CV Project: Pneumonia Detection

Dinesh Solunke, Nivedita Rana, Pittu Chiradeep Krishna, Sharma Shekhar Jeevendra Nath, Ronit Kumar Gouda

(https://github.com/ChiruKrish/CapstoneProject_AIML.git)

PROBLEM STATEMENT

What is Pneumonia?

Pneumonia is an infection in one or both lungs. Bacteria, viruses, and fungi cause it. The infection causes inflammation in the air sacs in your lungs, which are called alveoli. Pneumonia accounts for over 15% of all deaths of children under 5 years old internationally. In 2017, 920,000 children under the age of 5 died from the disease. It requires review of a chest radiograph (CXR) by highly trained specialists and confirmation through clinical history, vital signs and laboratory exams. Pneumonia usually manifests as an area or areas of increased opacity on CXR. However, the diagnosis of pneumonia on CXR is complicated because of a number of other conditions in the lungs such as fluid overload (pulmonary edema), bleeding, volume loss (atelectasis or collapse), lung cancer, or post-radiation or surgical changes. Outside of the lungs, fluid in the pleural space (pleural effusion) also appears as increased opacity on CXR. When available, comparison of CXRs of the patient taken at different time points and correlation with clinical symptoms and history are helpful in making the diagnosis. CXRs are the most commonly performed diagnostic imaging study. A number of factors such as positioning of the patient and depth of inspiration can alter the appearance of the CXR, complicating interpretation further. In addition, clinicians are faced with reading high volumes of images every shift.

Pneumonia Detection Now to detection Pneumonia we need to detect inflammation of the lungs. In this project, you’re challenged to build an algorithm to detect a visual signal for pneumonia in medical images. Specifically, your algorithm needs to automatically locate lung opacities on chest radiographs.

SUMMARY

Pneumonia is a condition with infection in lung parenchyma, the cause for which can be either bacterial or viral. A number of patients die due to incorrect diagnosis of pneumonia. The primary modality for diagnosis of pneumonia is chest X-rays(CXR). The pneumonia on CXR presents as opacity. But the presence of opacity in CXR is not specific to pneumonia but can be due to other lung conditions as well, like pulmonary edema, bleeding, volume loss (atelectasis or collapse), lung cancer, or post-radiation or surgical changes and pleural effusion. So, the problem statement at hand needed us to build pneumonia detection system, which can aid in correct diagnosis of pneumonia. The data consists of the chest X-ray images of the patient. The intent was to build an object detection model capable of first, classifying the image as pneumonia or not pneumonia and second, localising the site of pneumonia within the lung in case of pneumonia image, with good accuracy. This was done in two parts, exploring data analysis (EDA) and model building. The data was freely available on Kaggle and was collected from Radiological Society of North America (RSNA) and had approximately 30000 images. After exploring the given data files, classes and images of different classes,checking for duplicate and null values, the images were visualised of different classes. Object detection model were then build using region proposal based algorithm (Faster-RCNN) and also without region proposal based algorithm(YOLO). The proposed method achieved mAP(mean of Average Precision) values of 0.55 and 0.4 using Faster R-CNN and Yolo, respectively

EDA and Pre-processing

The data has been provided in both csv and dicom file format. Three data sets have been provided consisting of train label data, class label data and test data.

1.1 Shape and info of data in csv format

- Label Data:

  Shape of the data was found to be (30227, 6), implying 30227 no of rows/patient images with 6 columns. The six columns were found to be patientId, x, y, width, height and Target. Of these numerical attributes were x, y, width and height.Target was a categorical attribute. It was observed that the target attribute had two values. The value “0” denotes no disease or not pneumonia and value “1” denotes pneumonia.

- Class Details:

  Shape of the class_label data frame was found to be (30227, 2), implying 30227 no. of rows/patient images with 2 columns. The two columns were found to be patientId and class label.class label was a categorical attribute with three values No Lung Opacity/Not Normal, Normal and Lung opacity.

1.2 Checking null Values

The values for bounding box coordinates (x,y, width and height) were null for all the patients with target value “0” and bounding box coordinates (x,y, width and height) were defined only for target “1”. However, since “0” denotes no disease, no bounding box and hence no coordinates were expected. So, such values were not dropped considering null values.

1.3 Checking duplicate Values

While checking for duplicate values, it was found that some of the patientIDs were same. However, the bounding box coordinates for the patientID with same value were different. Implying having more than one bounding box for some patients.

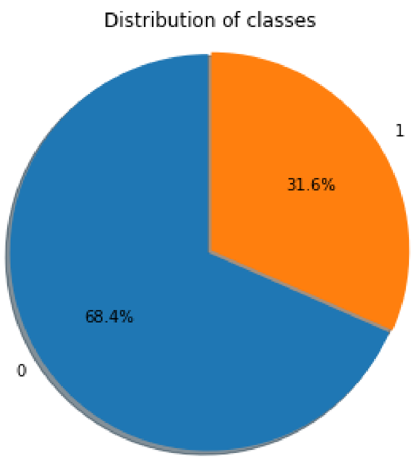
1.4 Checking for no. of bounding boxes

Patient IDs were grouped to check the number of bounding boxes for each unique p atient ID

No. of bounding boxes	No. of patients
1	23286
2	3266
3	119
4	13

1.5 Checking for target balancing

- 68.39% - Out of 30227, 20672 are identified as Negative with Pneumonia
- 31.61% - Out of 30227, 9555 are identified as Positive with Pneumonia



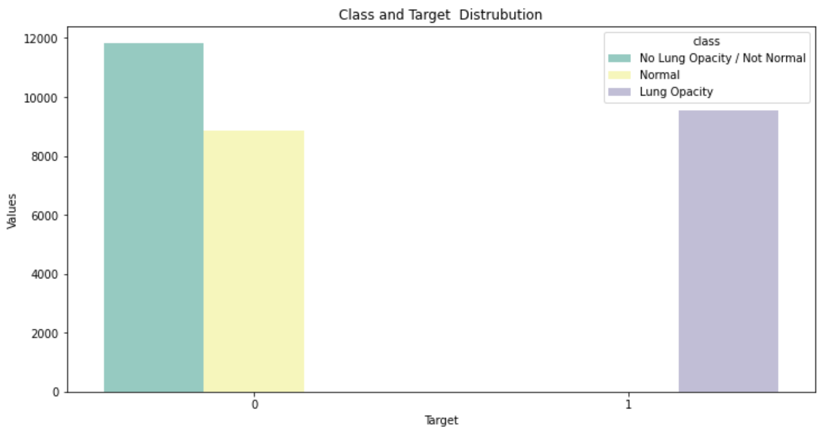
- From the pie chart above it is evident that the dataset is imbalanced which needs to be fixed before model fitting as 31.6% of the dataset have positive pneumonia evidence and the rest show negative evidence of pneumonia detection.

