CAPSTONE PROJECT REPORT

BY

Essa Ibrahim Shariff

Duvvuru Lokesh

Debolina Dutta

Bibhunandan Mohanta

Mohammed Hisham M

INDEX

SUMMARY OF PROBLEM STATEMENT, DATA AND FINDINGS

In medicine, the next frontier for AI is anomaly localization in medical imaging. Localization of anomalies refers to both predicting anomalies and their boundaries. Automatic detection algorithms to locate inflammation in an image can help physicians make better clinical decisions. In this project, we analyze data with the knowledge of EDA. We build a detection model and present our findings based on the evaluations with the RSNA Pneumonia Detection Challenge dataset.

OVERVIEW OF PNEUMONIA

Pneumonia is a form of an acute respiratory infection that affects the lungs. The lungs comprise small sacs called alveoli that fill up with oxygen as a healthy person breathes. The alveoli are filled with pus and fluid when a person has pneumonia, making breathing difficult, and reducing oxygen intake.
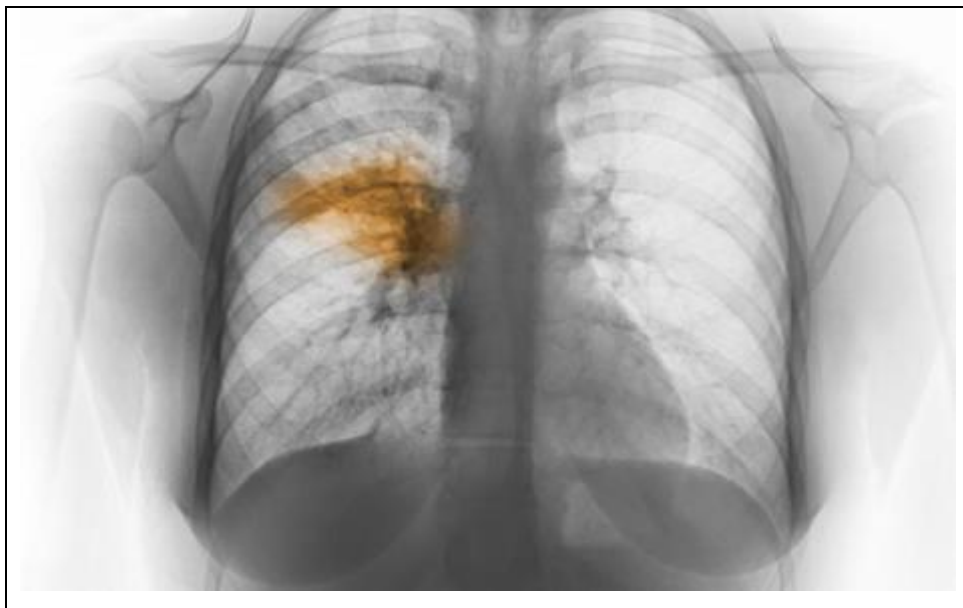

Figure 1- X-Ray Image of Lungs with pneumonia

The single most significant bacterial cause of death in children worldwide is pneumonia. In 2017, pneumonia killed 808,694 children under the age of 5, accounting for 15 percent of all deaths by children under five. Children and families worldwide are afflicted by pneumonia, but it is most common in South Asia and sub-Saharan Africa. It can be avoided with easy procedures and managed with low-cost, low-tech treatment and care.

In 2015 spending for maternal, infant, and child survival, the cost of antibiotic care for all children with pneumonia estimate at about US$ 109 million per year among 66 countries. The expense requires antibiotics and diagnostics for the treatment of pneumonia.

CAUSES AND TRANSMISSION

According to WHO, pneumonia is caused by several infectious agents, including viruses, bacteria, and fungi. The most common are:

- Streptococcus pneumoniae – the most common cause of bacterial pneumonia in children;
- Haemophilus influenzae type b (Hib) – the second most common cause of bacterial pneumonia;
- the respiratory syncytial virus is the most common viral cause of pneumonia;
- in infants infected with HIV, Pneumocystis jiroveci is one of the most common reasons for pneumonia, responsible for at least one-quarter of all pneumonia deaths in HIV-infected infants.

Spreading of Pneumonia happens in many ways. The viruses and bacteria commonly found in a child's nose or throat can infect the lungs while breathing. They may also spread via air-borne droplets from a cough or sneeze. Besides, pneumonia may spread through blood, especially during and shortly after birth. More research needs to be done on the different pathogens causing pneumonia and how they are transmitted, as this is of critical importance for treatment and prevention.

TREATMENT AND PREVENTION

Pneumonia is treated with antibiotics. Amoxicillin-dispersible tablets are the antibiotic of choice. In most pneumonia cases, oral antibiotics are needed, which are mostly administered at a health clinic. These cases may also be diagnosed and treated at the neighborhood level by qualified community health professionals with affordable oral antibiotics. Only for severe cases of pneumonia is hospitalization recommended.

DIAGNOSTIC PROCEDURE

The doctor will diagnose pneumonia based on your medical history, a physical exam, and test results. Sometimes pneumonia is hard to analyze because symptoms may be the same as a cold or flu. The patient may not realize that his/her condition is more severe until it lasts longer than these other conditions.

If the doctor thinks the patient may have pneumonia, they may do one or more of the following tests.

➢ Chest X-ray to look for inflammation in the patient's lungs. A chest X-ray is often used to diagnose pneumonia.

➢ Blood tests, such as a complete blood count (CBC), determine whether the patient's immune system is fighting an infection.

➢ Pulse oximetry to measure how much oxygen is in his/her blood. Pneumonia can keep the patient's lungs from moving enough oxygen into his/her blood. A small sensor called a pulse oximeter is attached to the patient's finger or ear to calculate the levels.

DATA DESCRIPTION

In 2018, RSNA organized an AI challenge to detect pneumonia, one of the leading causes of mortality worldwide, as part of its efforts to help improve artificial intelligence (AI) instruments for radiology. RSNA Pneumonia dataset consists of 29684 thousand images. All the images are in Dicom format. There are 3000 images for testing and the remaining for training.

Dicom images: The images are in a particular format called DICOM files (*. dcm). They contain a mix of header metadata as well as pixel data underlying raw image arrays.

There are three classes in the dataset - Normal, Not normal/No opacity, and Lung opacity. Normal class indicates there is no anomaly in the lungs. Not normal/No opacity demonstrates to those who do not have pneumonia, but the image still has some abnormality. Sometimes, this finding could mimic the appearance of the right pneumonia. Lung opacity class indicates there is definite pneumonia in the lungs. Finally, these three classes are divided into two target variables, 0 and 1. The images with lung opacity come under target 1 and 0 for the other two classes.

Along with the images, two csv files are provided. A detailed class info file consists of the image name and the class it belongs to. The train labels file consists of the bounding box coordinates belonging to each image. Bounding box coordinates are given in the following format:

- 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.

With these bounding box coordinates, the target column is provided, which discriminates classes into categories of 0 and 1.

