AIML Capstone Project

Pneumonia Detection Challenge

Final Report

Submitted by,

Group I - Oct – 22B

Group Members	
Name	**Mobile**
Deep	9886729655
Ishwarya Jayaraman	9629809364
Zacharia	9847623955
Pooja Nagare	9175931271
Sree Latha	6305838547
Asish Bargotra	8816082470

Mentor: Ankit Chhabra

October 2023

Table of Contents:

1. Overview:

Pneumonia is a form of acute respiratory infection that affects the lungs. The lungs are made up of small sacs called alveoli, which fill with air when a healthy person breathes. When an individual has pneumonia, the alveoli are filled with pus and fluid, which makesbreathing painful and limits oxygen intake.

Following are some of the key facts about Pneumonia which needs at most attention to address the problem proactively.

- Pneumonia accounts for 15% of all deaths of children under 5 years old,killing 808 694 children in 2017.
- Pneumonia can be caused by viruses, bacteria, or fungi.
- Pneumonia can be prevented by immunization, adequate nutrition, and byaddressing environmental factors.
- Pneumonia caused by bacteria can be treated with antibiotics, but only onethird of children with pneumonia receive the antibiotics they need.

Per WHO, Pneumonia is the single largest infectious cause of death in children worldwide. Pneumonia killed 808 694 children under the age of 5 in 2017, accountingfor 15% of all deaths of children under five years old. Pneumonia affects children and families everywhere but is most prevalent in South Asia and sub-Saharan Africa. Children can be protected from pneumonia, it can be prevented with simple interventions, and treated with low-cost, low-tech medication and care.

The WHO and UNICEF integrated Global action plan for pneumonia and diarrhea (GAPPD) aims to accelerate pneumonia control with a combination of interventions to protect, prevent, and treat pneumonia in children with actions to:

- protect children from pneumonia including promoting exclusive breastfeeding and adequate complementary feeding.
- prevent pneumonia with vaccinations, hand washing with soap, reducing household air pollution, HIV prevention and cotrimoxazole prophylaxis for HIV-infected and exposed children.
- treat pneumonia focusing on making sure that every sick child has access to the right kind of care -- either from a community-based health worker, or in a health facility if the disease is severe -- and can get the antibiotics and oxygenthey need to get well.

2. Abstract:

This project is aimed at detecting Pneumonia by locating the lung opacities on the Chest radiographs. This process can help identify the problem at an early stage as well as help the Clinical analysis much faster and drives better decision making. This process of Pneumonia detection will be done by looking at several thousand images of Chest radiographs taken from the past wherein the analysis and desired results were identified by the specialists. These past datapoints will become the indicators and these images will be processed through the Computer Vision Technology of deep learning to capture every detail by which a Deep Learning Algorithm will be built.

The new patient's data will be fed to this model which detects the Lung Opacity indication along with its location such that Clinical specialists will be able to confirm diagnosis quickly and can help in taking respective decisions quickly to move forward with the next steps of the treatment.

Deep neural networks models have conventionally been designed and experiments were performed upon them by human experts in a continuing trial and error method. This process demands enormous time, knowhow, and resources. To overcome this problem, a novel but simple model is introduced to automatically perform optimal classification tasks with deep neural network architecture.

The Neural network architecture was specifically designed for Pneumonia image classification tasks. The proposed technique is based on the CNN algorithm, utilizing a set of neurons to convolve on a given sample images to extract relevant features from them. This is demonstrated through validating the accuracy of the detection along with the objective to reduce the loss while the network is learning the details.

As part of this project, we will be demonstrating the outcome with 4 different model architecture along with their outcome in each of the model and provide the commentary for each of the models developed.

3. Project Objective

The objective of this capstone project is to build a Pneumonia detection system to locate the position of the inflammation in Chest Radiography.

Based on the dataset provided, we will have to take the following steps to work towards building the final model and validate and results.

- Validate the images and respective bounding box coordinates.
- Extract all the features from the images and build the csv file for processing further.
- Perform Exploratory Data Analysis to validate all the data points and build insights.
- Based on the project objective – isolate the dataset and accordingly images as well which are fully unique by removing the duplicate records in the dataset based on target variable.
- Build the model with different architectures and showcase the accuracy and showcase the right model to approach the problem description.

4. Summary of problem statement, data, and findings

A. Problem Statement

In this capstone project, the goal is to build a pneumonia detection system, to locate the position of inflammation in an image. Tissues with sparse material, such as lungs which are full of air, do not absorb the X-rays and appear black in the image. Dense tissues such as bones absorb X-rays and appear white in the image. While we are theoretically detecting “lung opacities”, there are lung opacities that are not pneumonia related. In the data, some of these are labeled “Not Normal No Lung Opacity”. 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.

Dicom original images: Medical images are stored in a special format called DICOM files (*.dcm). They contain a combination of header metadata as well as underlying raw image arrays for pixel data.

Here’s the backstory and why solving this problem matters.

Pneumonia accounts for over 15% of all deaths of children under 5 years old internationally. In 2015, 920,000 children under the age of 5 died from the disease. 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.

While common, accurately diagnosing pneumonia is a tall order. 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.**

To improve the efficiency and reach of diagnostic services, the Radiological Society of North America (RSNA®) has reached out to Kaggle's machine learning community and collaborated with the US National Institutes of Health, The Society of Thoracic Radiology, and MD.ai to develop a rich dataset for this challenge. (Description Source)

Data Source: https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/data

B. Data set

In addition to zip files which contain the DICOM images for training and testing there are three csv files. Which contain the following information.

- **stage_2_train.csv** - the training set. Contains "patientId's" and bounding box / target information.
- **stage_2_sample_submission.csv** - a sample submission file in the correct format. Contains "patientId's" for the test set. Note that the sample submission contains one box per image, but there is no limit to the number of bounding boxes that can be assigned to a given image.
- **stage_2_detailed_class_info.**csv - provides detailed information about the type of positive or negative class for each image.
- **stage_2_train_images.**zip - directory which contains training DICOM image set.
- **stage_2_test_images.zip** - directory which contains test DICOM image set.

C. Initial Findings

- The entire dataset contains 22.53% samples as Pneumonia positive rows while 77.47% contains Pneumonia negative rows.
- Out of the 77.47% of people, there is a chunk of people who may not have Pneumonia but still they don't have normal lungs.
- In the dataset where patients have Pneumonia, the majority of the entries with either 2 or 1 bounding box of the affected area.
- Initial problem is to train a binary image classifier to classify the Xray images to pneumonia positive and negative. CNN are proven models which are best suited for image classification.

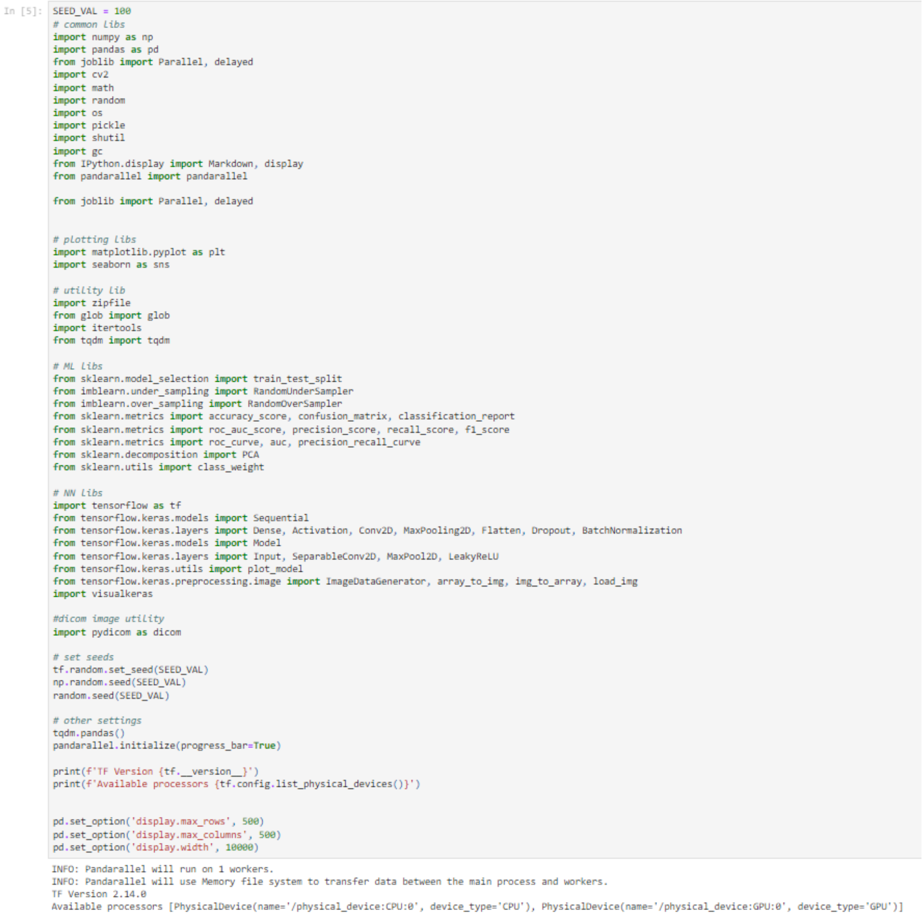
D. Observations from the CSV

Based on analysis, some of the observations:

- Training data is having a set of **patientId's** 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 evidence of Pneumonia are either of Normal or No Lung Opacity / Not Normal class.
- About 23,286 **patientId's** (~87% of them) provided have 1 bounding box while 13 patients have 4 bounding boxes!!!!

