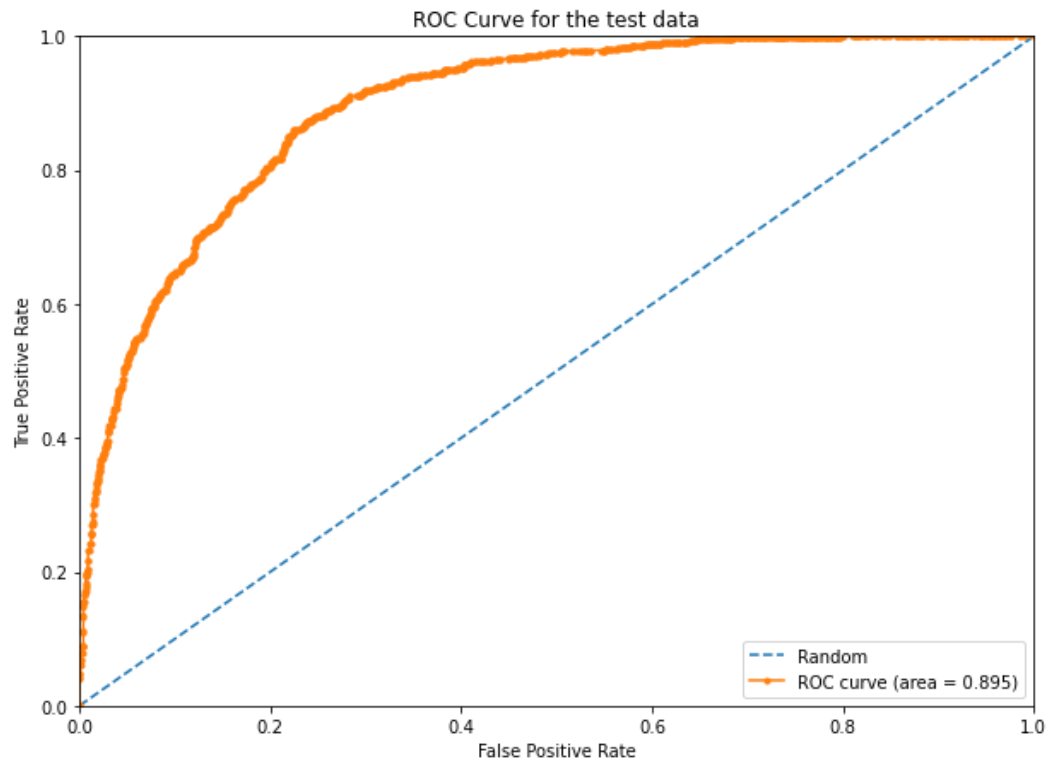


Figure 8: ROC for test data

The confusion matrix for predicted labels:

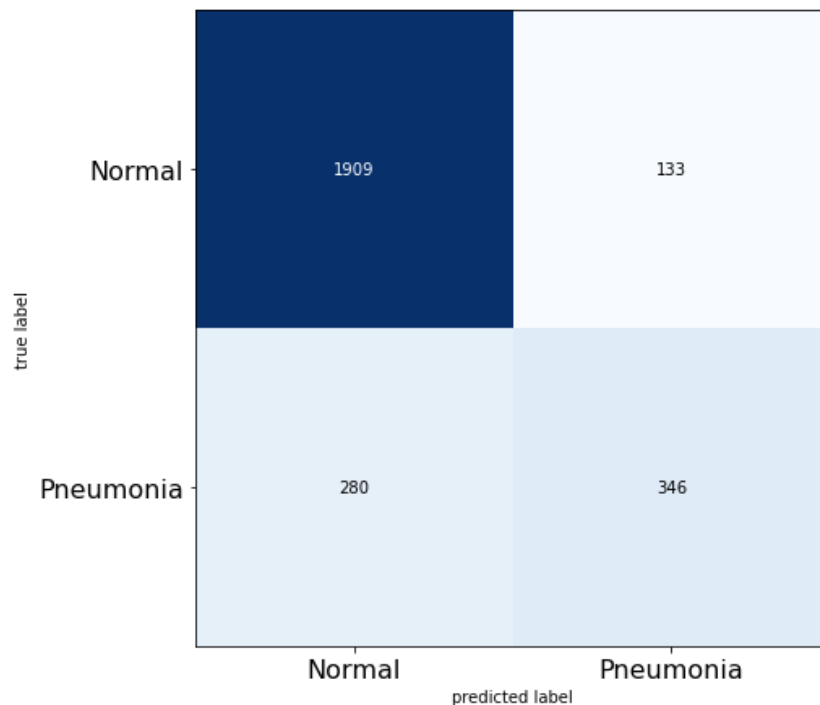


Figure 9: CM for predicted class

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.