OVERVIEW OF THE FINAL PROCESS

EDA AND PREPROCESSING

There are two datasets; each dataset has 30227 rows, and the trian_labels.csv has six columns while class_info.csv has two columns (See Figure 2).

```
print("Bounding Box Shape:", bbox_df.shape)
print("Class info Shape:", class_df.shape)

Bounding Box Shape: (30227, 6)
Class info Shape: (30227, 2)
```

Figure 2- Shape of the data

From Figure 3 we can see that train_labels.csv file contains patientId, which is a unique value per patient. Each patientId has one target column and four values: the corresponding abnormality bounding box defined by the upper-left-hand corner 'x' and 'y' coordinate and its corresponding width and height. The target column has two values 0 and 1. 0 is for No Lung Opacity / Not Normal, Normal, and 1 is for Lung opacity.

	patientId	x	y	width	height	Target
0	0004cfab-14fd-4e49-80ba-63a80b6bddd6	NaN	NaN	NaN	NaN	0
1	00313ee0-9eaa-42f4-b0ab-c148ed3241cd	NaN	NaN	NaN	NaN	0
2	00322d4d-1c29-4943-afc9-b6754be640eb	NaN	NaN	NaN	NaN	0
3	003d8fa0-6bf1-40ed-b54c-ac657f8495c5	NaN	NaN	NaN	NaN	0
4	00436515-870c-4b36-a041-de91049b9ab4	264.0	152.0	213.0	379.0	1

Figure 3- Bounding Box data

In the class_info.csv file, there are two columns patientId and class column that describe the three conditions of lungs (See Figure 4).

	patientId	class
0	0004cfab-14fd-4e49-80ba-63a80b6bddd6	No Lung Opacity / Not Normal
1	00313ee0-9eaa-42f4-b0ab-c148ed3241cd	No Lung Opacity / Not Normal
2	00322d4d-1c29-4943-afc9-b6754be640eb	No Lung Opacity / Not Normal
3	003d8fa0-6bf1-40ed-b54c-ac657f8495c5	Normal
4	00436515-870c-4b36-a041-de91049b9ab4	Lung Opacity

Figure 4- Class data

5

The frequency of patients in each class and their respective percentages are shown in Figure 5. 23.5 percent of the patients are Normal, and the remaining are Not Normal and Lung opacity.

```
data['class'].value_counts()
[ ]

Lung Opacity                        16957
No Lung Opacity / Not Normal        11821
Normal                               8851
Name: class, dtype: int64


data['class'].value_counts()*(100.0)/len(data.index)

Lung Opacity                        45.063648
No Lung Opacity / Not Normal        31.414600
Normal                              23.521752
Name: class, dtype: float64
```

Figure 5 - Count and Percentage of patients in each class

The bounding box dataset has missing values in x, y, height, and width column. A total of 20672 missing values are present in each column(See Figure 6).

```
def missing_data(data):
    null_data = data.isnull().sum()
    num_rows = len(data.index)
    percent_null = 100.*null_data/num_rows
    return pd.concat([null_data, percent_null.round(1)], axis=1, keys=['Missing', 'PercentMissi

missing_data(data)
```

	Missing	PercentMissing
patientId	0	0.0
class	0	0.0
x	20672	54.9
y	20672	54.9
width	20672	54.9
height	20672	54.9
Target	0	0.0

Figure 6- Missing Values

Figure 7 shows that the number of missing values is equal the number of No Lung Opacity (target=0). This indicates that the missing values are not random missing or errors, they are the images of lungs of normal person (or person without Pneumonia).

```
print("Target 0")
print("="*20)
print(data[data.Target == 0].count())
print("")
print("Target 1")
print("="*20)
print(data[data.Target == 1].count())

Target 0
====================
patientId    20672
class        20672
x            20672
y            20672
width        20672
height       20672
Target       20672
dtype: int64
```

Figure 7- Count patients with Target 0

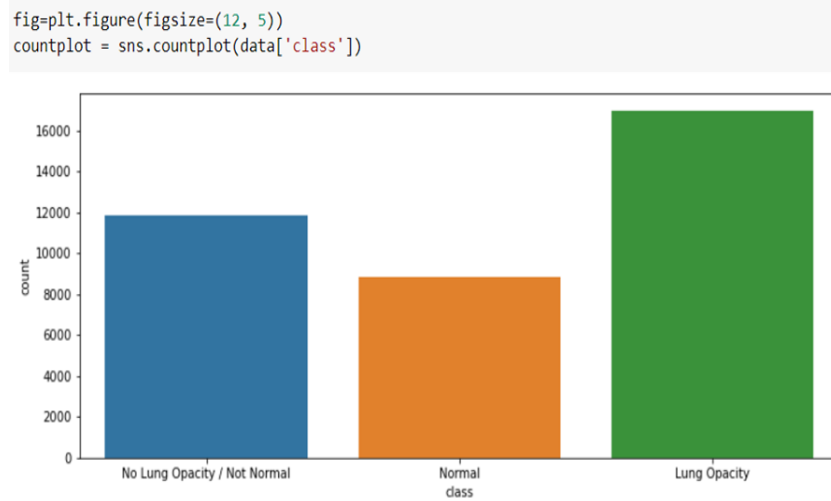So, the remaining 16957 are the positive means Lung Opacity case. Figure 8 shows the count plot of the three classes.

```
fig=plt.figure(figsize=(12, 5))
countplot = sns.countplot(data['class'])
```


Figure 8 - Bar diagram of patients in each classes

There are 26684 training images and 3000 test images. Visualizations of the few samples are shown in Figure 9.
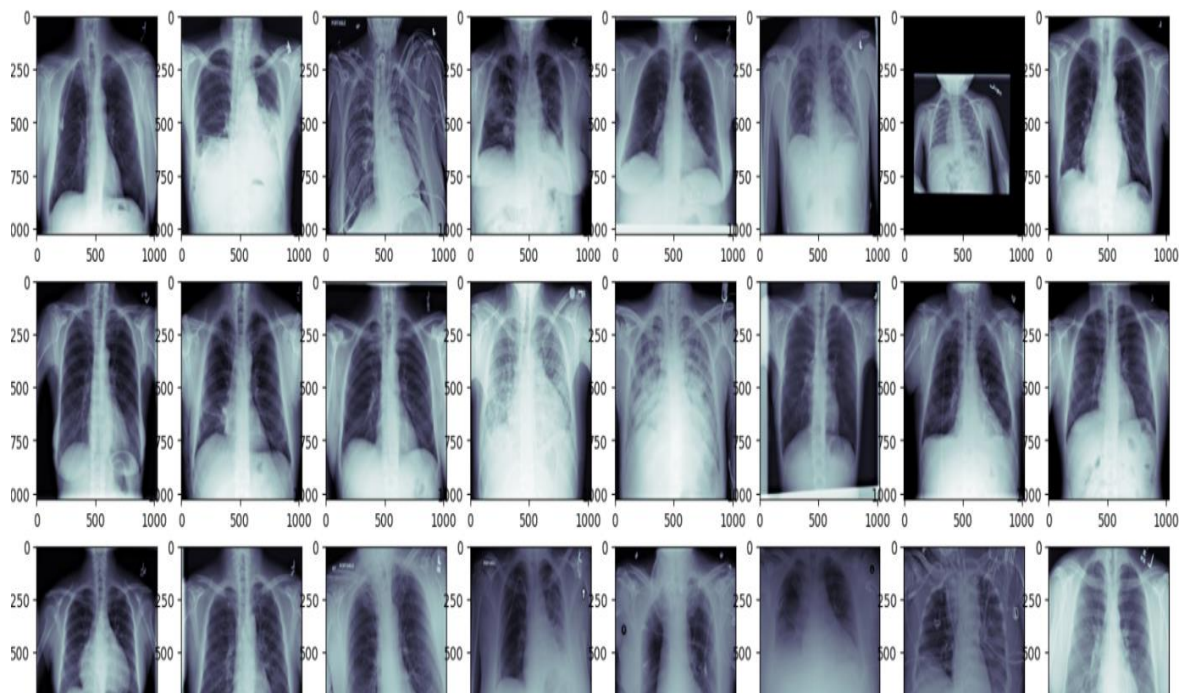