E. Import the Data

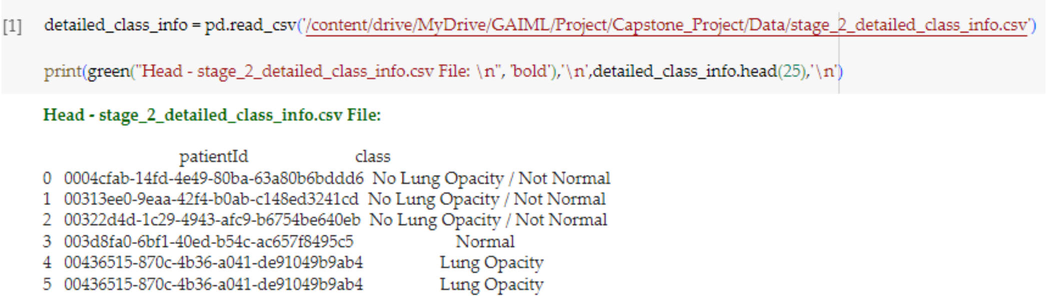
- Import the required libraries.



```
In [5]: SEED_VAL = 100
        # common Libs
        import numpy as np
        import pandas as pd
        from joblib import Parallel, delayed
        import cv2
        import math
        import random
        import os
        import pickle
        import shutil
        import gc
        from IPython.display import Markdown, display
        from pandarallel import pandarallel

        from joblib import Parallel, delayed


        # plotting Libs
        import matplotlib.pyplot as plt
        import seaborn as sns

        # utility Lib
        import zipfile
        from glob import glob
        import itertools
        from tqdm import tqdm

        # ML Libs
        from sklearn.model_selection import train_test_split
        from imblearn.under_sampling import RandomUnderSampler
        from imblearn.over_sampling import RandomOverSampler
        from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
        from sklearn.metrics import roc_auc_score, precision_score, recall_score, f1_score
        from sklearn.metrics import roc_curve, auc, precision_recall_curve
        from sklearn.decomposition import PCA
        from sklearn.utils import class_weight

        # NN Libs
        import tensorflow as tf
        from tensorflow.keras.models import Sequential
        from tensorflow.keras.layers import Dense, Activation, Conv2D, MaxPooling2D, Flatten, Dropout, BatchNormalization
        from tensorflow.keras.models import Model
        from tensorflow.keras.layers import Input, SeparableConv2D, MaxPool2D, LeakyReLU
        from tensorflow.keras.utils import plot_model
        from tensorflow.keras.preprocessing.image import ImageDataGenerator, array_to_img, img_to_array, load_img
        import visualkeras

        #dicom image utility
        import pydicom as dicom

        # set seeds
        tf.random.set_seed(SEED_VAL)
        np.random.seed(SEED_VAL)
        random.seed(SEED_VAL)

        # other settings
        tqdm.pandas()
        pandarallel.initialize(progress_bar=True)

        print(f'TF Version {tf.__version__}')
        print(f'Available processors {tf.config.list_physical_devices()}')


        pd.set_option('display.max_rows', 500)
        pd.set_option('display.max_columns', 500)
        pd.set_option('display.width', 10000)
```

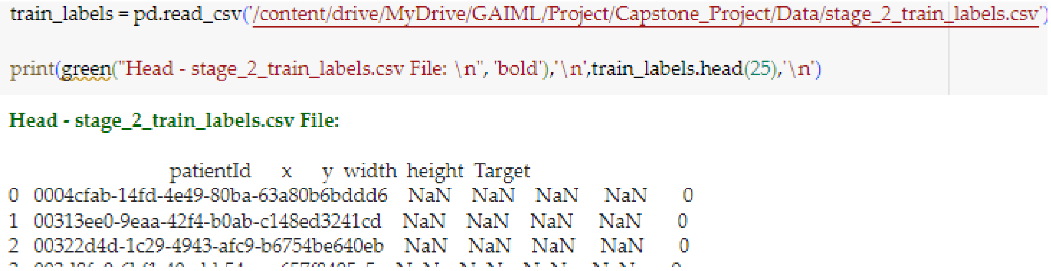
```
INFO: Pandarallel will run on 1 workers.
INFO: Pandarallel will use Memory file system to transfer data between the main process and workers.
TF Version 2.14.0
Available processors [PhysicalDevice(name='/physical_device:CPU:0', device_type='CPU'), PhysicalDevice(name='/physical_device:GPU:0', device_type='GPU')]
```

- Read the stage_2_detailed_class_info.csv & stage_2_train_labels.csv file and import the data using Pandas.



```
[1]  detailed_class_info = pd.read_csv('/content/drive/MyDrive/GAIML/Project/Capstone_Project/Data/stage_2_detailed_class_info.csv')

     print(green("Head - stage_2_detailed_class_info.csv File: \n", 'bold'),'\n',detailed_class_info.head(25),'\n')
```

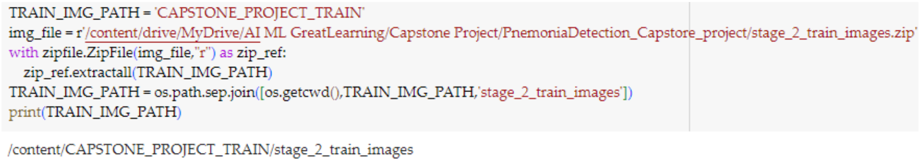
```
Head - stage_2_detailed_class_info.csv File:

                          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
5  00436515-870c-4b36-a041-de91049b9ab4                  Lung Opacity
```



```
train_labels = pd.read_csv('/content/drive/MyDrive/GAIML/Project/Capstone_Project/Data/stage_2_train_labels.csv')

print(green("Head - stage_2_train_labels.csv File: \n", 'bold'),'\n',train_labels.head(25),'\n')
```

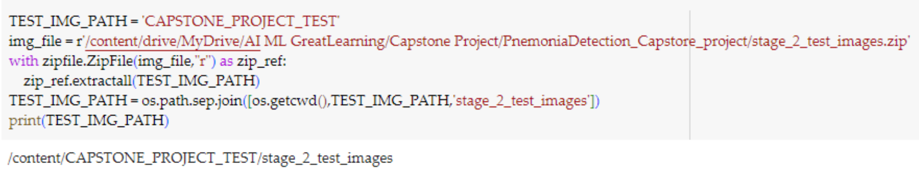
```
Head - stage_2_train_labels.csv File:

                          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
```

- Extract the stage_2_train_images.zip & stage_2_test_images.zip.



```
TRAIN_IMG_PATH = 'CAPSTONE_PROJECT_TRAIN'
img_file = r'/content/drive/MyDrive/AI ML GreatLearning/Capstone Project/PnemoniaDetection_Capstore_project/stage_2_train_images.zip'
with zipfile.ZipFile(img_file,"r") as zip_ref:
   zip_ref.extractall(TRAIN_IMG_PATH)
TRAIN_IMG_PATH = os.path.sep.join([os.getcwd(),TRAIN_IMG_PATH,'stage_2_train_images'])
print(TRAIN_IMG_PATH)
```

```
/content/CAPSTONE_PROJECT_TRAIN/stage_2_train_images
```



```
TEST_IMG_PATH = 'CAPSTONE_PROJECT_TEST'
img_file = r'/content/drive/MyDrive/AI ML GreatLearning/Capstone Project/PnemoniaDetection_Capstore_project/stage_2_test_images.zip'
with zipfile.ZipFile(img_file,"r") as zip_ref:
   zip_ref.extractall(TEST_IMG_PATH)
TEST_IMG_PATH = os.path.sep.join([os.getcwd(),TEST_IMG_PATH,'stage_2_test_images'])
print(TEST_IMG_PATH)
```

```
/content/CAPSTONE_PROJECT_TEST/stage_2_test_images
```

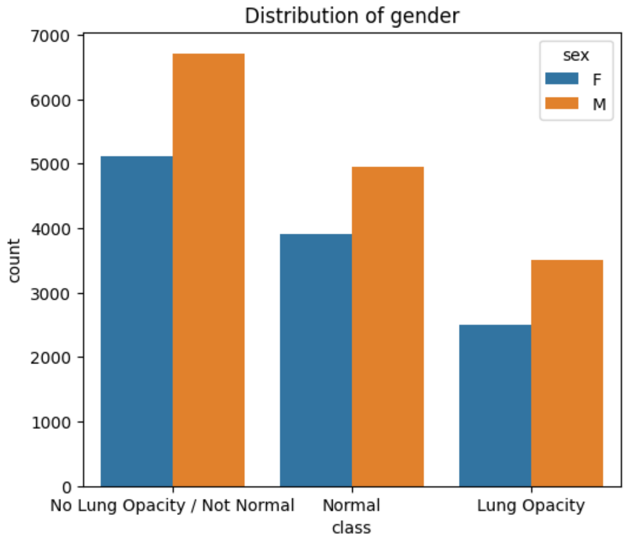
F. Data fields

- **patientId** - A patientId. Each patientId 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.
- **class** - One of 'Lung Opacity', 'Normal' or 'No Lung Opacity / Not Normal'

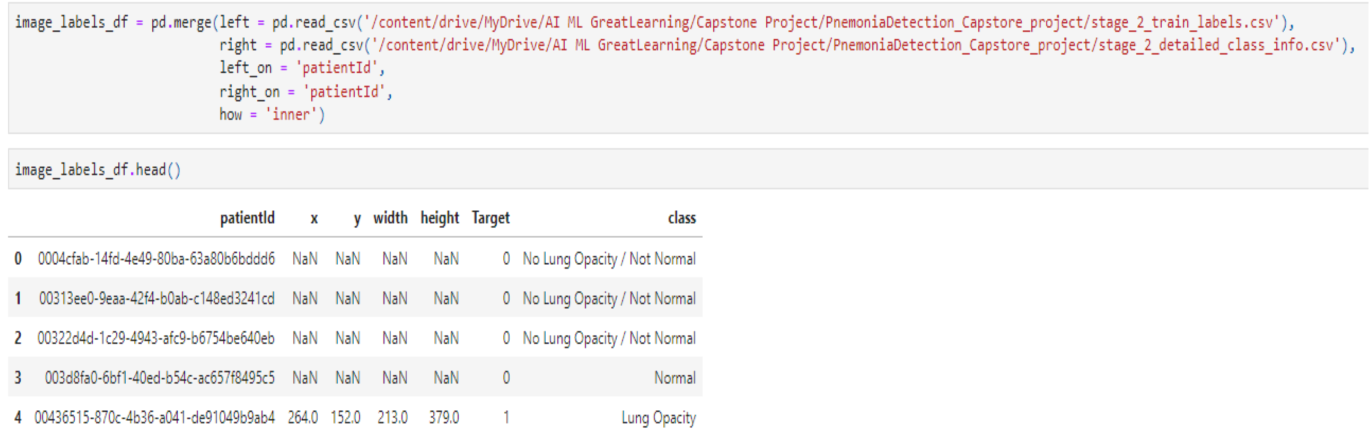
5. Overview of the final processing – EDA Inference & Data pre-processing

A. Exploratory data analysis

- Based on the sample dataset provided – we see that there are 3 class values presents. Our objective would be to look at images having Lung Opacity and identify the right bounding box to locate the area of inflammation, hence we may need to club them into Lung Opacity and others as another single category.



