FDA Submission

Your Name: Thevenin Julie

Name of your Device: PneumoDetect Pro

Algorithm Description

1. General Information

Intended Use Statement:

PneumoDetect Pro is intended for the automated classification of chest X-ray images to aid in the detection of pneumonia. The algorithm analyzes the images and provides a binary output indicating the presence or absence of pneumonia.

Indications for Use:

PneumoDetect Pro is indicated for use by trained healthcare professionals as a decision support tool to assist in the interpretation of chest X-ray images for the presence or absence of pneumonia. The device is not intended to replace the judgment of a qualified radiologist but aims to enhance the diagnostic process.

Device Limitations:

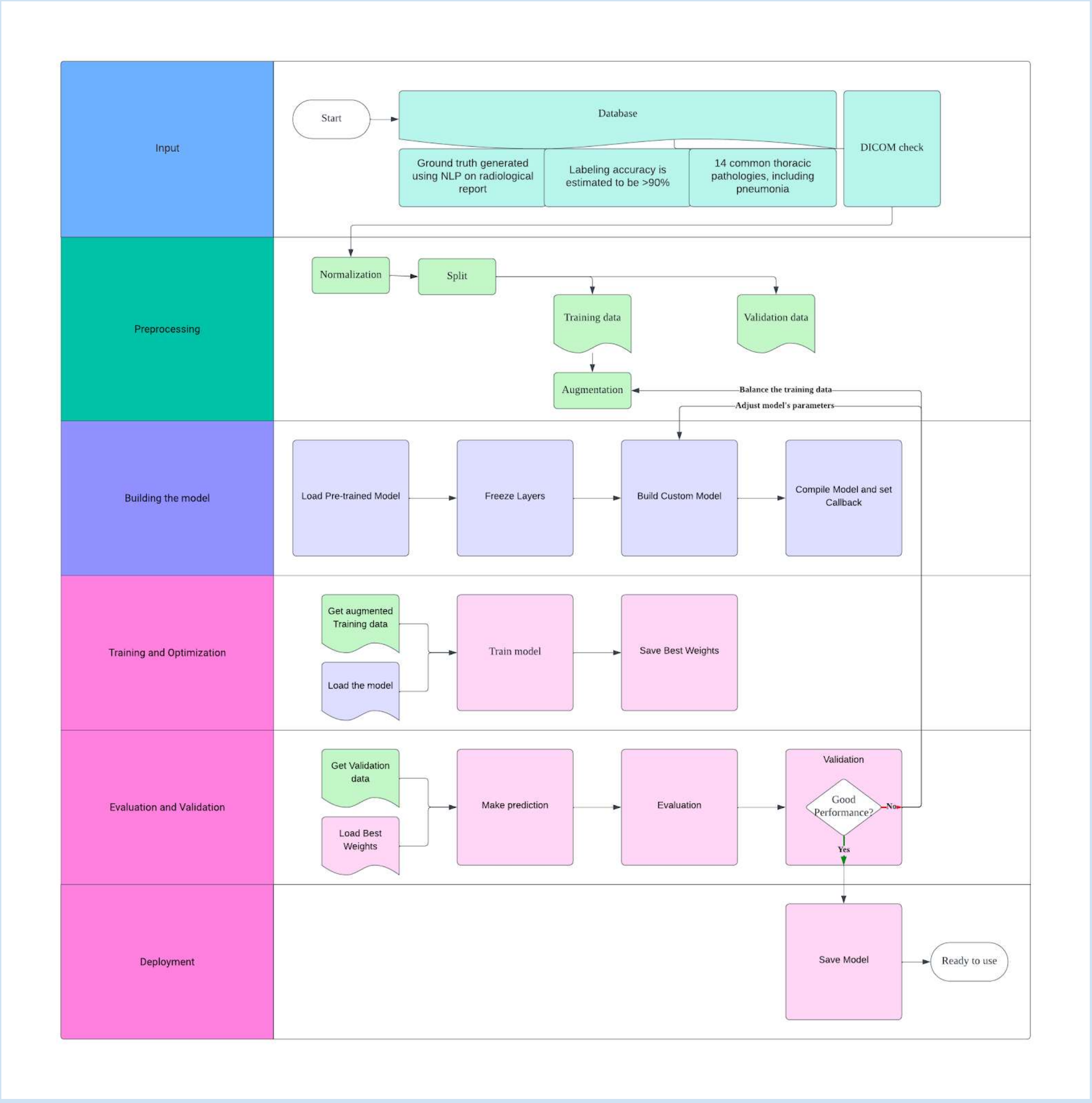
PneumoDetect Pro may have limitations in cases with ambiguous or overlapping radiological findings. Healthcare professionals need to consider clinical history, patient demographics, and other relevant information when interpreting results. The device is not designed to diagnose other thoracic pathologies beyond pneumonia.
The algorithm is designed to handle a variety of patient characteristics, including a balanced distribution of age and gender. However, it may exhibit limitations in cases with ambiguous radiological findings or certain comorbidities.