Figure 9- Sample visuals of chest X-Ray images

The images, along with the bounding box, is presented in Figure 10. The figure indicates that some images have more than one bounding box, whereas some do not even have one.
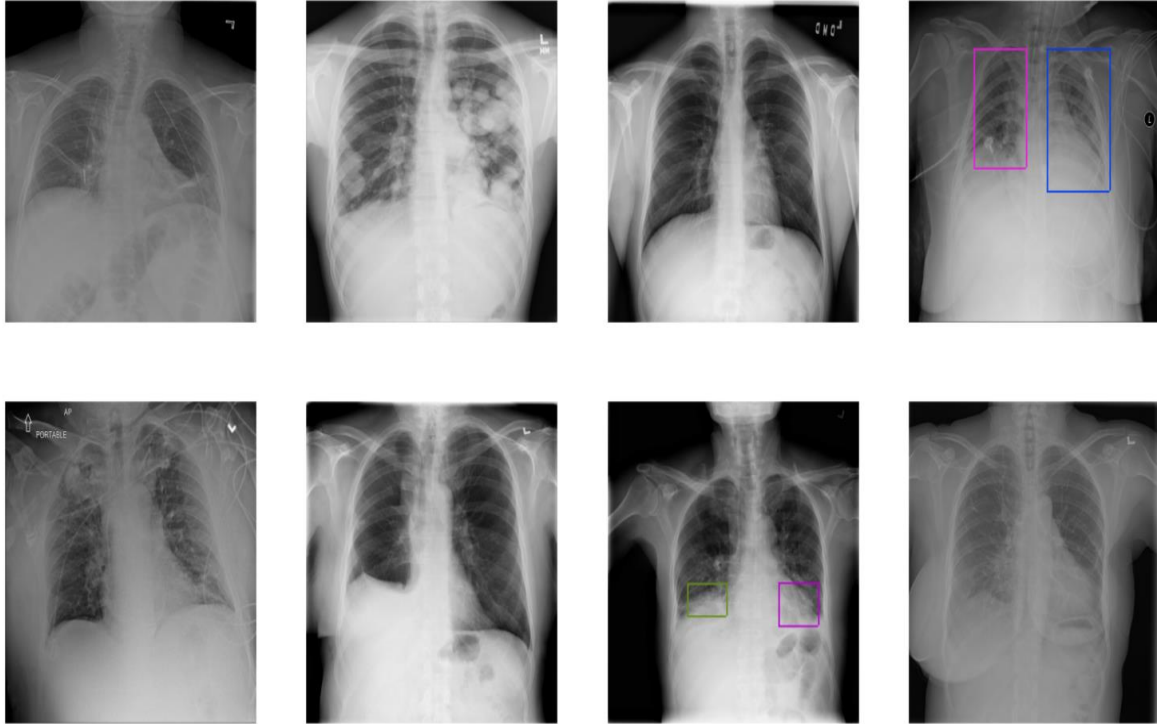
Figure 10- Sample visuals of chest X-Ray images along with Bounding Boxes

As we can determine from the above visualizations, the task in hand is a regression problem. There is a need for building a feasible model that can regress the bounding box in the images. Moreover, there is an extra class that is Not normal/ No lung opacity. This class shows there is an anomaly in the lungs, which can be easily misread as pneumonia. So, there is a need to examine that class a little more briefly. A visualization showing all three classes together is shown Figure 11.
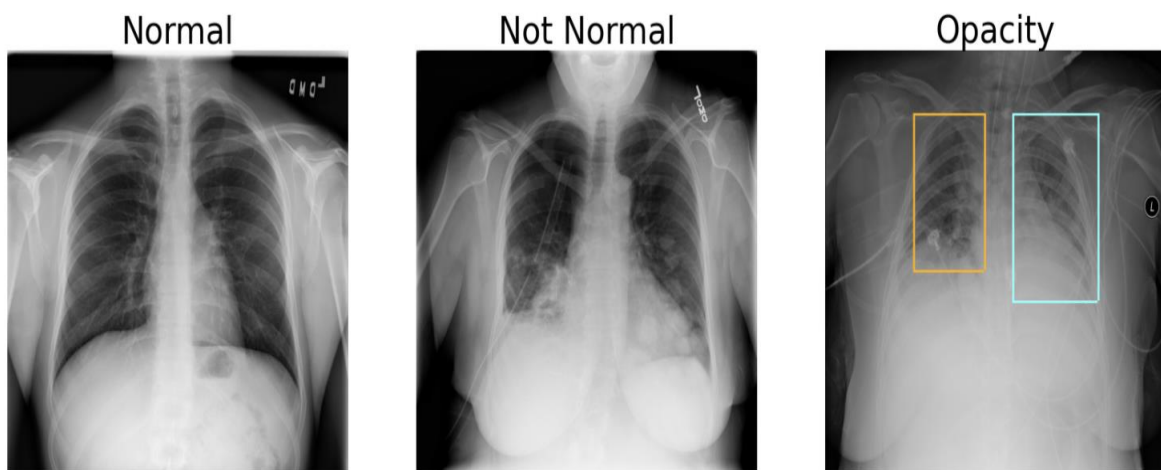

Figure 11 - Sample visuals of chest X-Ray images along with Bounding Boxes for the three classes.

A bar plot of the target classes is shown in Figure 12.

```
In [12]:
fig, ax = plt.subplots(nrows=1,figsize=(12,6))
tmp = comb_bounding_box_df.groupby('Target')['class'].value_counts()
df = pd.DataFrame(data={'Exams': tmp.values}, index=tmp.index).reset_index()
sns.barplot(ax=ax,x = 'Target', y='Exams',hue='class',data=df, palette='Set2')
plt.title("Chest exams class and Target")
plt.show()
```
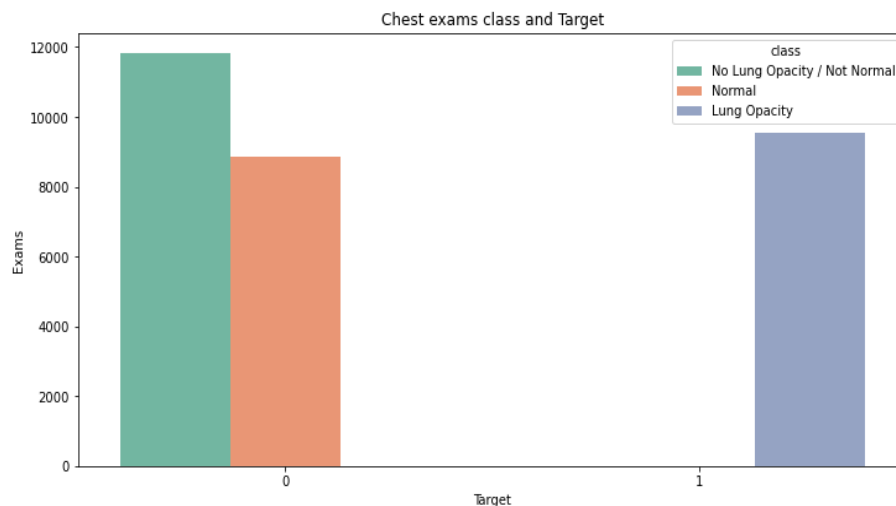

Figure 12 - Bar diagram representing the three classes in the two target columns

The information regarding the patients is available in the metadata of the Dicom images. Visualizations of that data may give a better understanding of the pneumonia disease itself. Figure 13 shows the data frame of the extracted dicom data.

```
[53]:
image_meta_df['class'] = detail_class_df['class']
image_meta_df['patientId'] = detail_class_df['patientId']
image_meta_df.head()
```

Out[53]:

	PatientAge	BodyPartExamined	ViewPosition	PatientSex	path	class	patientId
0	8	CHEST	PA	M	../input/rsna-pneumonia-detection-challenge/st...	No Lung Opacity / Not Normal	0004cfab-14fd-4e49-80ba-63a80b6bddd6
1	16	CHEST	AP	M	../input/rsna-pneumonia-detection-challenge/st...	No Lung Opacity / Not Normal	00313ee0-9eaa-42f4-b0ab-c148ed3241cd
2	62	CHEST	PA	M	../input/rsna-pneumonia-detection-challenge/st...	No Lung Opacity / Not Normal	00322d4d-1c29-4943-afc9-b6754be640eb
3	65	CHEST	AP	M	../input/rsna-pneumonia-detection-challenge/st...	Normal	003d8fa0-6bf1-40ed-b54c-ac657f8495c5
4	26	CHEST	AP	F	../input/rsna-pneumonia-detection-challenge/st...	Lung Opacity	00436515-870c-4b36-a041-de91049b9ab4

