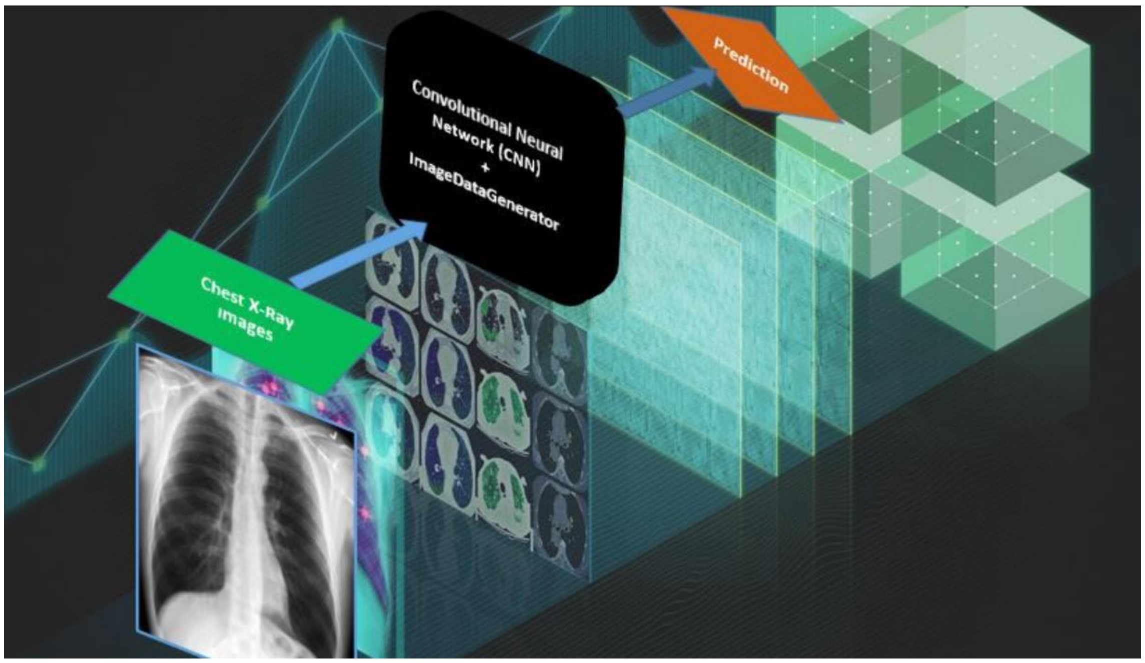


Pneumonia Detection Challenge

Final Report

AIML capstone project

ABSTRACT
Building model that helps clinicians to detect Pneumonia from the DICOM image taken for the patient

AIML-Jul 21A-Group 6 – CV 1

	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	NA

1 Project Repository - Links

GitHub repository:

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

Reports:

Interim Report:

https://github.com/ashish090798/GL-Capstone-Project/blob/c01a1695d41e41318b860b2a21441927a8d23767/AIML-Jul%2021A-Group%206%20%E2%80%93%20CV1-InterimReport.pdf

EDA:

1. https://github.com/ashish090798/GL-Capstone-Project/blob/489017b481d251869beb2c7a6888f57288b9c880/Capstone_Project_RSNA_Pneumonia_EDA_&_Data_prep.ipynb
2. https://github.com/ashish090798/GL-Capstone-Project/blob/c01a1695d41e41318b860b2a21441927a8d23767/Capstone_Project_RSNA_Pneumonia_DenseNet_121.ipynb

UNet VGG16:

https://github.com/ashish090798/GL-Capstone-Project/blob/489017b481d251869beb2c7a6888f57288b9c880/Capstone_Project_RSNA_Pneumonia_unet_vgg16.ipynb

MASK RCNN:

https://github.com/ashish090798/GL-Capstone-Project/blob/5a2a5e94d40b3fa71d11472a497842af5be77eca/capstone_project_rsna_pneumonia_maskrcnn_final.ipynb

MASK RCNN Kaggle:

https://www.kaggle.com/code/ashishtiwari98/object-detection-maskrcnn-rsna-pneumonia

Model deployment files:

https://github.com/ashish090798/GL-Capstone-Project/tree/main/Deployment%20Files

AIML-Jul 21A-Group 6 – CV 1

Table of Contents

1 PROJECT REPOSITORY - LINKS 2

2 SUMMARY OF THE PROBLEM STATEMENT, DATA AND FINDINGS 6

2.1 PROBLEM STATEMENT 6

2.2 DATA & FINDINGS 7

3 SUMMARY OF THE APPROACH TO EDA AND PRE-PROCESSING 11

3.1 APPROACH TO EDA AND PRE-PROCESSING 11

3.2 EDA AND PRE-PROCESSING – STEPS AND RESULTS 11

3.3 EDA CONCLUSIONS 22

4 DECIDING MODELS AND MODEL BUILDING 24

4.1 MODEL APPROACH 24

4.2 MODEL IDENTIFICATION 24

4.3 MODEL METRICS 25

4.4 MODEL TRAINING SUMMARY 25

5 IMPROVING MODEL PERFORMANCE 27

5.1 FEATURE EXTRACTION 27

5.2 DATA MANIPULATION/AUGMENTATION 27

5.3 MODEL IMPROVEMENTS 28

6 COMPARISON OF MODELS, BENCHMARKING AND VISUALIZATION 33

6.1 APPROACH TO SELECT FINAL MODEL BASED ON OUTCOMES 33

6.2 MODEL EVALUATION 36

7. **Model Deployment** 37

7.1 Basic approach 37

7.2 Model architecture 37

7.3 Model testing 38

8 LIMITATIONS AND SUGGESTED IMPROVEMENTS 409

7.1 LIMITATIONS OF OUR SOLUTION 409

7.2 SUGGESTED IMPROVEMENTS 409

9 OUR LEARNING 40

10 FINAL NOTE. 43

11 ACKNOWLEDGEMENTS 44

Figure 1: Sample Data 9
Figure 2: Detailed class info 12
Figure 3: Train labels 12
Figure 4: Training column values info 12
Figure 5: Training missing values 13
Figure 6: Train class distribution 13
Figure 7: Train class distribution values 14
Figure 8: Merged train labels and detailed csv 14
Figure 9: Chest Examination distribution 15
Figure 10: Train and Test distribution 15
Figure 11: Unique Patient Ids 16
Figure 12: Pneumonia image display 19
Figure 13: Pneumonia image display with bounding boxes 19
Figure 14: Highest number places, pneumonia detected in an image 20
Figure 15: Normal image 20
Figure 16:Train feature View position 21
Figure 17: Test feature View position 21
Figure 18: Patient age distribution 21
Figure 19: Gender distribution 22
Figure 20: Intersection Over Union (IOU) 25
Figure 21: Base model initial results 26
Figure 22: Base model precision, recall 26
Figure 23: Dropout 29
Figure 24: Gradient descent 30
Figure 32: Top 3 best performing models 33
Figure 33: UNet VGG16 best model prediction 34
Figure 34: MASK RCNN hyper parameter combination performance 35
Figure 35: MASK RCNN performance graph 35
Figure 36: MASK RCNN top 3 best performing models 35
Figure 37: MASK RCNN prediction GT vs Predictions 36

2 Summary of the problem statement, Data and findings

2.1 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.