- Input data has 26684 training images and 3000 testing images. Post merging all the data from the two .csv files the summary table looks like below.

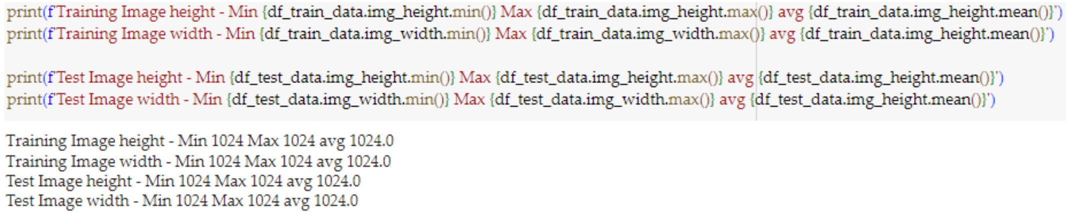


```
image_labels_df = pd.merge(left = pd.read_csv('/content/drive/MyDrive/AI ML GreatLearning/Capstone Project/PnemoniaDetection_Capstore_project/stage_2_train_labels.csv'),
                           right = pd.read_csv('/content/drive/MyDrive/AI ML GreatLearning/Capstone Project/PnemoniaDetection_Capstore_project/stage_2_detailed_class_info.csv'),
                           left_on = 'patientId',
                           right_on = 'patientId',
                           how = 'inner')
```

```
image_labels_df.head()
```

	patientId	x	y	width	height	Target	class
0	0004cfab-14fd-4e49-80ba-63a80b6bddd6	NaN	NaN	NaN	NaN	0	No Lung Opacity / Not Normal
1	00313ee0-9eaa-42f4-b0ab-c148ed3241cd	NaN	NaN	NaN	NaN	0	No Lung Opacity / Not Normal
2	00322d4d-1c29-4943-afc9-b6754be640eb	NaN	NaN	NaN	NaN	0	No Lung Opacity / Not Normal
3	003d8fa0-6bf1-40ed-b54c-ac657f8495c5	NaN	NaN	NaN	NaN	0	Normal
4	00436515-870c-4b36-a041-de91049b9ab4	264.0	152.0	213.0	379.0	1	Lung Opacity

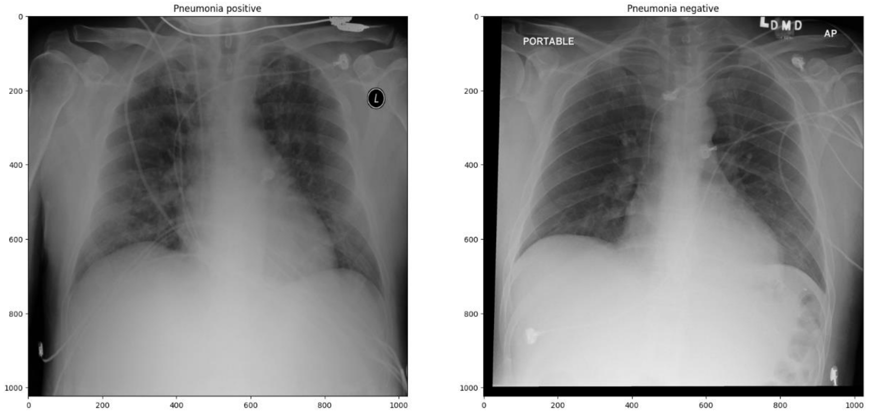
- Image sizes are the same across all images, and they are all 1024x1024 pixels. They are all grey images.



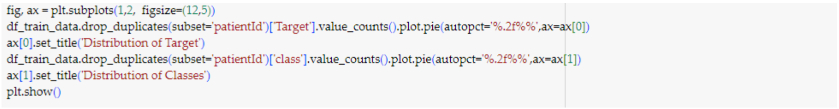
```
print(f'Training Image height - Min {df_train_data.img_height.min()} Max {df_train_data.img_height.max()} avg {df_train_data.img_height.mean()}')
print(f'Training Image width - Min {df_train_data.img_width.min()} Max {df_train_data.img_width.max()} avg {df_train_data.img_height.mean()}')

print(f'Test Image height - Min {df_test_data.img_height.min()} Max {df_test_data.img_height.max()} avg {df_test_data.img_height.mean()}')
print(f'Test Image width - Min {df_test_data.img_width.min()} Max {df_test_data.img_width.max()} avg {df_test_data.img_height.mean()}')
```

```
Training Image height - Min 1024 Max 1024 avg 1024.0
Training Image width - Min 1024 Max 1024 avg 1024.0
Test Image height - Min 1024 Max 1024 avg 1024.0
Test Image width - Min 1024 Max 1024 avg 1024.0
```

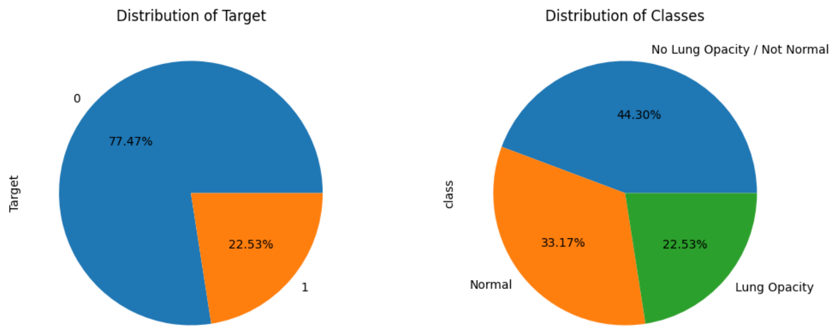
- From the above picture it can be seen that Pneumonia infected (or positive) lungs has more black/grey regions in comparison to not infected (or negative) lungs. We need our models to capture this difference in images.
- Images from the Pneumonia infected and normal lung x-ray.



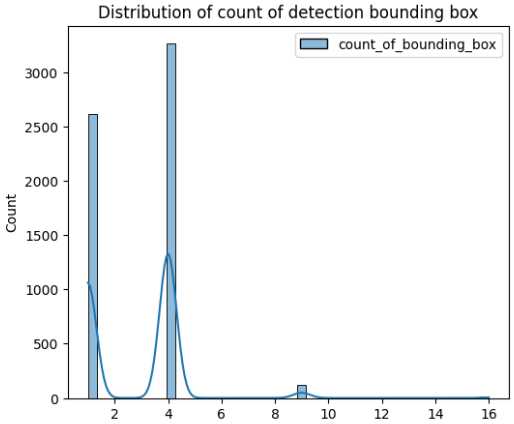
- Distribution of Target - the positive class is a minority class with only 22.5% samples coming from that set.



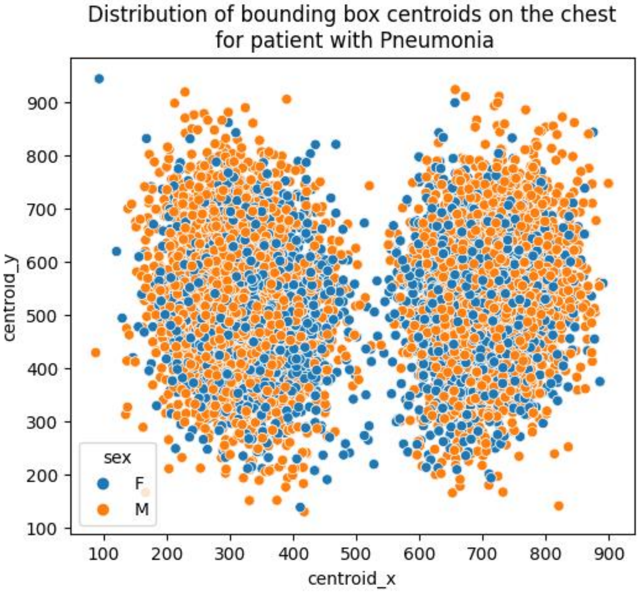
```
fig, ax = plt.subplots(1,2,  figsize=(12,5))
df_train_data.drop_duplicates(subset='patientId')['Target'].value_counts().plot.pie(autopct='%.2f%%',ax=ax[0])
ax[0].set_title('Distribution of Target')
df_train_data.drop_duplicates(subset='patientId')['class'].value_counts().plot.pie(autopct='%.2f%%',ax=ax[1])
ax[1].set_title('Distribution of Classes')
plt.show()
```



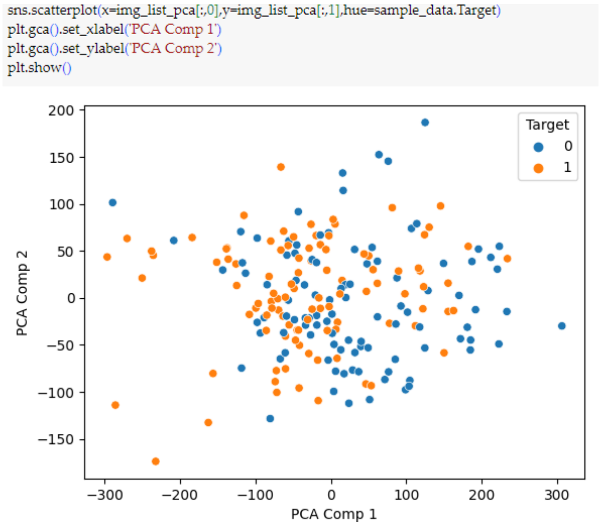
- Distribution of number of infection areas shows that there are 4 groups, chest Xray with 1, 4, 9 and 16 areas of infection. As seen from the plot below, most of the X-rays have 4 infection areas.