Clinical Impact of Performance:

The utilization of PneumoDetect Pro is anticipated to enhance the efficiency of pneumonia diagnosis by providing a rapid and consistent assessment. Nevertheless, Healthcare professionals must apply their clinical judgment and view the algorithm's output as an adjunctive tool in the diagnostic process. Therefore, it is recommended that professionals utilize PneumoDetect Pro as the initial step in the diagnostic workflow. Positive cases identified by the algorithm will be prioritized, creating an initial batch for thorough examination by the healthcare professional. Subsequently, the negative cases will follow in the diagnostic workflow. This approach ensures that positive cases are promptly identified and addressed, aligning to optimize the diagnostic process.

2. Algorithm Design and Function

****Algorithm Flowchart****



DICOM Checking Steps:
It is crucial to adhere to specific checking steps to ensure the compatibility of the algorithm with diverse patient characteristics in the dataset. The DICOM checking steps include:

1. Proper Image Acquisition Type (DX):

The DICOM wrapper verifies that the DICOM images processed by the algorithm correspond to the X-ray imaging modality.

2. Proper Body Part in Acquisition (Chest):

The DICOM wrapper confirms that the DICOM images are focused on the anatomical region specified as the "Chest."

3. Proper Image Acquisition Orientation: (PA/AP)

The DICOM wrapper doesn’t validate that the orientation of the DICOM images aligns with the orientation encountered because the model was trained on the two orientations.

Pre-Processing and Augmentation

1. Image normalization and resizing

The image should be normalized with pixel values between 0 and 1.
Because we are using VGG16 the image is resized in the following format (1,224,224,3).

Few augmentations of the given data have been made to get a variety of possible cases.
Different techniques were applied.
First, some basic ImageDataGenerator Augmentation Techniques:

2. Horizontal Flipping:

- This technique mimics the potential variations in patient positioning during X-ray imaging, where left and right orientations may differ.
- It enhances the model's ability to generalize to X-ray images with diverse anatomical orientations.

3. Rotation:

- This technique simulates the variability in the angle at which X-ray images are captured during routine medical imaging procedures.
- It improves the model's robustness to images captured at different angles, contributing to its adaptability in real-world scenarios.
- Parameters: random rotation between 0-20°

4. Zoom:

- This technique represents the potential differences in the scale and framing of chest X-ray images during acquisition.
- It augments the dataset with variations in image scale, allowing the model to learn features at different levels of magnification.
- Parameters: random zoom with a maximum of 0.1

Other custom Augmentation Functions were used:

5. Brightness Adjustment:

- This technique emulates the variations in image brightness due to differences in X-ray equipment or imaging conditions.
- It introduces diversity in brightness levels, ensuring the model is exposed to a range of illumination scenarios commonly encountered in real-world chest X-ray imaging.
- Parameters:
    - scaling factor for intensity: alpha = 1.5
    - scalar added to each pixel: beta = 30
    - formula: alpha * image + beta