2.2 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

Table 1: Stage 2 Train Labels CSV

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

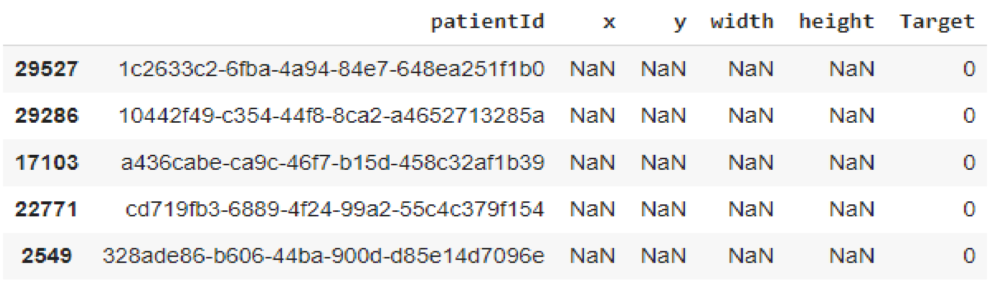
Data Field	Description
patientId	Patient Id. Each patientId corresponds to a unique image
class	Contains one of the below values in each of the rows: • No Lung Opacity / Not Normal • Normal • Lung Opacity

Table 2: Stage 2 detailed class info CSV

- 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.



	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

Figure 1: Sample Data

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.

3 Summary of the Approach to EDA and Pre-processing

3.1 Approach to EDA and Pre-processing

Major steps taken as part of EDA are:

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

3.2 EDA and Pre-processing – Steps and Results

Loading data:

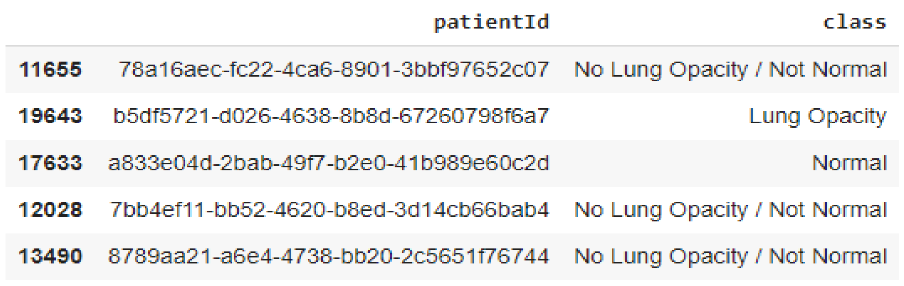
The tabular data provided in the form of csv files are loaded. There are two files:

- 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

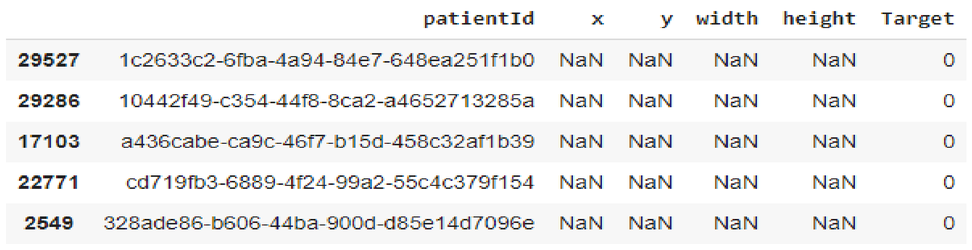
In Detailed class info file, the detailed information about the type of class associated with a certain patient are given. Sample data is given below:



	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

Figure 2: Detailed class info

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.



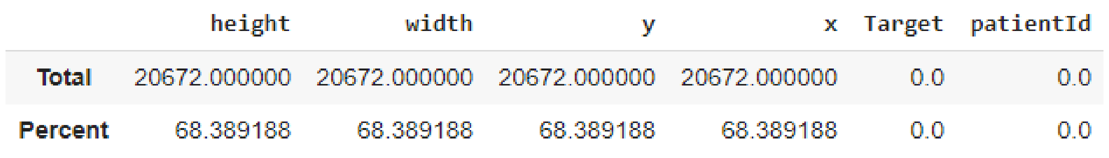
	patientId	x	y	width	height	Target
29527	1c2633c2-6fba-4a94-84e7-648ea251f1b0	NaN	NaN	NaN	NaN	0
29286	10442f49-c354-44f8-8ca2-a4652713285a	NaN	NaN	NaN	NaN	0
17103	a436cabe-ca9c-46f7-b15d-458c32af1b39	NaN	NaN	NaN	NaN	0
22771	cd719fb3-6889-4f24-99a2-55c4c379f154	NaN	NaN	NaN	NaN	0
2549	328ade86-b606-44ba-900d-d85e14d7096e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

Figure 3: Train labels

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.

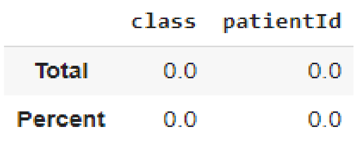


	height	width	y	x	Target	patientId
Total	20672.000000	20672.000000	20672.000000	20672.000000	0.0	0.0
Percent	68.389188	68.389188	68.389188	68.389188	0.0	0.0

Figure 4: Training column values info

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

Figure 5: Training missing values

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:

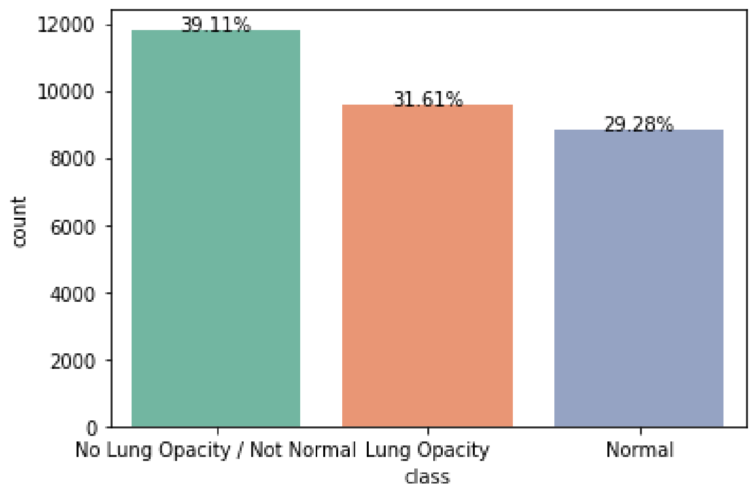
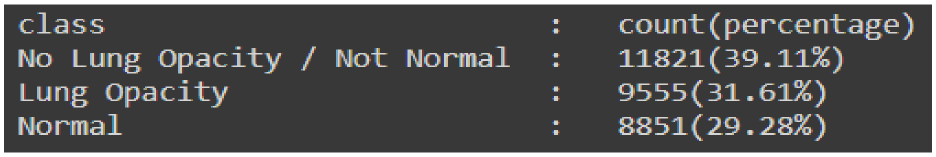