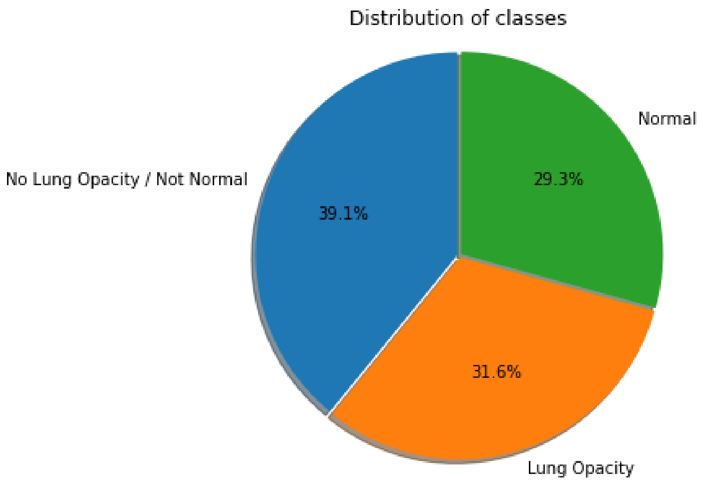
1.6 Merging the two data frames of labels and class label

The two data frames were merged for the common attribute “Patient id”

Class label Vs Target

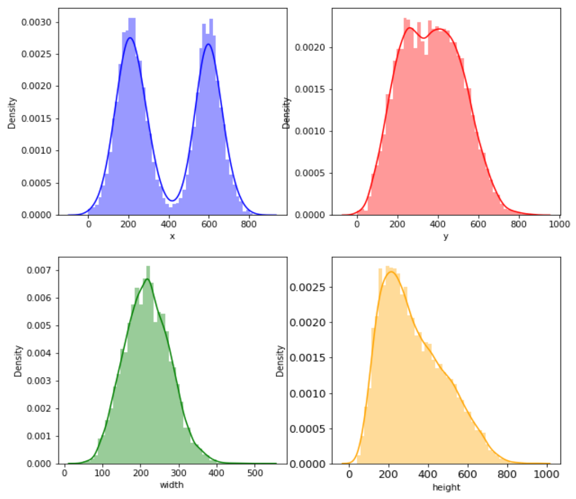
The patients target values were further mapped to three class labels. However it was observed that all the patients with target value “1” had only one class label “Lung opacity”. The other two class labels were associated with only Target “0”. This observation is in line with the fact that pneumonia on X-ray image is diagnosed as lung opacity.





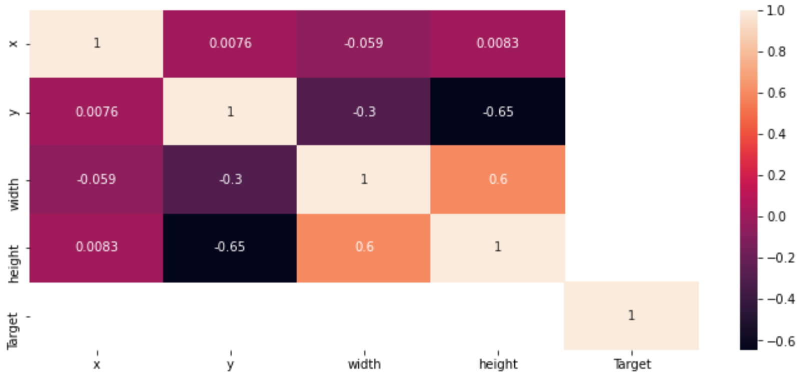
- 39.1% - Out of 30227, 11821 are identified as 'No Lung Opacity / Not Normal'
- 31.6% - Out of 30227, 9555 are identified as 'Lung Opacity'
- 29.3% - Out of 30227, 9555 are identified as 'Normal'

1.7 Checking the distribution of numerical attributes



The attribute "x" seemed to be sum of two normal distributions, "y" attribute also showed two peaks, "width" was found to be normally distributed and "height" was right skewed.

1.8 Correlation of attributes



	x	y	width	height	Target
x	1	0.0076	-0.059	0.0083	
y	0.0076	1	-0.3	-0.65	
width	-0.059	-0.3	1	0.6	
height	0.0083	-0.65	0.6	1	
Target					1

The attributes did not have any correlation amongst themselves

1.9. Observations from the final dataset prepared

Based on analysis above, some of the inferences found are:

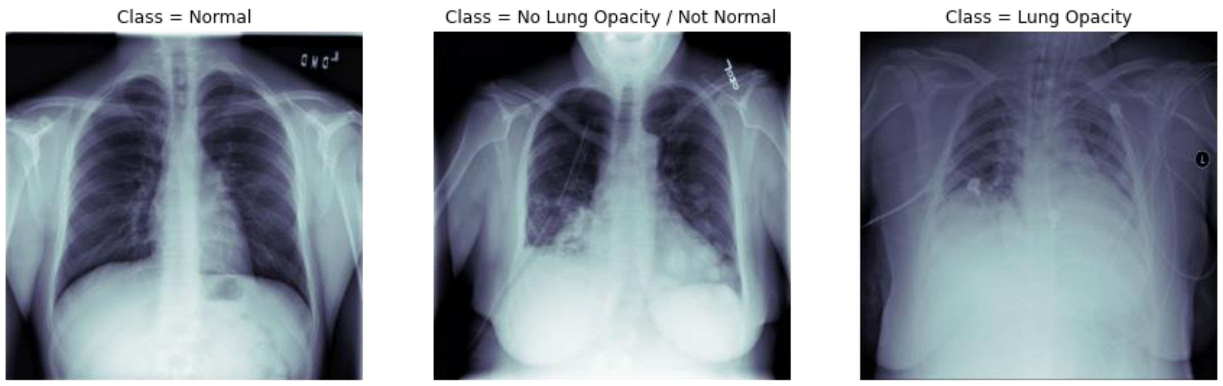
- Training data is having a set of patientIds and bounding boxes.
- Bounding boxes are defined as follows: x, y, width and height.
- There are multiple records for patients. Number of duplicates in patientID = 3,543.

- There is also a binary target column i.e. Target indicating there was evidence of pneumonia or no definitive evidence of pneumonia.
- Class label contains: No Lung Opacity/Not Normal, Normal and Lung Opacity.
- Chest examinations with Target = 1 i.e. ones with evidence of Pneumonia are associated with Lung Opacity class.
- Chest examinations with Target = 0 i.e. those with no definitive effidence of Pneumonia are either of Normal or No Lung Opacity / Not Normal class.
- About 23,286 patientIds (~87% of them) provided have 1 bounding boxes while 13 patients have 4 bounding boxes.

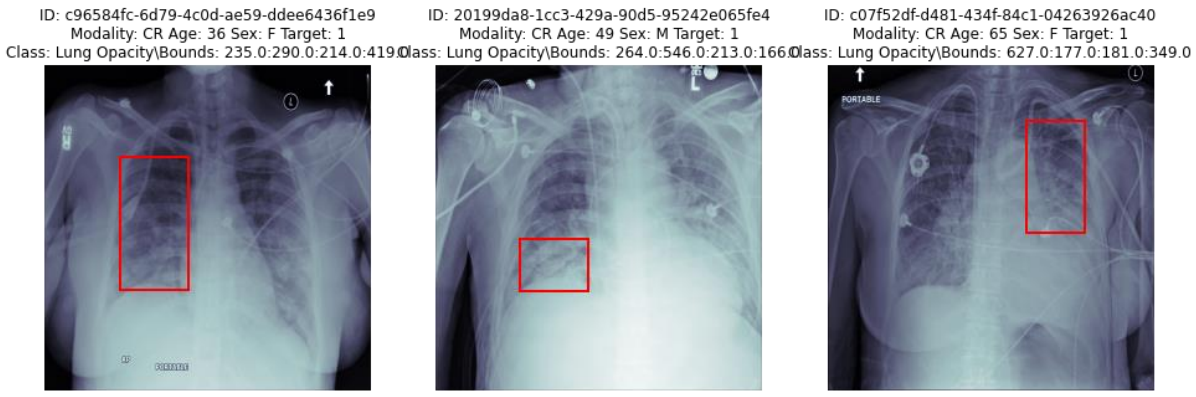
2.1 Analysing the dicom images

Some of the dicom images in the training data were visualised both with and without bounding boxes. Also images belonging to three different class labels were visualised.

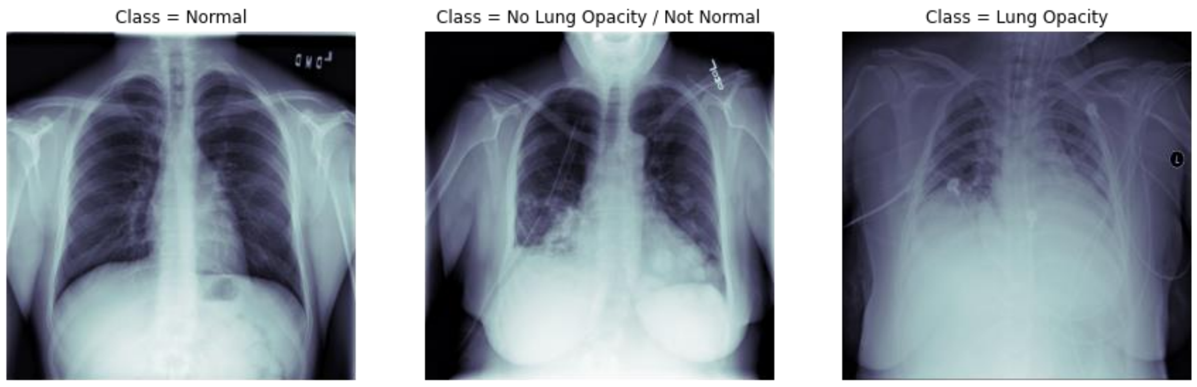
2.1.1 Without Pneumonia



2.1.2 With Pneumonia and bounding box


ID: c96584fc-6d79-4c0d-ae59-ddee6436f1e9
Modality: CR Age: 36 Sex: F Target: 1
Class: Lung Opacity\Bounds: 235.0:290.0:214.0:419.0
ID: 20199da8-1cc3-429a-90d5-95242e065fe4
Modality: CR Age: 49 Sex: M Target: 1
Class: Lung Opacity\Bounds: 264.0:546.0:213.0:166.0
ID: c07f52df-d481-434f-84c1-04263926ac40
Modality: CR Age: 65 Sex: F Target: 1
Class: Lung Opacity\Bounds: 627.0:177.0:181.0:349.0
PORTABLE

