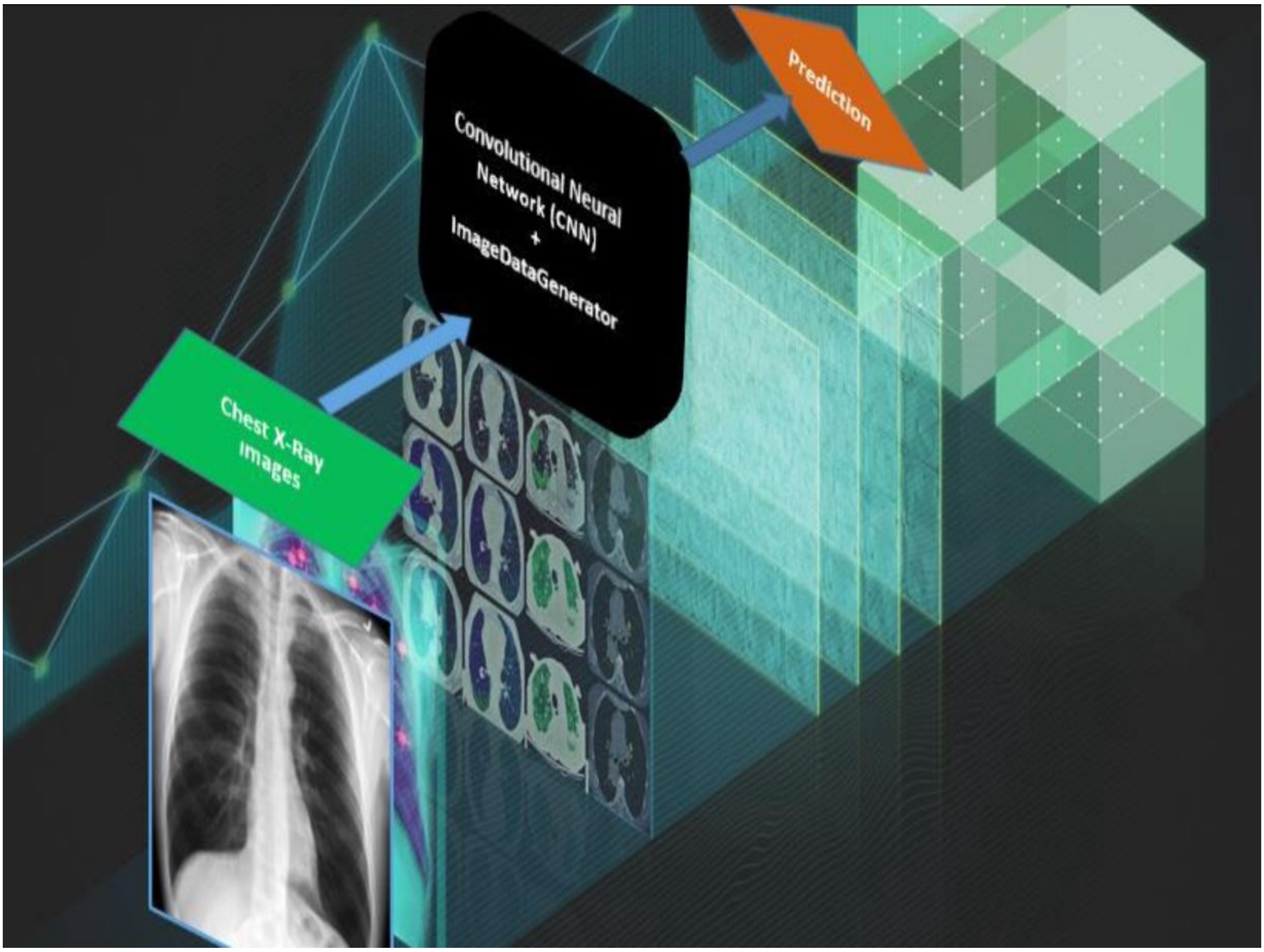
AIML-Jul 21A-Group 6 – CV 1



Capstone Project – Pneumonia Detection Challenge

Interim Report

	Name	Mail Id
Team	Ashish Tiwari	ashishtiwari2114@gmail.com
	Kanishk Sanger	kanishksanger2410@gmail.com
	Ninad Mahajan	ninad.mahajan99@gmail.com
	Avinash Kumra Sharma	avinashzen123@gmail.com
	Tushar Bisht	tussods@gmail.com
Mentor	**Mr. Rohit Raj**	rohit.raj2017@gmail.com

GitHub repository:

https://github.com/ashish090798/GL-Capstone-Project

Contents

1. Summary of the problem statement, Data and findings 4
1.1 Problem Statement 4
1.2 Data & Findings 5
2. Summary of the Approach to EDA and Pre-processing 9
2.1 Approach to EDA and Pre-processing 9
2.2 EDA and Pre-processing – Steps and Results 9
2.3 EDA - Conclusions 19
3. Deciding Models and Model Building............................20
3.1 Model Approach............................20
3.2 Model Identification 20
3.3 Model Metrics 21
3.4 Model Training Summary 22
4. Improving Model Performance............................24
4.1 Feature Extraction 24
4.2 Data Manipulation/Augmentation............................24
4.3 Model Improvements............................25
5. Next Steps............................30

1. Summary of the problem statement, Data and findings

1.1 Problem Statement

Pneumonia is a global health problem that does not understand social or cultural strata, causing millions of deaths each year. In several conferences and articles, it is called "the silent killer", a nickname that reflects the little social and political awareness towards this disease that without receiving the same attention as other pathologies her number of affectations year after year are forceful. In developing countries, the affectation of this lung disease makes it one of the deadliest among children under 5 years of age, causing 15% of the deaths recorded each year

In the United States, pneumonia accounts for over 500,000 visits to emergency departments and over 50,000 deaths in 2015, keeping the ailment on the list of top 10 causes of death in the country. Official Report. Pneumonia diagnosing requires a review of a chest radiograph (CXR), clinical history, vital signs, and laboratory exams. When interpreting an X-ray by a professional, his capacity and experience are key.

Chest Radiographs basically is the process of taking an image, in other words the X-ray passes through the body and reaches a detector on the other side. Tissues with sparse material, such as lungs, which are full of air, do not absorb X-rays and appear black in the image. Dense tissues such as bones absorb X-rays and appear white in the image. (1) Black = Air (2) White = Bone (3) Grey = Tissue or Fluid. Pneumonia usually appears as an area of increased lung opacity on CXR.

The objective is to build an algorithm that can detect visual signals for pneumonia in medical images. Specifically, the algorithm needs to automatically locate lung opacities on chest radiographs, but only the opacities that look like pneumonia, and discard other types of opacities like the ones caused by fluid overload (pulmonary edema), bleeding, volume loss (atelectasis or collapse), lung cancer, post-radiation or surgical changes. Outside of the lungs, fluid in the pleural space (pleural effusion) also appears as increased opacity on CXR. A pneumonia opacity is a part of the lungs that looks darker on a radiograph and has a shape that indicates that pneumonia is (or may be) present.