Figure 6: Train class distribution



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

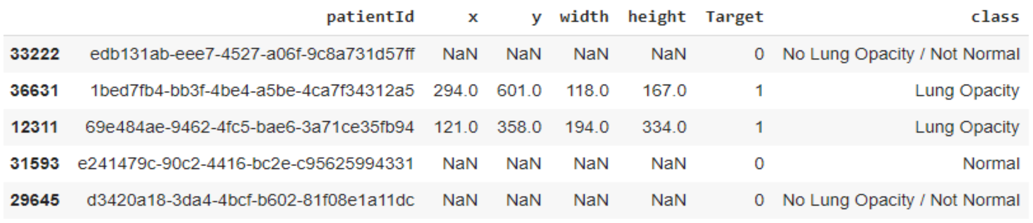
Figure 7: Train class distribution values

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 of 30227 which is **9555**
- In the train set, the percent of data with value for Target = 1 is 30.9%
- 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.

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

Figure 8: Merged train labels and detailed csv

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

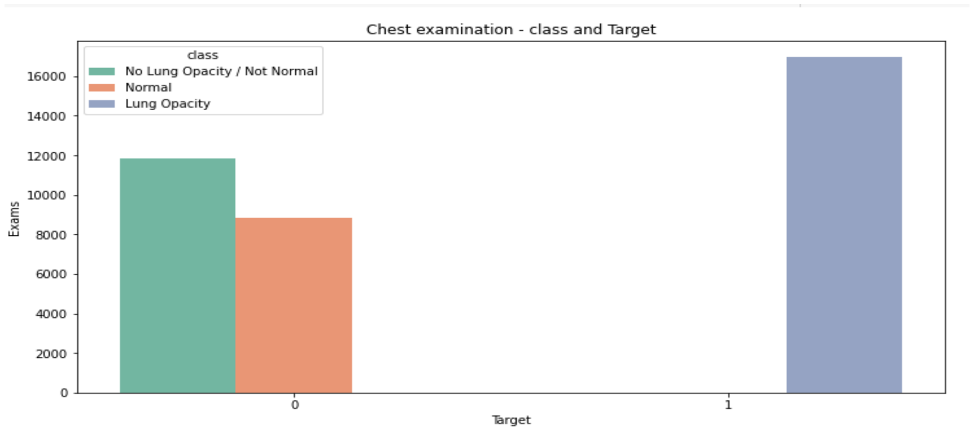


Figure 9: Chest Examination distribution

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
```

Figure 10: Train and Test distribution

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

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

Figure 11: Unique Patient Ids

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:

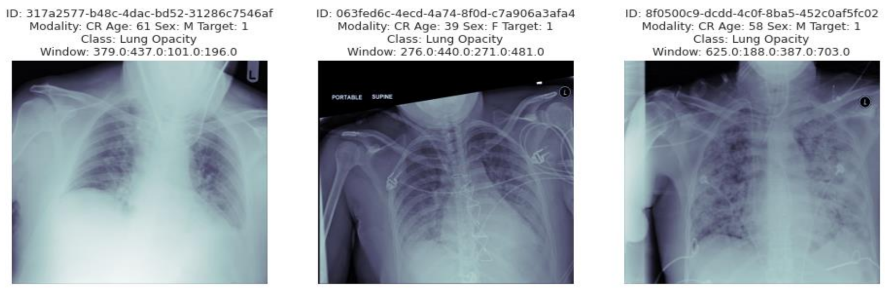

ID: 317a2577-b48c-4dac-bd52-31286c7546af
Modality: CR Age: 61 Sex: M Target: 1
Class: Lung Opacity
Window: 379.0:437.0:101.0:196.0
ID: 063fed6c-4ecd-4a74-8f0d-c7a906a3afa4
Modality: CR Age: 39 Sex: F Target: 1
Class: Lung Opacity
Window: 276.0:440.0:271.0:481.0
PORTABLE
SUPINE
ID: 8f0500c9-dcdd-4c0f-8ba5-452c0af5fc02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

Figure 12: Pneumonia image display

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:

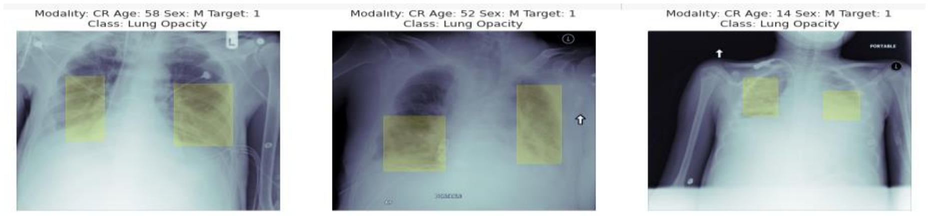

Modality: CR Age: 58 Sex: M Target: 1
Class: Lung Opacity
Modality: CR Age: 52 Sex: M Target: 1
Class: Lung Opacity
Modality: CR Age: 14 Sex: M Target: 1
Class: Lung Opacity
PORTABLE

Figure 13: Pneumonia image display with bounding boxes

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.

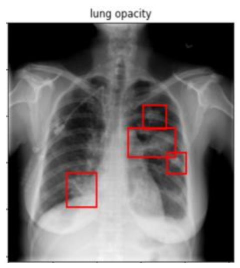


Figure 14: Highest number places, pneumonia detected in an image

<u>Plotting DICOM images with Target = 0 (without Pneumonia)</u>

Sample images plotted with Target = 0 are as below:

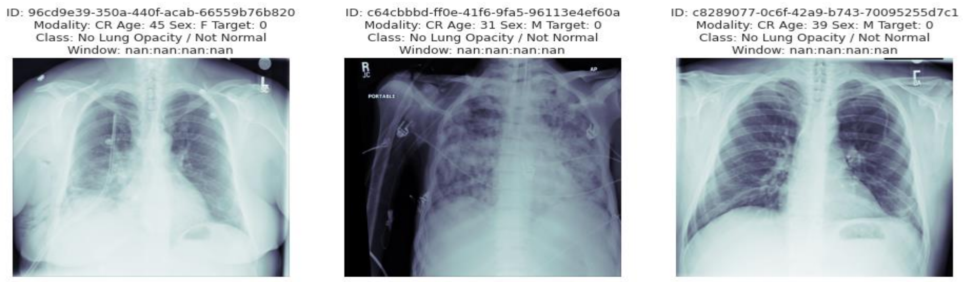

ID: 96cd9e39-350a-440f-acab-66559b76b820
Modality: CR Age: 45 Sex: F Target: 0
Class: No Lung Opacity / Not Normal
Window: nan:nan:nan:nan
ID: c64cbbbd-ff0e-41f6-9fa5-96113e4ef60a
Modality: CR Age: 31 Sex: M Target: 0
Class: No Lung Opacity / Not Normal
Window: nan:nan:nan:nan
ID: c8289077-0c6f-42a9-b743-70095255d7c1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

Figure 15: Normal image

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

<u>Inferences from DICOM Metadata extracted from given images</u>

- 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%
```

