FDA Submission

Your Name: Ting Lu

Name of your Device: AttnPnoNet

Algorithm Description

1. General Information

Intended Use Statement: Assist radiological diagnosis of pneumonia

Indications for Use: In Screening Studies: Exclude patients unlikely to have pneumonia. In Diagnostic Studies: Deprioritize the review of predicted negative cases and assist radiologists in prioritizing cases with a high probability of pneumonia for initial review, enabling faster diagnosis and treatment for positive patients.

Device Limitations: The device can't replace the diagnosis decision of radiologist, only as a reference.

Clinical Impact of Performance:

In non-emergency scenarios, leveraging the algorithm to aid in the radiological diagnosis of pneumonia is recommended.

With a specificity of 32%, the algorithm reliably identifies negative cases (no pneumonia), while achieving a precision of 31% for positive predictions (pneumonia).

The model demonstrates a noteworthy recall rate of 81%, accurately pinpointing true positive instances (pneumonia). Consequently, when the algorithm yields a negative prediction, it instills confidence due to its robust capability in identifying positive patients. This heightened recall rate renders the predictions particularly suitable for supporting screening initiatives and streamlining radiologists' workflows. Prioritizing the review of predicted positive cases can optimize efficiency in radiological assessments.

2. Algorithm Design and Function

DICOM Checking Steps:

- is examined body part the chest area?
- is the image scan position either posterior/anterior (PA) or anterior/posterior (AP)?
- is the modality a digital radiography (DX)?

Preprocessing Steps:

- Images are resized to 224x224 to fit the input dimension of the pre-trained VGG16 CNN model.
- Rescaled pixel values of image by standardization (substract mean and divide by standard deviation) because VGG16 pretrained weights expect normalized data as input

CNN Architecture:

We start from a classic Convolutional Neural Network (CNN) model VGG16, which has been pre-trained on a large image dataset and learned to extract features from images for classification tasks.

Then, we built a new model on top of this pre-trained VGG16 model, called the attention model. The goal of the attention model is to make the neural network focus not just on the entire image equally but selectively on important parts of the image, enabling more effective learning and feature extraction from images.

To achieve this goal, we added some special layers after the convolution and pooling layers of VGG16, including a Global Average Pooling layer (GAP). The role of global average pooling is to take the average of all pixel values in each feature map, transforming it into a fixed-length vector. This vector contains the average importance of each feature map and better reflects the importance level of different parts in an image. By adjusting the network structure and training process, we enable this new model to automatically learn and determine which parts are crucial in an image, thereby enhancing its performance for tasks like image classification or other related tasks.

Note: regarding neural network model structure, I'm inspired by @Claudia and do some modification on layers and hyperparameters.

Attention Layer

```
Model: "attention_layer"
_________________________________________________________________________________________________
Layer (type)                 Output Shape              Param #     Connected to
=================================================================================================
feature_input (InputLayer)   (None, 7, 7, 512)    0
_________________________________________________________________________________________________
features_batch_norm (BatchNorma (None, 7, 7, 512)    2048        feature_input[0][0]
_________________________________________________________________________________________________
conv2d_9 (Conv2D)            (None, 7, 7, 128)    65664       features_batch_norm[0][0]
_________________________________________________________________________________________________
conv2d_10 (Conv2D)           (None, 7, 7, 32)     4128        conv2d_9[0][0]
_________________________________________________________________________________________________
conv2d_11 (Conv2D)           (None, 7, 7, 16)     528         conv2d_10[0][0]
_________________________________________________________________________________________________
average_pooling2d_3 (AveragePoo (None, 7, 7, 16)     0           conv2d_11[0][0]
_________________________________________________________________________________________________
AttentionMap2D (Conv2D)      (None, 7, 7, 1)      17          average_pooling2d_3[0][0]
_________________________________________________________________________________________________
conv2d_12 (Conv2D)           (None, 7, 7, 512)    512         AttentionMap2D[0][0]
_________________________________________________________________________________________________
multiply_3 (Multiply)        (None, 7, 7, 512)    0           conv2d_12[0][0]
                                                              features_batch_norm[0][0]
_________________________________________________________________________________________________
global_average_pooling2d_5 (Glo (None, 512)          0           multiply_3[0][0]
_________________________________________________________________________________________________
global_average_pooling2d_6 (Glo (None, 512)          0           conv2d_12[0][0]
_________________________________________________________________________________________________
RescaleGAP (Lambda)          (None, 512)          0           global_average_pooling2d_5[0][0]
                                                              global_average_pooling2d_6[0][0]
_________________________________________________________________________________________________
dropout_5 (Dropout)          (None, 512)          0           RescaleGAP[0][0]
_________________________________________________________________________________________________
dense_5 (Dense)              (None, 128)          65664       dropout_5[0][0]
_________________________________________________________________________________________________
dropout_6 (Dropout)          (None, 128)          0           dense_5[0][0]
_________________________________________________________________________________________________
dense_6 (Dense)              (None, 1)            129         dropout_6[0][0]
=================================================================================================
Total params: 138,690
Trainable params: 137,154
Non-trainable params: 1,536
```