4.4 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.

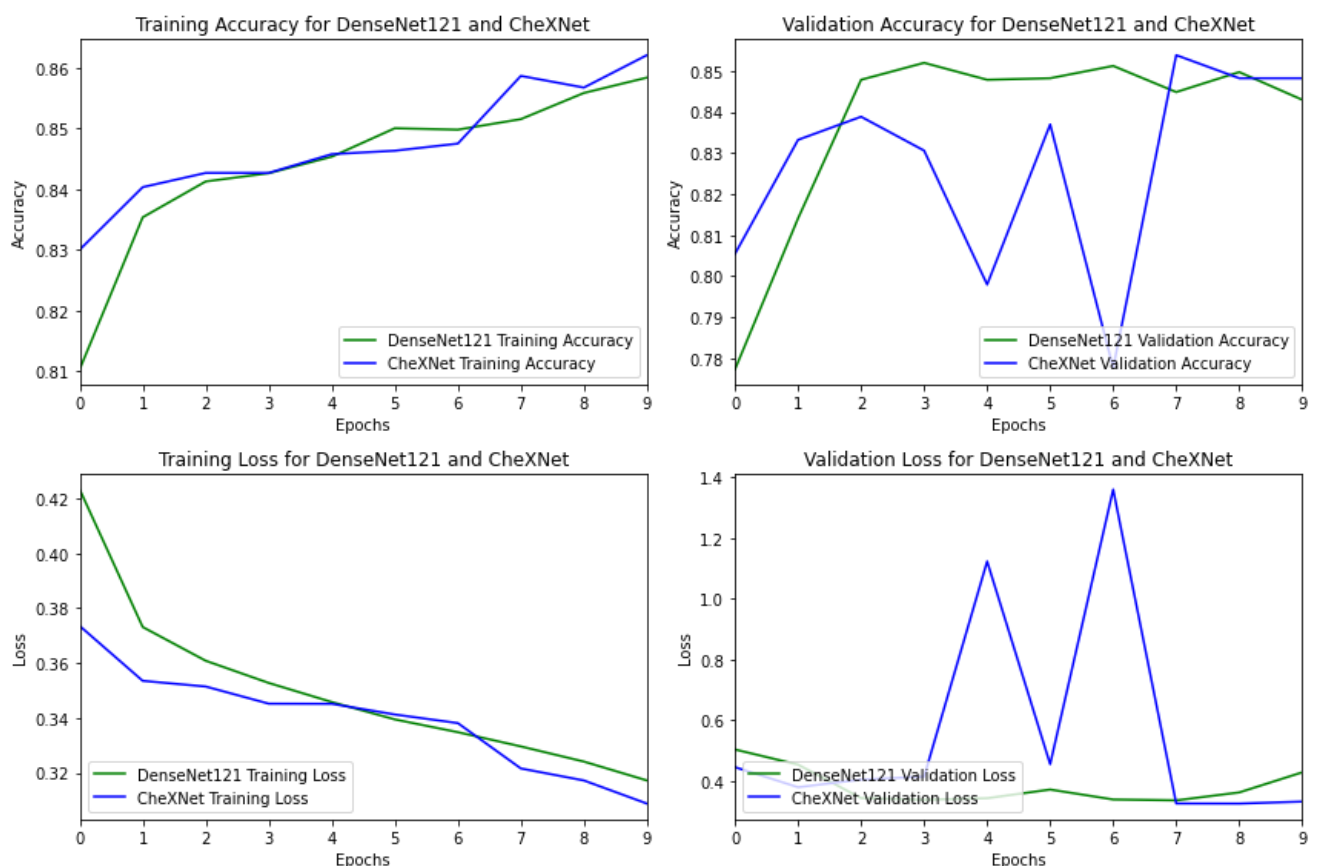


Figure 10:DenseNet vs Chexnet loss & accuracy

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.

4.5 Results Summary

S. No.	Model	Class	F1-Score	Accuracy	Recall	Precision
1	DenseNet 121	Normal	0.9	84%	0.93	0.87
2	DenseNet 121	Pneumonia	0.62	84%	0.55	0.72
3	DenseNet121+CheXnet weights	Normal	0.9	85%	0.93	0.87
4	DensetNet121+CheXnet weights	Pneumonia	0.63	85%	0.55	0.72

5. 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.
- Apply transfer-learning methods.