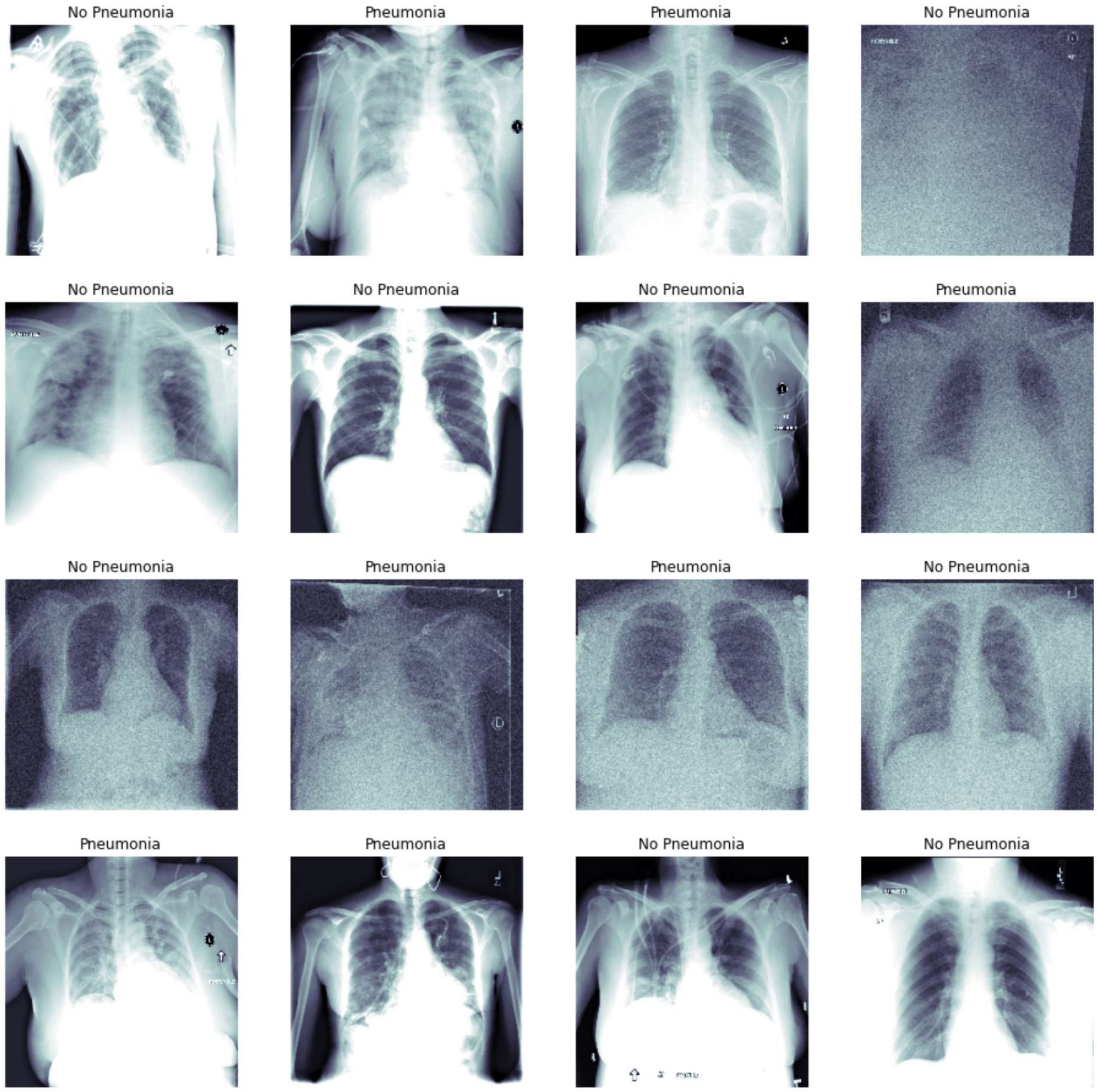
6. Gaussian Noise Addition:

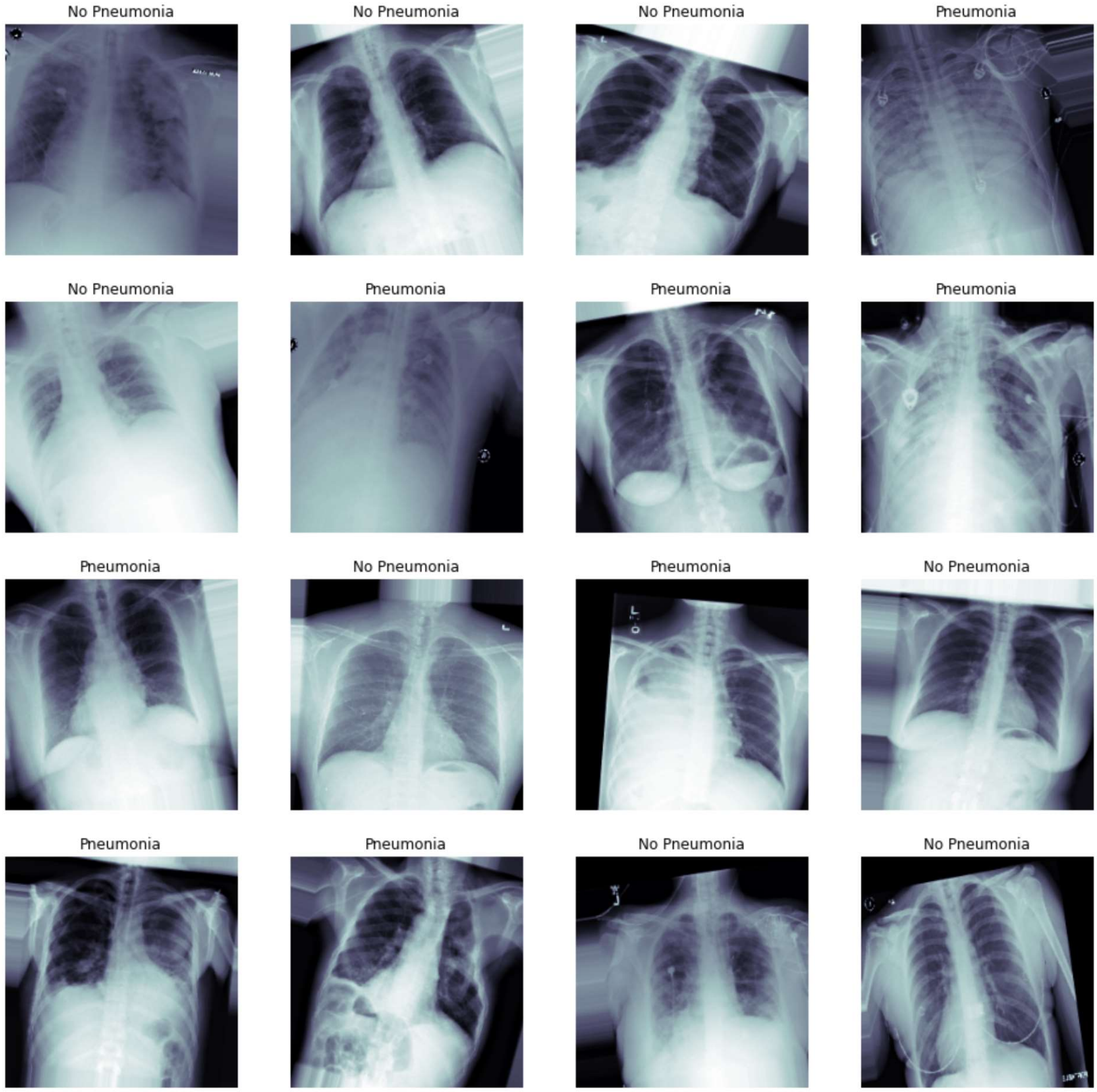
- This technique represents the inherent noise present in medical images, which can be caused by factors such as equipment imperfections.
- It introduces noise to the training data, allowing the model to learn to distinguish between signal and noise, improving its robustness to variations in image quality.
- Generate Gaussian noise with mean 0 and standard deviation 25

These augmentation techniques collectively contribute to diversifying the training dataset by simulating real-world variations in chest X-rays. The combination of horizontal flipping, rotation, and zoom mimics potential differences in patient positioning, imaging angles, and scale. Additionally, custom augmentation functions for brightness adjustment and Gaussian noise ensure the model is exposed to variations in image illumination and noise levels, further enhancing its ability to generalize to diverse and realistic chest X-ray scenarios.

