FDA Submission

Sidrah Madiha Siddiqui

Pneumonia Classifier from chest x-rays

Algorithm Description

1. General Information

Intended Use Statement: "Pneumonia Classifier from chest x-rays" is designed to assist in detecting presence of Pneumonia from chest x-rays. It is intended for professional use by radiologist.

Indications for Use: This algorithm is intended for use on patients from the ages of 25-65 who have been administered a chest x-ray study for screening of Pneumonia.

Device Limitations:

• The model was not permitted to use patient history, which has been shown to decrease radiologist diagnostic performance in interpreting chest radiographs.

• The device is not tested for the patients who are suspected of carrying more than one or two of the fourteen common thoracic pathologies include Atelectasis, Consolidation, Infiltration, Pneumothorax, Edema, Emphysema, Fibrosis, Effusion, Pneumonia, Pleural thickening, Cardiomegaly, Nodule, Mass and Hernia.
• The device may perform slowly without GPU.
• The algorithm is not tested for patients with previous history of Pneumonia.

Clinical Impact of Performance: The threshold was chosen to increase recall and allowing some margin for False Positives because the purpose of the device is to increase recall for assisting radiologist in detecting truly positive case in less amount of time (if this includes some number of False Positives that will not penalize the performance of classifier in this clinical context). The F1 score at this threshold was found to be ~0.67 which is better than average radiologist F1 score which is ~0.435 as stated in Research Paper: [CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays]

2. Algorithm Design and Function

Algorithm Design:

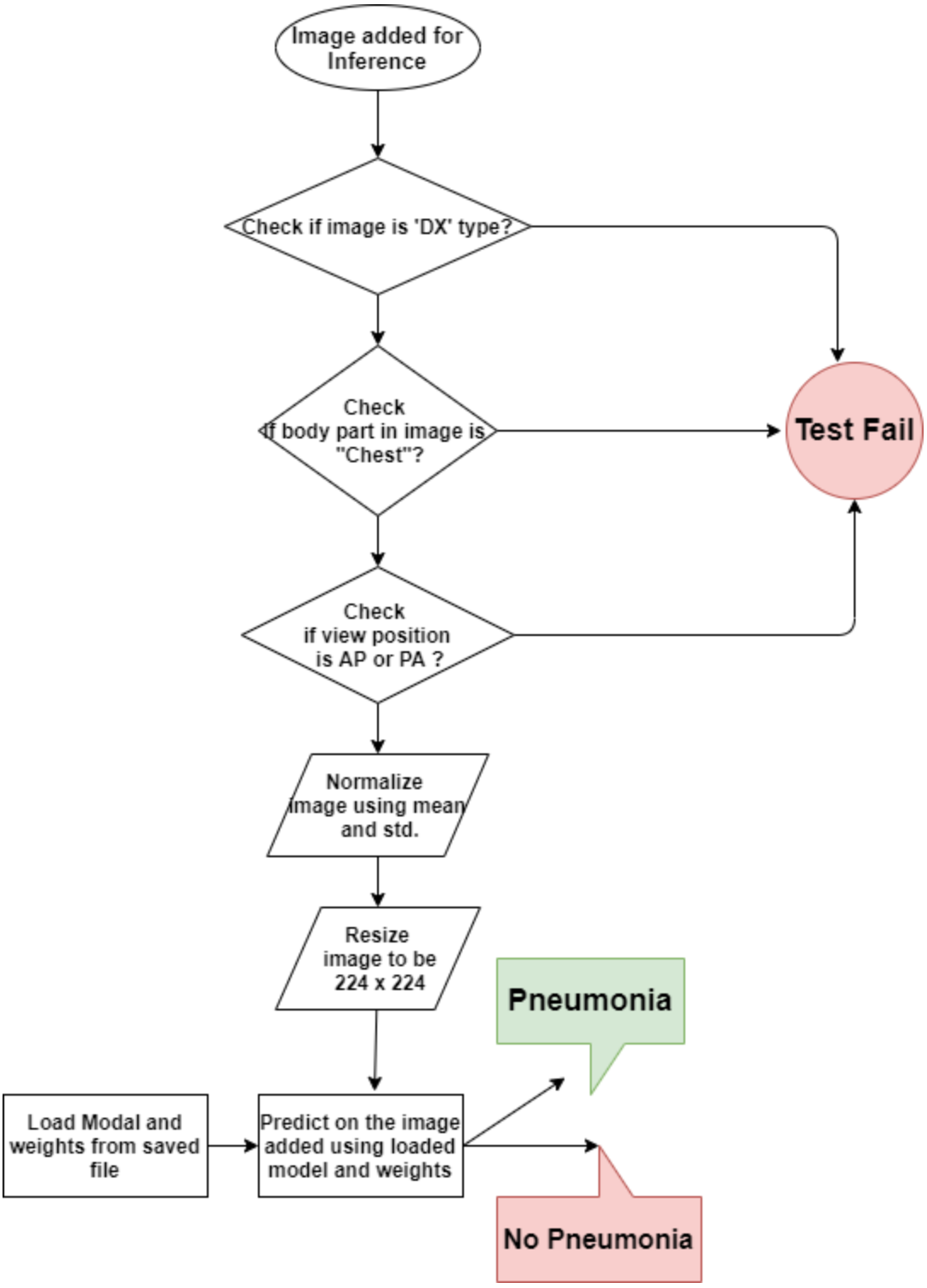
All pre-existing CNN and max pooling layers are intact and frozen:

```
Model: "vgg16"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 224, 224, 3)       0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0
_________________________________________________________________
```