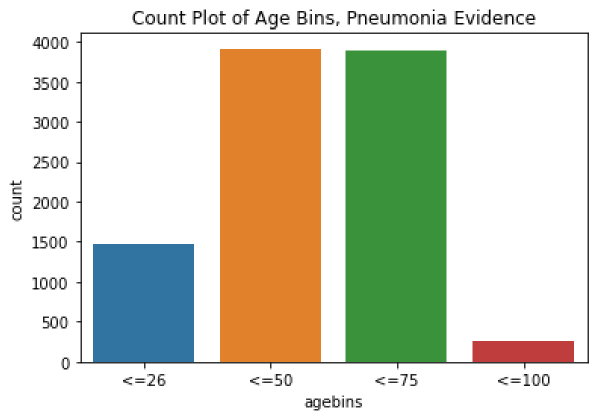
2.1.3 Different class labels



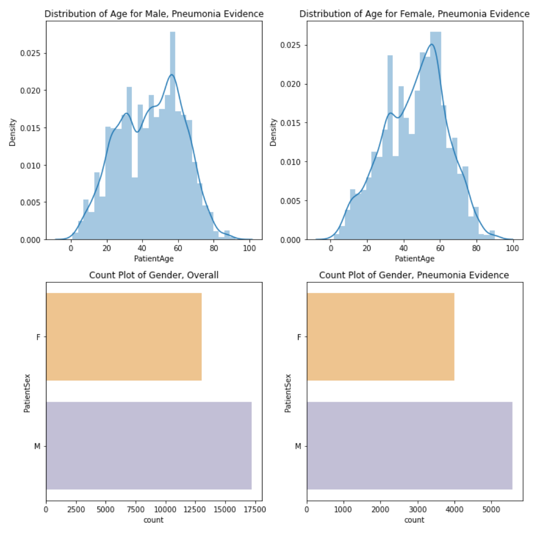
2.2 Getting metadata of each patient

Metadata information like patient’s age, sex etc were added to the train and test data

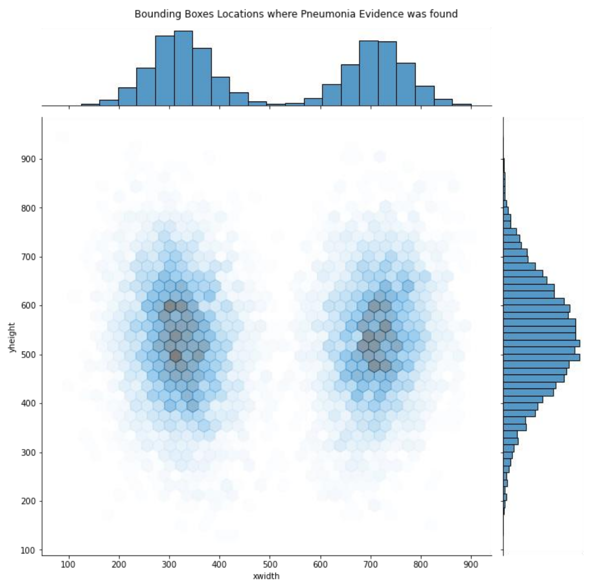
2.2.1 Distribution of patients with pneumonia in different age groups



2.2.2 Creating distribution of age with pneumonia evidence, by gender and count plot of gender



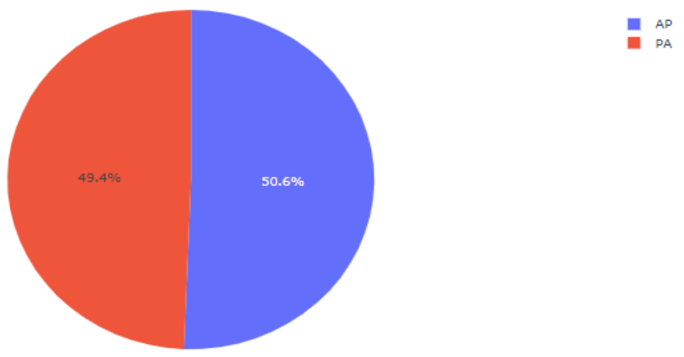
2.2.3 Getting the centre position of each bounding box



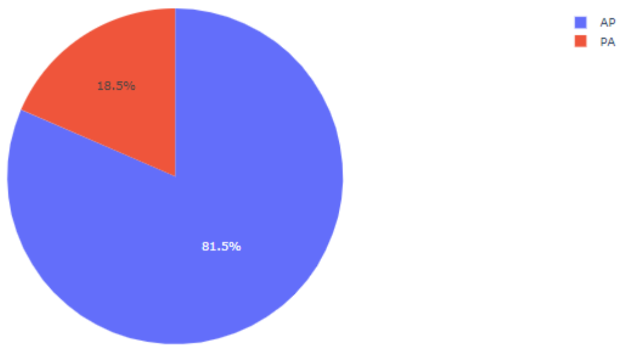
It can be noted that most of the defects were present in the centre of the lungs

2.2.4 Distribution of View Position

- **Posterior/Anterior (PA):** In PA, X-Ray beam hits the posterior (back) part of the chest before the anterior (front) part. While obtaining the image patient is asked to stand with their chest against the film.
- **Anterior/Posterior (AP):** At times it's not possible for radiographers to acquire a PA chest X-ray. This is usually because the patient is too unwell to stand. AP projection images are of lower quality than PA images. Heart size is exaggerated (cardiothoracic ratio approximately 50%)



Overall distribution of AP and PA



Distribution of AP and PA in patients with evidence of lung

Biasing in presence of pneumonia can be observed in AP images, this may be due to better image quality in case of PA images than AP

MODEL BUILDING

Since this is an object detection problem, algorithms like Region-based Convolutional Neural Networks(or R-CNN), Fast-RCNN, Faster-RCNN, RetinaNets, SSD (single shot detector), YOLO (you only look once) can be used.