Incorporating these augmentation details provides a comprehensive view of the strategies employed to enrich the training data. The augmented dataset, reflecting a broad spectrum of real-world scenarios, contributes significantly to the algorithm's overall performance and reliability. By training on a diverse and augmented dataset, the algorithm is better equipped to generalize its learnings, making it more robust and capable of delivering reliable results when applied to a wide range of chest X-ray images in clinical settings. The emphasis on capturing true positives is reinforced through these augmentation strategies, enhancing the algorithm's clinical utility and effectiveness.

Display of augmented data sample:




No Pneumonia
No Pneumonia
No Pneumonia
Pneumonia
No Pneumonia
Pneumonia
Pneumonia
Pneumonia
Pneumonia
No Pneumonia
Pneumonia
No Pneumonia
Pneumonia
Pneumonia
No Pneumonia
No Pneumonia

CNN Architecture:

VGG16 architecture with weights trained on the ImageNet dataset. The last layers were fine-tuned for pneumonia classification.

3. Algorithm Training

The model architecture is based on the VGG16 architecture with fine-tuning for pneumonia classification. Training parameters, including optimizer, loss function, and metrics, were carefully selected for optimal performance. The training process over ten epochs demonstrates the iterative improvement of the model.

Model Architecture:
- Base Model: VGG16 architecture with weights trained on the ImageNet dataset.
- Layers up to 'block5_pool' were frozen, while the last layers were fine-tuned for pneumonia classification.

Custom Model Architecture (Sequential):

- Flattening layer with 25088 output nodes.
- Dropout layer with a dropout rate of 0.5.
- Dense layer with 1024 neurons and 'relu' activation.
- Dropout layer with a dropout rate of 0.5.
- Dense layer with 512 neurons and 'relu' activation.
- Dropout layer with a dropout rate of 0.5.
- Dense output layer with 1 neuron and 'sigmoid' activation.

Training Parameters:
- Total Trainable Parameters: 28,576,257
- Non-trainable Parameters: 12,354,880
- **Optimizer**: Adam with a learning rate of 0.0001.
- **Loss Function**: Binary cross-entropy.
- **Metrics**: Binary accuracy.

Training Process (Epochs 1-10):
- Epochs: 10
- Batch Size: 20
- Training Time per Epoch: Varies, approximately 8-13 seconds per epoch.

- **Checkpointing**: ModelCheckpoint monitors validation loss and saves the best weights.
- **Early Stopping:** Monitors validation loss and stops training if no improvement after 20 epochs.

Training Performance Visualization

```
input_3 False
block1_conv1 False
block1_conv2 False
block1_pool False
block2_conv1 False
block2_conv2 False
block2_pool False
block3_conv1 False
block3_conv2 False
block3_conv3 False
block3_pool False
block4_conv1 False
block4_conv2 False
block4_conv3 False
block4_pool False
block5_conv1 False
block5_conv2 False
block5_conv3 True
block5_pool True
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
model_3 (Model)              (None, 7, 7, 512)         14714688
_________________________________________________________________
flatten_3 (Flatten)          (None, 25088)             0
_________________________________________________________________
dropout_7 (Dropout)          (None, 25088)             0
_________________________________________________________________
dense_7 (Dense)              (None, 1024)              25691136
_________________________________________________________________
dropout_8 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_8 (Dense)              (None, 512)               524800
_________________________________________________________________
dropout_9 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_9 (Dense)              (None, 1)                 513
=================================================================
Total params: 40,931,137
Trainable params: 28,576,257
```