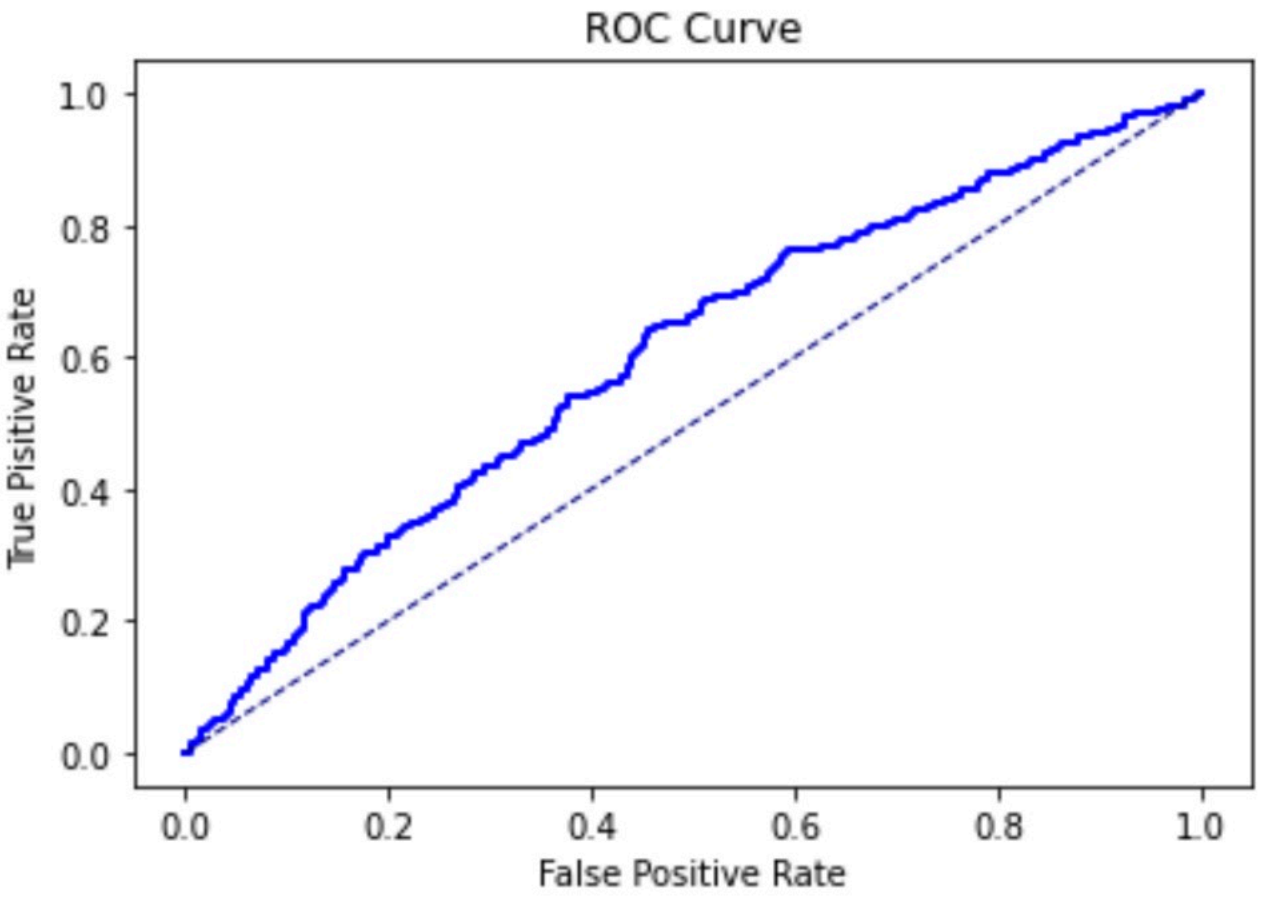
Overall VGG16 adding attention layer model architecture

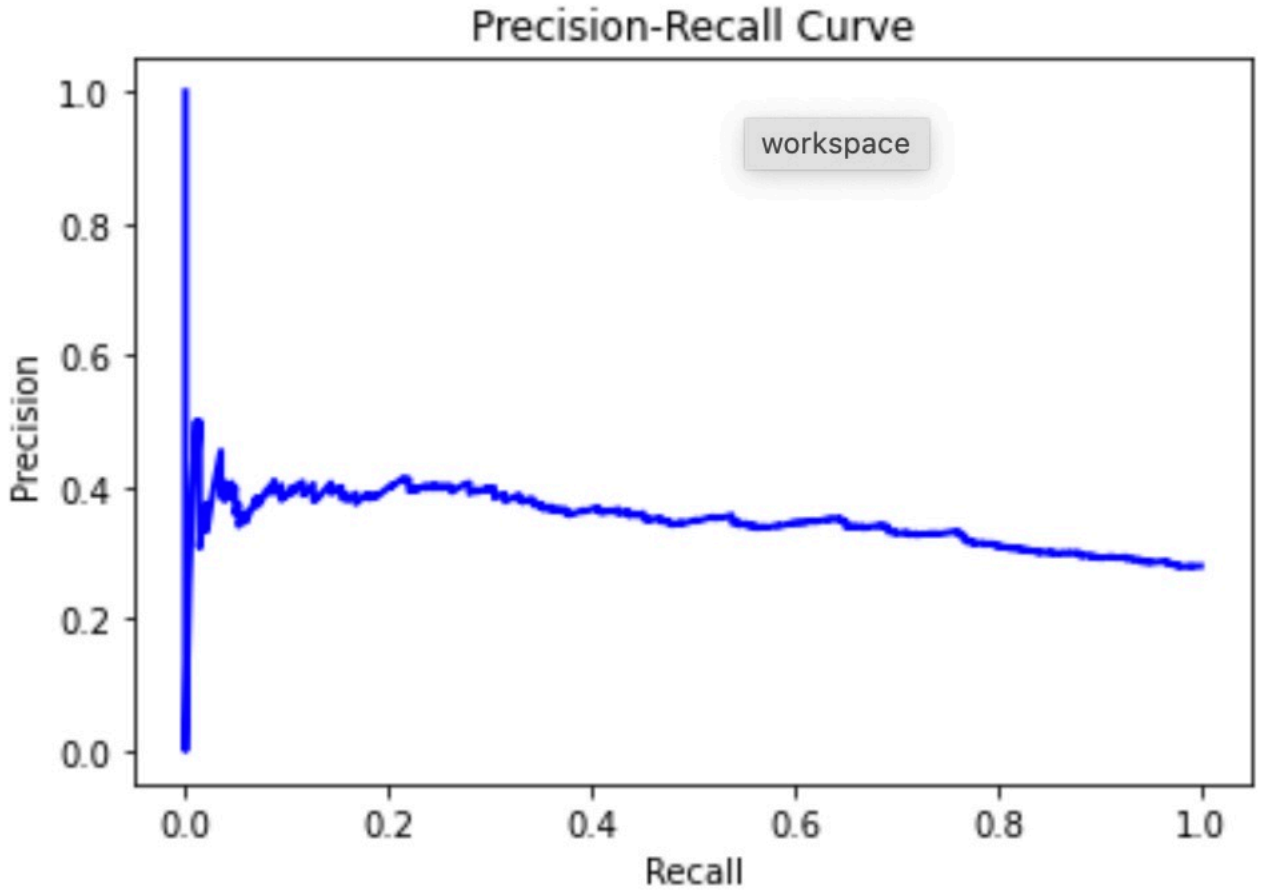
```
Model: "Attention_CNN_model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
vgg16 (Model)                (None, 7, 7, 512)         14714688
_________________________________________________________________
attention_layer (Model)      (None, 1)                 138690
=================================================================
Total params: 14,853,378
Trainable params: 137,154
Non-trainable params: 14,716,224
_________________________________________________________________
```