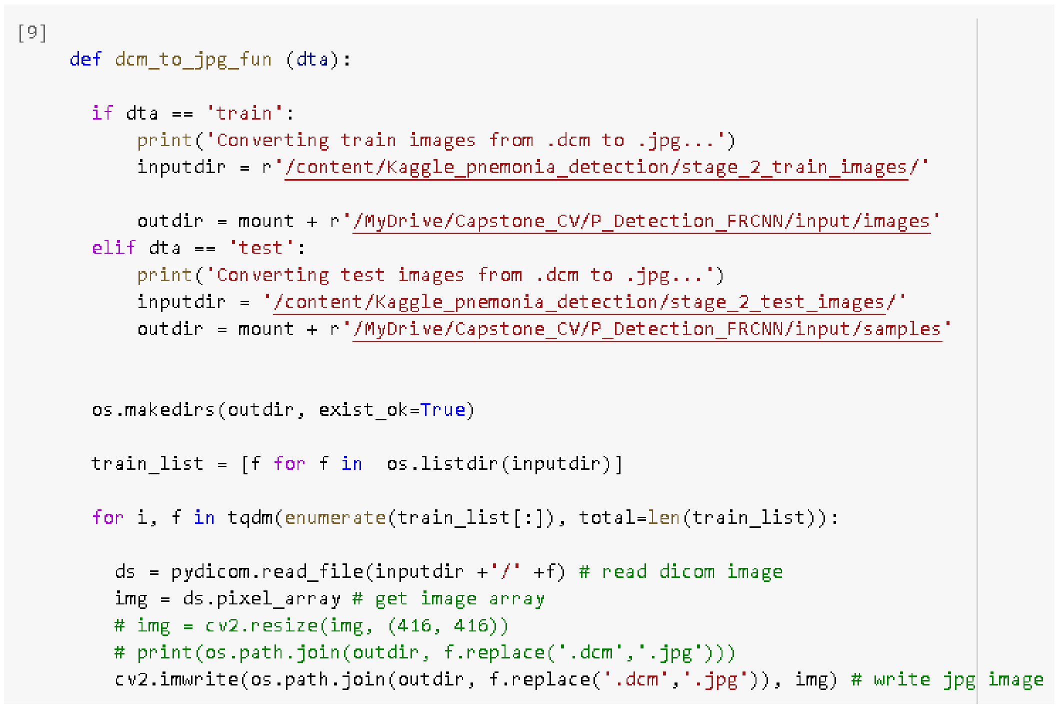
The main advantage of using Region-based Convolutional Neural Networks algorithm is that it really fast. However, the weakness of this technique is that the position of the bounding boxes is not very accurate.

YOLO algorithm is better algorithm that tackles the issue of predicting accurate bounding boxes while using the convolutional sliding window technique. This algorithm provides high accuracy while running in real time.

Faster-RCNN

Data Preparation:

Step-I: Convert images ‘dcm’ to ‘jpg’ format.



```
[9]
    def dcm_to_jpg_fun (dta):

      if dta == 'train':
          print('Converting train images from .dcm to .jpg...')
          inputdir = r'/content/Kaggle_pnemonia_detection/stage_2_train_images/'

          outdir = mount + r'/MyDrive/Capstone_CV/P_Detection_FRCNN/input/images'
      elif dta == 'test':
          print('Converting test images from .dcm to .jpg...')
          inputdir = '/content/Kaggle_pnemonia_detection/stage_2_test_images/'
          outdir = mount + r'/MyDrive/Capstone_CV/P_Detection_FRCNN/input/samples'


      os.makedirs(outdir, exist_ok=True)

      train_list = [f for f in  os.listdir(inputdir)]

      for i, f in tqdm(enumerate(train_list[:]), total=len(train_list)):

        ds = pydicom.read_file(inputdir +'/' +f) # read dicom image
        img = ds.pixel_array # get image array
        # img = cv2.resize(img, (416, 416))
        # print(os.path.join(outdir, f.replace('.dcm','.jpg')))
        cv2.imwrite(os.path.join(outdir, f.replace('.dcm','.jpg')), img) # write jpg image
```

Above image show python code to convert images. Advantage of doing conversion is to make model easier and lighter while training.

Step-II: Read input data (Label info, class info and images)

Reading input data is divided into three different steps.

1. Create a function to collate all information into one. In this for each patient we are getting image, gt box (ground truth box) , labels and patientid in one location.
2. Then we are creating training and validation dataset. Training data will be used to train the model and we will test our model on validation data set.
3. We will pass our data in batch (size = 8) during training process. This is preserve resources like RAM and GPU memory.

Step-III: Validation

In this section we will discuss on metrics used for measuring performance of our model.First, we get IoU and then we calculate mAP. Detailed explanation is below

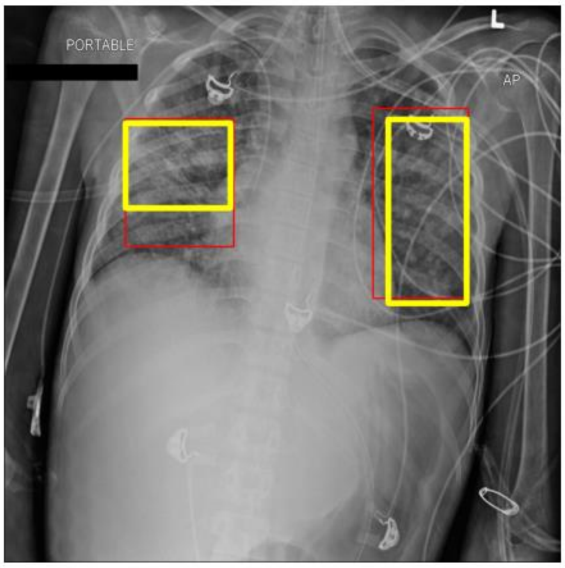
Intersection over Union (IoU):

Intersection over Union is an evaluation metric used to measure the accuracy of an object detector on a particular dataset. Any algorithm that provides predicted bounding boxes as output can be evaluated using IoU. To apply Intersection over Union to evaluate an (arbitrary) object detector we need:

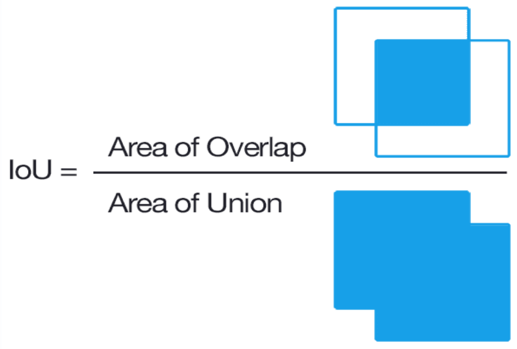
1. The ground-truth bounding boxes (i.e., the hand labeled bounding boxes from the testing set that specify where in the image our object is).