After the last pooling layer, three dense layers are added with dropout added to each layer except the last to avoid overfitting:

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
model_1 (Model)              (None, 7, 7, 512)         14714688
_________________________________________________________________
flatten_1 (Flatten)          (None, 25088)             0
_________________________________________________________________
dropout_1 (Dropout)          (None, 25088)             0
_________________________________________________________________
dense_1 (Dense)              (None, 1024)              25691136
_________________________________________________________________
dropout_2 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_2 (Dense)              (None, 512)               524800
_________________________________________________________________
dropout_3 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_3 (Dense)              (None, 256)               131328
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 257
=================================================================
Total params: 41,062,209
Trainable params: 28,707,329
Non-trainable params: 12,354,880
_________________________________________________________________
```

Algorithm Function:



DICOM Checking Steps: DICOM is checked first to ensure that the type of x-ray is DX, the body part in x-ray is 'Chest' and the view position is either AP or PA.

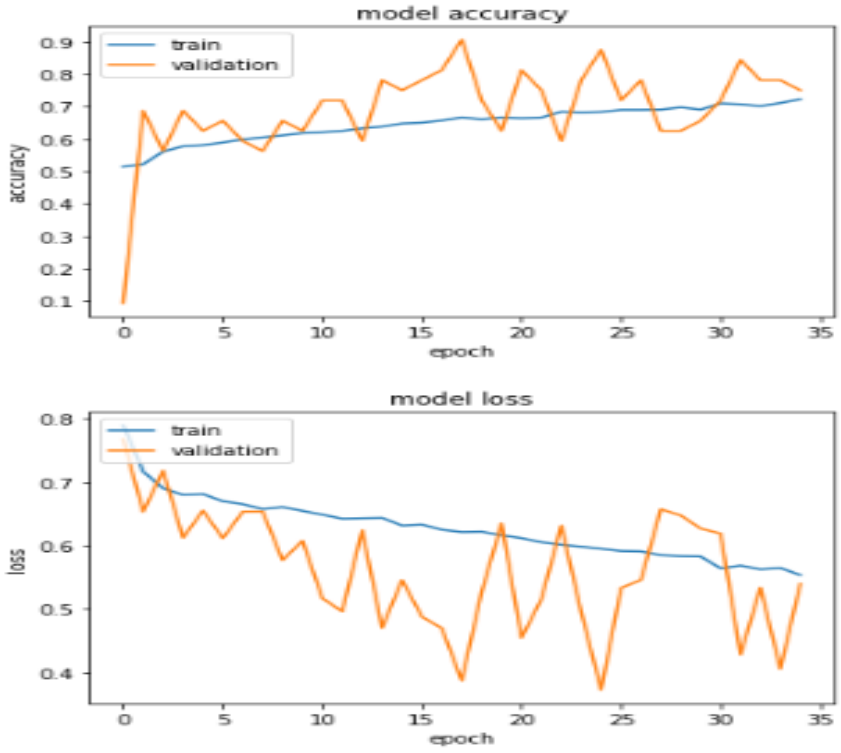
Preprocessing Steps: The image is preprocessed to be normalized using standardization and resized to be of height and width of 224 x 224