- Plot of centroid of the infecting detecting bounding boxes show the distribution of infection locations on the Xray. It can be seen that the infection is most evenly distributed across left and right lungs.



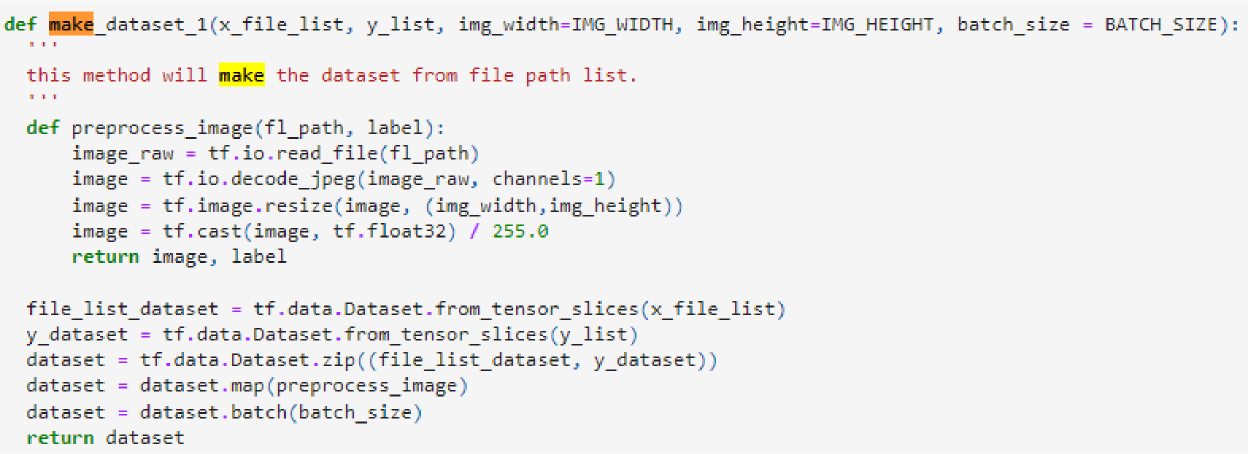
- Performing a principal component decomposition of pixel values of a sample image reveals that there is no significant feature within the pixels that can separate the two target values (0 for pneumonia -ve and 1 for pneumonia +ve). It's a very complex feature space and it's difficult to separate between two classes.



```
sns.scatterplot(x=img_list_pca[:,0],y=img_list_pca[:,1],hue=sample_data.Target)
plt.gca().set_xlabel('PCA Comp 1')
plt.gca().set_ylabel('PCA Comp 2')
plt.show()
```

B. Preprocessing of Data

- For the binary classification of data, we have decided to scale all the images to 256*256 size. We first extract the image data from DICOM images and store them as grey scale JPG files for easy processing. We also have designed a data generator which will pull data from a folder of JPG images, resize it, normalize the pixel values to be within 0 to 1 and then supply it to the CNN model.

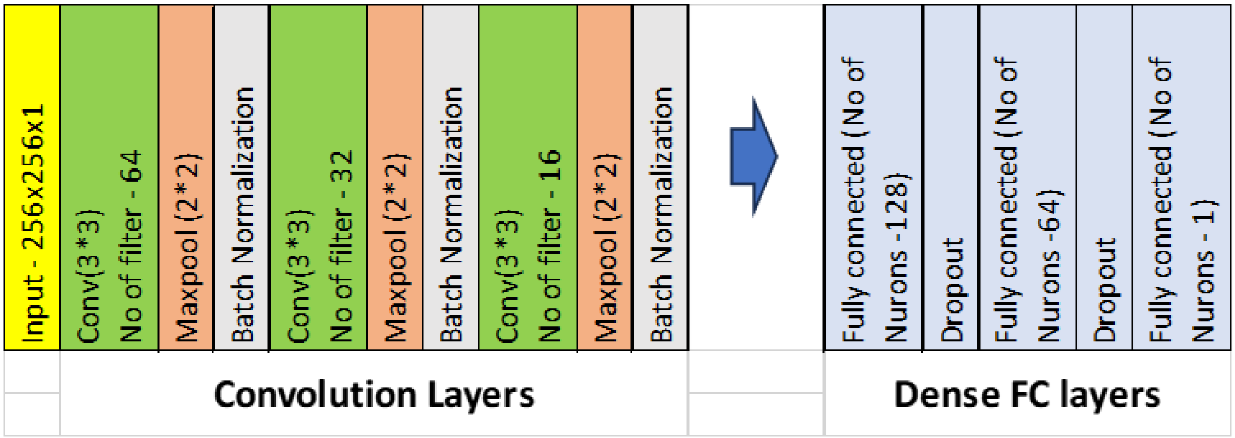


```
def make_dataset_1(x_file_list, y_list, img_width=IMG_WIDTH, img_height=IMG_HEIGHT, batch_size = BATCH_SIZE):
  '''
  this method will make the dataset from file path list.
  '''
  def preprocess_image(fl_path, label):
      image_raw = tf.io.read_file(fl_path)
      image = tf.io.decode_jpeg(image_raw, channels=1)
      image = tf.image.resize(image, (img_width,img_height))
      image = tf.cast(image, tf.float32) / 255.0
      return image, label

  file_list_dataset = tf.data.Dataset.from_tensor_slices(x_file_list)
  y_dataset = tf.data.Dataset.from_tensor_slices(y_list)
  dataset = tf.data.Dataset.zip((file_list_dataset, y_dataset))
  dataset = dataset.map(preprocess_image)
  dataset = dataset.batch(batch_size)
  return dataset
```

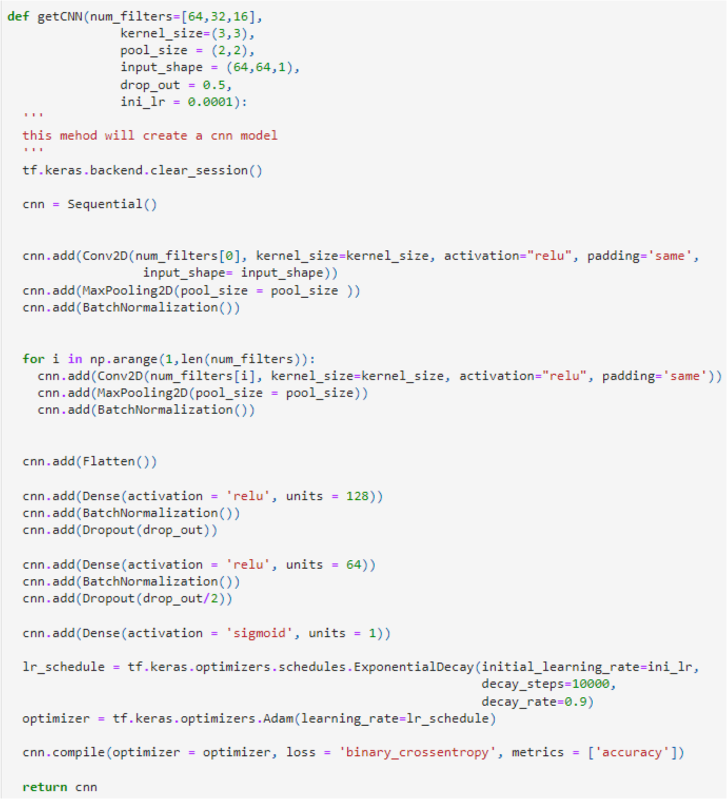
6.Step-by-step walk through the solution

Step 1. Models and Model Building

Since this is an Image classification task, the best model to use is CNN. We have designed a simple 3-layer CNN architecture as shown below.