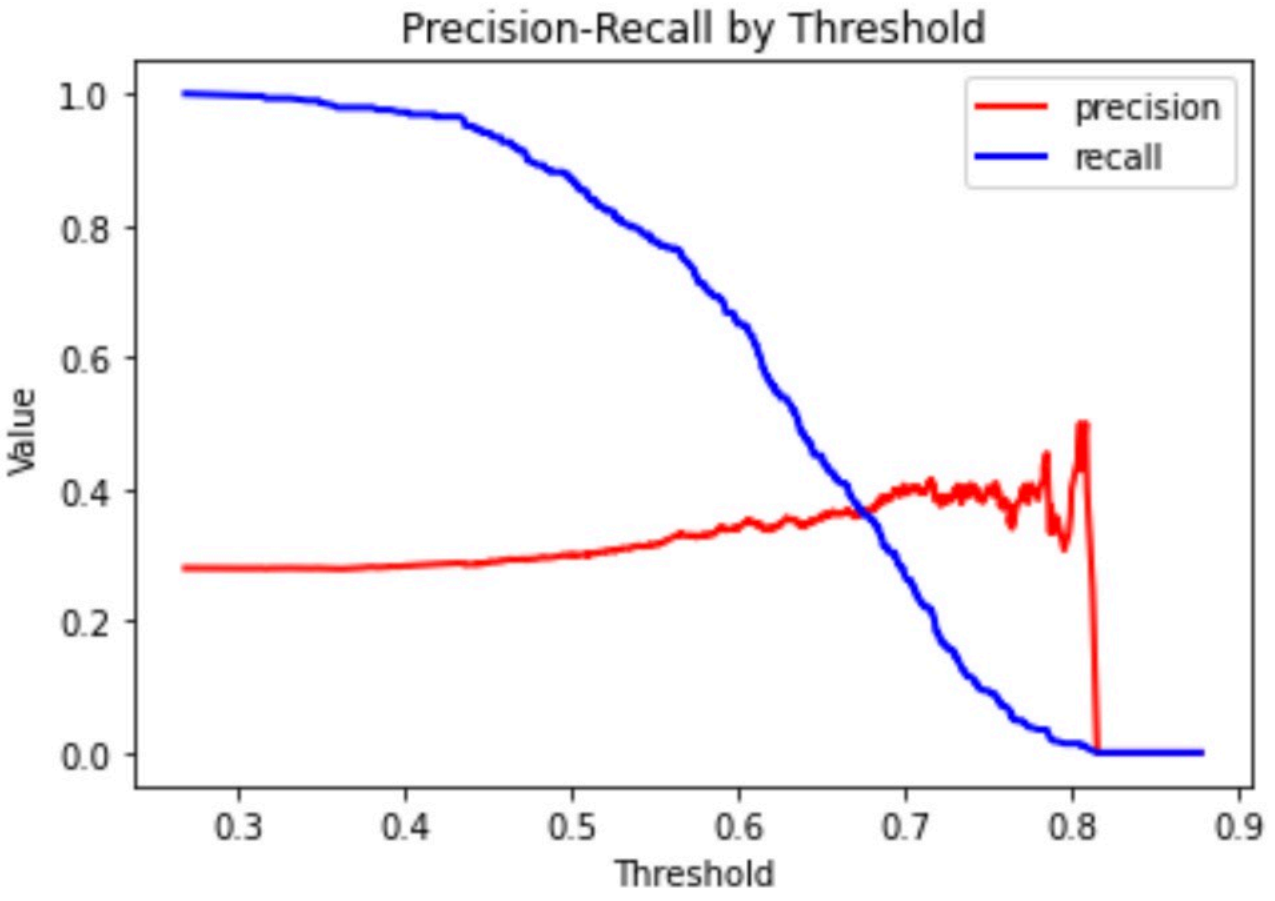
3. Algorithm Training

Parameters:

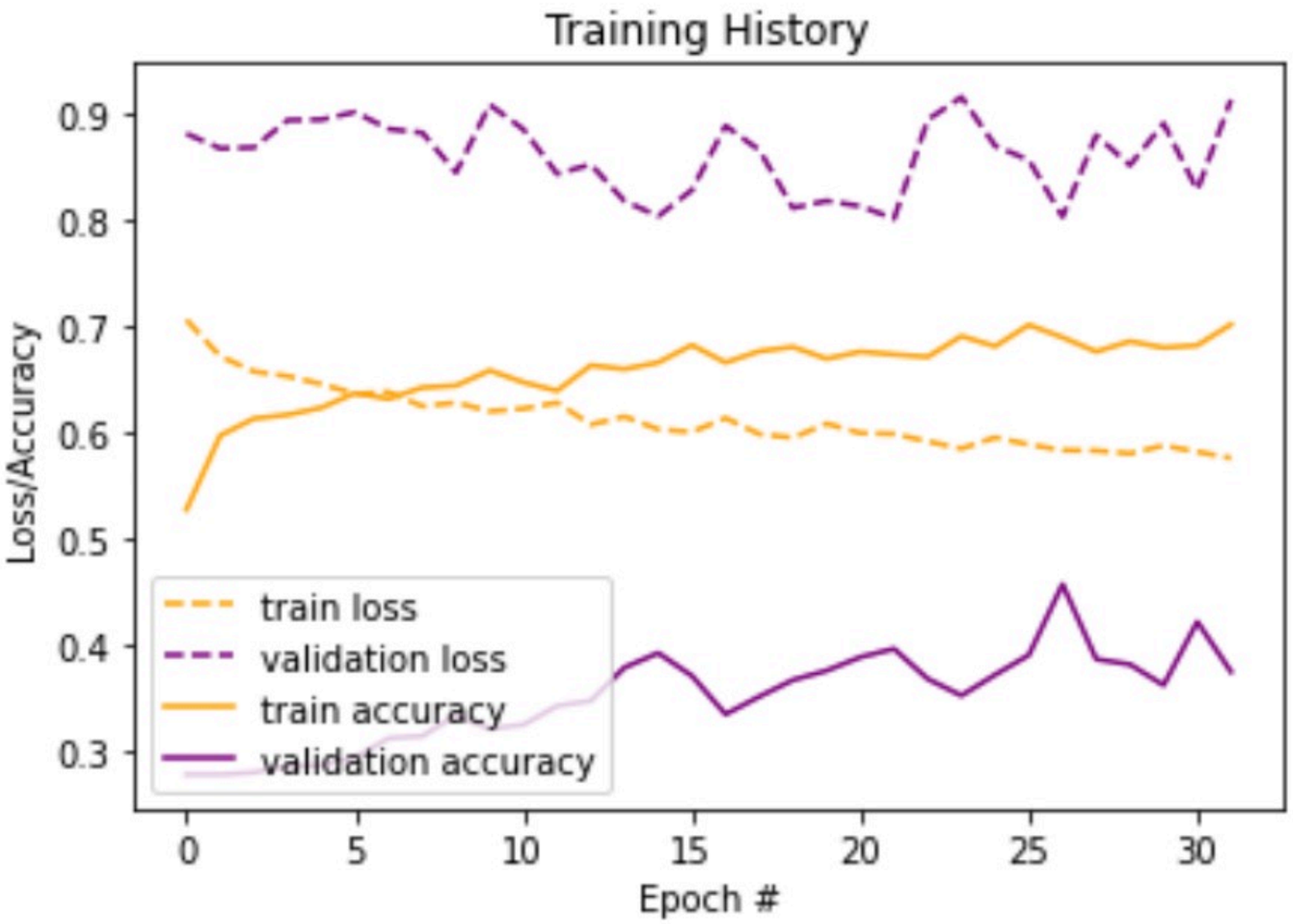
- Image augmentation during training: Rescaled 1/255, Centered & Std normalized sample-wise, Horizontal flips, Height/Width shift range 0.05, Zoom range 0.15
- Batch size:
  - Training: 64
  - Validation: 1024
  - Prediction: 64
- Optimizer learning rate: Adam 0.0001 (1e-4)
- Layers of pre-existing architecture:
  - Frozen: First 17 layers of VGG16
  - Fine-tuned: All dense layers of VGG16 and attention
- Layers added to pre-existing architecture: Batch Normalization, Conv2D, Locally Connected 2D, Conv2D, Multiply, Global Avg Pooling, Global Avg Pooling, RescaleGAP, Dropout, Dense, Dropout, Dense