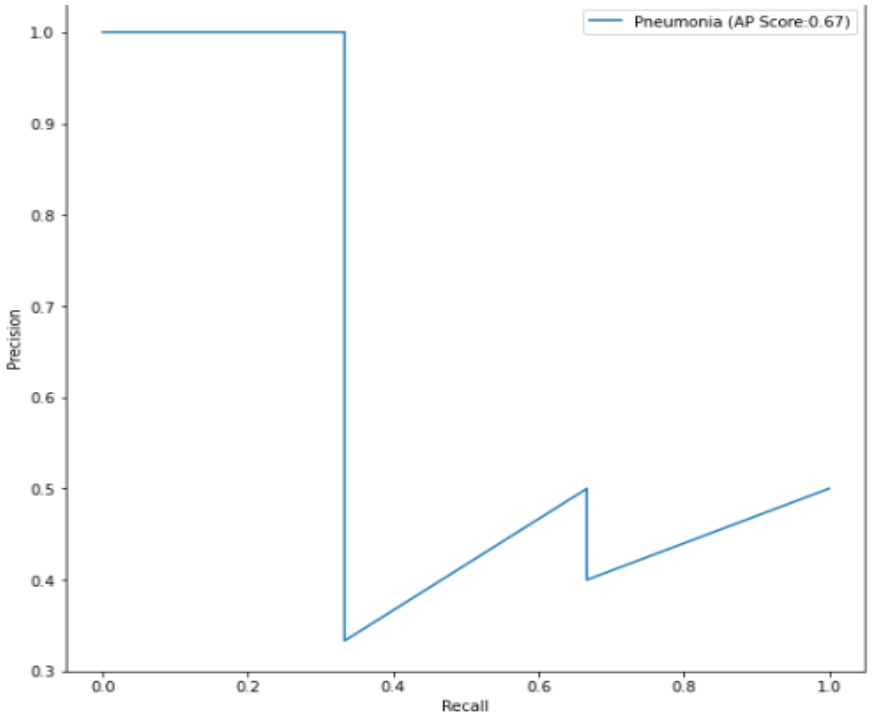
CNN Architecture: VGG16 model is used up to the last pooling layer as shown in figure then the model is fine-tuned using four dense layer each (but the last) accompanied with a dropout layer to avoid overfitting

3. Algorithm Training

Parameters:

- Types of augmentation used during training: rescaling by dividing each pixel by 255, added horizontal flip, height shift range and width shift range by factor of 0.1, rotation range by 10 degree, shear range by 0.1, zoom range by 0.1
- Batch size: 32
- Optimizer learning rate: 0.0001
- Layers of pre-existing architecture that were frozen: All convolution, max pooling layers were frozen
- Layers of pre-existing architecture that were fine-tuned: None, new dense layers were added for fine-tuning.
- Layers added to pre-existing architecture: four dense layer each (but the last) accompanied with a dropout layer to avoid overfitting.

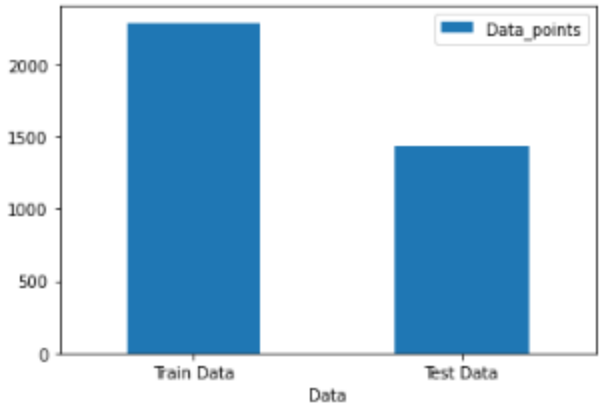




Final Threshold and Explanation:

The threshold was chosen to be 0.48 where recall is 1.0 to increase recall and allowing some margin for False Positives because the purpose of the device is to increase recall for assisting radiologist in detecting truly positive case in less amount of time.

4. Databases

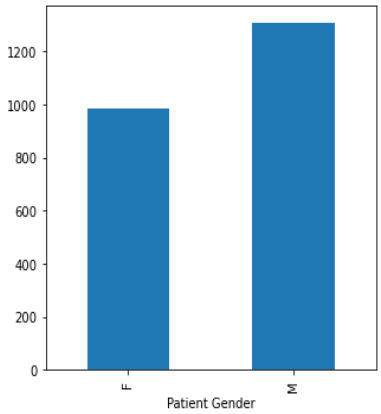


Description of Training Dataset:

1. Size of the training dataset: 2290
2. The number of positive cases and the its radio to the number of negative cases: 1:1 (equal number of positive and negative cases)
3. The patient demographic data (as it is available) :

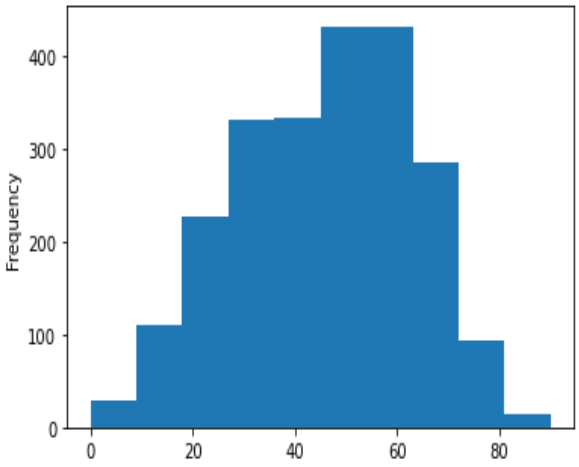
   - w.r.t to gender distribution:

```
In [25]: # train_data['Pneumonia'].sum().plot.bar(x='Pneumonia')
         fig, ax = plt.subplots(figsize=(5,5))
         train_data.groupby(['Patient Gender']).count()['Pneumonia'].plot.bar()

Out[25]: <matplotlib.axes._subplots.AxesSubplot at 0x7faaa904f9d0>
```



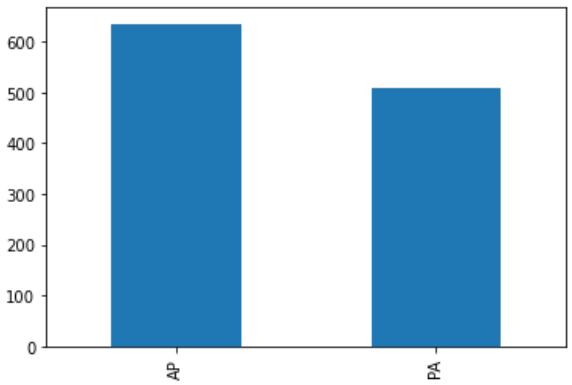
   - w.r.t to age group distribution:

```
In [28]: train_data.loc[all_xray_df['Patient Age'] > 100, 'Patient Age'] = 0
         train_data['Patient Age'].plot.hist()