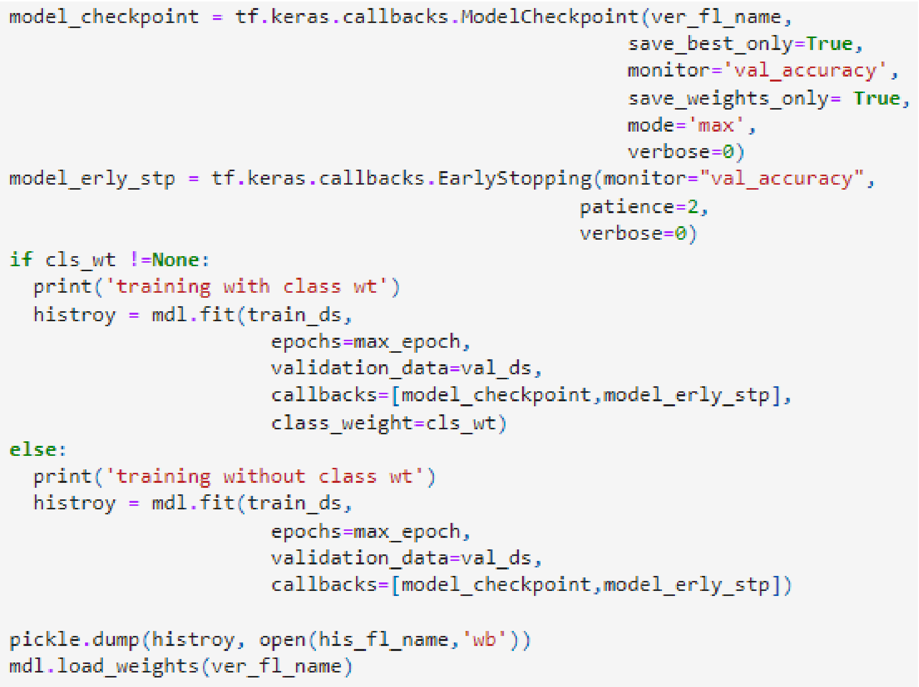
We have designed a flexible method (as seen below) which can take various input parameters to generate various architecture of CNN.



```
def getCNN(num_filters=[64,32,16],
               kernel_size=(3,3),
               pool_size = (2,2),
               input_shape = (64,64,1),
               drop_out = 0.5,
               ini_lr = 0.0001):
  '''
  this mehod will create a cnn model
  '''
  tf.keras.backend.clear_session()

  cnn = Sequential()


  cnn.add(Conv2D(num_filters[0], kernel_size=kernel_size, activation="relu", padding='same',
                input_shape= input_shape))
  cnn.add(MaxPooling2D(pool_size = pool_size ))
  cnn.add(BatchNormalization())


  for i in np.arange(1,len(num_filters)):
    cnn.add(Conv2D(num_filters[i], kernel_size=kernel_size, activation="relu", padding='same'))
    cnn.add(MaxPooling2D(pool_size = pool_size))
    cnn.add(BatchNormalization())


  cnn.add(Flatten())

  cnn.add(Dense(activation = 'relu', units = 128))
  cnn.add(BatchNormalization())
  cnn.add(Dropout(drop_out))

  cnn.add(Dense(activation = 'relu', units = 64))
  cnn.add(BatchNormalization())
  cnn.add(Dropout(drop_out/2))

  cnn.add(Dense(activation = 'sigmoid', units = 1))

  lr_schedule = tf.keras.optimizers.schedules.ExponentialDecay(initial_learning_rate=ini_lr,
                                                              decay_steps=10000,
                                                              decay_rate=0.9)
  optimizer = tf.keras.optimizers.Adam(learning_rate=lr_schedule)

  cnn.compile(optimizer = optimizer, loss = 'binary_crossentropy', metrics = ['accuracy'])

  return cnn
```

For all these studies the batch size of 32 is used and the learning rate of 1e-3. The model uses an Adam optimizer and uses Exponential Decay of the learning rate. A max of 5 epoch was set as this is an evaluation phase.

There are also two callbacks associated with monitoring the training activity (as seen below). One callback is defined to store the best models based on validation accuracy on hard disk. The other callback continuously monitors the validation accuracy and if the validation accuracy decreases 2 times the training is stopped.



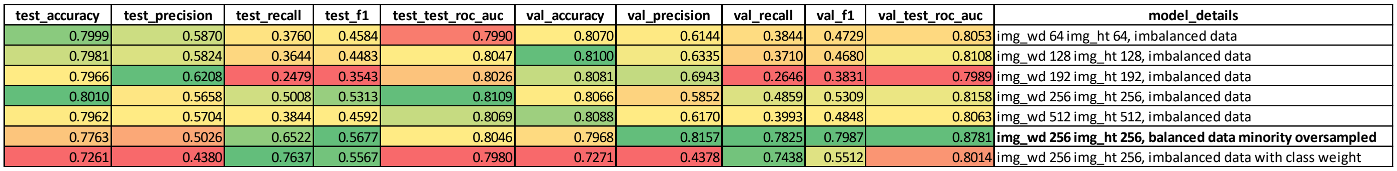
```
model_checkpoint = tf.keras.callbacks.ModelCheckpoint(ver_fl_name,
                                                      save_best_only=True,
                                                      monitor='val_accuracy',
                                                      save_weights_only= True,
                                                      mode='max',
                                                      verbose=0)
model_erly_stp = tf.keras.callbacks.EarlyStopping(monitor="val_accuracy",
                                                  patience=2,
                                                  verbose=0)
if cls_wt !=None:
  print('training with class wt')
  histroy = mdl.fit(train_ds,
                    epochs=max_epoch,
                    validation_data=val_ds,
                    callbacks=[model_checkpoint,model_erly_stp],
                    class_weight=cls_wt)
else:
  print('training without class wt')
  histroy = mdl.fit(train_ds,
                    epochs=max_epoch,
                    validation_data=val_ds,
                    callbacks=[model_checkpoint,model_erly_stp])

pickle.dump(histroy, open(his_fl_name,'wb'))
mdl.load_weights(ver_fl_name)
```

Training data is split into 80% training, 10% validation and 10% test. Since this is medical application, we have chosen F1 score and Recall as the preferred measure to evaluate models.

Model is trained in the following step.

- We first determine what size of the image will give the best performance measure.
- Since the is class imbalance, we try to adjust for imbalance by
  1. Minority oversampling.
  2. Training with class weight.

Below table shows a comparative study of performance of various models on test and validation set.



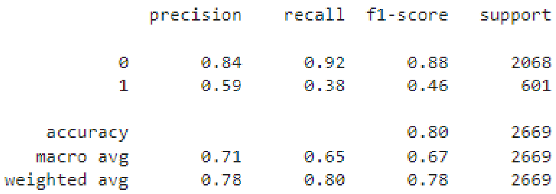
test_accuracy	test_precision	test_recall	test_f1	test_test_roc_auc	val_accuracy	val_precision	val_recall	val_f1	val_test_roc_auc	model_details
0.7999	0.5870	0.3760	0.4584	0.7990	0.8070	0.6144	0.3844	0.4729	0.8053	img_wd 64 img_ht 64, imbalanced data
0.7981	0.5824	0.3644	0.4483	0.8047	0.8100	0.6335	0.3710	0.4680	0.8108	img_wd 128 img_ht 128, imbalanced data
0.7966	0.6208	0.2479	0.3543	0.8026	0.8081	0.6943	0.2646	0.3831	0.7989	img_wd 192 img_ht 192, imbalanced data
0.8010	0.5658	0.5008	0.5313	0.8109	0.8066	0.5852	0.4859	0.5309	0.8158	img_wd 256 img_ht 256, imbalanced data
0.7962	0.5704	0.3844	0.4592	0.8069	0.8088	0.6170	0.3993	0.4848	0.8063	img_wd 512 img_ht 512, imbalanced data
0.7763	0.5026	0.6522	0.5677	0.8046	0.7968	0.8157	0.7825	0.7987	0.8781	**img_wd 256 img_ht 256, balanced data minority oversampled**
0.7261	0.4380	0.7637	0.5567	0.7980	0.7271	0.4378	0.7438	0.5512	0.8014	img_wd 256 img_ht 256, imbalanced data with class weight

Observation

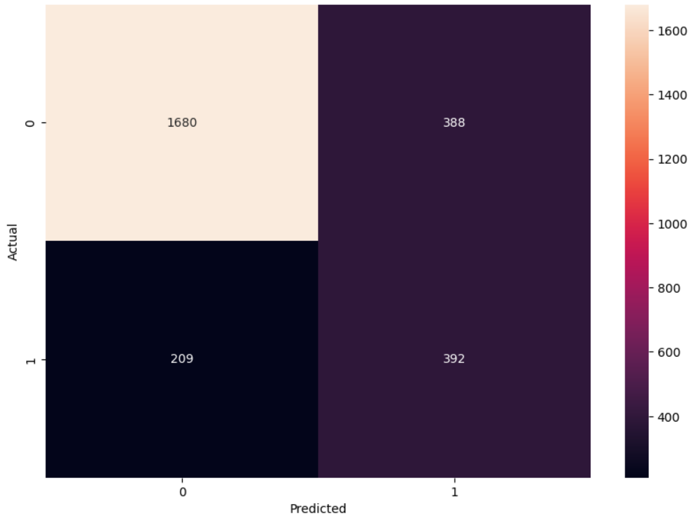
From the above it can be seen that the ‘minority over sampled' model is overall a better model with a better test f1 and recall score in comparison to other models.

class weight-based models perform better for the minority class but have significant reduction in accuracy.

Below are pictures which show the confusion matrix and classification report on the test data for the best model.

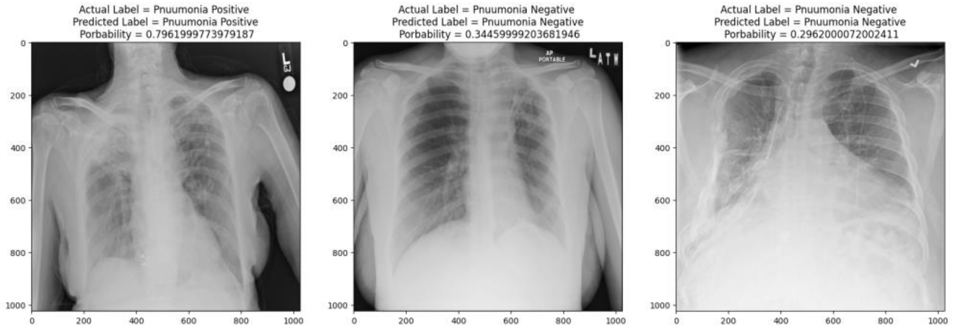


```
              precision    recall  f1-score   support

           0       0.84      0.92      0.88      2068
           1       0.59      0.38      0.46       601

    accuracy                           0.80      2669
   macro avg       0.71      0.65      0.67      2669
weighted avg       0.78      0.80      0.78      2669
```



From the above it can be seen that model does much better in majority class (Pneumonia -ve) not so good for the minority class (Pneumonia +ve).