Figure 16:Train feature View position

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%
```

Figure 17: Test feature View position

- 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:

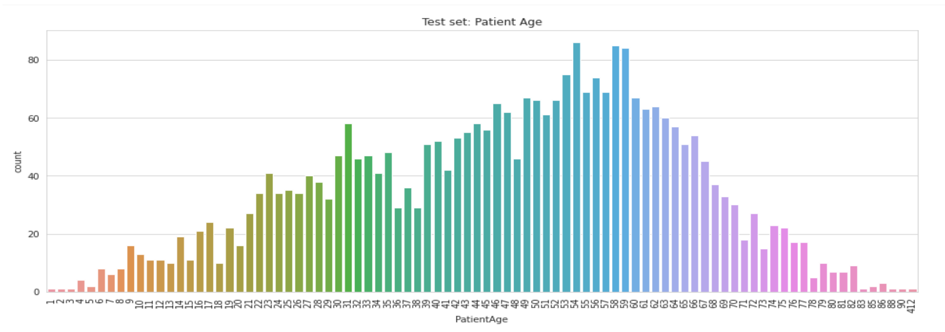


Figure 18: Patient age distribution

- 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:

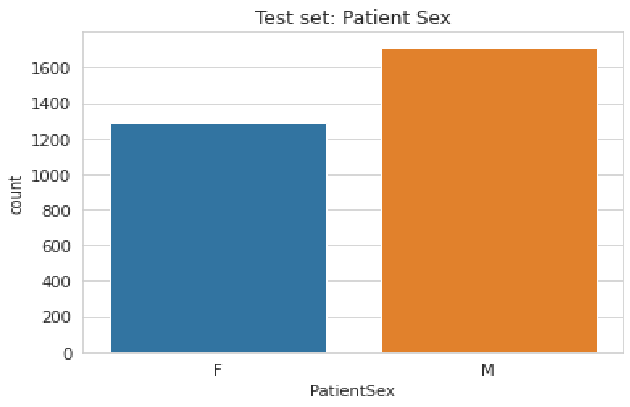


Figure 19: Gender distribution

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

3.3 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.

4 Deciding Models and Model Building

4.1 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 the restricted hardware resources, time taken for model training, memory consumption etc., of 26684 images data set, we have used 2000 images for training (Opacity - 1500 and normal - 500) and 500 images for validation (Opacity - 350, normal - 150).

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.

4.2 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.

4.3 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, simply called mAP. We have used mean IOU as our metric.

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.

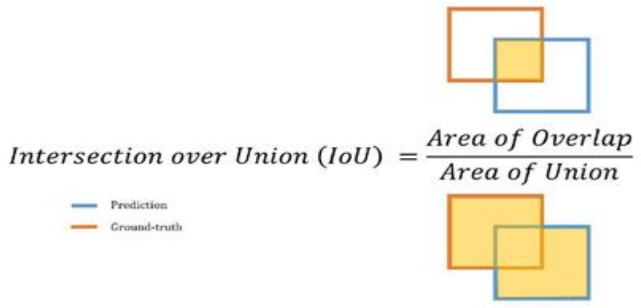
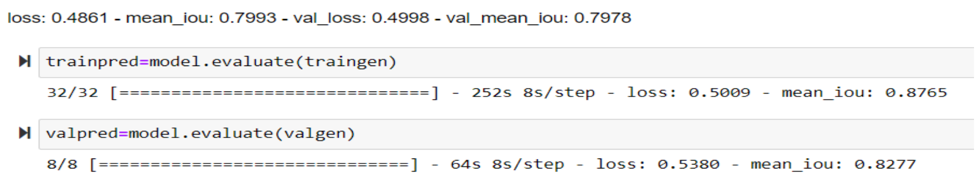

Intersection over Union (IoU) = Area of Overlap / Area of Union
Prediction
Ground-truth

Figure 20: Intersection Over Union (IOU)

4.4 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 of mean IOU during model training are 0.7993 and 0.7978 respectively. During the model evaluation, the mean IOU on the training and validation data are 0.8765 and 0.8277 respectively. The mean IOU for both train and validation are deviated with .05; the model is an overfit.



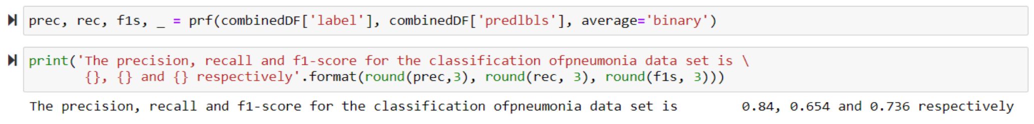
```
loss: 0.4861 - mean_iou: 0.7993 - val_loss: 0.4998 - val_mean_iou: 0.7978

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
```

Figure 21: Base model initial results

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
```

Figure 22: Base model precision, recall

5 Improving Model Performance

5.1 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 for feature extraction. Considering the number of parameters and faster performance, we used MASK RCNN model architecture with ResNet50 as backbone for feature extraction.

5.2 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 feature 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 predefined 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.

5.3 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.

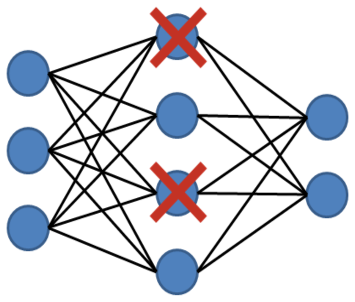


Figure 23: Dropout

- 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**

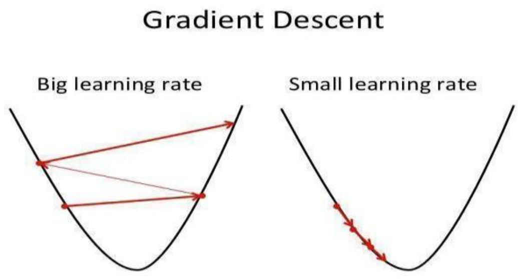


Figure 24: Gradient descent

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, Adagrad, RMSProp, SGD are chosen as the optimizers to train the model as these are one of the best ones to minimize the loss value in most neural network tasks.