Out[28]: <matplotlib.axes._subplots.AxesSubplot at 0x7faaa8ff8cd0>
```



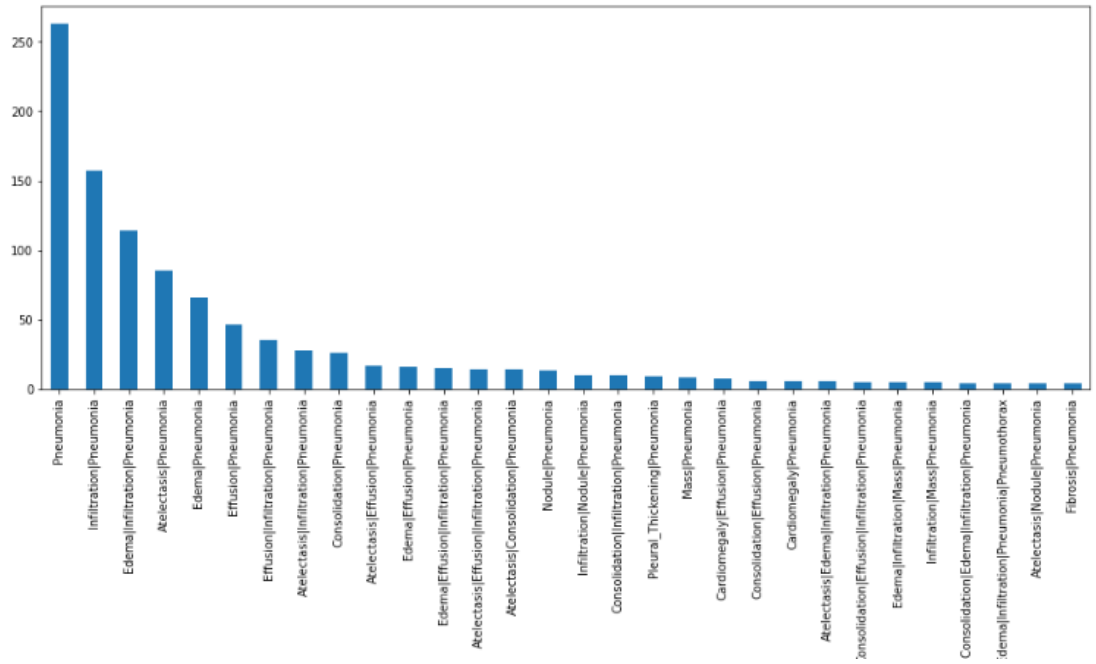
4. The radiologic techniques used and views taken