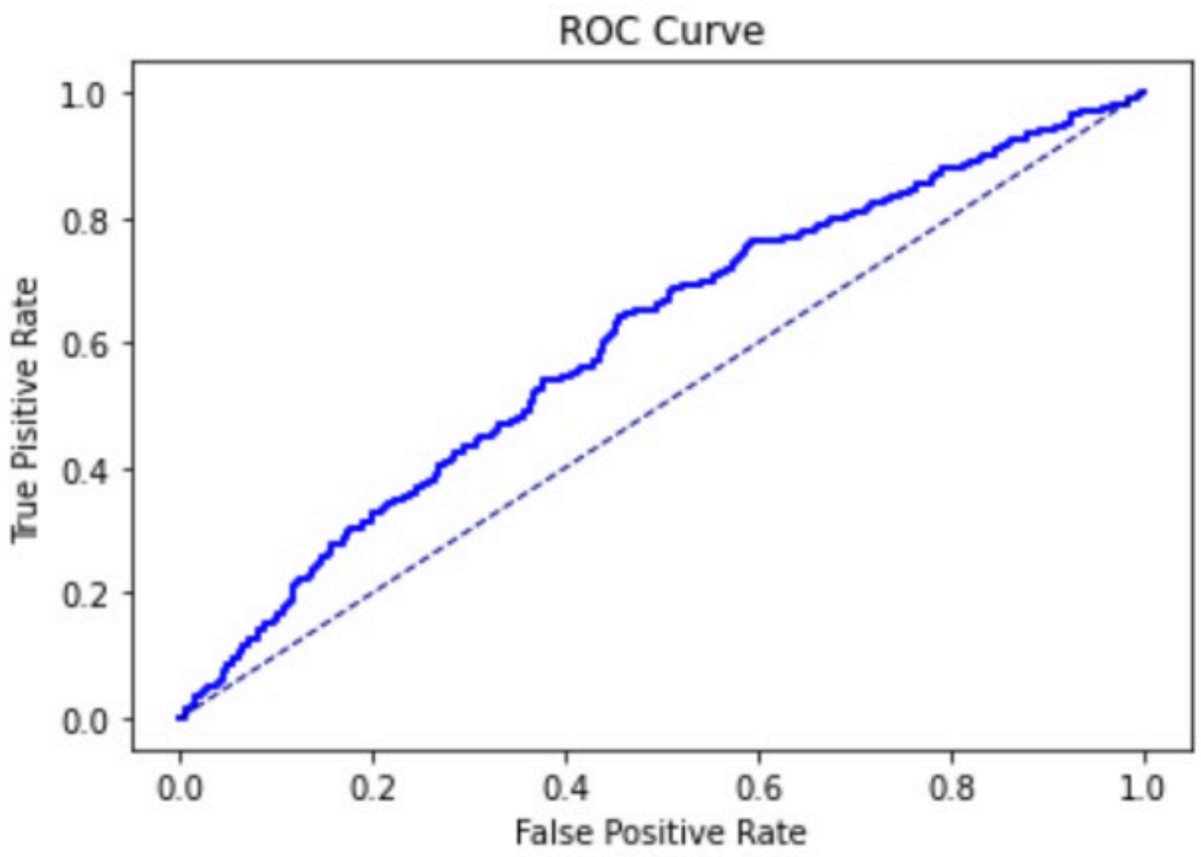

ROC Curve
True Pisitive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

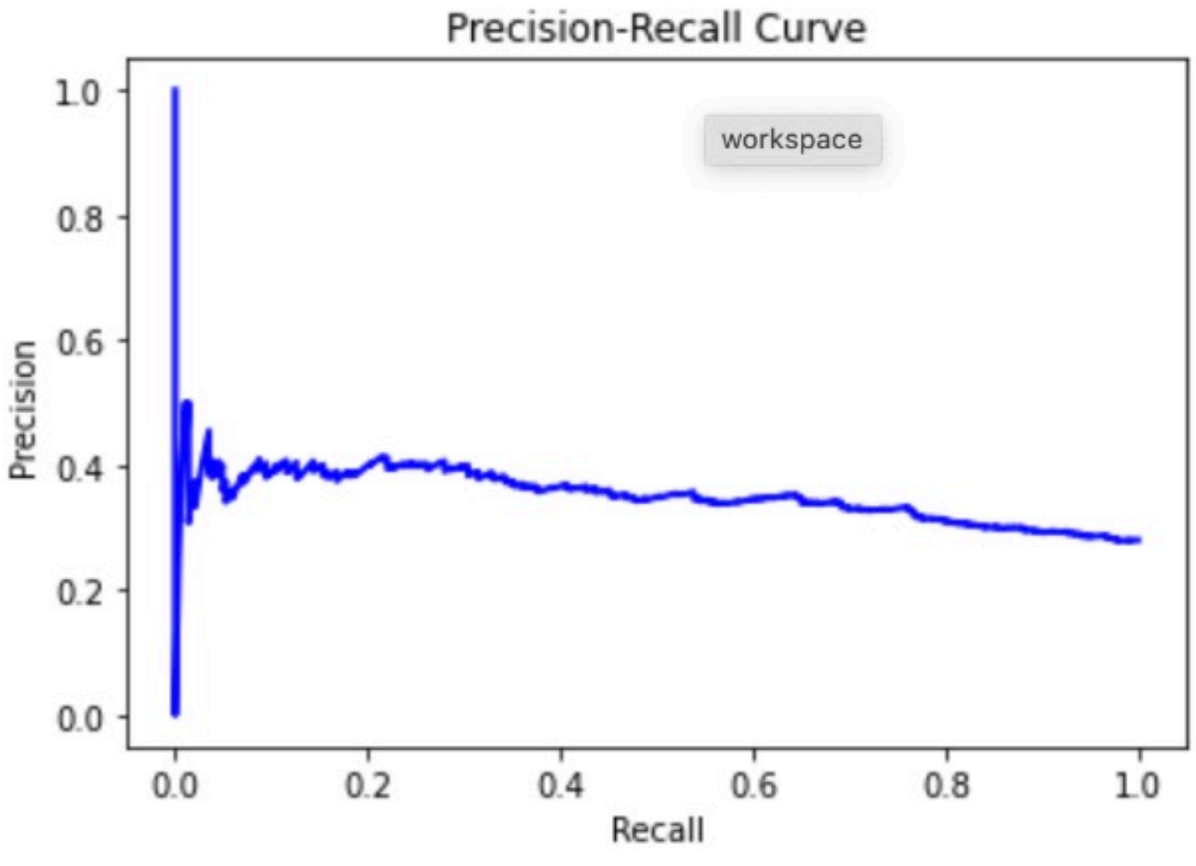

Precision-Recall Curve
Precision
Recall
workspace
0.0
0.2
0.4
0.6
0.8
1.0