Non-trainable params: 12,354,880

Epoch 1/10
20/20 [==============================] - 13s 625ms/step - loss: 0.9812 - binary_accuracy: 0.5150 - val_loss: 0.6299 - val_binary_accuracy: 0.6900

Epoch 00001: val_loss improved from inf to 0.62991, saving model to xray_class_my_model.best.hdf5
Epoch 2/10
20/20 [==============================] - 9s 432ms/step - loss: 0.8766 - binary_accuracy: 0.5425 - val_loss: 0.7786 - val_binary_accuracy: 0.2800

Epoch 00002: val_loss did not improve from 0.62991
Epoch 3/10
20/20 [==============================] - 9s 447ms/step - loss: 0.8157 - binary_accuracy: 0.5325 - val_loss: 0.6907 - val_binary_accuracy: 0.5200

Epoch 00003: val_loss did not improve from 0.62991
Epoch 4/10
20/20 [==============================] - 10s 485ms/step - loss: 0.8303 - binary_accuracy: 0.5275 - val_loss: 0.6820 - val_binary_accuracy: 0.5400

Epoch 00004: val_loss did not improve from 0.62991
Epoch 5/10
20/20 [==============================] - 11s 535ms/step - loss: 0.7792 - binary_accuracy: 0.5508 - val_loss: 0.7053 - val_binary_accuracy: 0.5000

Epoch 00005: val_loss did not improve from 0.62991
Epoch 6/10
20/20 [==============================] - 10s 477ms/step - loss: 0.7145 - binary_accuracy: 0.5725 - val_loss: 0.6902 - val_binary_accuracy: 0.5100

Epoch 00006: val_loss did not improve from 0.62991
Epoch 7/10
20/20 [==============================] - 9s 475ms/step - loss: 0.7227 - binary_accuracy: 0.5550 - val_loss: 0.8105 - val_binary_accuracy: 0.2600

Epoch 00007: val_loss did not improve from 0.62991
Epoch 8/10
20/20 [==============================] - 9s 457ms/step - loss: 0.7338 - binary_accuracy: 0.5250 - val_loss: 0.7457 - val_binary_accuracy: 0.3600

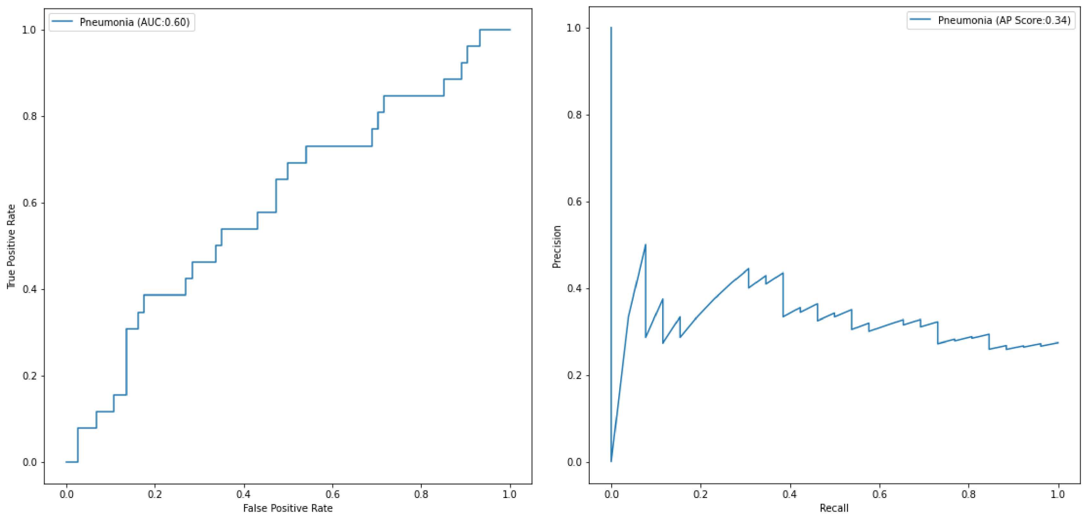
Epoch 00008: val_loss did not improve from 0.62991
Epoch 9/10
20/20 [==============================] - 10s 522ms/step - loss: 0.6879 - binary_accuracy: 0.5736 - val_loss: 0.6805 - val_binary_accuracy: 0.5600