```
In [29]: pd.value_counts(train_data.loc[train_data['Pneumonia'] == 1.0, 'View Position'] ).plot.bar()
Out[29]: <matplotlib.axes._subplots.AxesSubplot at 0x7faaa8eb2a50>
```



5. The co-occurrence frequencies of pneumonia with other diseases and findings:

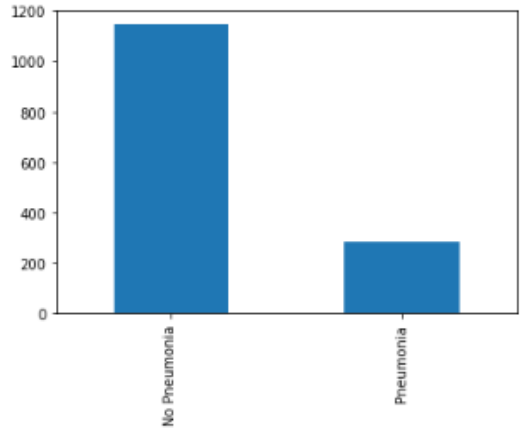
```
In [30]: plt.figure(figsize=(16,6))
         train_data[train_data.Pneumonia==1]['Finding Labels'].value_counts()[0:30].plot(kind='bar')
Out[30]: <matplotlib.axes._subplots.AxesSubplot at 0x7faaa8e1e2d0>
```



Description of Validation Dataset:

1. Size of the testing dataset: 1430
2. The number of positive cases and the its radio to the number of negative cases: negative cases are 25% more than positive cases.

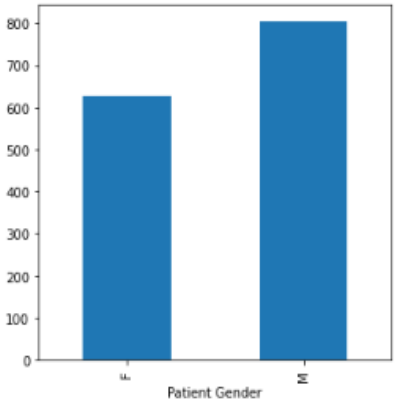
```
In [31]: pd.value_counts(val_data['pneumonia_class']).plot(kind="bar")
Out[31]: <matplotlib.axes._subplots.AxesSubplot at 0x7faaa8d6fd50>
```



3. The patient demographic data (as it is available) :

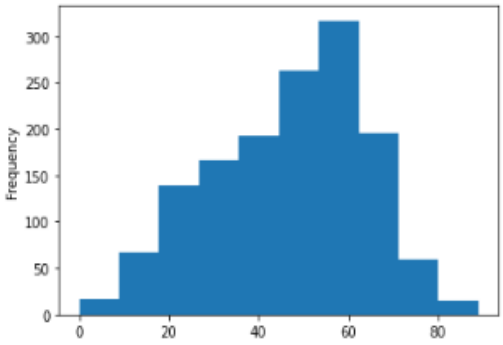
   - w.r.t to gender distribution:

```
In [32]: # train_data['Pneumonia'].sum().plot.bar(x='Pneumonia')
         fig, ax = plt.subplots(figsize=(5,5))
         val_data.groupby(['Patient Gender']).count()['Pneumonia'].plot.bar()
Out[32]: <matplotlib.axes._subplots.AxesSubplot at 0x7faaa8dd0c90>
```



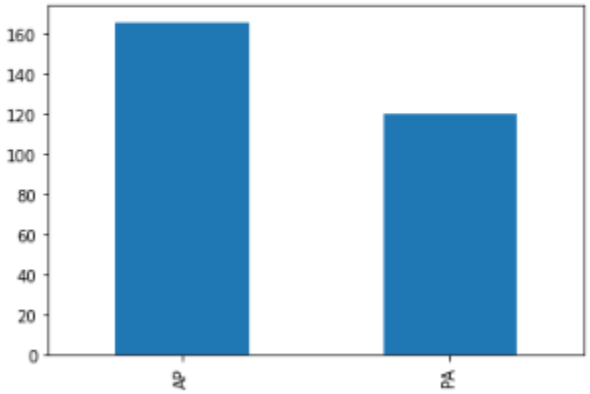
   - w.r.t to age group distribution:

```
In [34]: val_data.loc[val_data['Patient Age'] > 100, 'Patient Age'] = 0
         val_data['Patient Age'].plot.hist()