Figure 13 - Data Frame of the extracted dicom data

Gender is one of the variables present in the data which can be explored. We can infer that the dataset has more male examples from the below images than the female examples. In this case,

9

there are more men with pneumonia, around 4800 compared to around 3300 women with pneumonia (See Figure 14).

```
[56]:   sns.countplot(image_meta_df['PatientSex'])
        plt.title("Patient Sex")
        plt.show()
```
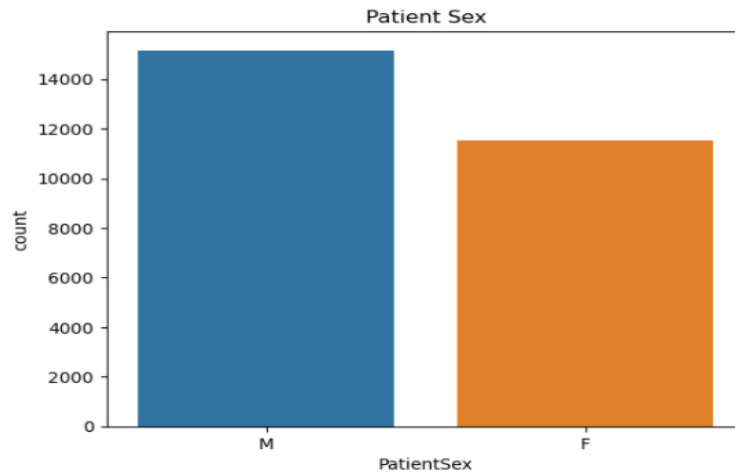

Figure 14 – Bar Diagram representing the gender of the patients

The abnormality in the lungs is very high in men compared to women. There is a massive difference in the Not normal/ No lung opacity class between males and females (See Figure 15).

```
[55]:   tmp = image_meta_df.groupby(['class', 'PatientSex'])['patientId'].count()
        df1 = pd.DataFrame(data={'Exams': tmp.values}, index=tmp.index).reset_index()
        tmp = df1.groupby(['Exams','class', 'PatientSex']).count()
        df3 = pd.DataFrame(data=tmp.values, index=tmp.index).reset_index()
        fig, (ax) = plt.subplots(nrows=1,figsize=(6,6))
        sns.barplot(ax=ax, x = 'PatientSex', y='Exams', hue='class',data=df3)
        plt.title("Patient Sex and class")
        plt.show()
```
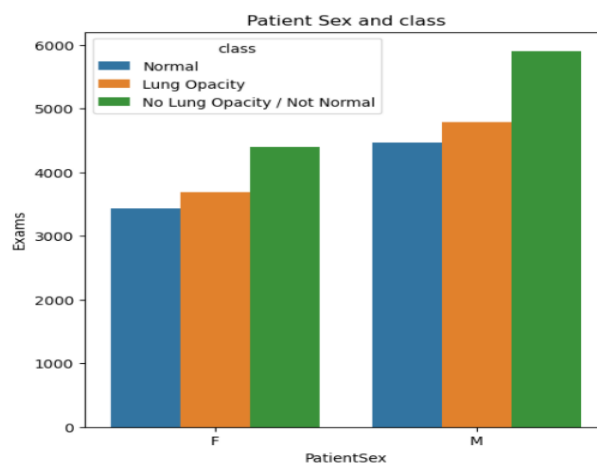

Figure 15 - Bar Diagram representing the distribution of the three classes with respect to their gender

```
[52]:  DCM_TAG_LIST = ['PatientAge', 'BodyPartExamined', 'ViewPosition', 'PatientSex']
       def get_tags(in_path):
           c_dicom = pydicom.read_file(in_path, stop_before_pixels=False)
           tag_dict = {c_tag: getattr(c_dicom, c_tag, '')
                      for c_tag in DCM_TAG_LIST}
           tag_dict['path'] = in_path
           return pd.Series(tag_dict)
       image_meta_df = image_df.apply(lambda x: get_tags(x['path']), 1)
       # show the summary
       image_meta_df['PatientAge'] = image_meta_df['PatientAge'].map(int)
       image_meta_df['PatientAge'].hist()
       image_meta_df.drop('path',1).describe(exclude=np.number)
```

Out[52]:

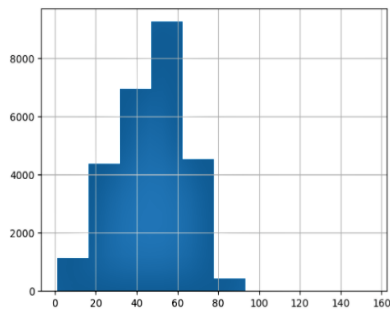|	BodyPartExamined	ViewPosition	PatientSex
count	26884	26884	26884
unique	1	2	2
top	CHEST	PA	M
freq	26884	14511	15166

Figure 16 – Histogram of age distribution and

Similarly, we can explore the other variables such as body parts examined and age. Figure 16 gives the histogram of the age distribution. We can see that most of the patients are above 40years and below 65 years. The detailed plots are used to examine further Similarly, we can explore the other variables such as body parts examined and age (See Figure 17).

```
[54]:  fig, (ax) = plt.subplots(nrows=1,figsize=(16,6))
       sns.countplot(image_meta_df['PatientAge'], ax=ax)
       plt.title("Patient Age")
       plt.xticks(rotation=90)
       plt.show()
```
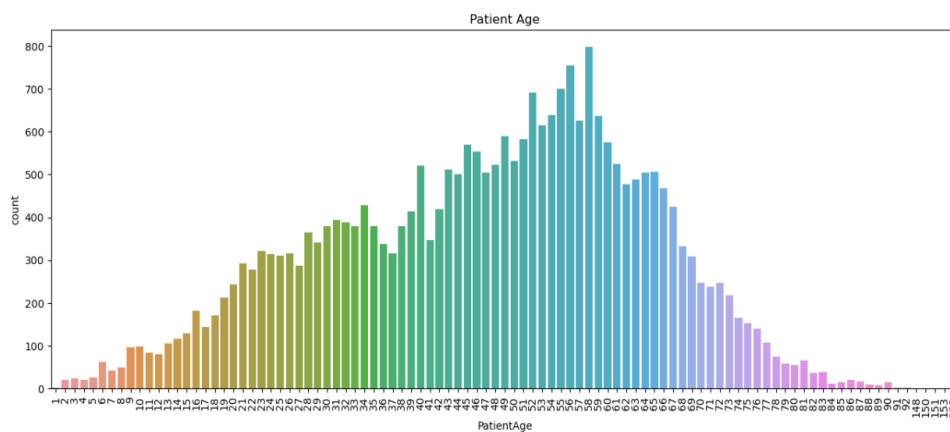

Figure 17 – Bar plot for showing the distribution of age.

From the above visualization, we can see that patients between 50 and 60 are more affected, with the dominant age being 58. The figure clearly shows that pneumonia is more prominent at an older age. Moreover, we can also see that the provided dataset consists of people aged between 40 and 65. We can explore a bit further by comparing Age with the class. The visualization that summarizes that relationship is plotted in below picture.