Below are some random predictions on test data using the best model ('minority over sampled').


Actual Label = Pnuumonia Positive
Predicted Label = Pnuumonia Positive
Porbability = 0.7961999773979187
Actual Label = Pnuumonia Negative
Predicted Label = Pnuumonia Negative
Porbability = 0.34459999203681946
Actual Label = Pnuumonia Negative
Predicted Label = Pnuumonia Negative
Porbability = 0.2962000072002411
0
200
400
600
800
1000
AP PORTABLE

Step 2 : Improving model performance.

We deploy 3 strategies to improve model performance.

1. Data augmentation – random transformation was applied to provided images to generate more images from the minority class that was used to train the binary classifier.
2. Parameter optimization – CNN has numerous parameters; we select the most important 2 number of filters and batch size and run a small DOE to select the best parameter.
3. Probability threshold optimization – by adjusting probability at which model predicts a positive class we try to fine tune the model.

Step 3: Using transfer learning model

We use 6 transfer learning model and attempt to improve the performance of the model

1. VGG16 Model

2. VGG19 Model

3. MobilenetV2 Model

4. Inception v3 Model

5. ResNet Model

6. Efficient Net

Step 4: RCNN model

We try two flavors of RCNN

1. Region proposal and then CNN to classify the region

2. RCNN - using UNET masking

Region proposal and then CNN to classify the region

This will be done the following steps

1. will use Selective search process to generate most probable image regions.

2. then calculate IOU w.r.t the ground truth bounding boxes. Select only those proposed bounding boxes which have IOU >0.3 as positive and rest as negative.

3. will then crop the image regions and resize them to 224 * 224 size and store them as positive or negative samples.

4. since this will generate a greater number of negative samples then positive samples, we will store only 30 negative sample and all positive samples.

5. from the samples will then build a balanced dataset and train a CNN to binary classify the regions

6. while predicting will use the same process to first generate regions of interest and then classify them into positive and negative.

We deploy this strategy during prediction:

1. Predict if the image is pneumonia positive using the best binary classifier model

2. if positive use the above RCCN model to predict the region with the highest probability

Mask RCNN using UNET

We will generate Mask RCNN using the following steps

1. Generate ground truth mask.

2. Develop a UNET architecture – using pretrained encoder of mobile net, decoder of mobile net and followed by a fully connected NN.

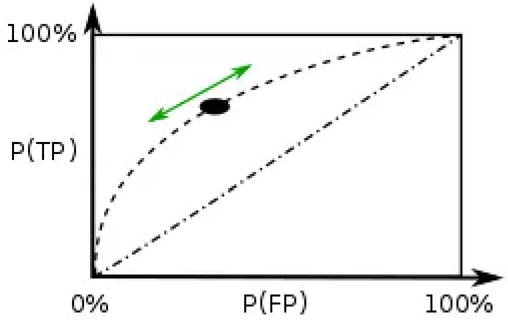
3. Will use this UNET to build classifier will a loss function of DICE loss.

7.Model evaluation

Binary Classification:

Objective is to classify chest Xray images into pneumonia positive and negative. Since this is a medical application, we think the following metric would be important a medical application.

- Recall = True positive / (True positive + False Negative). In medical application recall is a better metric because if recall for positive class improves false negative values will decrease. That is better in the case of medical application as reduction in false negative is important so the rate of true positive cases are surely identified.
- ROC AUC: A Receiver Operating Characteristic curve or ROC curve is created by plotting the True Positive (TP) against the False Positive (FP) at various threshold settings. The ROC curve is generated by plotting the cumulative distribution function of the True Positive in the y-axis versus the cumulative distribution function of the False Positive on the x-axis. The area under the ROC curve (ROC AUC) is the single-valued metric used for evaluating the performance.



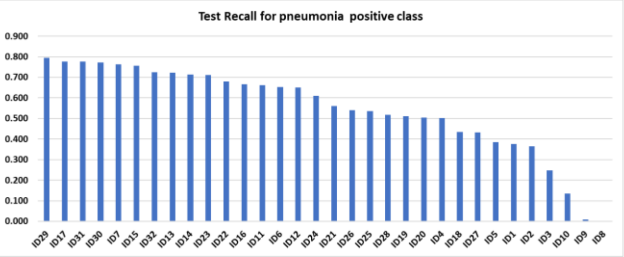
The dashed curved line is the ROC Curve

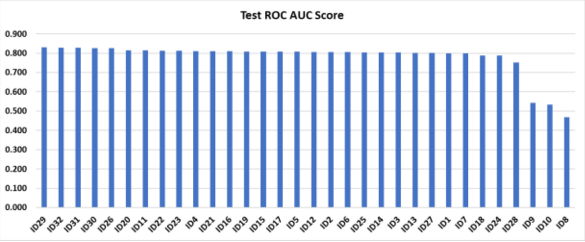
Since our dataset it highly imbalanced so ROC AUC is better matric as it shows models ability to distinguish between positive and negative classes.

We will primarily use these two parameters to assess model performance of various models that we have tried.

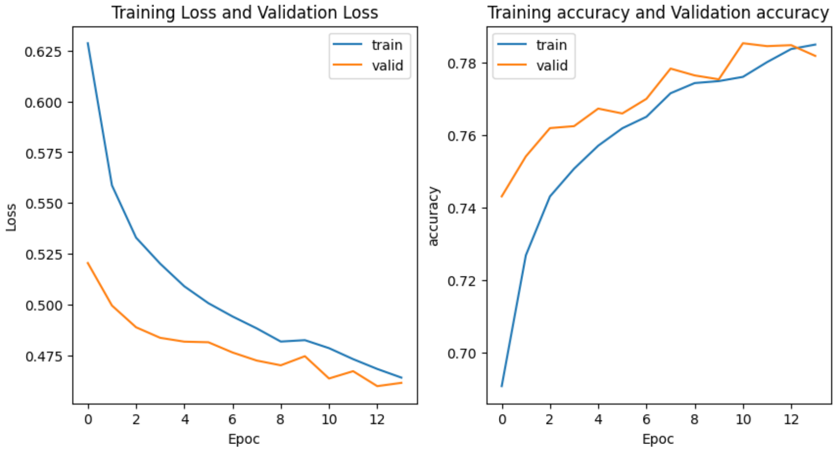
We have 33 different combinations of models. We started with our own custom designed CNN and then fine-tuned them. Then we also tried to parameter optimize them and improve their performance. Finally, we attempted transfer learning using some industry best practice models.

- Below chart shows the comparative results of all models for Recall and ROC AUC scores.

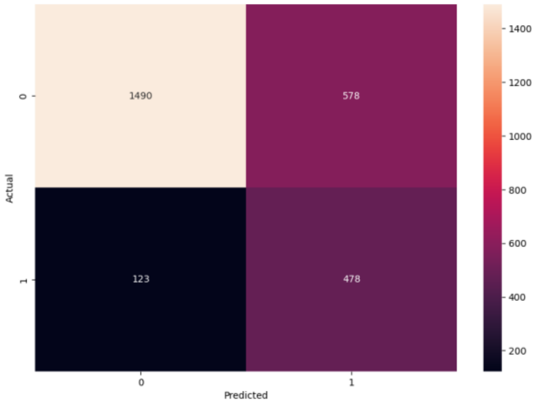




- From the above charts it can be seen that Model number 29 is the best model with both the criterion. It has a recall score of 0.79 and ROC AUC of 0.83. Below are training and validation loss curves.



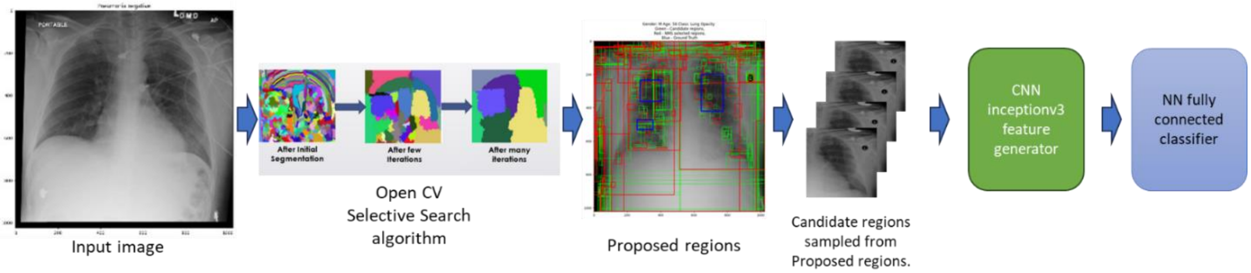
- As can be seen that training and validation scores are near to each other, so it can be concluded that model is not overfitting. Below chart shows the confusion matric on the test set.



RCNN Model

The objective of this model is to identify the regions of pneumonia infection within the Xray image of the patient. We developed two solutions first a **region proposal** based on image gradients and then using CNN to classify the region and second a **mask RCNN - using UNET**.

Region proposal network is a two-stage network we use selective search-based algorithm to propose regions and then apply a transfer learning CNN to generate features and then deploying a fully connected NN as the classifier. This architecture is shown below.