- **Custom Loss Function**
  We have used Binary Cross Entropy loss in conjunction with the Mean 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 used for tuning:

Model specific parameters that are used to tune in the project are provided below.

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

Mask RCNN:

- 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
- Detection Minimum confidence, RPN threshold

Hyperparameter Tuning evaluation

To compare model performance, we used the following metrics:

- Mean IOU
- IOU loss
- Binary cross entropy loss
- Precision/Recall
- F1 Score

6 Comparison of Models, Benchmarking and Visualization

6.1 Approach to select final model based on outcomes

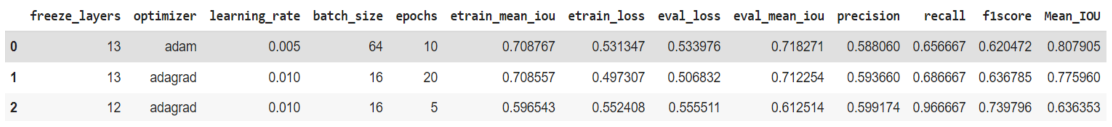
We ran hyper parameter tuning for the two models UNet VGG16 and MASK RCNN. Below we describe more details on the model execution and its evaluation.

UNet VGG16:

The hyper parameters considered for fine tuning the UNet VGG16 model are freeze layers of VGG16, optimizer, learning rate, batch size and epochs. We tried for multiple runs and with multiple values for each parameter. Following are some visualizations of the run. As part of training the MASK RCNN, we changed the approach in dataset selection, preprocessing techniques.

We performed 5 different runs with many different hype parameter combinations (200+), and took top 30 hyper parameter combinations that are performing well.

Following are the predictions for the top 3 chosen hyper parameter combinations of UNet VGG16.



	freeze_layers	optimizer	learning_rate	batch_size	epochs	etrain_mean_iou	etrain_loss	eval_loss	eval_mean_iou	precision	recall	f1score	Mean_IOU
0	13	adam	0.005	64	10	0.708767	0.531347	0.533976	0.718271	0.588060	0.656667	0.620472	0.807905
1	13	adagrad	0.010	16	20	0.708557	0.497307	0.506832	0.712254	0.593660	0.686667	0.636785	0.775960
2	12	adagrad	0.010	16	5	0.596543	0.552408	0.555511	0.612514	0.599174	0.966667	0.739796	0.636353

Figure 25: Top 3 best performing models

From the results captured in the above table, we have the best model performance with a mean IOU of 0.808 and an F1 score of 0.621 when below hyper parameters are selected.

- Freeze layers = 13
- optimizer = Adam
- LR = 0.005
- batch size = 64
- epoch = 10

Hence, we choose the best model for UNet VGG16 with the above hyper parameters

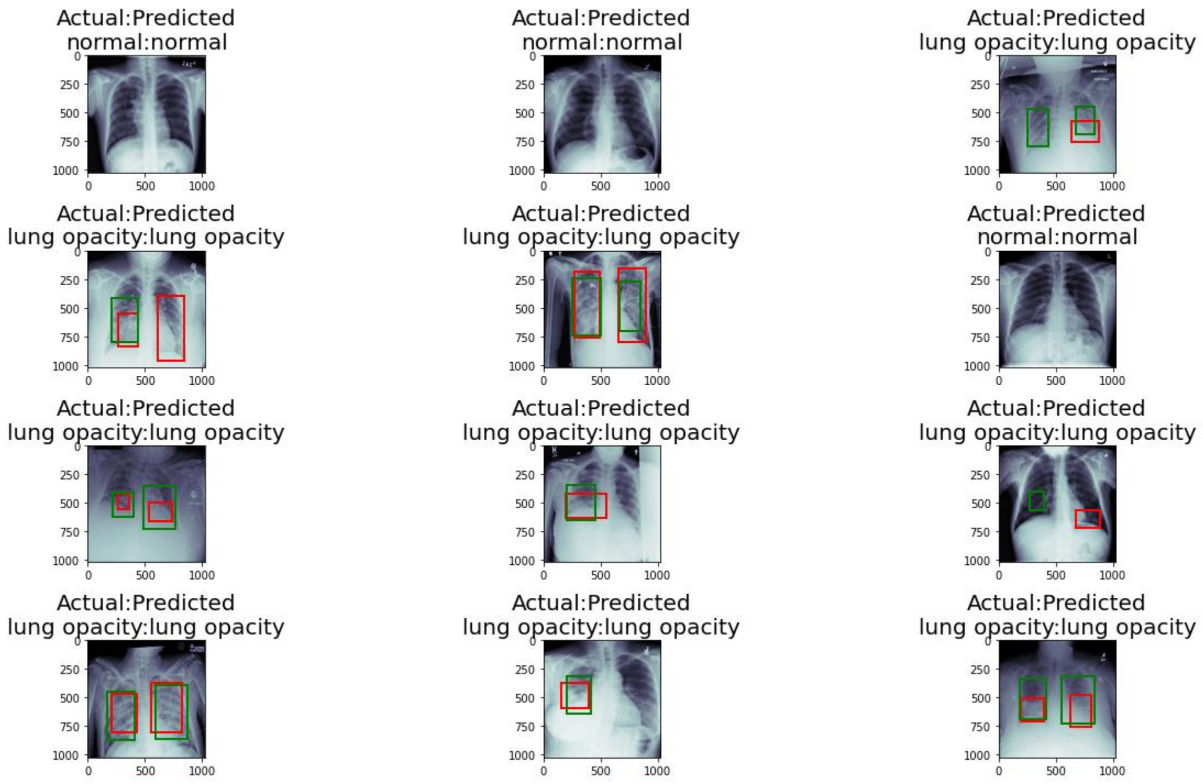
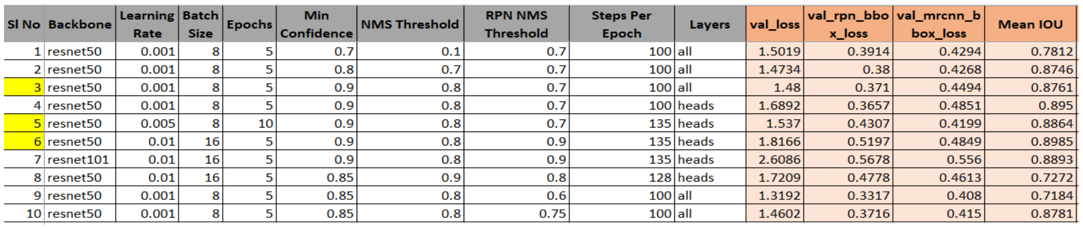


Figure 26: UNet VGG16 best model prediction

MASK RCNN:

The hyper parameters considered for fine tuning the Mask RCNN model are freeze layers, learning rate, batch size, epochs, Detection Minimum Confidence, Detection NMS threshold, RPN NMS threshold and Steps Per Epoch. We tried 10 different combinations of hyper parameter tuning for the above list.