From Figure 18, we can infer that Abnormalities in the lungs are more prominent in the people age between 45 and 65. So of no surprise, the lung opacities are maximum in that age group.
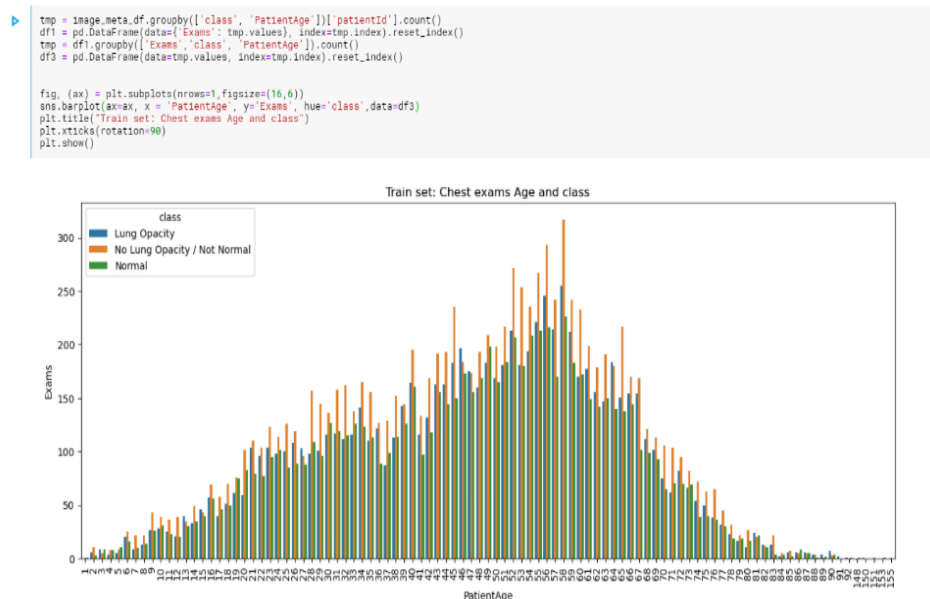
```python
tmp = image_meta_df.groupby(['class', 'PatientAge'])['patientId'].count()
df1 = pd.DataFrame(data={'Exams': tmp.values}, index=tmp.index).reset_index()
tmp = df1.groupby(['Exams','class', 'PatientAge']).count()
df3 = pd.DataFrame(data=tmp.values, index=tmp.index).reset_index()

fig, (ax) = plt.subplots(nrows=1, figsize=(16,6))
sns.barplot(ax=ax, x = 'PatientAge', y='Exams', hue='class',data=df3)
plt.title("Train set: Chest exams Age and class")
plt.xticks(rotation=90)
plt.show()
```


Figure 18 - Bar plot for showing the distribution of age with different class.

To summarize the Exploratory data analysis. We can say we are dealing with class imbalance, as most images do not have the bounding boxes. We can say that there is a need for the bounding box regressor model. The metadata provided can give insight to the disease as a whole. The visualizations of the original data and dicom metadata are extremely useful for the stakeholders and model evaluation.

STEP-BY-STEP WALK THROUGH THE SOLUTION

Based on the findings from exploratory data analysis and problem statement, it is evident that the model should be a bounding box regressor that can identify the lung opacities, in turn predicting pneumonia. The model should have the ability to localize and identify the opacities. So, based on that, we came up with the following models.

CNN MODEL WITH RESNET BLOCKS:

This model consists of a series of residual blocks in the middle with downsampling then followed by output block, which leads to upsampling.

Approach

Firstly, a convolutional neural network is used to segment the image, using the bounding boxes directly as a mask. The connected components are used to separate multiple areas of predicted pneumonia. Finally, a bounding box is simply regressed around every connected component.

Network

The network consists of several residual blocks with convolutions and downsampling blocks with max pooling. At the end of the network, a single upsampling layer converts the output to the same shape as the input.

As the input to the network is 256 by 256 (instead of the original 1024 by 1024) and the network downsamples several times without any meaningful upsampling (the final upsampling is to match in 256 by 256 mask), the final prediction is very crude. If the network down samples four times, the final bounding boxes can only change with at least 16 pixels.

Results

This model is initially trained for 20 epochs. This gives the mean IOU of 0.76 on the validation set, and model is a good fit. Accuracy increases slightly as epochs increased, and loss gradually decreases. Results for the above network on the initial network are as shown in the below graphs.
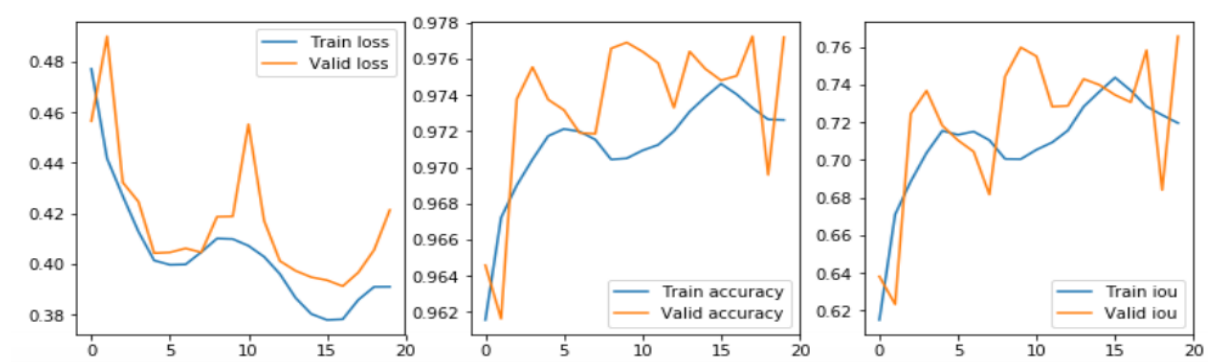

Figure 19- Graphs of Loss, Accuracy and IoU of train and test data of CNN model with RESNET Blocks.

U-NET WITH MOBILENET BACKBONE

This model is predominantly used for the image segmentation problems. In this case, the UNet model is modified for pneumonia detection problem using the bounding boxes' coordinates to construct the masks.

Approach

The mobilenet model is loaded from the Keras library with imagenet competition weights and the top of the network is frozen. The image is given as an input with the corresponding bounding box, and feature maps are produced and downsampled gradually to fit the bounding box and then returned to the original shape of the input image. This process allows the network to propagate context information to higher resolution layers.

Network

This network consists of u-shaped architecture, as the name suggests. During the downsampling, features are generated gradually. Whereas on upsampling, the features are duplicated to that of the original image with mask segmentation.

This network takes an image input of 224*224 pixels with three channels red, green, and blue. It has around 3 million parameters. A standard convolution network consisting of repeated use of convolutions, each followed by a rectified linear unit (ReLU) and a max-pooling process, is the contracting part. The spatial information is decreased during the contraction, while feature information is increased. Via a series of up-convolutions and concatenations with high-resolution characteristics from the contracting path, the expansive path incorporates the function and spatial details.

Results

During the initial run of 5 epochs, the u-net model showed promising results with a mean IOU of 0.88. While loss goes up and gradually decreases, the model's accuracy seems to take a downfall with the training. The results for the initial run are shown in the graphs below. This model's drawback is its training time is very long, nearly 50mins per epoch on GPU.
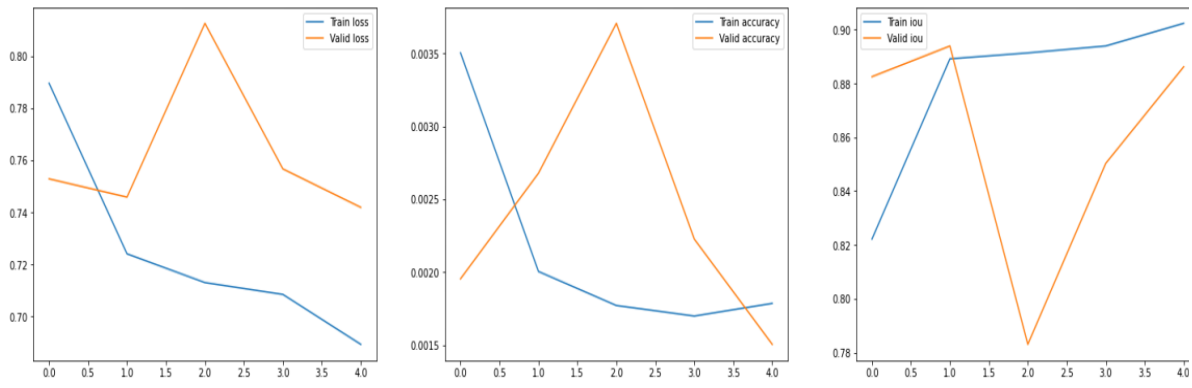
Figure 20- Graphs of Loss, Accuracy and IoU of train and test data of U-NET model.