Out[34]: <matplotlib.axes._subplots.AxesSubplot at 0x7faaa8d2ead0>
```



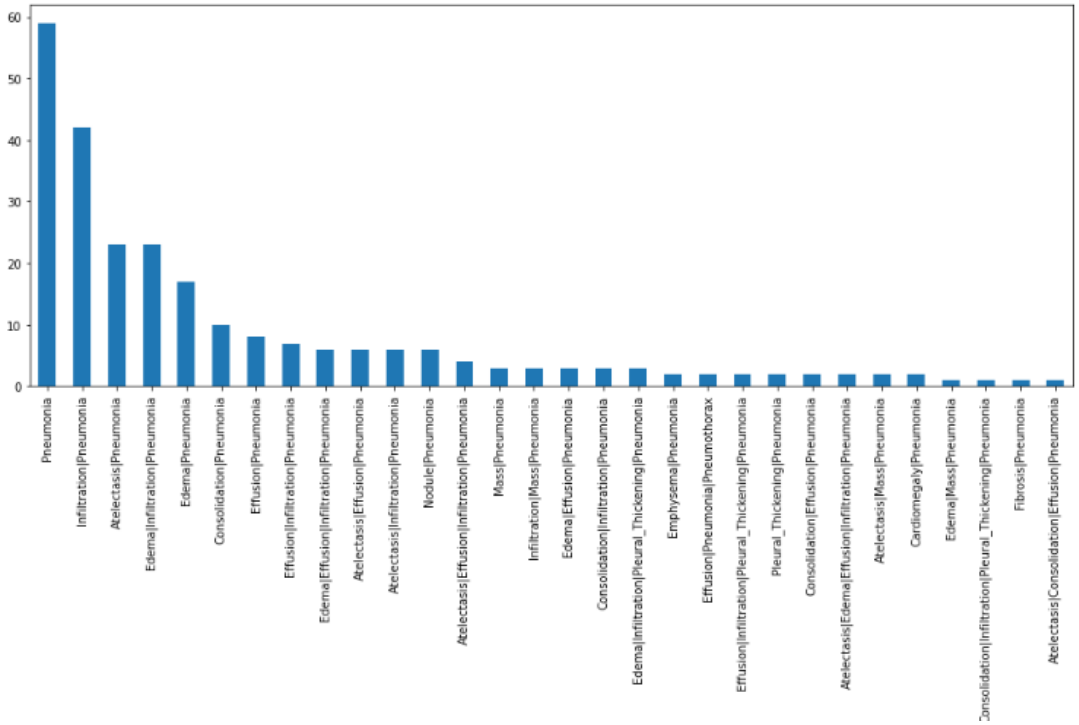
4. The radiologic techniques used and views taken

```
In [38]: pd.value_counts(val_data.loc[val_data['Pneumonia'] == 1.0, 'View Position'] ).plot.bar()

Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7faaa8c388d0>
```



5. The co-occurrence frequencies of pneumonia with other diseases and findings:

```
In [39]: plt.figure(figsize=(16,6))
         val_data[val_data.Pneumonia==1]['Finding Labels'].value_counts()[0:30].plot(kind='bar')
Out[39]: <matplotlib.axes._subplots.AxesSubplot at 0x7faaa8bbec50>
```



5.

Ground Truth:

Ground truth is established using the text-mined fourteen disease image labels (where each image can have multi-labels), mined from the associated radiological reports using natural language processing. The text-mined disease labels are expected to have accuracy >90%.

6. FDA Validation Plan

Patient Population Description for FDA Validation Dataset:

For FDA Validation Plan for this algorithm we will be needing chext x-rays of patients with the following demographics:

- Age range of patients: 20-70

- Sex: Male and Female in almost same proportion

- Body Part Imaged: Chest

- Prevalence of disease of interest: almost 20% of the whole test data.

- Any other diseases that should be included or excluded as comorbidities in the population: some comorbidity that can exist with Pneumonia are Infiltration, Edema+Infiltration, Atelectasis, Edema, Effusion

Ground Truth Acquisition Methodology:

Annotations can be obtained independently from four practicing radiologists, who can be asked to label all 14 pathologies. The radiologists can have varied level of experience like 4, 7, 25, and 28 years of experience.

Algorithm Performance Standard:

We assess the performance of both radiologists and our device on the test set for the pneumonia detection task. For each of the images in the test set,

we have 4 labels from four practicing radiologists and 1 label from our device. We will have a final ground truth for each label by assigning a weight to each label of radiologist based on experience level of that radiologist.

We will compute the F1 score for each individual radiologist and for our device against final labels as ground truth. We will report the mean of the 4 resulting F1 scores for each radiologist and for our device, along with the average F1 across the radiologists.To determine whether our model performance is statistically significantly higher than radiologist performance, we also calculate the difference between the average F1 score of our model and the average F1 score of the radiologists.