For different values for hyper parameters, the mean IOU and validation loss for all combinations are very close to one another. After close observation, with row 5 hyper parameters, the model had performed well with an overall validation loss of 1.53 and a mean IOU of 0.886.



Sl No	Backbone	Learning Rate	Batch Size	Epochs	Min Confidence	NMS Threshold	RPN NMS Threshold	Steps Per Epoch	Layers	val_loss	val_rpn_bbox_loss	val_mrcnn_bbox_loss	Mean IOU
1	resnet50	0.001	8	5	0.7	0.1	0.7	100	all	1.5019	0.3914	0.4294	0.7812
2	resnet50	0.001	8	5	0.8	0.7	0.7	100	all	1.4734	0.38	0.4268	0.8746
3	resnet50	0.001	8	5	0.9	0.8	0.7	100	all	1.48	0.371	0.4494	0.8761
4	resnet50	0.001	8	5	0.9	0.8	0.7	100	heads	1.6892	0.3657	0.4851	0.895
5	resnet50	0.005	8	10	0.9	0.8	0.7	135	heads	1.537	0.4307	0.4199	0.8864
6	resnet50	0.01	16	5	0.9	0.8	0.9	135	heads	1.8166	0.5197	0.4849	0.8985
7	resnet101	0.01	16	5	0.9	0.8	0.9	135	heads	2.6086	0.5678	0.556	0.8893
8	resnet50	0.01	16	5	0.85	0.9	0.8	128	heads	1.7209	0.4778	0.4613	0.7272
9	resnet50	0.001	8	5	0.85	0.8	0.6	100	all	1.3192	0.3317	0.408	0.7184
10	resnet50	0.001	8	5	0.85	0.8	0.75	100	all	1.4602	0.3716	0.415	0.8781

Figure 27: MASK RCNN hyper parameter combination performance

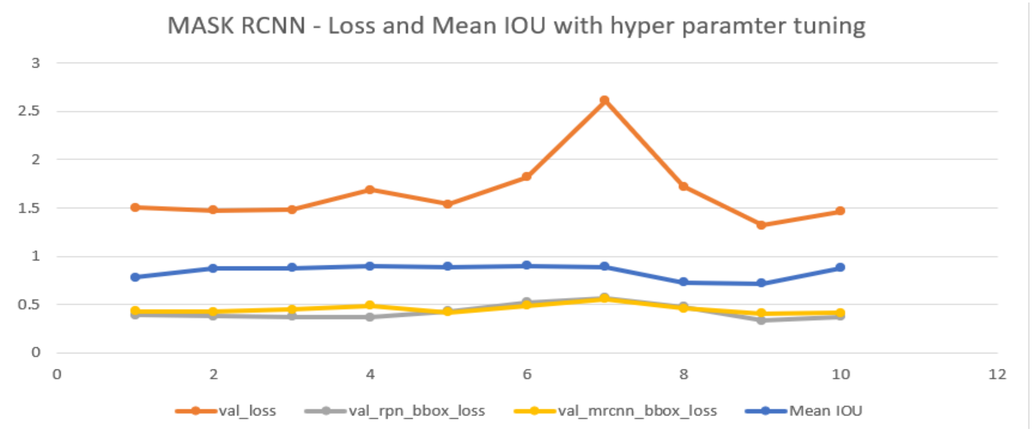
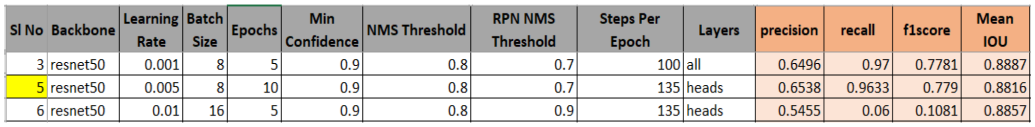


Figure 28: MASK RCNN performance graph

From the top 3 best performing model with the chosen hyper parameter combinations, we calculated Precision/Recall and F1 scores.



Sl No	Backbone	Learning Rate	Batch Size	Epochs	Min Confidence	NMS Threshold	RPN NMS Threshold	Steps Per Epoch	Layers	precision	recall	f1score	Mean IOU
3	resnet50	0.001	8	5	0.9	0.8	0.7	100	all	0.6496	0.97	0.7781	0.8887
5	resnet50	0.005	8	10	0.9	0.8	0.7	135	heads	0.6538	0.9633	0.779	0.8816
6	resnet50	0.01	16	5	0.9	0.8	0.9	135	heads	0.5455	0.06	0.1081	0.8857

Figure 29: MASK RCNN top 3 best performing models

Based on the evaluation, the hyper parameter combinations in 5th row had resulted in high Precision, Recall and F1 score which are 0.651, 0.95 and 0.772 respectively. Hence, we choose this as our best model for MASK RCNN and following are the corresponding hyper parameter values.

- backbone = resnet50
- learning_rate = 0.005
- batch_size = 8
- epochs = 10

- det_min_conf = 0.9
- det_nms_th = 0.8
- rpn_nms_th = 0.7
- steps_per_epoch = 135
- layers = heads

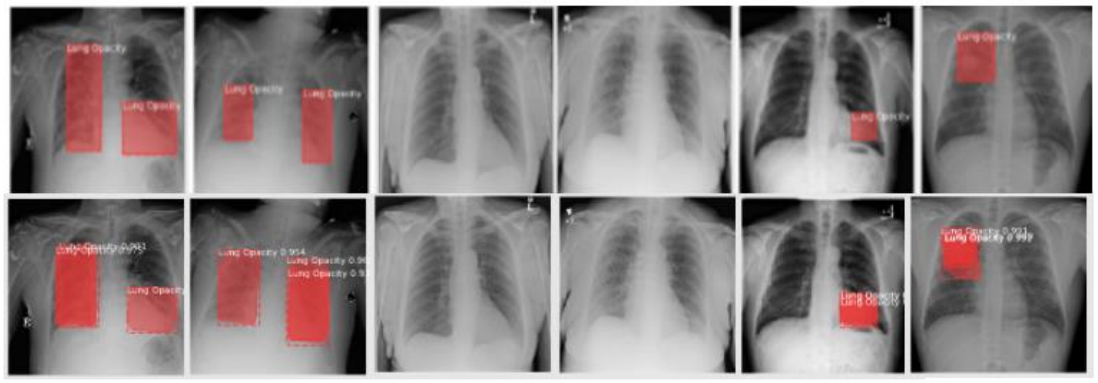


Figure 30: MASK RCNN prediction GT vs Predictions

6.2 Model Evaluation

On evaluating the models, we observe that Mean IOU for MASK RCNN is **0.90** which is higher than UNet VGG16 whose mean IOU is 0.81.