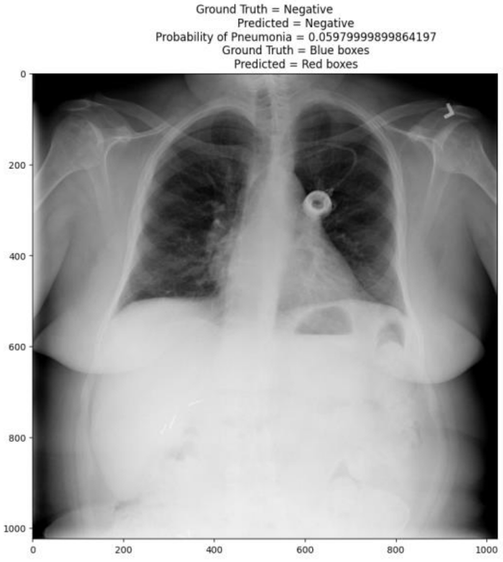

Input image
PORTABLE
AP
After Initial Segmentation
After few iterations
After many iterations
Open CV Selective Search algorithm
Proposed regions
Candidate regions sampled from Proposed regions.
CNN inceptionv3 feature generator
NN fully connected classifier

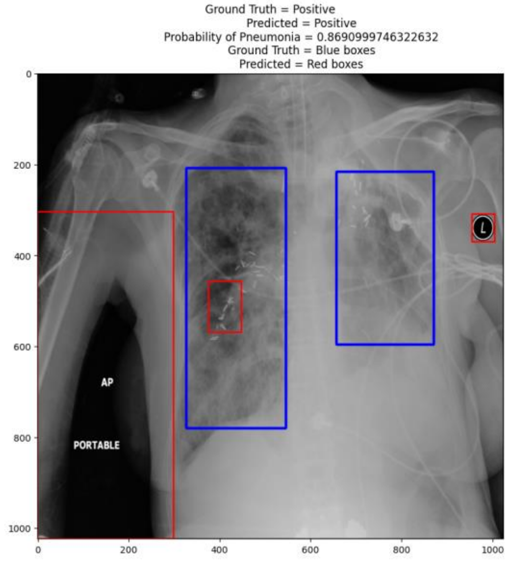
When predicting using this model we use a two stage

1. Predict if the image is pneumonia positive using the best model developed in step 2 of this milestone above.

2. if positive use the above RCCN model to predict the region with the highest probability

Below is the output from the two sample test cases.

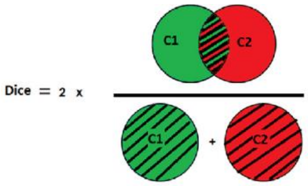




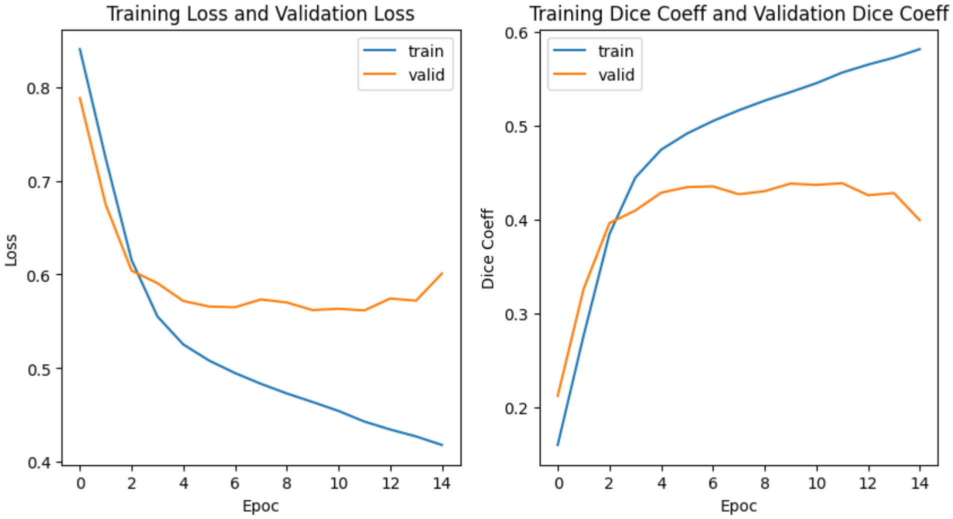
Mask RCNN - using UNET:

This is an image segmentation-based network developed using UNET architecture which leverages Mobile net v2 as the pre trained encoder and decoder network would be trained along with a fully connected layer. These models are evaluated based on DICE coefficient. Definition of DICE coefficient as shown below.

Below picture shows training and validation DICE coefficient



Below picture shows training and validation curve. It shows that there is some overfitting as shown gap between the gap between the training and validation curve.



The test average dice coefficient we were able to achieve was 0.1146. which is quite less in comparison to one achieved by the Kaggle competition winners of 0.25475.

8. Comparison to benchmark

Our benchmark for this object detection model is to classify the image correctly as pneumonia positive and negative and identify the region of infection.

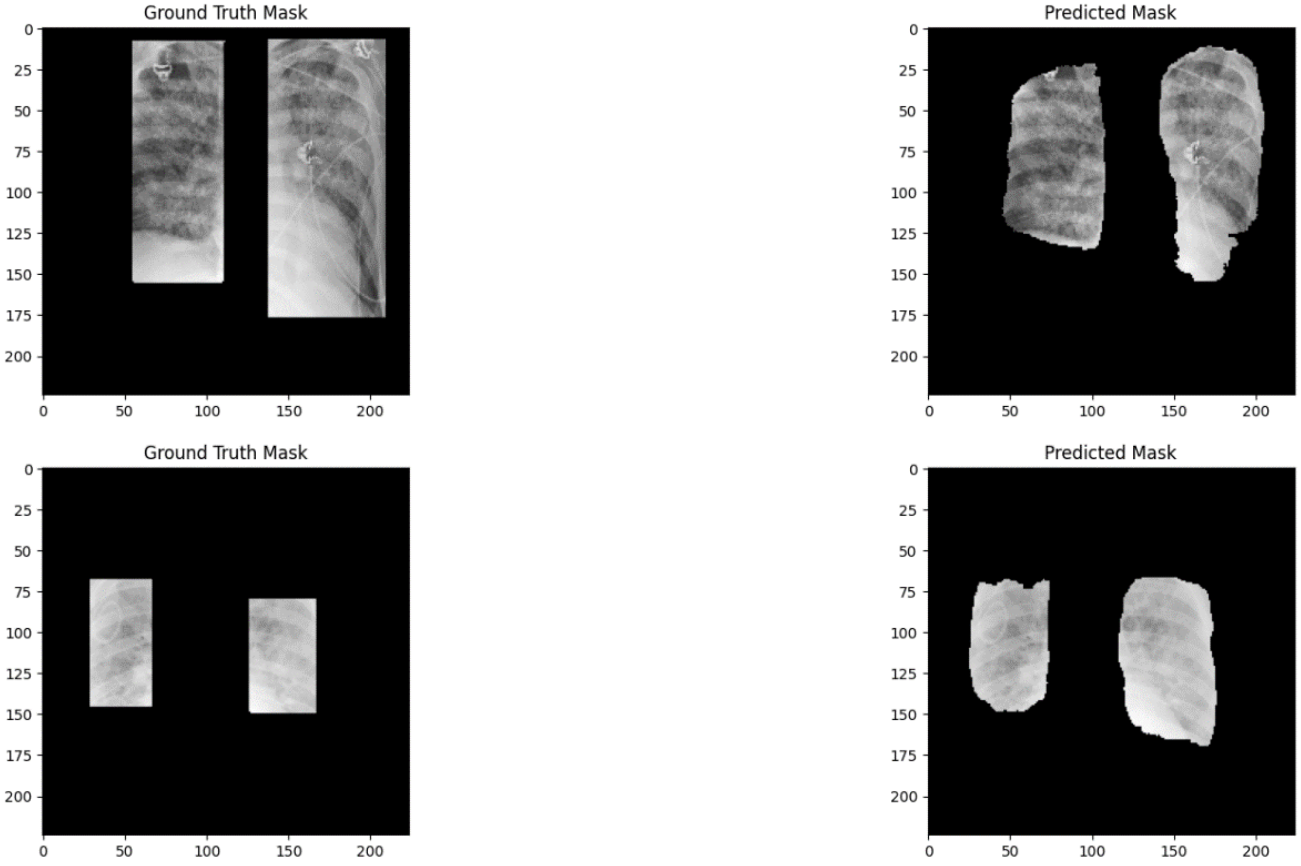
There are several competitions on Kaggle for the object detection where the objects are detected, localized, and labelled for very generic categories. For the Kaggle competition on this topic winners achieved a dice coefficient of 0.25475.

With the given time and availability of resources we could train our model to achieve above 83% ROC AUC for binary classifier and an average 0.1146 DICE coefficient. That means, we are able to detect the pneumonia positive X ray images more accurately then detect the region of infection.

Visualization of Model Predictions

Below are some predictions from this model.