As the objective is to detect and draw a bounding box on each of the pneumonia opacities, where each image can have 0 or many opacities, and the training set is already classified, it will be analyzed as a supervised classification. The neural networks seem to be the best bet, specially a FCNN (Fully Convolutional Neural Network).

To improve the efficiency and reach of diagnostic services, our aim is to build a deep learning model to detect a visual signal for pneumonia in medical images assisting medical practitioners. Model needs to automatically locate lung opacities on chest radiographs, and hence help clinicians to detect and diagnose pneumonia with high accuracy & efficiency.

1.2 Data & Findings

One of the ways to diagnose pneumonia is to analyze the Chest X-Ray. The dataset available contains these CXR images in **DICOM** format.

DICOM – Digital Imaging and Communications in Medicine is known as an international standard for medical images and everything that is related to them. DICOM images are known to have high quality, since the diagnosis requires as clear information as possible. This format is used in a variety of medical domains like radiology, cardiology and so on.

It contains several important bits of information: for example, unique patient ID, position of the body when the scan was taken, patient gender, age and so on.

The input files provided are:

- stage_2_train_labels.csv - the training set, contains patientIds and bounding box/target information.
- Data Fields in stage_2_train.csv are shown in the below table

Data Field	Description
patientId	Patient Id. Each patient Id corresponds to a unique image
x	the upper-left x coordinate of the bounding box
y	the upper-left y coordinate of the bounding box
width	the width of the bounding box
height	the height of the bounding box
Target	the binary Target, indicating whether this sample has evidence of pneumonia

- stage_2_detailed_class_info.csv - provides detailed information about the type of each image

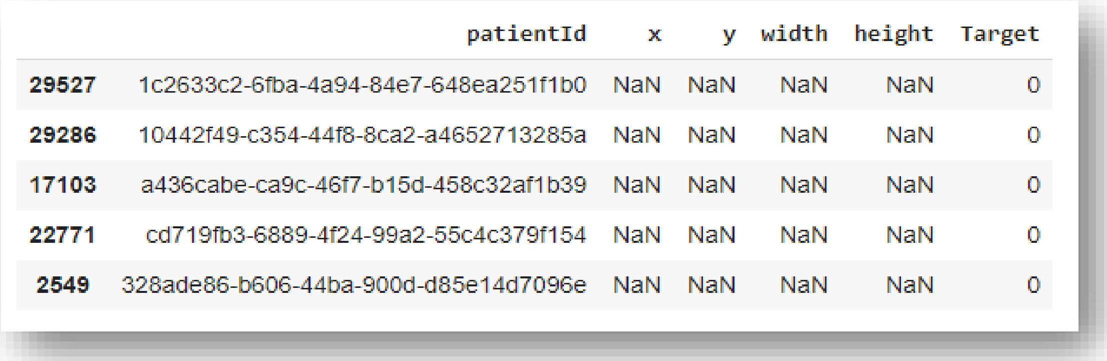
Data Field	Description
patientId	Patient Id. Each patientId corresponds to a unique image
class	Contains one of the below values in each of the rows: • No Lung Opacity / Not Normal • Normal • Lung Opacity

- Training images are provided in stage_2_train_images.zip
- Test images are provided as stage_2_test_images.zip

The training data is provided as a set of patientIds and bounding boxes. Bounding boxes are defined as follows: x-min y-min width height. There is also a binary target column, Target, indicating pneumonia or normal. The objective is to identify if the patient is having pneumonia i.e., predict whether pneumonia exists in a given image. It is done by predicting bounding boxes around areas of the lung. Samples without bounding boxes are negative and contain no definitive evidence of pneumonia. Samples with bounding boxes indicate evidence of pneumonia.

When making predictions, as many bounding boxes as required are to be predicted. There should be only ONE predicted row per image. This row may include multiple bounding boxes.

Sample Data:



	patientId	x	y	width	height	Target
29527	1c2633c2-6fba-4a94-84e7-648ea251f1b0	NaN	NaN	NaN	NaN	0
29286	10442f49-c354-44f8-8ca2-a46527132285a	NaN	NaN	NaN	NaN	0
17103	a436cabe-ca9c-46f7-b15d-458c32af1b39	NaN	NaN	NaN	NaN	0
22771	cd719fb3-6889-4f24-99a2-55c4c379f154	NaN	NaN	NaN	NaN	0
2549	328ade86-b606-44ba-900d-d85e14d7096e	NaN	NaN	NaN	NaN	0

Data Findings:

1. stage_2_train_labels.csv: The CSV file contains PatientId, bounding box details with (x, y) coordinates and width and height that encapsulates the box. It also contains the Target variable. For target variable 0, the bounding box values have NaN values.
2. There are only 26684 images in the image directory, but the csv file contains 30227 rows. There are more rows than the images, which indicates there are duplicate entries for the patientId.

3. Patient Ids are duplicated in different rows of stage_2_train_labels.csv with multiple bounding boxes values. From the data it is clear that the patient is identified with pneumonia at multiple areas in the lungs.
4. We observe that of the total 30227 rows, 9555 rows have non null bounding box values. So, all bounding boxes are either defined or not defined.
5. The total number of patientIds that are identified with Pneumonia are 9555 and it matches the non-null values. It can be inferred from this that all pneumonia data set has bounding boxes defined and for normal patients, no bounding boxes exist
6. The “target” data field has two values “0” and “1” denoting “Normal” and “affected by Pneumonia” respectively.

2. Summary of the Approach to EDA and Pre-processing

2.1 Approach to EDA and Pre-processing

Major steps taken as part of EDA are:

- The DICOM data is explored
- The meta information from the DICOM files are extracted
- Various features of the DICOM images grouped by age, sex are visualized

2.2 EDA and Pre-processing – Steps and Results

Loading the data

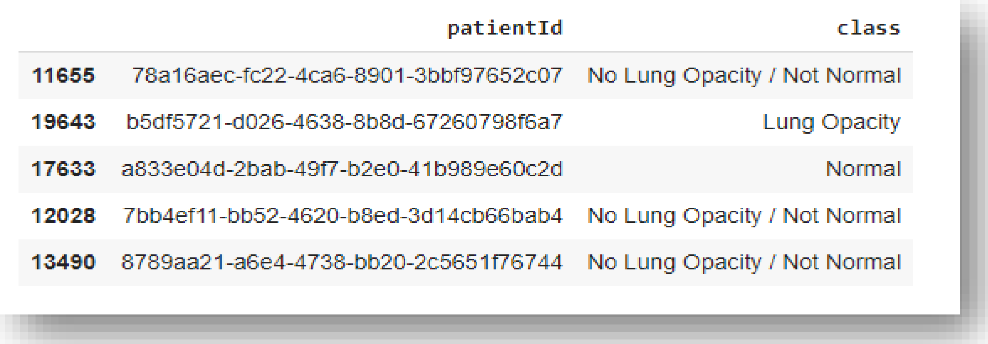
The tabular data provided in the form of csv files are loaded. There are two files:

- Detailed class info - stage_2_detailed_class_info.csv
- Train labels - stage_2_train_labels.csv

Number of rows and columns in these csv files are:

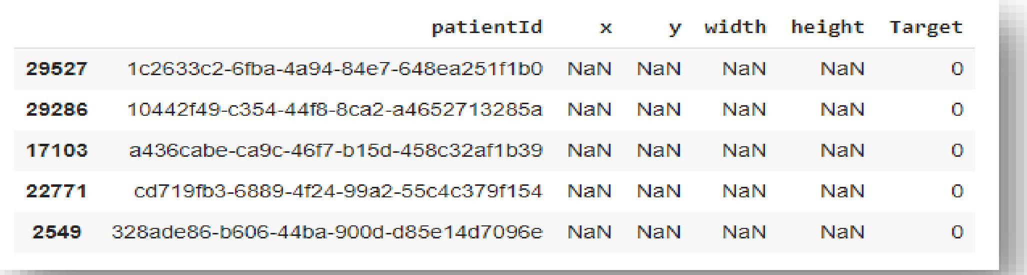
- Detailed class info - rows: 30227, columns: 2
- Train labels - rows: 30227, columns: 6

In **Detailed class info** file, the detailed information about the type of class associated with a certain patient are given. Sample data is given below:



	patientId	class
11655	78a16aec-fc22-4ca6-8901-3bbf97652c07	No Lung Opacity / Not Normal
19643	b5df5721-d026-4638-8b8d-67260798f6a7	Lung Opacity
17633	a833e04d-2bab-49f7-b2e0-41b989e60c2d	Normal
12028	7bb4ef11-bb52-4620-b8ed-3d14cb66bab4	No Lung Opacity / Not Normal
13490	8789aa21-a6e4-4738-bb20-2c5651f76744	No Lung Opacity / Not Normal

In **train labels** file, the patient ID and the window (x min, y min, width and height of the) containing evidence of pneumonia are given. Below image shows sample data.

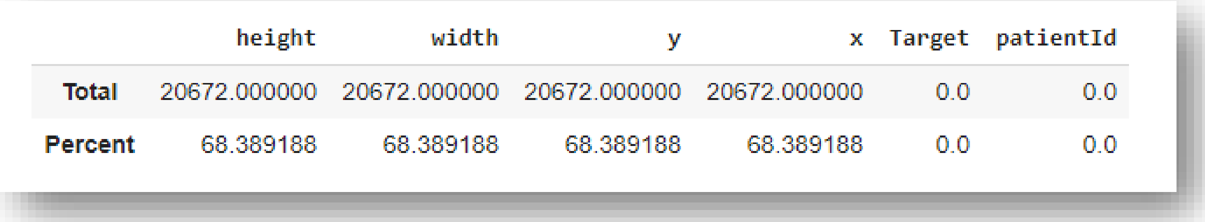


	patientId	x	y	width	height	Target
29527	1c2633c2-6fba-4a94-84e7-648ea251f1b0	NaN	NaN	NaN	NaN	0
29286	10442f49-c354-44f8-8ca2-a4652713285a	NaN	NaN	NaN	NaN	0
17103	a436cabe-ca9c-46f7-b15d-458c32af1b39	NaN	NaN	NaN	NaN	0
22771	cd719fb3-6889-4f24-99a2-55c4c379f154	NaN	NaN	NaN	NaN	0
2549	328ade86-b606-44ba-900d-d85e14d7096e	NaN	NaN	NaN	NaN	0

Checking Missing values

Missing information is checked in both the input files.

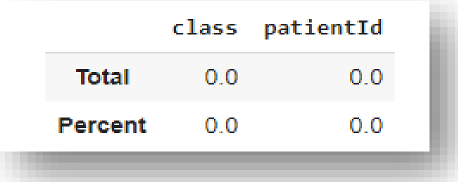
Train labels file: There are 20672 rows that have NaN values, of the total 30227 entries.



	height	width	y	x	Target	patientId
Total	20672.000000	20672.000000	20672.000000	20672.000000	0.0	0.0
Percent	68.389188	68.389188	68.389188	68.389188	0.0	0.0

68.38% of values are missing for x, y, height and width in train labels for target 0 (not Lung opacity)

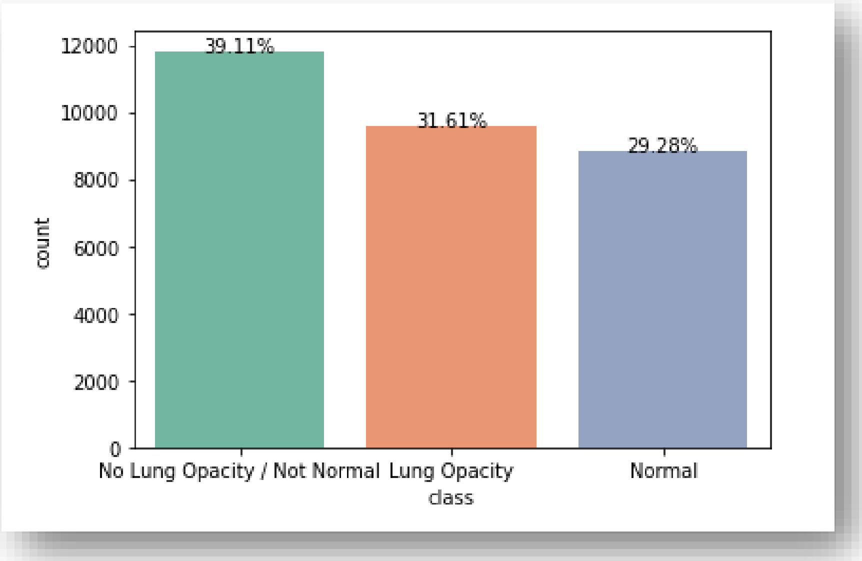
Detailed class info file: There are no missing values

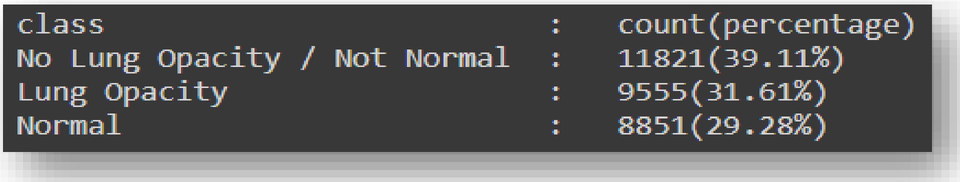


	class	patientId
Total	0.0	0.0
Percent	0.0	0.0

Checking class distribution in Detailed class info file

The class distribution of the three classes – No Lung Opacity / Not Normal, Lung Opacity and Normal is depicted below:





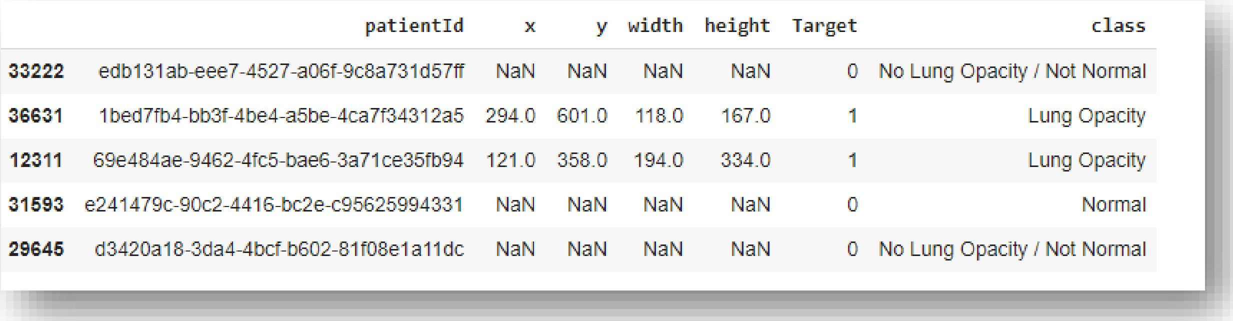
```
class                           :    count(percentage)
No Lung Opacity / Not Normal    :    11821(39.11%)
Lung Opacity                    :    9555(31.61%)
Normal                          :    8851(29.28%)
```

Observations:

- The count of "No Lung Opacity / Not Normal" is higher than the other two classes.
- Count of "Lung Opacity" class is 31.61% of the total count 30227 which is **9555**
- No Lung Opacity / Not Normal and Normal have together the same percent (68.39%) as the percent of missing values for target window in class details information.
- In the train set, the percent of data with value for Target = 1 is therefore 30.9%.

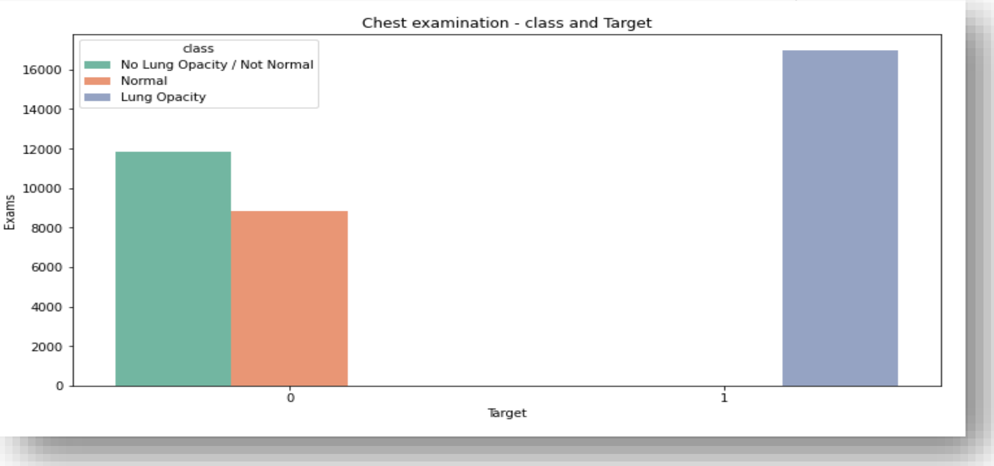
Merging train labels and Detailed class info datasets to get more insights

The two datasets (train labels and Detailed class info) are merged using Patient ID as the merge criteria. The training dataset looks as below after merging.



	patientId	x	y	width	height	Target	class
33222	edb131ab-eee7-4527-a06f-9c8a731d57ff	NaN	NaN	NaN	NaN	0	No Lung Opacity / Not Normal
36631	1bed7fb4-bb3f-4be4-a5be-4ca7f34312a5	294.0	601.0	118.0	167.0	1	Lung Opacity
12311	69e484ae-9462-4fc5-bae6-3a71ce35fb94	121.0	358.0	194.0	334.0	1	Lung Opacity
31593	e241479c-90c2-4416-bc2e-c95625994331	NaN	NaN	NaN	NaN	0	Normal
29645	d3420a18-3da4-4bcf-b602-81f08e1a11dc	NaN	NaN	NaN	NaN	0	No Lung Opacity / Not Normal

Number of examinations for each class detected, grouped by Target value is plotted

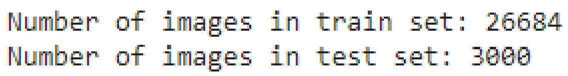


Inferences:

- All chest examinations with Target = 1 (pathology detected) associated with class: Lung Opacity.
- The chest examinations with Target = 0 (no pathology detected) are either of class: Normal or class: No Lung Opacity / Not Normal.

Exploring DICOM image files - Reading training & test files

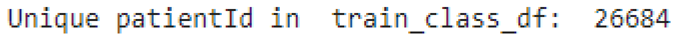
The input DICOM images provided in the folders - stage_2_train_images, stage_2_test_images - are read



```
Number of images in train set: 26684
Number of images in test set: 3000
```

Observations

- The files names are the patient's ID
- Only a reduced number of images are present in the training set (26684), compared with data in train labels dataset (30227)



```
Unique patientId in  train_class_df:  26684
```

- Number of unique patientIds are equal to the number of DICOM images in the train set

Extracting the inner details of single DICOM image and processing the information

Single Image is processed for extracting below DICOM information

```
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
```

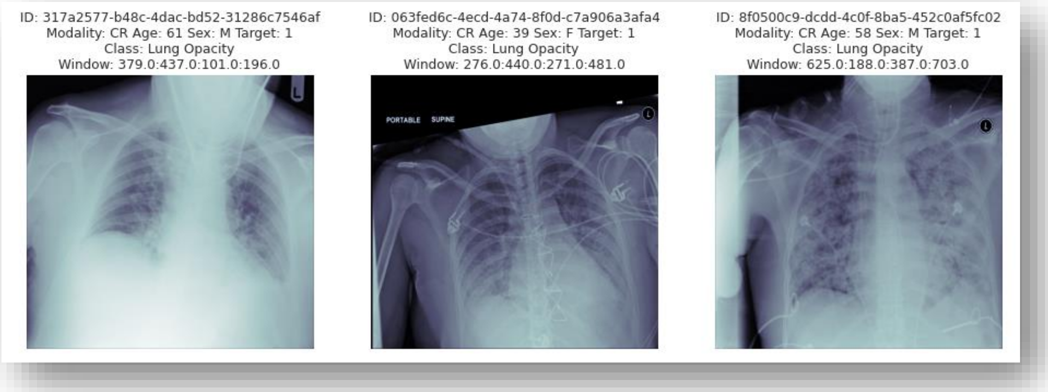
It is observed that some useful information is available in the DICOM metadata with predictive values, for example:

- Patient sex
- Patient age
- Modality
- Body part examined
- View position
- Rows & Columns
- Pixel Spacing

After extracting DICOM information, DICOM images are plotted for further study

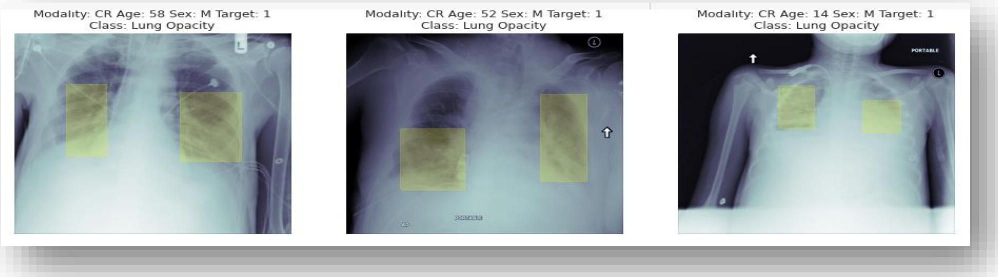
Plotting DICOM images with Target = 1(with Pneumonia)

Sample images plotted with Target = 1 are as below:


ID: 317a2577-b48c-4dac-bd52-31286c7546af
Modality: CR Age: 61 Sex: M Target: 1
Class: Lung Opacity
Window: 379.0:437.0:101.0:196.0
ID: 063fed6c-4ecd-4a74-8f0d-c7a906a3afa4
Modality: CR Age: 39 Sex: F Target: 1
Class: Lung Opacity
Window: 276.0:440.0:271.0:481.0
PORTABLE SUPINE
ID: 8f0500c9-dcdd-4c0f-8ba5-452c0af5fc02
Modality: CR Age: 58 Sex: M Target: 1
Class: Lung Opacity
Window: 625.0:188.0:387.0:703.0

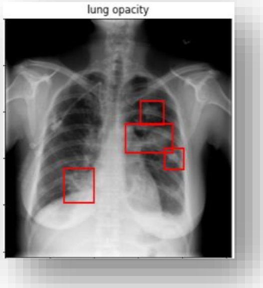
Next step is to represent the images with the overlay boxes superposed. For this, the whole dataset with Target = 1 has to be parsed and all coordinates of the windows showing a Lung Opacity on the same image have to be gathered.

Sample DICOM Images with boxes superposed showing chest areas affected by Pneumonia is as below:


Modality: CR Age: 58 Sex: M Target: 1
Class: Lung Opacity
Modality: CR Age: 52 Sex: M Target: 1
Class: Lung Opacity
Modality: CR Age: 14 Sex: M Target: 1
Class: Lung Opacity
PORTABLE

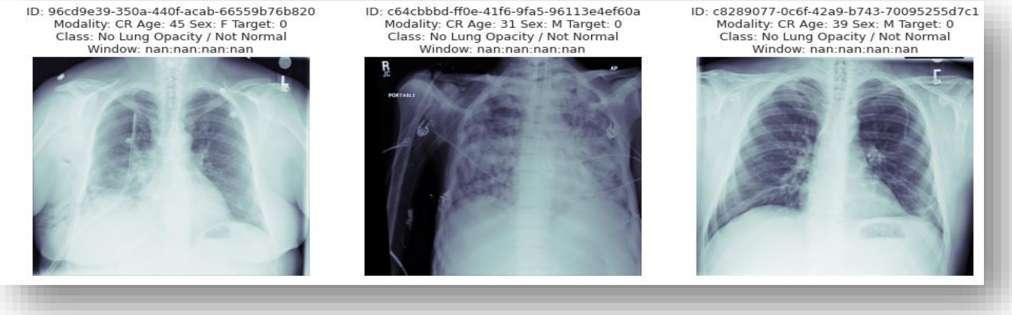
For some of the images with Target=1, we could see multiple areas (boxes) with Lung Opacity.

In the given dataset, the lung opacity is identified in lungs at a maximum of 4 areas.



Plotting DICOM images with Target = 0 (with out Pneumonia)

Sample images plotted with Target = 0 are as below:


ID: 96cd9e39-350a-440f-acab-66559b76b820
Modality: CR Age: 45 Sex: F Target: 0
Class: No Lung Opacity / Not Normal
Window: nan:nan:nan:nan
ID: c64cbbbd-ff0e-41f6-9fa5-96113e4ef60a
Modality: CR Age: 31 Sex: M Target: 0
Class: No Lung Opacity / Not Normal
Window: nan:nan:nan:nan
PORTABLE
AP
ID: c8289077-0c6f-42a9-b743-70095255d7c1
Modality: CR Age: 39 Sex: M Target: 0
Class: No Lung Opacity / Not Normal
Window: nan:nan:nan:nan

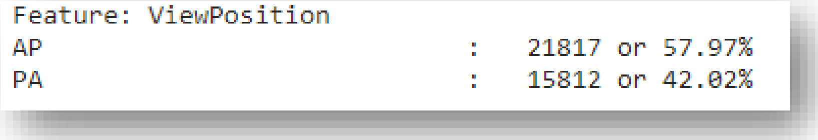
Adding metadata information from DICOM data to training and test datasets

Once DICOM images are plotted and checked, the next step is to parse the DICOM meta information and add it to the train and test datasets

Inferences from DICOM Metadata extracted from given images

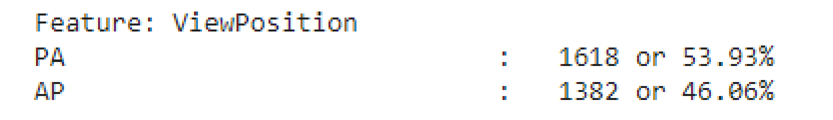
- Only one **modality**, “CR” - Computer Radiography is used
- As per the data given, **body part examined** is only “Chest”
- **View Position** is a radiographic view associated with the Patient Position. Both AP and PA body positions are present in the data. The meaning of these view positions is: AP - Anterior/Posterior; PA - Posterior/Anterior.