The precision, recall and f1 score for UNet VGG16 are 0.59, 0.66 and 0.62 respectively and compared to MASK RCNN, Mask RCNN has high recall of **0.90** and a precision of **0.71** with an f1 score is **0.78**. The recall is high which means the model is able to predict most of the positive samples (both opacity and normal samples) correctly, however, the precision is low which indicates the true positive samples of all the positive samples is low. And the F1 score is also very high compared to the UNet VGG16 model.

Best Model:

From the above comparison results we can clearly say MASK RCNN out performed UNet VGG16 and hence selected as the BEST model. The file name with the best model is "capstone_project_rsna_pneumonia_maskrcnn_final.ipynb".

Model deployment-

As we know that Flask enables us to create web applications very easily.

DIRECTORY STRUCTURE

First let's have a look at our directory structure, this will give us a broader picture of the overall project and it is also useful to know it when you working with flask, I have saved the project under a main directory called **'deployment files'.**

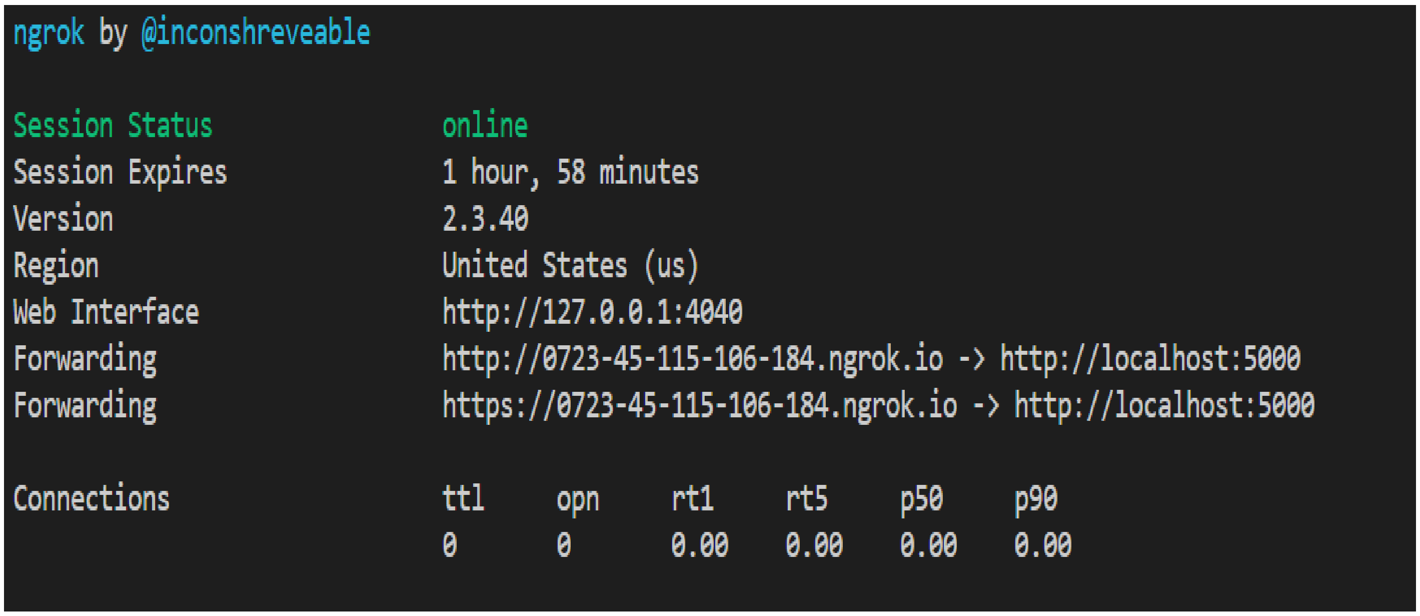
Requirement-

For deployment, we have imported some necessary libraries. which is given below-

- Flask==1.1.2,
- keras==2.9.0 ,
- matplotlib,
- numpy,opencv,
- python_headless,
- pandas, Pillow==9.2.0,
- pydicom==2.3.0,
- scikit-image,
- scikit-learn,
- seaborn,
- tensorflow==2.9.1

DEPLOYMENT and OUTPUT :

Now let us have a look at the final result, how everything looks like once we deployed it, first we deployed it in my local machine and later we will deploy it in a public server.



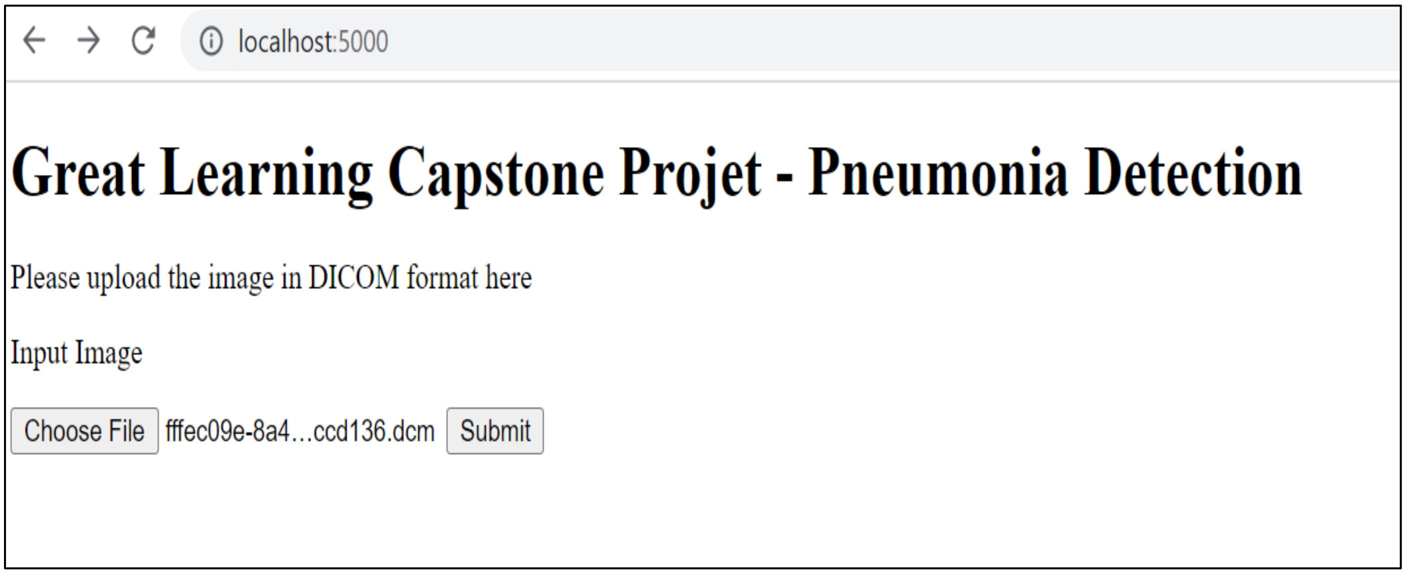
```
ngrok by @inconshreveable

Session Status                online
Session Expires               1 hour, 58 minutes
Version                       2.3.40
Region                        United States (us)
Web Interface                 http://127.0.0.1:4040
Forwarding                    http://0723-45-115-106-184.ngrok.io -> http://localhost:5000
Forwarding                    https://0723-45-115-106-184.ngrok.io -> http://localhost:5000

Connections                   ttl     opn     rt1     rt5     p50     p90
                              0       0       0.00    0.00    0.00    0.00
```

Final Output:

This is our local host where we need to choose the image file and submit it and it will give the output.