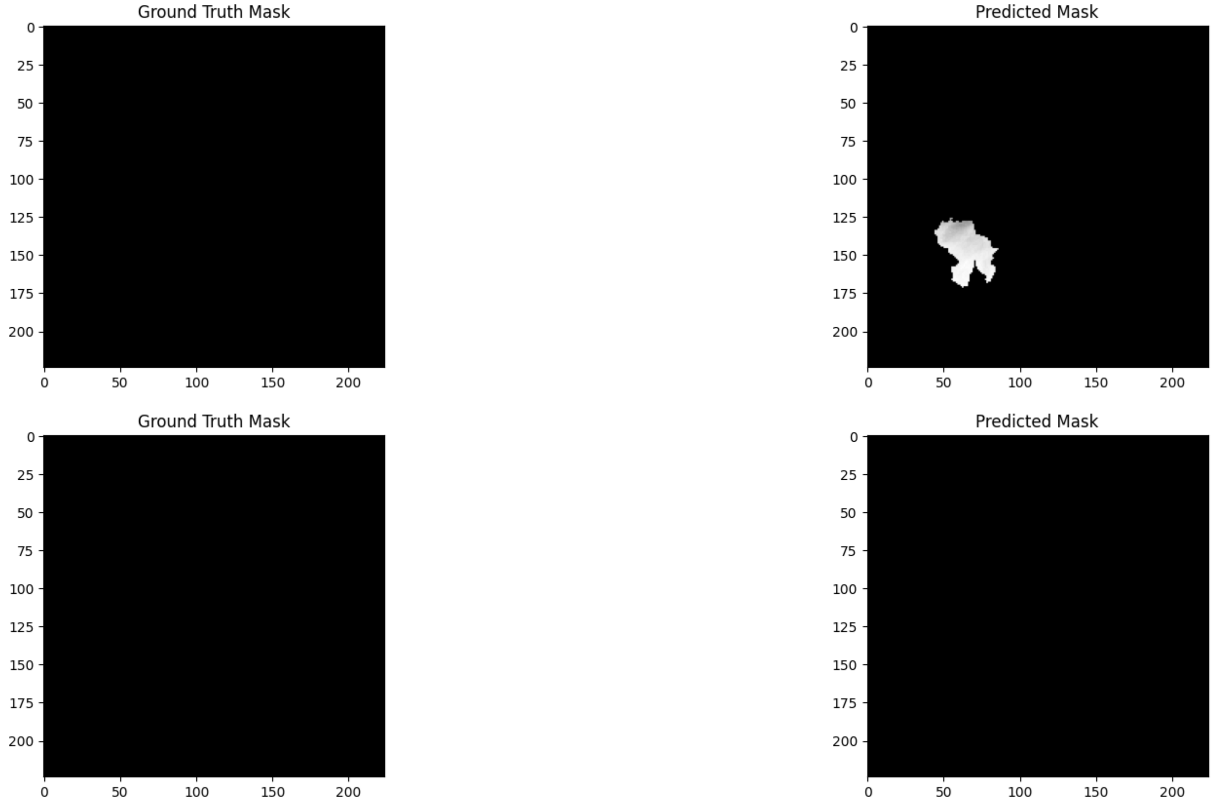
Positive case prediction.



Negative case prediction

As seen from above the model seems to perform well to predict regions where infections are present.

From the above it can be seen that in -ve cases, the model works reasonably well and does not detect any region of infection and in one case detecting a small region of infection. Since this is a medical application, we think it a good model as it is conservative and its predicting in places where there may be little chance of occurrence of pneumonia, which can be corrected by subsequent doctor review.



9.Implications

The impact of a solution in the domain or business, in the context of a Pneumonia Detection Capstone Project, can be significant, particularly in the healthcare domain. Here are some potential effects and recommendations:

Improved Accuracy and Efficiency in Diagnosis:

A well-trained Pneumonia Detection model can assist healthcare professionals, particularly radiologists, in making more accurate and timely diagnoses. It can help reduce the chances of misdiagnoses or overlooking pneumonia cases, potentially saving lives.

Faster Turnaround Time:

Automation of the initial screening process using AI can significantly speed up the evaluation of medical images. This can lead to faster decision-making and treatment initiation, which is critical in pneumonia cases.

Enhanced Resource Allocation:

With an AI model handling the initial screening, radiologists can focus their expertise on more complex cases and other important tasks, optimizing resource allocation and improving overall healthcare efficiency.

Recommendations:

I recommend integrating the Pneumonia Detection model into the clinical workflow, but as an initial screening tool rather than a replacement for human expertise. This recommendation aligns with ethical considerations and ensures that the model's predictions are always validated by a medical professional.

Continuous Monitoring and Improvement:

Continuously monitor the model's performance and gather feedback from healthcare professionals to improve its accuracy and reliability. Regularly update the model as more data becomes available and as technology advances.

Ethical Considerations:

Ensure that patient data privacy and ethical guidelines are strictly followed. Maintain a strong commitment to patient consent and data security throughout the implementation of the solution.

Confidence Level:

The confidence level in the recommendations depends on the robustness of the model's performance, the quality of data used for training and testing, and the validation through real-world testing. If the model has demonstrated high accuracy and reliability, the confidence in its recommendations can be relatively high. However, it's essential to acknowledge that no model is infallible, and its recommendations should always be considered as a supportive tool rather than a definitive diagnosis.

Regulatory Compliance:

Ensure that the solution complies with relevant healthcare regulations, which can vary by region. This includes compliance with data protection laws and medical device regulations if applicable.

Cost-Benefit Analysis:

Evaluate the cost-effectiveness of implementing the solution. Consider the initial investment in technology and ongoing operational costs versus the potential benefits in terms of time savings, improved accuracy, and patient outcomes.

Stakeholder Involvement:

Engage healthcare professionals, administrators, and IT experts in the implementation process. Their expertise and perspectives are crucial in making informed decisions. In summary, the Pneumonia Detection solution can positively impact the healthcare domain by enhancing diagnostic accuracy, efficiency, and resource allocation. However, it is essential to implement it ethically, in collaboration with healthcare professionals, and with continuous monitoring for improvement. Confidence in the solution's recommendations should be relatively high if it has been rigorously tested and validated.

10. Limitations

The limitations of a Pneumonia Detection model, or any machine learning model in the real world, can be numerous and are crucial to understand in order to make informed decisions and improvements. Here are some common limitations and potential enhancements:

Limited Generalization:

Limitation: Pneumonia Detection models may struggle to generalize to diverse populations, age groups, or data from different healthcare institutions. They might not perform well in cases that differ significantly from the training data.

Enhancement: Collect a more diverse and representative dataset for training. Implement domain adaptation techniques to make the model more robust to variations in data sources.

Data Imbalance:

Limitation: Data may be imbalanced, meaning there are far more negative (non-pneumonia) cases than positive (pneumonia) cases. This can lead to bias and reduced performance.

Enhancement: Use data augmentation techniques to balance the dataset. Explore methods like oversampling or generating synthetic data to ensure the model learns from sufficient positive cases.

Lack of Interpretability:

Limitation: Deep learning models, especially CNNs, can lack interpretability, making it challenging to understand why a model makes a particular prediction.

Enhancement: Implement interpretability techniques like Grad-CAM to visualize which regions in an image influenced the model's decision. Consider using explainable AI (XAI) techniques for better transparency.

Uncertainty Estimation:

Limitation: Models often provide a binary prediction (pneumonia or not) without quantifying uncertainty. In real-world applications, it can be valuable to know the model's confidence in its prediction.

Enhancement: Implement uncertainty estimation techniques such as dropout, Bayesian neural networks, or Monte Carlo dropout to provide probabilistic predictions.

Ethical Concerns:

Limitation: There can be ethical concerns related to bias, fairness, and privacy. Models may not perform equally well for all demographic groups, and patient privacy should be a top priority.

Enhancement: Conduct thorough bias and fairness audits on your model. Implement fairness-aware learning techniques and ensure strict adherence to data privacy regulations.

Hardware and Infrastructure:

Limitation: Training and deploying deep learning models often require significant computational resources, which may not be readily available in all healthcare settings.

Enhancement: Explore cloud-based solutions or edge computing for more accessible deployment. Optimize model architectures for efficiency.

Continuous Monitoring:

Limitation: Models can degrade over time as data distributions change or as they encounter previously unseen scenarios.

Enhancement: Implement continuous monitoring and retraining of the model using updated data. Set up alert mechanisms for model performance deterioration.

Clinical Validation:

Limitation: Real-world clinical validation may be a lengthy and complex process, involving regulatory hurdles.

Enhancement: Collaborate with healthcare professionals to establish rigorous clinical validation protocols and work closely with regulatory authorities to expedite the approval process.

False Positives and Negatives:

Limitation: Models may produce false positives (incorrectly diagnosing pneumonia) or false negatives (missing actual cases).

Enhancement: Fine-tune the model to optimize sensitivity and specificity based on the clinical requirements. Post-process model predictions to reduce false positives or negatives.

Robustness to Image Quality:

Limitation: The model may not perform well on images with varying qualities or artifacts.

Enhancement: Augment the dataset with various image qualities and consider preprocessing techniques to enhance image quality.

In summary, addressing these limitations involves a combination of data management, model development, transparency, and a strong focus on ethical and practical considerations. Continuous improvement, validation, and collaboration with healthcare professionals and domain experts are essential to enhance the solution's real-world performance.

11. Closing Reflections

Here are some common lessons learned from project experiences and potential improvements for future projects:

Understand the Problem:

It's crucial to have a deep understanding of the problem you are trying to solve before diving into a project. This understanding should include the domain, stakeholders, and specific challenges.

Data Quality Matters:

Data quality is often more important than the complexity of the model. Poor-quality data can lead to suboptimal results. Collecting, cleaning, and preprocessing data should be a major focus.

Communication is Key:

Effective communication among team members, stakeholders, and end-users is essential. Ensure everyone is on the same page regarding project goals, progress, and expectations.

Scope Management:

Scope creep can lead to project delays and budget overruns. Clearly define project scope and objectives and be disciplined about managing changes to the scope.

Testing and Validation:

Rigorous testing and validation of models or solutions are critical. Ensure that the solution works as intended in real-world scenarios.

Ethical Considerations:

Understand the ethical implications of your project, especially in fields like healthcare. Respect privacy, ensure fairness, and prioritize the well-being of individuals involved.

What to Do Differently Next Time:

Detailed Planning:

Invest more time in project planning and defining objectives. Clearly outline the project's scope, timeline, and deliverables.

Team Collaboration:

Foster stronger collaboration among team members, assign responsibilities clearly, and encourage open communication throughout the project.

Documentation:

Maintain thorough documentation of project processes, decisions, and outcomes. This will make it easier to learn from past experiences and share knowledge with others.

Continuous Learning:

Stay up to date with the latest technologies and methodologies in your field. Apply new knowledge and best practices to future projects.

Feedback Integration:

Actively seek feedback from project stakeholders and end-users. Use this feedback to make improvements and adjustments in real-time.

Risk Management:

Be proactive in identifying and mitigating risks. Develop contingency plans to address unexpected challenges.