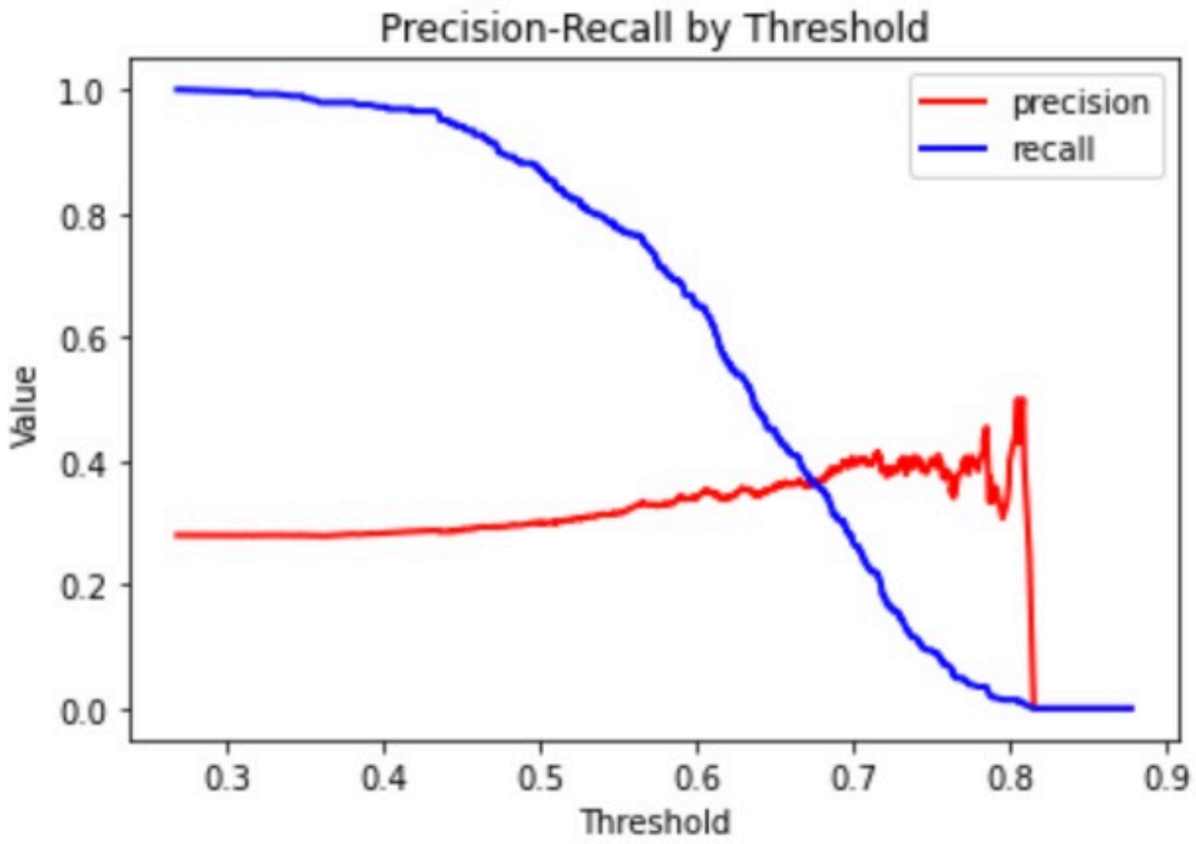

Precision-Recall by Threshold
precision
recall
Value
Threshold
0.0
0.2
0.4
0.6
0.8
1.0
0.3
0.4
0.5
0.6
0.7
0.8
0.9

Training loss & Validation loss

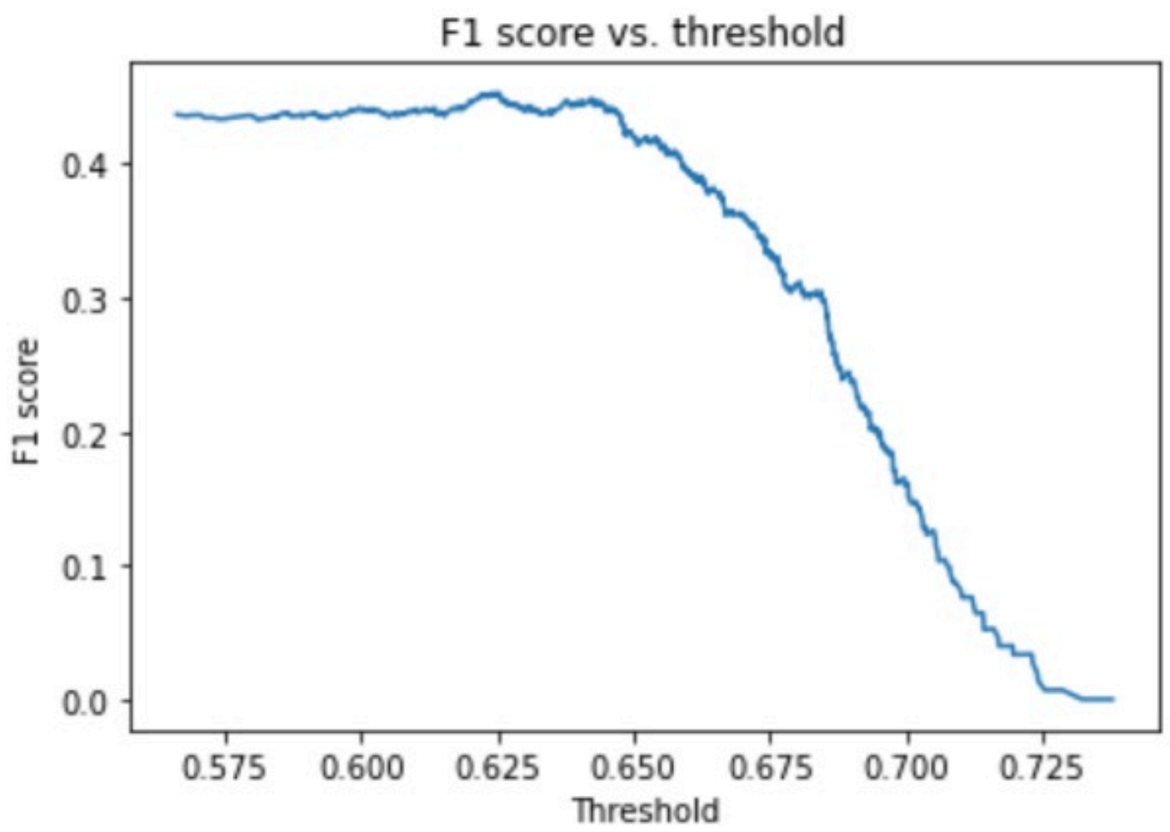








Final Threshold and Explanation:



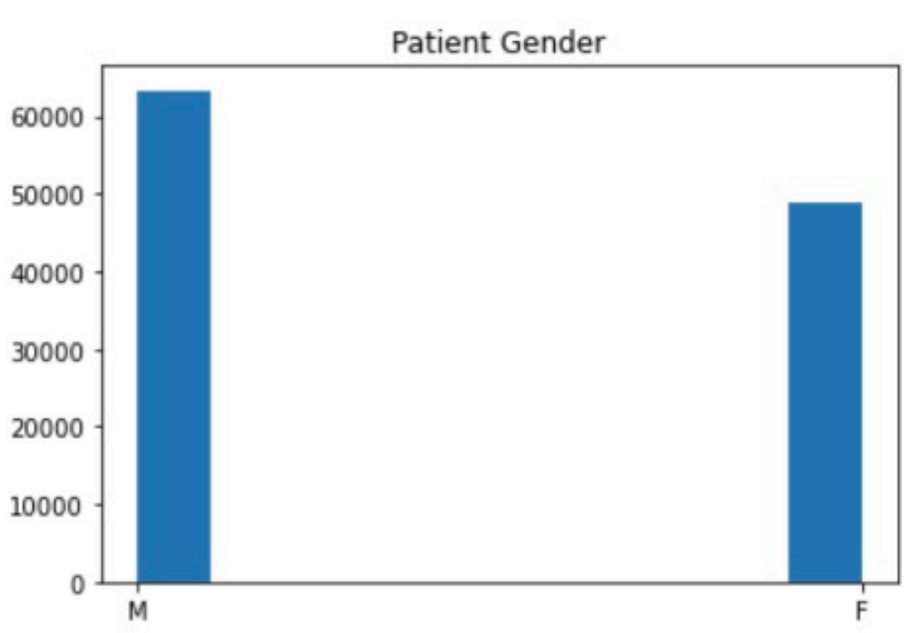

```
Precision: 0.3155006858710562 3
Recall: 0.8041958041958042
Threshold: 0.62521714
F1 Score: 0.4532019704433497 6
```

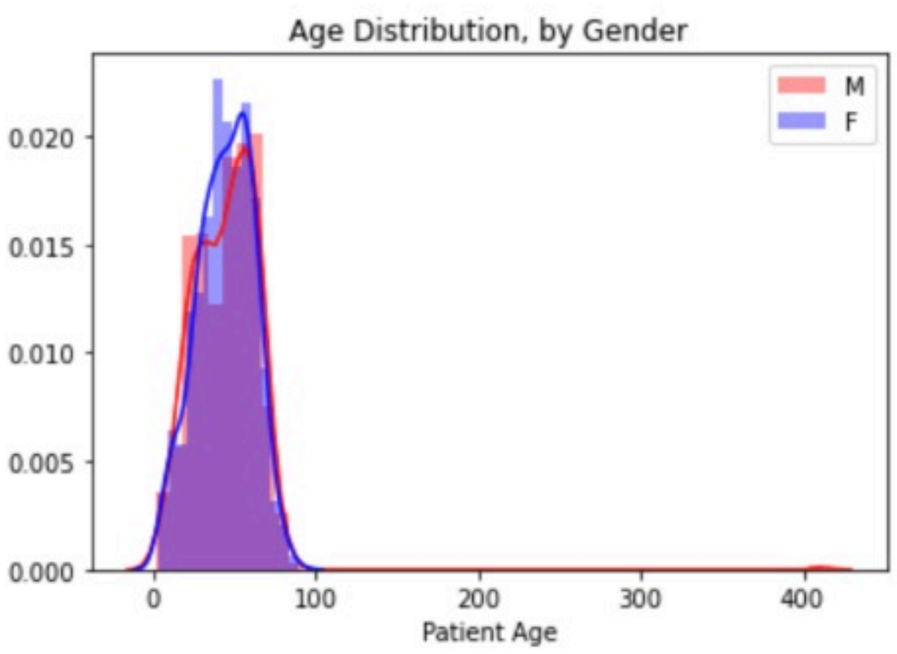
With the best threshold at 0.62, the model exhibits a high Recall (0.81), indicating its ability to accurately predict positive cases (pneumonia) among all actual positive cases. This is crucial for minimizing false negatives, ensuring fewer cases of pneumonia are missed. However, the Precision (0.31) is relatively low, suggesting a considerable proportion of false positives when the model predicts positive cases. This could lead to overdiagnosis. Additionally, the Specificity (0.32) is also not very high, indicating a lower accuracy in predicting negative cases (no pneumonia) among all actual negative cases. Consequently, when applying this model, a trade-off between Recall and Precision needs to be considered to determine appropriate thresholds, and further refinement is necessary to enhance the overall performance of the model.

4. Databases

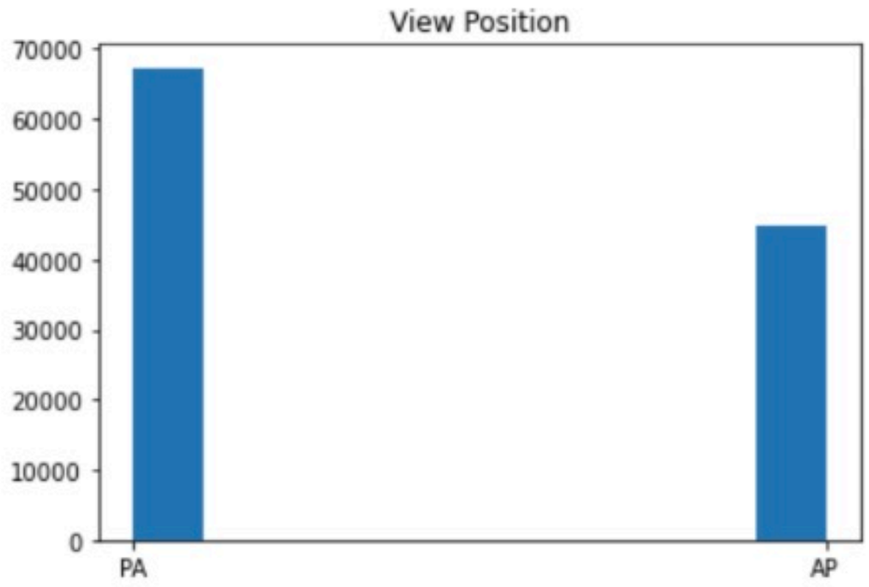
For EDA process, metadata for all images (Data_Entry_2017.csv) containing Image Index, Finding Labels, Follow-up #, Patient ID, Patient Age, Patient Gender, View Position, Original Image Size, and Original Image Pixel Spacing.