  While checking the View Positions distribution in the training dataset, we got below inference:



```
Feature: ViewPosition
AP                                    :   21817 or 57.97%
PA                                    :   15812 or 42.02%
```

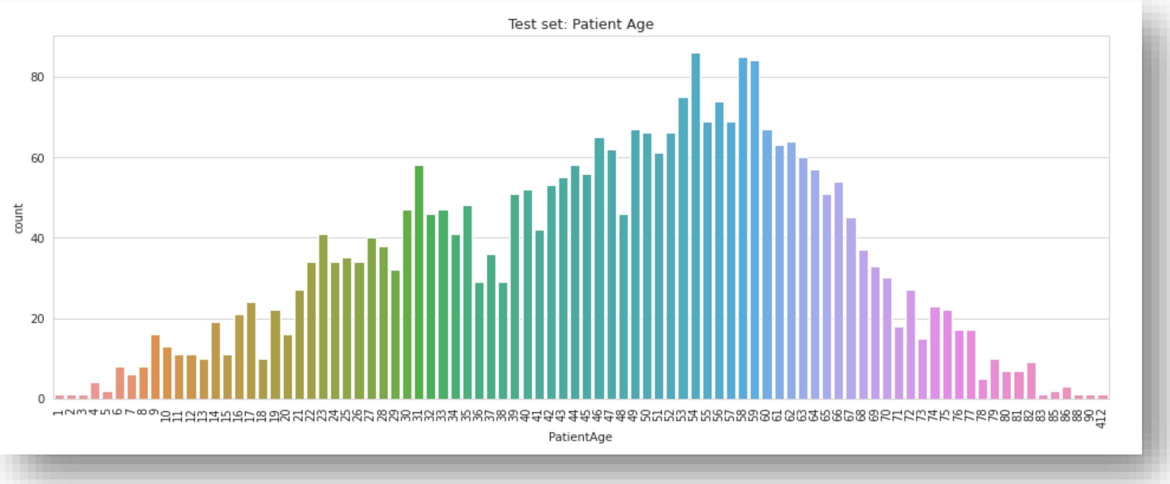
  While checking the View Positions distribution in the test dataset, we got below inference:



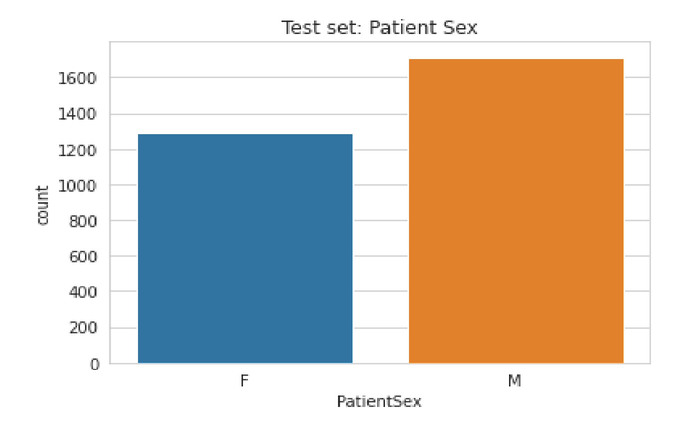
```
Feature: ViewPosition
PA                                    :   1618 or 53.93%
AP                                    :   1382 or 46.06%
```

- Both train and test have only WSD Conversion Type Data. The meaning of this Conversion Type is WSD: Workstation
- Only {Rows: Columns}, {1024:1024} are present in both train and test datasets

- Distribution of patient age for the test data set is as below:



- We can observe a few datasets has the age equal to 412. This could be a typo or a mistake in the value. As the dataset size is small compared to the total data and is not used for model training, we ignore this typo and use the data for training.
    - Distribution of Patient Sex for the test data is shown as below:



- stage_2_sample_submission.csv - Contains patientIds for the test set. Each row in sample submission represents one bounding box per image. This is the input .csv file for creating test dataset

2.3 EDA - Conclusions

After exploring both the tabular and DICOM data, we draw following conclusions.

1. Discovered duplicate patientIds in the tabular data, an indication that the patient infected with Pneumonia at more places in the lungs.
2. There are patients who are infected with Pneumonia at more than one place in the lungs
3. Able to extract meta information from the DICOM data for more detailed analysis
4. Further analyze the distribution of the data with the newly added features from DICOM metadata

All these findings are useful for building a model.

3. Deciding Models and Model Building

3.1 Model Approach

The objective of our model is to detect the object classification along with identification of the object location. In simple words, the aim of detection technique is to determine the classification as well as the localization of the infection. The solution requires model building based on the Classification model approach by using Computer Vision - Object Detection techniques.

The data set is huge and due to restricted hardware resources, time taken for model training, memory consumption etc., of 26684 images data set, we plan to use 2000 images for training and 500 images for validation.

Bounding Box for Object Classification

The primary goal will be the detection of bounding boxes consisting of a binary classification e.g. the presence or absence of pneumonia. However, in addition to the binary classification, the dataset without pneumonia is further categorized into normal or no lung opacity / not normal. This extra third class indicates that while pneumonia was determined not to be present, there was nonetheless some type of abnormality in the image; and oftentimes this finding may mimic the appearance of true pneumonia. This extra class is provided as supplemental information to help improve algorithm accuracy if needed; generation of this separate class will not be a formal metric used to evaluate performance.

3.2 Model Identification

There are multiple approaches to solve an object detection problem. They are broadly classified into a 2-stage, a single stage without anchor boxes and a single stage with anchor boxes. There are numerous algorithms available for object detection with subtle variations. The popular models in this area are Mask RCNN, YOLO, SSD. As part of the project our aim is to work on 1 state-of-the-art model along with the base U-Net model.

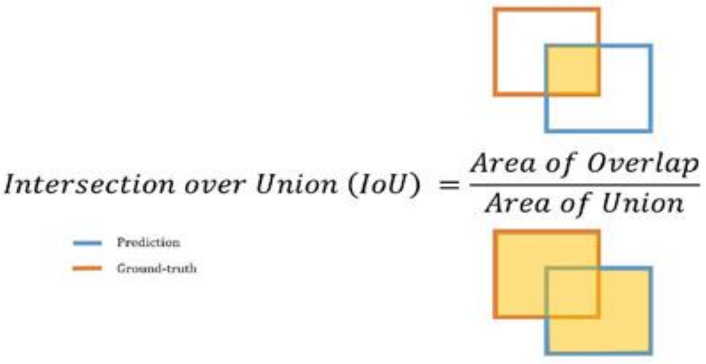
Base Model: As a base model we choose to utilize the very popular UNET architecture for semantic segmentation. The UNET was developed by Olaf Ranneberger et al. for Bio Medical Image Segmentation. The architecture contains two parts. First part is the encoder, which is used to capture the context in the image. It is crucial for extracting the right features and hence we have used the most popular ImageNet competition winner, VGG16. The second part is the decoder, symmetric to the encoder, which is used to enable precise location using transposed convolutions technique. The encoder uses transfer learning while the model will be trained for decoder block.