2. The predicted bounding boxes from our model.

Below I have included a visual example of a ground-truth bounding box versus a predicted bounding box:



Computing Intersection over Union can therefore be determined by:



In the numerator we compute the **area of overlap** between the *predicted* bounding box and the *ground-truth* bounding box.

The denominator is the **area of union**, or more simply, the area encompassed by *both* the predicted bounding box and the ground-truth bounding box.

Dividing the area of overlap by the area of union yields our final score — *the Intersection over Union.*

Mean average precision (mAP):

Mean average precision (mAP) is the most meaningful metric for object detectors, instance, and semantic segments. It incorporates the trade-off between precision and recall, and in doing so, takes into account both types of errors, false positives (FP) and false negatives (FN). This property makes mAP applicable for most use cases.

For calculating Precision and Recall, we will use IoU to get precision and Recall. The most used threshold is 0.5 - i.e. If the IoU is > 0.5, it is considered a True Positive, else it is considered a false positive.

In our current model we are using 0.5 as or threshold to calculate precision. Precision is calculated for 7 different thresholds. Once we have all threshold, it is divided by number of threshold and mean is calculated which give us mAP.

"mAP of current faster RCNN model is 0.5534"

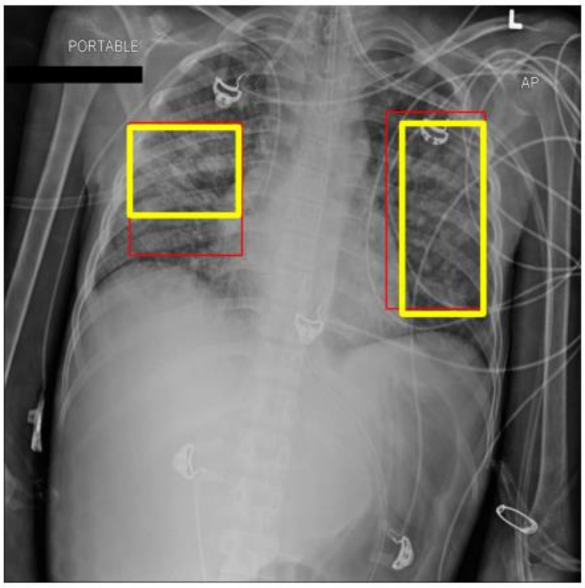
Hyper Parameters:

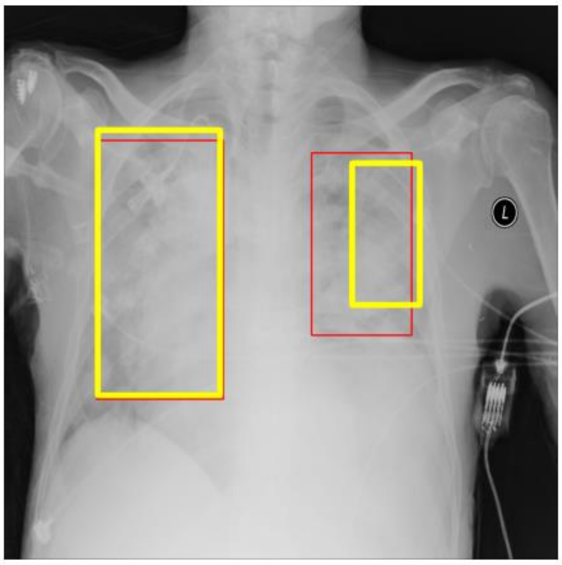
To train the model, we are using following hyper parameters:

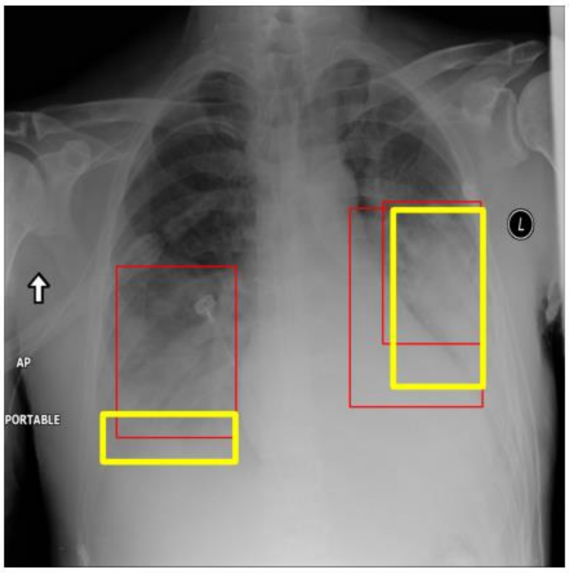
Number of iterations = 50
Learning rate = 0.01
Batch Size = 8
Optimizer = SGD
Momentum = 0.9
weight_decay = 0.5

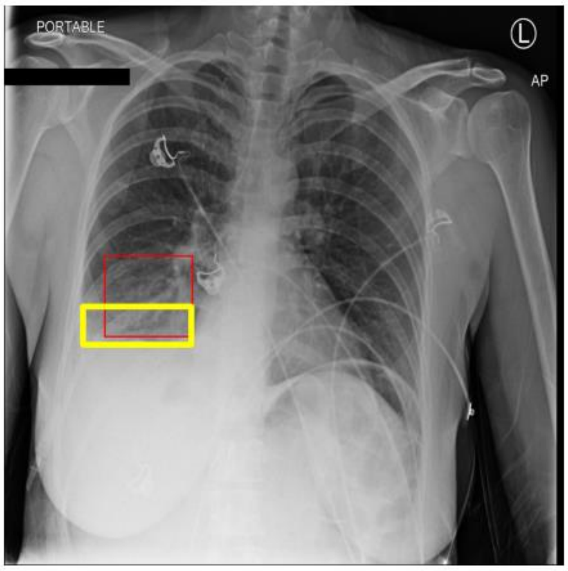
Model Performance:

Some examples of the predictions done on validation data are given below. In the images you can see Ground truth (in yellow) and predicted box (in red)





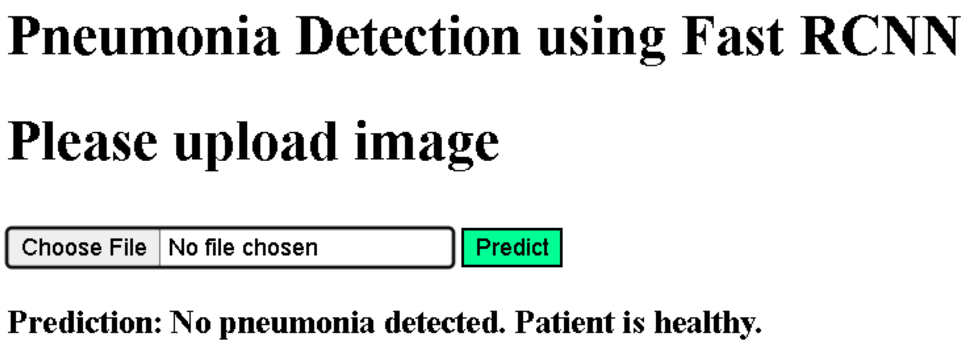




FLASK App:

We have created Flask application which will enable user to check images individually and get the predictions. This app will predict 2 class

1. "No pneumonia detected. Patient is healthy"

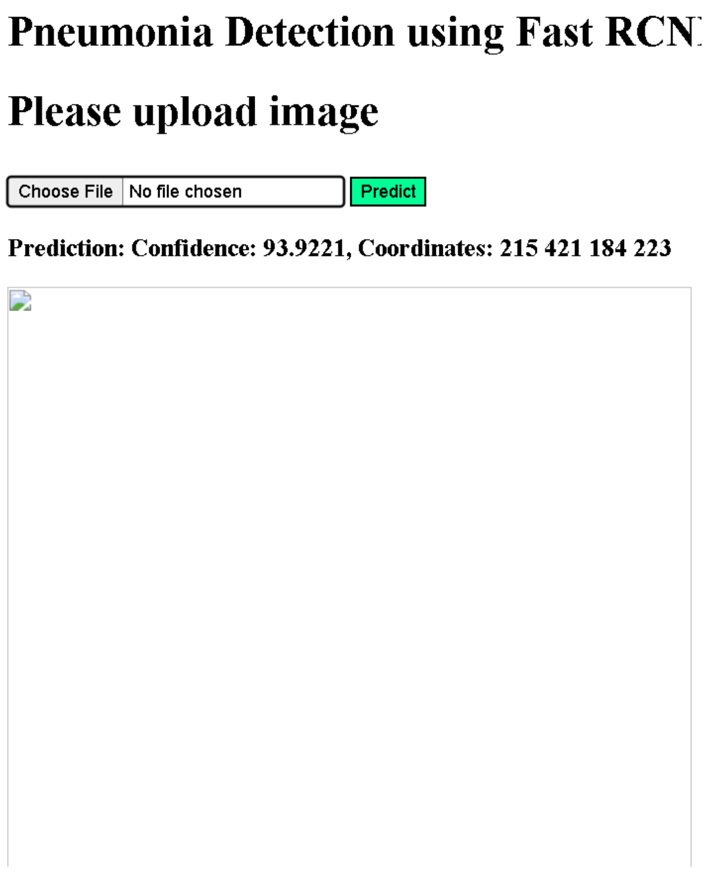


Pneumonia Detection using Fast RCNN

Please upload image

Prediction: No pneumonia detected. Patient is healthy.

2. Pneumonia detected (Confidence and coordinates) (Image not visible because of credential issues)



Pneumonia Detection using Fast RCN

Please upload image

Prediction: Confidence: 93.9221, Coordinates: 215 421 184 223

YOLO

The YOLO model and its other requirements were downloaded and yolo directory was created

1. **Conversion of images** from dicom format to jpg format: The data was in two folders TRAIN(stage_2_train_images) and TEST(stage_2_test_images).Both train and test images were converted to jpg format and saved as separate folders.

2. **Splitting the train data** into train(80%) and validation data (20%) using random train_test_split

3.**Creating data.yaml file**

The data.yaml, is the dataset configuration file that defines:

1. the dataset root directory and relative paths to train/val/test image directories (or paths to *.txt files with image paths).
2. the number of classes.
3. a list of class names.

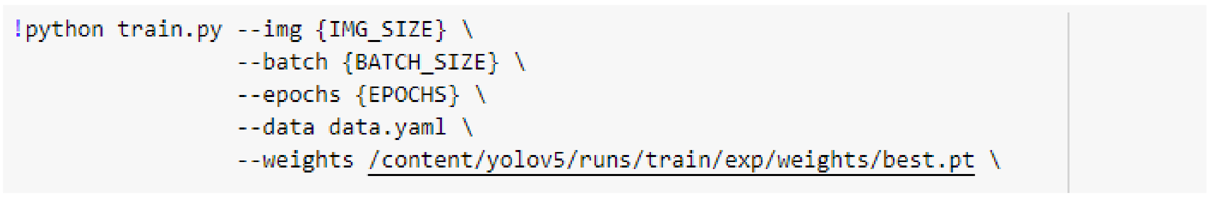
4. **Prepare Bounding Box Coordinated** for YOLOv5

For every image with **bounding box(es)** a `.txt` file with the same name as the image will be created in the format shown below:

- One row per object.
- Each row is class x_center y_center width height format.
- Box coordinates must be in normalized xywh format (from 0 - 1). We can normalize by the boxes in pixels by dividing x_center and width by image width, and y_center and height by image height.
- Class numbers are zero-indexed (start from 0).