DENSENET 121

We have an input image in a typical Convolutional Neural Network, which is then passed through the network to get a projected output mark in a way where the forward pass is pretty straightforward. In a DenseNet, each layer is connected to every other layer, hence the name Densely Connected Convolutional Network.

Approach

L(L+1)/2 direct connections exist for L layers. The feature maps of all the previous layers are used as inputs for each layer, and their own feature maps are used as inputs for each subsequent layer. The concatenation of feature maps from previous layers is the input of a layer within DenseNet.

Network

There are four Dense Blocks with varying layer numbers in each architecture. For instance, in the four dense blocks, DenseNet-121 has [6,12,24,16] layers. We can see that a 7x7 stage 2 Conv Layer followed by a 3x3 stride-2 MaxPooling layer consists of the first part of the DenseNet architecture. A Classification Layer that accepts the feature maps of all network layers to perform the classification follows the fourth dense block.

Downsampling and upsampling are added to the dense layers to learn the mask implementation of the given input images of 224*224 with three channels (RGB). This network has 6million parameters.

Results

During the initial run of the model, it gave better results than U-net. Here accuracy and loss are better accompanied by 0.88 mean IOU. Training time takes longer than other models but

15

less than u-net, around 37mins for each epoch. The results from the initial training run are shown in the graph below.
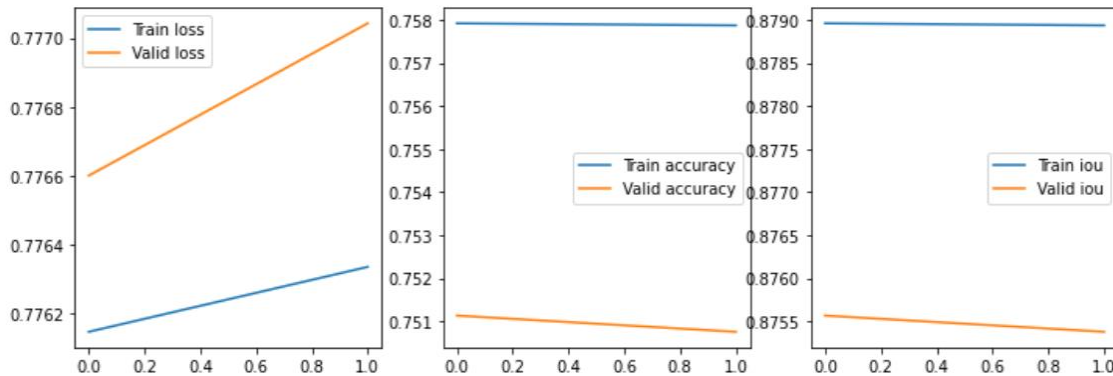

Figure 21- Graphs of Loss, Accuracy and IoU of train and test data of DENSENET 121 model.

Mask RCNN

The Mask R-CNN is the Faster R-CNN extension, which adds an output model for each detected object to predict a mask. Mask R-CNN, especially compared to a simple or even state-of-the-art deep convolutional neural network model, is a sophisticated model to implement.

Approach

Matterport's Mask R-CNN is adapted for the problem at hand. This model generates bounding boxes and segmentation masks for each instance of an object in the image. It's based on Feature Pyramid Network (FPN) and a ResNet101 backbone.

Network

The first step is to install the library. Installation involves cloning the GitHub repository and running the installation script on the workstation. Coco dataset pretrained weights are loaded to use for transfer learning.
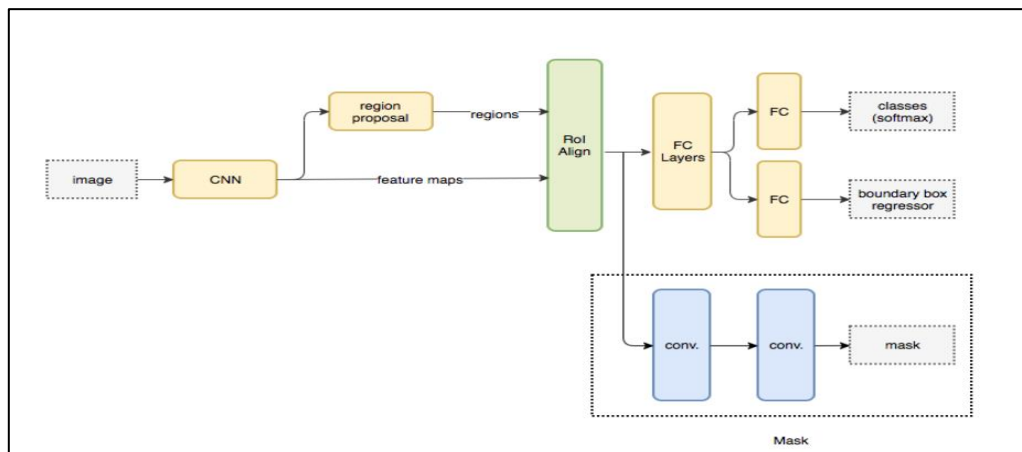

Figure 22- Flow chart of Mask RCNN

Mask RCNN architecture consists of two stages,

Stage 1: The first stage consists of two networks, a backbone network (ResNet, VGG, Inception, etc.) and a network of regional proposals. To offer a set of region proposals, these networks run once per picture. Proposals for a region are regions in the function map that contain the object. The image is converted from 1024x1024px x 3 (RGB) to a feature map of shape 32x32x2048. The **Feature Pyramid Network (FPN)** was an extension of the backbone network which can better represent objects at multiple scales.

It also uses the **Region Proposal Network (RPN)** which scans all FPN top to bottom and proposes regions which may contain objects. It uses anchors which are a set of boxes with predefined locations and scales itself according to the input images. Individual anchors are assigned to the ground-truth classes and bounding boxes. RPN generates two outputs for each anchor — anchor class and bounding box specifications. The anchor class is either foreground class or a background class.

Stage 2: In the second stage, the network predicts bounding boxes and object class for each of the proposed region obtained in stage1. Each proposed region can be of different size, whereas fully connected layers in the networks always require a fixed size vector to make predictions. These proposed regions' size is fixed by using either RoI pool (which is very similar to MaxPooling) or the RoIAlign method. The RoIAlign layer's output is then fed into Mask head, which consists of two convolution layers. It generates a mask for each RoI, thus segmenting an image in a pixel-to-pixel manner. The most inspiring things we found about Mask RCNN is that we could actually force different layers in neural network to learn features with different scales, just like the anchors and ROIAlign, instead of treating layers as black box

Results:

After the initial run of 5 epochs of training, training loss gradually reduces along with the rpn_class_loss, rpn_bbox_loss, mrcnn_class_loss, mrcnn_bbox_loss, and mrcnn_mask_loss. This model trains a little faster than the above models.

MODEL EVALUATION AND VISUALIZATIONS

As the problem statement is defined as the object detection, from the above findings its evident that we can use CNN custom model and MaskRcnn for the training as the other models are image detection and segmentation models.