YOLO V3: You Only Look Once is one of the most popular and a defacto model used for object detection. Darknet-53 model is used as the backbone which consists of 53 layers. It is further added 53 more layers making YOLO a 106-layer deep convolutional neural network. YOLO uses only convolutional layers and hence it is a Fully Convolutional network. We are researching to replace the backbone of the Darknet53 with DenseNet121 as a feature extractor for YOLO and train the model with YOLO architecture creating another version of state-of-the-art model. The reason to use DenseNet121 and more details are planned for the final report.

3.3 Model Metrics

While there are different model metrics available to measure the model performance, two of them are widely used for object detection models. Intersection Over Union or simply IOU and Mean Average Precision, called mAP.

Intersection Over Union: IOU is a measure of the magnitude of the overlap between two bounding boxes. It calculates the size of the overlap between the two objects, divided by the total area of the two objects combined. It can be visualized as follows.

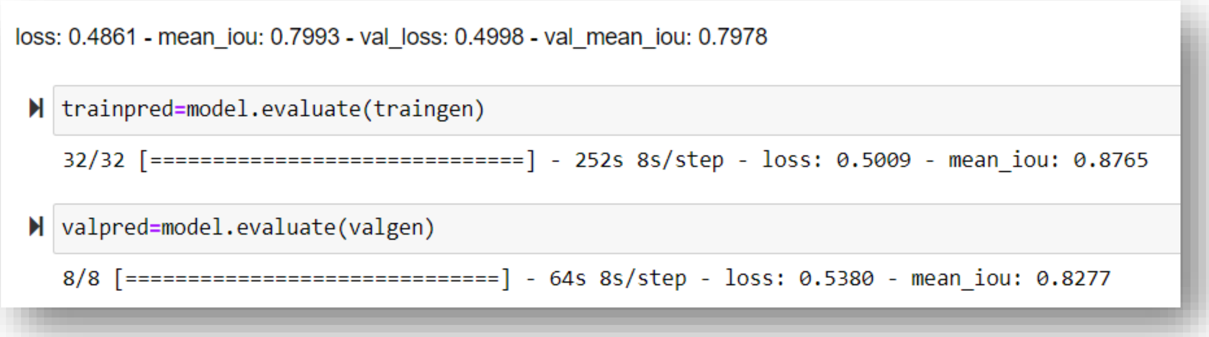

Intersection over Union (IoU) = Area of Overlap / Area of Union
Prediction
Ground-truth

Mean Average Precision: Before understanding mAP, let us understand Precision and Recall. In classification models, Precision is defined as the trueness of correct predictions and recall is defined as how good the prediction was. Precision and recall are not particularly useful metrics when used in isolation and f1 score is generally used. Due to the importance of both precision and recall, there is a precision-recall curve that shows the tradeoff between both the values for different thresholds. The average precision (AP) is a way to summarize the precision-recall curve into a single value representing the average of all precisions. The mean of the APs for all the classes is called the mAP which is evaluated against different values of IOU. As we have one class only, we can consider AP as mAP. Again, we plan to use this metrics for model evaluation and comparison and will be presented in our final report.

3.4 Model Training Summary

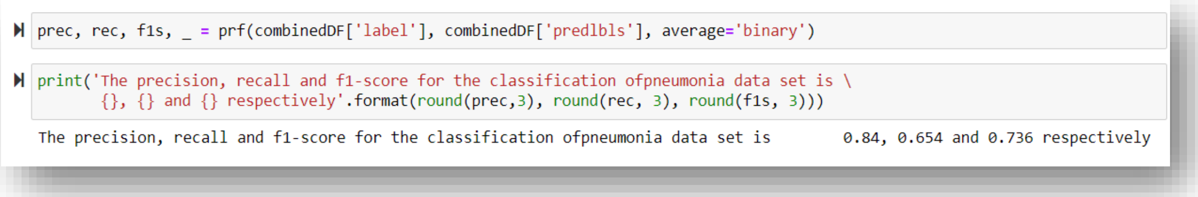
Summary of Model performance during model training is as follows.

UNET model: With initial hyperparameters, the model had performed decently. The training and validation mean IOU during model training are 79.93% and 79.78% respectively. And during the model evaluation, the mean IOU on the training and validation data are 87.65% and 82.77% respectively. The mean IOU for both train and validation are pretty close; hence the model is not an overfit.



```
loss: 0.4861 - mean_iou: 0.7993 - val_loss: 0.4998 - val_mean_iou: 0.7978

trainpred=model.evaluate(traingen)

32/32 [==============================] - 252s 8s/step - loss: 0.5009 - mean_iou: 0.8765

valpred=model.evaluate(valgen)

8/8 [==============================] - 64s 8s/step - loss: 0.5380 - mean_iou: 0.8277
```

For classification, we calculated the precision, recall and F1 score the predictions and have 0.84, 0.654 and 0.736 respectively.



```
prec, rec, f1s, _ = prf(combinedDF['label'], combinedDF['predlbls'], average='binary')

print('The precision, recall and f1-score for the classification ofpneumonia data set is \
        {}, {} and {} respectively'.format(round(prec,3), round(rec, 3), round(f1s, 3)))

The precision, recall and f1-score for the classification ofpneumonia data set is        0.84, 0.654 and 0.736 respectively
```

We will also evaluate the mean IOU for the bounding boxes predicted from the model in the final report.

Sl No	Model	Training	Validation	Precision	Recall
1	UNET (VGG16 backbone)	0.877	0.828	0.840	0.654
2	Yolo v3 (DenseNet121 backbone)				

4. Improving Model Performance

4.1 Feature Extraction

In computer vision, a feature is a measurable piece of data in the image which is a unique to that specific object. It may be a specific shape such as a line, edge or a distinct color in an image or an image segment. A good feature is used to distinguish objects from one another. Hence, feature extraction is a crucial task in the success for model performance.

Although the features are extracted with different variants of pre-trained CNN models, we are trying with different approaches for better model performance and faster results. For UNET, we used the VGG16 model as our feature extractor and trained the last few layers and extracted 4096 features. Considering the number of parameters and faster performance, we plan to use DenseNet model architecture for feature extraction.

4.2 Data Manipulation/Augmentation

Data augmentation implies increasing the amount of training data by applying transformations to both image and contour (so we could calculate the true bounding box). Data augmentation is typically applied as a pre-processing step to the model.

There are several data augmentation techniques like random cropping (with constraints), expansion, horizontal flip, image shearing, support bounding boxes, resize (with random interpolation), and color jittering (including brightness, hue, saturation, and contrast).

Data augmentation techniques can also be used to improve object detection models, although they improve single-stage detectors more than the multi-stage detectors as multi-stage detectors like Faster-RCNN use candidate object proposals that are sampled from large pool of generated ROIs, the detection results are produced by repetitive cropping of regions of future maps. Due to this cropping, multi-stage models do not use random cropped input images, thereby they do not require detailed geometric augmentations to be applied during training.