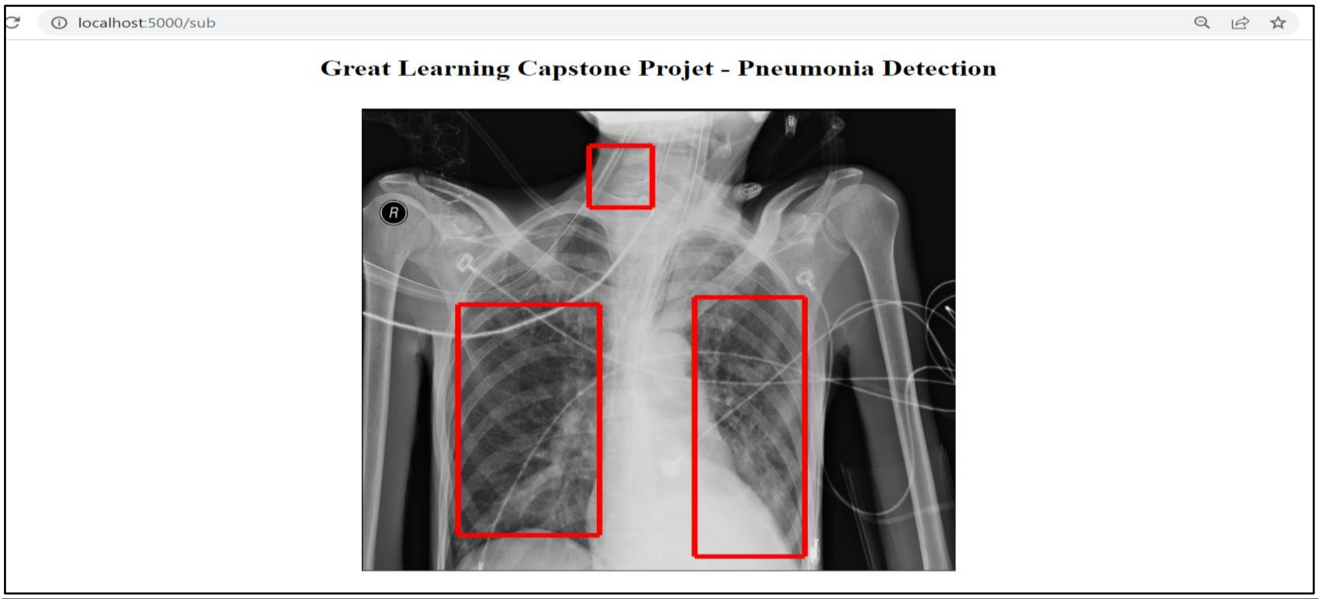
localhost:5000

Great Learning Capstone Projet - Pneumonia Detection

Please upload the image in DICOM format here

Input Image

Output-



localhost:5000/sub

Great Learning Capstone Projet - Pneumonia Detection

7 Limitations and Suggested improvements

7.1 Limitations of our solution

- The input data that was provided is challenging one to predict as all the images are similar. It was tough to differentiate between opacity and normal dataset.
- The data set is imbalanced with respect to the classes; hence we had chosen to use a negative sampling approach for our model building.
- Team had a short time working on different CV models, comparing the time taking on the SOTA models. Model consumes a lot of resources (GPU) and executions take longer time. To ensure we complete the model creation, we considered 2000 images for training, 500 for validation and inference.
- While using the MASK RCNN model, we find it tough to look for the right GitHub repository and the appropriate mapping of TensorFlow/Keras version. It didn’t work on Google Collab due to its nature of use of latest software versions. We chose to work on Kaggle to overcome this limitation, that provides the ability to select the versions of the software needed to build and train the model.
- As part of MASK RCNN training, the predictions are not coming correct even though the mean IOU is above 0.85. When we made a closure look, we identified that the model is predicting all images irrespective of the opacity or sometimes they are not predicting at all. Thus we had to change the data selection technique (negative sampling where we choose opacity as our main data and include negative samples with normal label data), preprocessing methods (data augmentation technique to restrict rotations, flips to horizontal) etc., for the model and adopt similar things for other models as well.

7.2 Suggested Improvements

- Dataset selection can be improved, by trying with different combinations
- Pre-processing can be improved, by trying with different image shapes
- Model training with different hyper parameters and values

8 Our Learnings

- Had an excellent opportunity to learn Computer Vision with SOTA models.
- Learned and used state of art models like UNet, MASK RCNN, YOLO V3 along with Transfer learning approach.
- Adopt different visualization techniques like line charts, bar charts, parallel coordinates, loading and displaying dicom images with bounding box representation.
- Handling imbalanced data and its implication, by adopting Negative sampling, using data augmentation technique (Albumentations, imgaug) and generating more opacity images.
- Use of other external tools like Weights and Biases to train models with defined hyper parameter values, and visualize data of model parameter executions.
- It has helped to look and gain awareness on the SOTA models like MASK RCNN, YOLO V3, SSD.

9 Business Value Derived

- Automating Pneumonia screening in chest radiographs, providing affected area details through the bounding box.
- Assist physicians to make better clinical decisions or even replace human intervention in certain functional areas of healthcare (eg, radiology).
- Guided by relevant clinical questions, powerful AI techniques can unlock clinically relevant information hidden in the massive amount of data, which in turn can assist clinical decision making.

10 Final Note

Many thanks to our Mentor – Mr. Rohit Raj for his expert guidance for completing this Capstone Project

Thanks to our Program Manager – Mugdha Deepala for her dedication and support in conducting AIML program smoothly as scheduled

Finally, Thanks to Great Learning for offering this wonderful course – AIML

11 Acknowledgements

UNet: Citation for UNet Architecture:

https://www.depends-on-the-definition.com/unet-keras-segmenting-images

Mask RCNN:

Thanks to Thanh Bui and Matter Plot Inc for providing the repositories that helped to execute the project smoothly

Matter Plot: https://github.com/matterport/Mask_RCNN

Thanh Bui: https://www.kaggle.com/tbui001/lung-opacity-classification-maskrcnn

```
@misc { matterport_maskrcnn_2017,
title={Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow},
  author={Waleed Abdulla},
  year={2017},
  publisher={Github},
  journal={GitHub repository},
  howpublished={\url{https://github.com/matterport/Mask_RCNN}},
}
```