Evaluate Mask R-CNN Model

The first step is to define a new configuration for evaluating the model. See the code below.

```python
# The following parameters have been selected to reduce running time fo
r demonstration purposes
# These are not optimal

class DetectorConfig(Config):
    """Configuration for training pneumonia detection on the RSNA pneum
onia dataset.
    Overrides values in the base Config class.
    """

    # Give the configuration a recognizable name
    NAME = 'pneumonia'

    # Train on 1 GPU and 8 images per GPU. We can put multiple images o
n each
    # GPU because the images are small. Batch size is 8 (GPUs * images/
GPU).
    GPU_COUNT = 1
    IMAGES_PER_GPU = 8

    BACKBONE = 'resnet50'

    NUM_CLASSES = 2  # background + 1 pneumonia classes

    IMAGE_MIN_DIM = 256
    IMAGE_MAX_DIM = 256
    RPN_ANCHOR_SCALES = (32, 64, 128, 256)
    TRAIN_ROIS_PER_IMAGE = 32
    MAX_GT_INSTANCES = 3
    DETECTION_MAX_INSTANCES = 3
    DETECTION_MIN_CONFIDENCE = 0.7
    DETECTION_NMS_THRESHOLD = 0.1

    STEPS_PER_EPOCH = 200

config = DetectorConfig()
config.display()


Configurations:
BACKBONE                      resnet50
BACKBONE_STRIDES              [4, 8, 16, 32, 64]
BATCH_SIZE                    8
BBOX_STD_DEV                  [0.1 0.1 0.2 0.2]
COMPUTE_BACKBONE_SHAPE        None
DETECTION_MAX_INSTANCES       3
DETECTION_MIN_CONFIDENCE      0.7
DETECTION_NMS_THRESHOLD       0.1
```

```
FPN_CLASSIF_FC_LAYERS_SIZE      1024
GPU_COUNT                       1
GRADIENT_CLIP_NORM              5.0
IMAGES_PER_GPU                  8
IMAGE_CHANNEL_COUNT             3
IMAGE_MAX_DIM                   256
IMAGE_META_SIZE                 14
IMAGE_MIN_DIM                   256
IMAGE_MIN_SCALE                 0
IMAGE_RESIZE_MODE               square
IMAGE_SHAPE                     [256 256   3]
LEARNING_MOMENTUM               0.9
LEARNING_RATE                   0.001
LOSS_WEIGHTS                    {'rpn_class_loss': 1.0, 'rpn_bbox_loss':
1.0, 'mrcnn_class_loss': 1.0, 'mrcnn_bbox_loss': 1.0, 'mrcnn_mask_loss'
: 1.0}
MASK_POOL_SIZE                  14
MASK_SHAPE                      [28, 28]
MAX_GT_INSTANCES                3
MEAN_PIXEL                      [123.7 116.8 103.9]
MINI_MASK_SHAPE                 (56, 56)
NAME                            pneumonia
NUM_CLASSES                     2
POOL_SIZE                       7
POST_NMS_ROIS_INFERENCE         1000
POST_NMS_ROIS_TRAINING          2000
PRE_NMS_LIMIT                   6000
ROI_POSITIVE_RATIO              0.33
RPN_ANCHOR_RATIOS               [0.5, 1, 2]
RPN_ANCHOR_SCALES               (32, 64, 128, 256)
RPN_ANCHOR_STRIDE               1
RPN_BBOX_STD_DEV                [0.1 0.1 0.2 0.2]
RPN_NMS_THRESHOLD               0.7
RPN_TRAIN_ANCHORS_PER_IMAGE     256
STEPS_PER_EPOCH                 200
TOP_DOWN_PYRAMID_SIZE           256
TRAIN_BN                        False
TRAIN_ROIS_PER_IMAGE            32
USE_MINI_MASK                   True
USE_RPN_ROIS                    True
VALIDATION_STEPS                50
WEIGHT_DECAY                    0.0001
```

The performance of a model for an object recognition task is often evaluated using the mAP and IOU.

We are predicting bounding boxes so we can determine how well the predicted and actual bounding boxes overlap. This can be calculated by dividing the area of the overlap by the total area of both bounding boxes, or the intersection divided by the union, referred to as "*intersection over union*," or IoU. A perfect bounding box prediction will have an IoU of 1. It is standard to assume a positive prediction of a bounding box if the IoU is greater than 0.5, e.g. they overlap by 50% or more.

19

Ground Truth vs Predictions

Image Detection
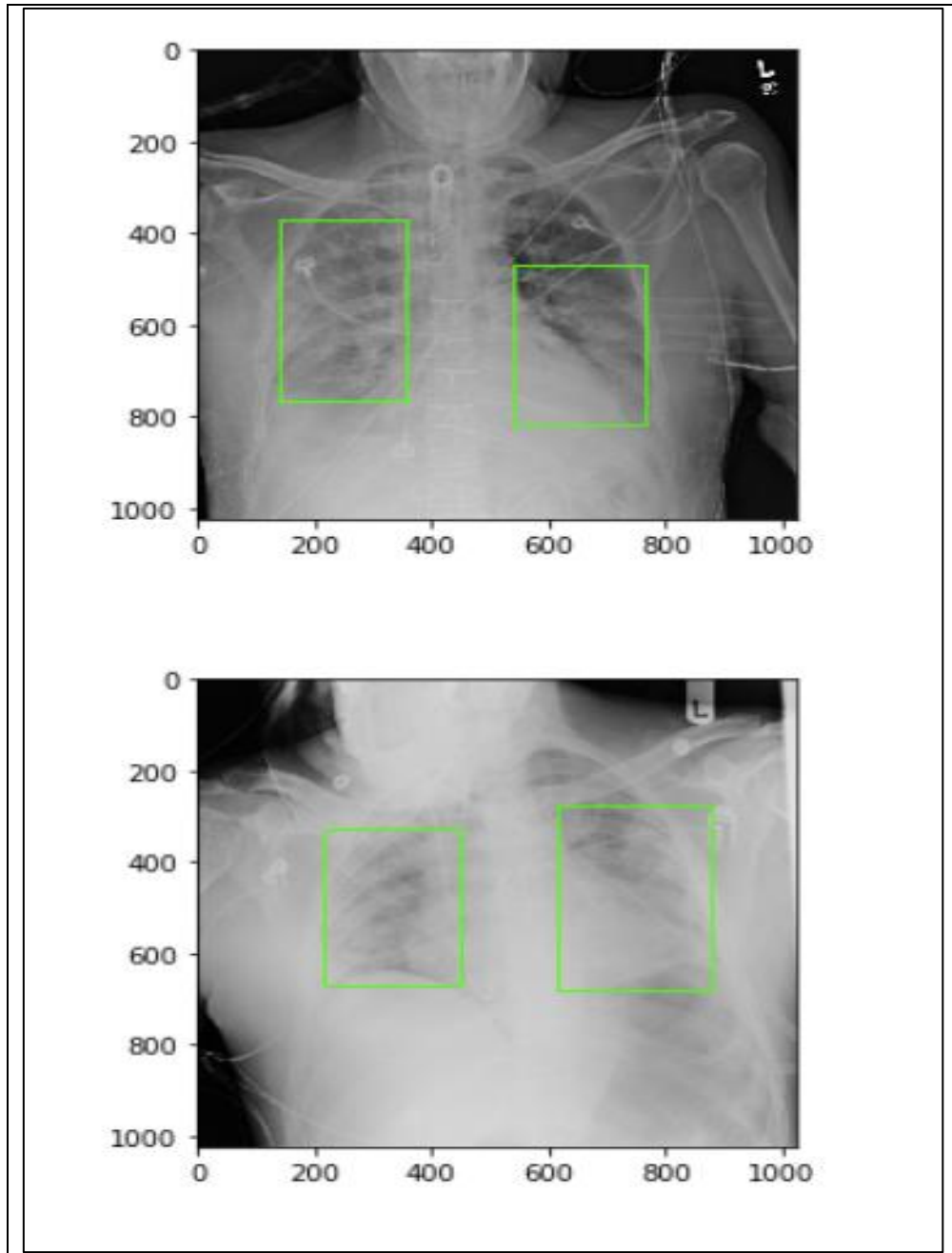
Model Hyperparameters

1) Set Higher Learning rate to train heads faster with 2 epochs.

2) Excluded last layers from training and trained on COCO datasets.

3) Original Image size is 1024x1024 with 3 channels which was scaled down to 256x256x3 to feed in the neural network.