There are several pre-defined image augmentation packages available for python like ImageDataGenerator, imgaug, Albumentations etc. that help us quickly perform the required augmentations.

We have considered performing image augmentation using the "Albumentations" package. This package is based on OpenCV, NumPy and imgaug.

This package efficiently implements a rich variety of image transform operations that are optimized for performance. Albumentations supports creating multiple images for computer vision tasks. We had used the Albumentations package and performed some techniques like rotation, horizontal flip, blur, brightness etc., on the images for model training.

4.3 Model Improvements

Model improvements can be achieved in fine tuning various hyper parameters. These hyper parameters vary from model to model. There are few basic parameters that are valid for every model like Learning rate, weight decay, optimizers selection and so on.

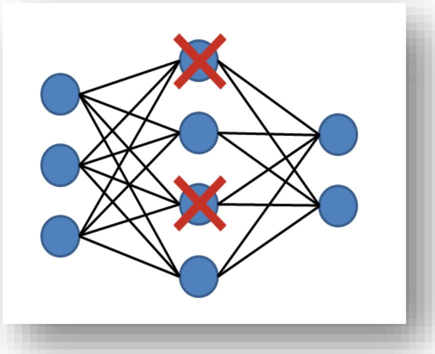
Hyperparameters are the variables which determine the network structure (Eg: Number of Hidden Units) and the variables which determine how the network is trained (Eg: Learning Rate Scheduler). Hyperparameters are set before training (before optimizing the weights and bias).

Hyperparameters related to Network Structure

Number of Hidden Layers and units

Hidden layers are the layers between input layer and output layer. Many hidden units within a layer with regularization techniques can increase accuracy. Smaller number of units may cause underfitting.

Dropout



Dropout is a regularization technique to avoid overfitting (increase the validation accuracy) thus increasing the generalizing power. Random neurons are cancelled

Generally, use a small dropout value of 20%-50% of neurons with 20% providing a good starting point. A probability too low has minimal effect and a value too high results in under-learning by the network.

Use a larger network. You are likely to get better performance when dropout is used on a larger network, giving the model more of an opportunity to learn independent representations.

Network Weight Initialization

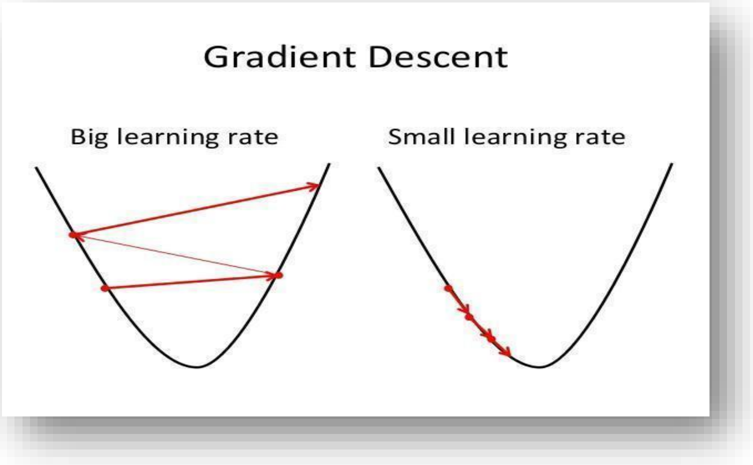
Ideally, it may be better to use different weight initialization schemes according to the activation function used on each layer. Mostly uniform distribution is used.

Activation function

Activation functions are used to introduce nonlinearity to models, which allows deep learning models to learn nonlinear prediction boundaries.
Sigmoid is used in the output layer while making binary predictions. Softmax is used in the output layer while making multi-class predictions.

Hyperparameters related to Training Algorithm

Learning Rate



The learning rate defines how quickly a network updates its parameters. Low learning rate slows down the learning process but converges smoothly. Larger learning rate speeds up the learning but may not converge. Usually a decaying Learning rate is preferred.

Sharp learning rate transition may cause the optimizer to re-stabilize the learning momentum in the following iterations.

Using a cosine scheduler (where the learning rate decreases slowly) with proper warmup (two epochs) can give better validation accuracy than using a step scheduler.

The learning rate is one of the hyperparameters that most affects performance. If we can only adjust one hyperparameter to obtain better results, then the best choice is the learning rate.

Momentum

Momentum helps to know the direction of the next step with the knowledge of the previous steps. It helps to prevent oscillations. A typical choice of momentum is between 0.5 to 0.9.

Number of epochs

Number of epochs is the number of times the whole training data is shown to the network while training. Increase the number of epochs until the validation accuracy starts decreasing even when training accuracy is increasing(overfitting).

Batch Size Normalization

Mini batch size is the number of sub samples given to the network after which parameter update happens. A good default for batch size might be 32. Batch size of 64, 128, 256 can also be tried

Optimizer

Adam is chosen as the optimizer since it’s just the best one to minimize loss value in most neural network tasks.

Custom Loss Function

We have used Binary Cross Entropy loss in conjunction with the IOU Loss.
We will also evaluate the model by modifying the loss function parameters, using only BCE loss or using only IOU loss and compare the model performance.

Hyper parameters planned for tuning:

Few of the model specific parameters that we are planning to tune as we move along in the project are provided below. This is only a subset of the hyper parameter tuning actions we have planned. We will update the list based on further progress of the project and provide in the final report.

UNET VGG16:

- Adding additional dropout or batch normalization layer(s)
- Play around with the last few layers for training with the custom images
- Reduced Learning Rate: Initial model used 0.01
- Use different optimizers
  - Adam optimizer which is one of the best optimizers currently
  - SGD optimizer
- Number of epochs, Batch size

Yolo V3:

- Custom architecture – evaluate different backbones
- Add additional dropout or batch normalization layer(s)
- Learning Rate and Optimizer hyper parameters will be modified
- Create Custom Loss functions
- Number of epochs, Batch size

Hyperparameter Tuning evaluation

To compare model performance, we are planning to compare the following metrics:

- Accuracy
- Mean IOU
- IOU loss
- Binary cross entropy loss
- Mean Average Precision

5. Next Steps

- Fine tune the base model UNET (VGG16 Backbone) to improve the metrics
- Finalize the approach to determine mAP (Mean Average Precision) or Mean IOU for our models.
- Train YOLO V3 with DenseNet121 as the backbone
- Optimize all the models by hyper parameter tuning considering following parameters
    - Learning Rate
    - Optimizer
    - Batch size
    - Number of epochs
    - Evaluating various options with custom layers
    - Creating mean IOU/mean average precision metrics
- Compare models by capturing following metrics as applicable:
    - Mean IOU/ Mean Average Precision
    - Binary cross entropy loss
- In addition to the models specified above (UNET-VGG16 and YOLOV3-DenseNet), we also plan to perform pneumonia detection using other models and capture our findings as part of the final submission.