Age and Gender:

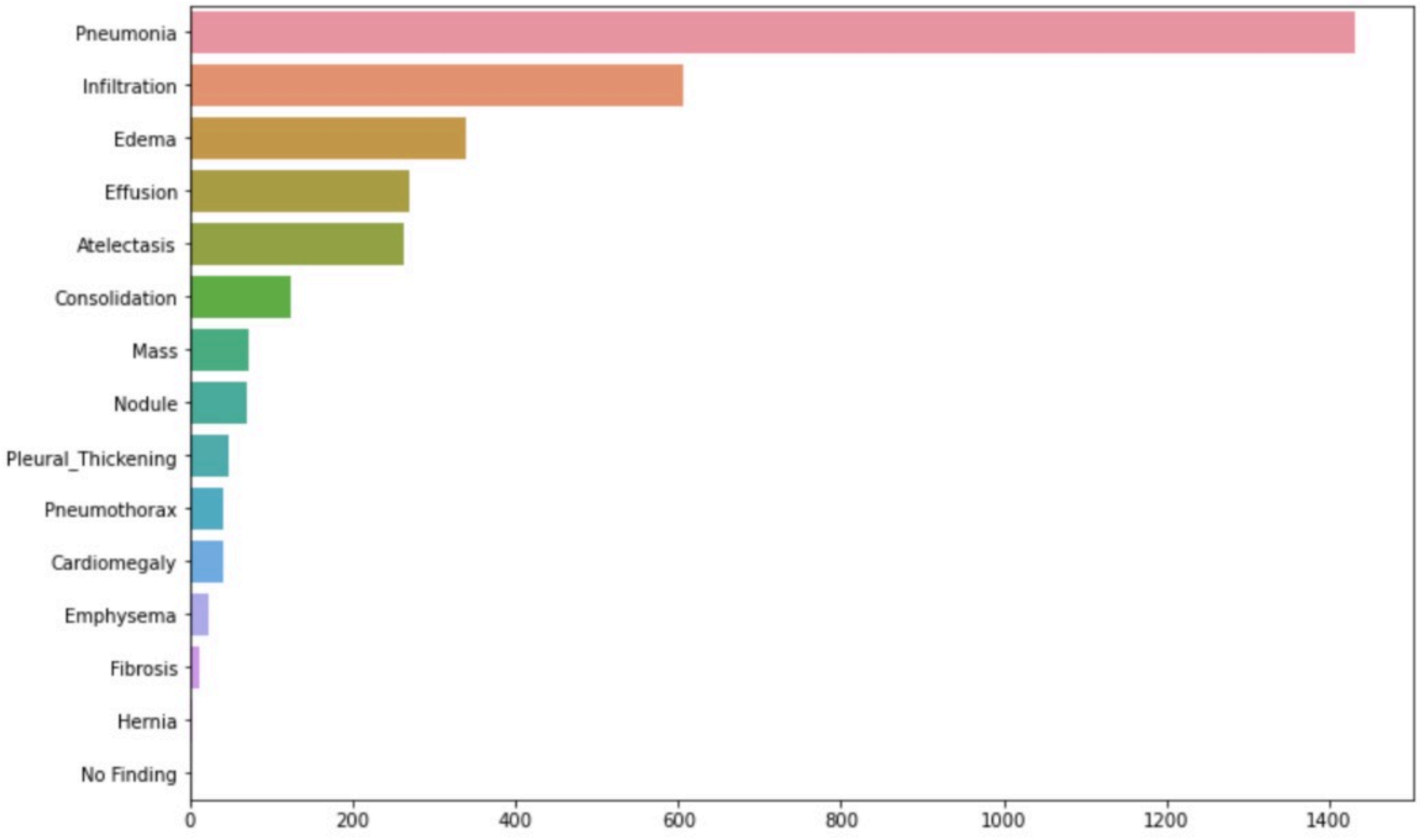




View Position:



Comorbidies:



Description of Training Dataset: The training dataset consisted of 2290 image files, with a 50/50 split of positive and negative pneumonia cases.

Description of Validation Dataset: The validation dataset consisted of 1430 image files, with a 20/80 split of positive and negative pneumonia cases to approach a more realistic distribution of pneumonia in the real world.

5. Ground Truth

Training and validation data (112,120 frontal-view chest X-ray PNG images in 1024*1024 resolution) were drawn from a comprehensive dataset curated by the NIH to address the scarcity of large X-ray datasets with accurate disease labels, vital for developing disease detection algorithms. While the original radiology reports remain inaccessible to the public, detailed information on the labeling process can be accessed at this paper.

The dataset comprises 112,120 X-ray images, each annotated with disease labels, derived from 30,805 unique patients. These labels were generated through Natural Language Processing (NLP) applied to radiological reports, capturing 14 common thoracic pathologies such as Atelectasis, Consolidation, and Pneumonia. One notable limitation of this dataset is its reliance on NLP-extracted labels, which may contain errors. However, the estimated accuracy of the NLP labeling exceeds 90%.

6. FDA Validation Plan

Patient Population Description for FDA Validation Dataset:

- Age: 0 to 100
- Gender: Female (56%) and Male (44%)
- Type of imaging modality: DX
- Body part imaged: Chest

Ground Truth Acquisition Methodology:

The gold standard: Sputum test or Pleural fluid culture, which is expensive and time consumed. The sliver standard: The comprehensive diagnostic results of three independent radiologists on X-ray reports.

Algorithm Performance Standard:

The performance of the model should be evaluated based on its F1 score compared to the silver standard. According to Rajpurkar et al. (2017), the average F1 score of radiologists is 0.387. To validate the effectiveness of this model, its F1 score should significantly exceed that of radiologists statistically.