4) This model consists of Time Distributed convolutional, Batch normalization layers and ResNet as a backbone architecture.

5) Ran the model in Inference mode setting GPU_COUNT=1 and IMAGES_PER_GPU=1.

6) Image Augmentation technique used to sharpen, blur and scale images.

```
LEARNING_RATE = 0.005
```

```
# Exclude the last layers because they require a matching
# number of classes
model.load_weights(COCO_WEIGHTS_PATH, by_name=True, exclude=[
    "mrcnn_class_logits", "mrcnn_bbox_fc",
    "mrcnn_bbox", "mrcnn_mask"])
```

7) Calculated IOU per image using helper function which predicted 0.75 pneumonia in patients.

Loss Function in Mask R-CNN:

Loss in Mask R-CNN is consisting of loss due to RPN (Regional Proposed Network) and loss due to classification, localization and segmentation mask.

1. Loss (RPN)= RPN_Class Loss + RPN_BBox Loss

2. Loss (Mask R-CNN) = Loss (class labels prediction) + Loss (Bounding Box prediction) + Loss (Mask Prediction)

Total Loss= Loss (RPN) + Loss (Mask R-CNN)

So, our optimization problem is to minimize the Total Loss

Scope of Improvement

1) Mask shape can be improved to the exact size of the infected area.

2) The implemented model RPN_Class Loss and RPN_BBox Loss has significant improvement per epoch, but Total Loss can be improved by changing some hyperparameters.

ResNet

ResNet, short for Residual Networks is a classic neural network used as a backbone for many computer vision tasks. This model was the winner of ImageNet challenge in 2015. The fundamental breakthrough with ResNet was it allowed us to train extremely deep neural networks with 150+layers successfully. Prior to ResNet training very deep neural networks was difficult due to the problem of vanishing gradients.

The main goal of the residual network is to build a deeper neural network. We can have two intuitions based on this:

- As we keep going deeper into implementing large number of layers, one should make sure not to degrade the accuracy and error rate. This can be handled by identity mapping.

- Keep learning the residuals to match the predicted with the actual

These are the functions of a Residual Network
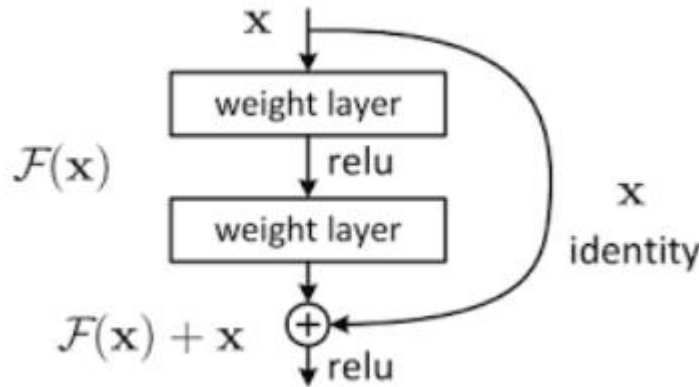
$$y = F(x, \{Wi\}) + x \; \text{---- (1)}$$

x and y are the input and output vectors of the layers considered. The function F(x, {Wi}) represents the residual mapping to be learned. For the example in Fig that has two layers,

$$F = W2\sigma(W1x) \text{ in which } \sigma \text{ denotes ReLU.}$$

$$y = F(x, \{Wi\}) + Wsx \; \text{---- (2)}$$

23

We can also use a square matrix Ws in Eqn. (1). But we will show by experiments that the identity mapping is sufficient for addressing the degradation problem and is economical, and thus Ws is only used when matching dimensions.

1) More layers are better but because of the vanishing gradient problem, model weights of the first layers cannot be updated correctly through the backpropagation of the error gradient (the chain rule multiplies error gradient values lower than one and then, when the gradient error comes to the first layers, its value goes to zero).

2) That is the objective of Resnet: preserve the gradient.

You can even forget the vanishing gradient problem and just look at an image of a Resnet network: the identity matrix transmits forward the input data that avoids the loose of information (the data vanishing problem). With ResNets, the gradients can flow directly through the skip connections backwards from later layers to initial filters.

Model Hyperparameters

1) Created a Data generator that takes input size, filenames and batch size.
2) Custom model with ResNet as a backbone architecture setting number of channels, image size, number of blocks and depth of a network.
3) Original Image size is 1024x1024 with 3 channels which is transformed into 256x256x3 to feed in the network for training.
4) Optimizers used are Adam, RMSProp and IOU as a metric to calculate bounding boxes on images.
5) callbacks used in Model hyperparameters for early stopping and reduce learning rate on plateau

Loss Functions in ResNet:

```python
# define iou or jaccard loss function
def iou_loss(y_true, y_pred):
    y_true = tf.reshape(y_true, [-1])
    y_pred = tf.reshape(y_pred, [-1])
    intersection = tf.reduce_sum(y_true * y_pred)
    score = (intersection + 1.) / (tf.reduce_sum(y_true) + tf.red
uce_sum(y_pred) - intersection + 1.)
    return 1 - score


# combine bce loss and iou loss
def iou_bce_loss(y_true, y_pred):
```

```python
    return 0.5 * keras.losses.binary_crossentropy(y_true, y_pred)
+ 0.5 * iou_loss(y_true, y_pred)


# mean iou as a metric
def mean_iou(y_true, y_pred):
    y_pred = tf.round(y_pred)
    intersect = tf.reduce_sum(y_true * y_pred, axis=[1, 2, 3])
    union = tf.reduce_sum(y_true, axis=[1, 2, 3]) + tf.reduce_sum
(y_pred, axis=[1, 2, 3])
    smooth = tf.ones(tf.shape(intersect))
    return tf.reduce_mean((intersect + smooth) / (union - interse
ct + smooth))
```

Scope of Improvement

1) This model gives better accuracy in training that is 97% and IOU of 73%
2) Performance of the model can be improved by increasing batch size and training for a greater number of epochs to increase IOU accuracy.

From the above inferences we feel that the MaskRCNN performs better as it helped us in achieving the Benchmark results.

COMPARISION TO BENCHMARK

At the beginning of the model building process we set out to achieve the IOU of 0.75 or more with the limited hardware and system resources available to run the model. From the above section it is evident that we achieved that benchmark with less epochs. But with the better resources and processors our benchmark would have been better. We were able to achieve the benchmark by trying different hyperparameters through one at a time method (OFAT). After reaching the benchmark we continued on and found out that computation cost is increasing which is not feasible with the system resources and GPU time.

IMPLICATIONS

Our solution has a huge advantage in the field of radiology given the better memory and computation resources it makes a huge difference in identifying the lung opacities from the lung X-rays. From the above findings we would recommend a training model that is dedicated to the fields of medicine and radiology which would give good accuracy by reducing the error rate and regulation of the training time.

25

LIMITATIONS

- As our model has huge number of parameters with large number of layers of Resnet as a backbone it is tend to take longer to train.

- Bounding boxes can be improved to the exact size of the infected area. So, the limitation is its very difficult to do the segmentation as the bounding boxes will include certain portion of uninfected area.

- The implemented model RPN_Class Loss and RPN_BBox Loss has significant improvement per epoch, but Total Loss can be improved by changing some hyperparameters.

- This model is computationally expensive. Which can be a huge drawback as the initial cost of implementation maybe higher.

CLOSING REFLECTIONS

As our approach resulted in reaching the benchmark, it can be illustrated that our approach to the problem is experimental. The lack of domain knowledge in the initial phase of the project is slow and inefficient. We studied the dataset with significantly less idea about the inner workings of pneumonia or opacities. So, we concur that a domain expert is essential for the more feasible and efficient solution.

We conclude that even though there have been substantial deep learning advances in radiology and medicine, we need better models and strategies that are majorly dedicated to those fields as the amount of data is vast. The margin of error allowed is very small or sometimes none.