5. **Defining Hyperparameters** like image size=1024, epochs and batch size

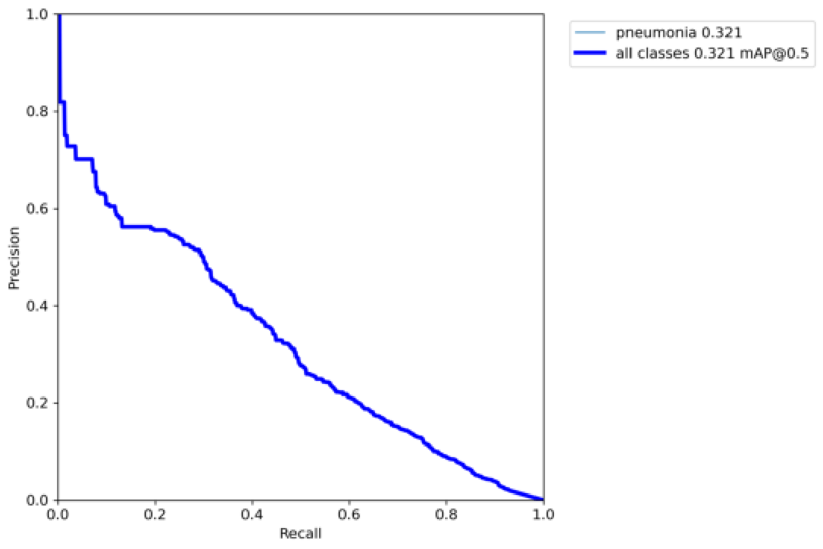
6. **Training:**Directory was again changed to yolov5 directory and the YOLO model was trained by giving arguments shown



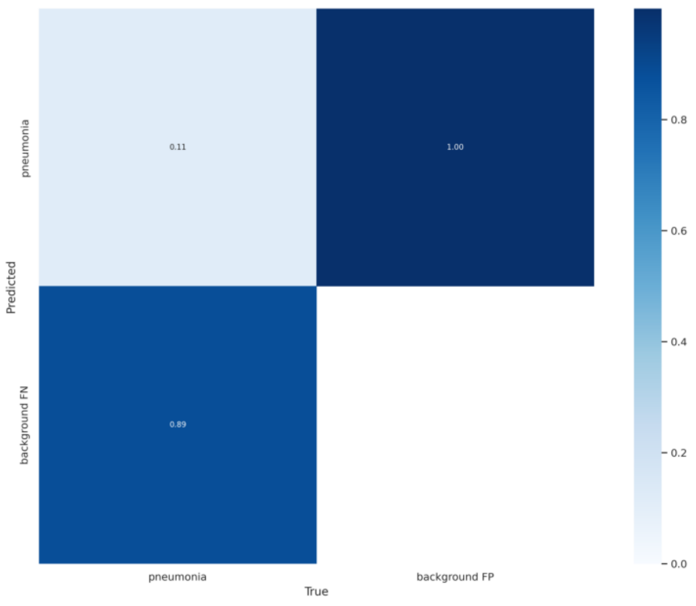
```
!python train.py --img {IMG_SIZE} \
                 --batch {BATCH_SIZE} \
                 --epochs {EPOCHS} \
                 --data data.yaml \
                 --weights /content/yolov5/runs/train/exp/weights/best.pt \
```

Metrics for calculating accuracy: Other than Precision and Recall, prediction accuracy of object detection was calculated using mAP: Mean average precision

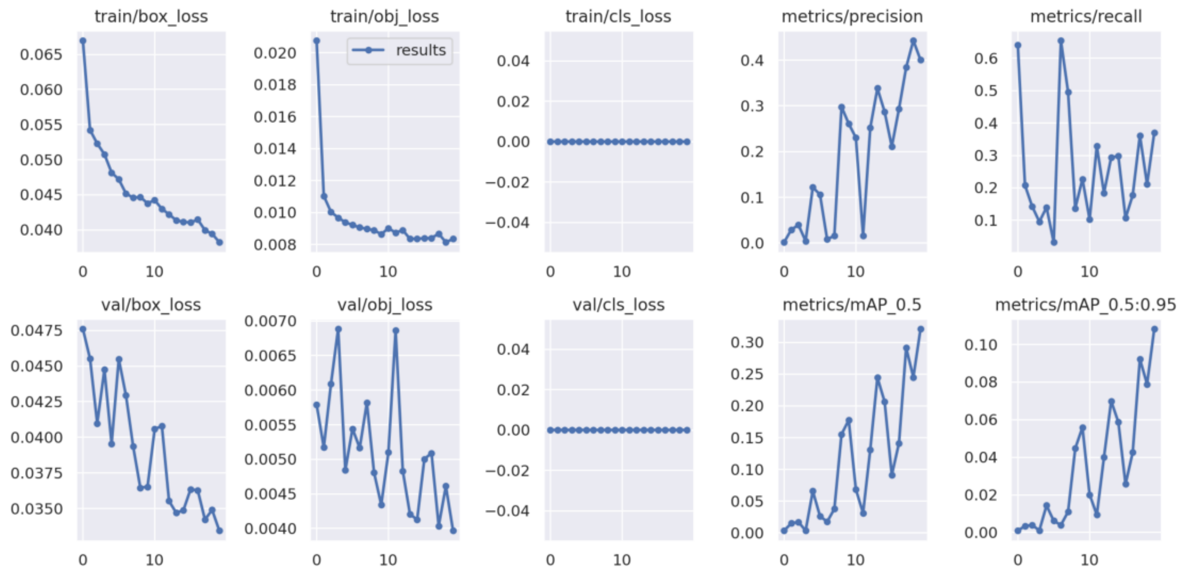
7. Plotting precision Recall curve



8. Plotting Confusion matrix



9. Printing Training details



Limitations:

The data if trained for more epochs would give better results. Due to time constraint it could be trained only for less number of epochs.

Conclusion:

The Faster R-CNN model could successfully classify and detect image coordinates with a mean Average of precision values of 0.5 approximately