Epoch 00009: val_loss did not improve from 0.62991
Epoch 10/10

20/20 [==============================] - 8s 423ms/step - loss: 0.7016 - binary_accuracy: 0.5825 - val_loss: 0.7771 - val_binary_accuracy: 0.3900

Epoch 00010: val_loss did not improve from 0.62991
100/100 [==============================] - 1s 8ms/step

Precision-recall curve



Final Threshold and Explanation:

Using the previous performance metrics and given the emphasis on maximizing true positives, the threshold for classification has been set at 0.40 during training, reflecting the priority of capturing as many actual positive cases of pneumonia as possible. This choice is reflected in the performance metrics, which prioritize higher recall at the expense of precision for identifying pneumonia cases.

This threshold allows to get:

- Accuracy: 0.452
- Precision: 0.310
- Recall: 0.692
- F1 Score: 0.429

4. Databases

Description of the given dataset:

- The dataset comprises 112,120 frontal-view chest X-ray images with an average age of 46.9 years.
- A balanced gender distribution with 63,340 males and 48,780 females.
- The majority of images are of the posterior-anterior (PA) view (67,310), with the remaining in the anterior-posterior (AP) view (44,810).
- Pneumonia-positive cases account for 1,431 instances, with an average age of 44.93 years.
- A notable ratio (0.39) of pneumonia cases occurred in individuals under 40 years old, emphasizing the algorithm's relevance across age groups.
- A balanced ratio of males (838) and females (593) were diagnosed with pneumonia, highlighting gender-inclusive performance.

The given dataset was split into a training and validation dataset with a ratio of 80/20.

Description of Training Dataset:

- The training dataset has a balanced representation of pneumonia-positive and pneumonia-negative cases (51% of pneumonia-positive cases).
- It has also a balanced representation of gender. The EDA shows that pneumonia touches men as much as women (57% of men).
- The EDA also shows that 60% of older people [40-80] are diagnosed with pneumonia. The training dataset represents this observation. (60% of people older than 40years old)
- The training dataset includes 50% of PA/AP.

Description of Validation Dataset:

- The validation dataset has 60% of people older than 40 years old.
- It has also a balanced representation of gender. The EDA shows that pneumonia touches men as much as women (55% of men).
- Non-pneumonia cases are 4 times bigger than the pneumonia cases to reflect the reality observed during the EDA phase. (20% of pneumonia cases)
- The validation dataset includes 57% of PA/AP.

5. Ground Truth

Ground truth labels were generated using Natural Language Processing (NLP) applied to radiological reports. The labeling accuracy is estimated to be >90%.
There are 14 common thoracic pathologies, including pneumonia.

6. FDA Validation Plan

Patient Population Description for FDA Validation Dataset:

The FDA validation dataset will encompass a diverse range of patient populations, including different demographics, comorbidities, and clinical scenarios. The dataset will be carefully selected to represent real-world variations in chest X-ray images.

Ground Truth Acquisition Methodology:

Ground truth labels for the FDA validation dataset will be obtained through an independent review by expert radiologists. These experts will assess the presence or absence of pneumonia in each chest X-ray image.

Algorithm Performance Standard:

The algorithm's performance will be evaluated based on a set of metrics, including sensitivity, specificity, positive predictive value (PPV), negative predictive value (NPV), and F1 score. The chosen performance metrics align with the goal of maximizing true positives while considering the trade-off with false positives. The standard for success is achieving a sensitivity and specificity comparable to or exceeding human-reader-level F1 scores for pneumonia detection.

Given the emphasis on maximizing true positives, the algorithm will be validated at the chosen threshold of 0.40. This threshold ensures a higher recall rate, prioritizing the identification of pneumonia cases. It is recognized that this choice may result in a certain level of false positives, and ongoing monitoring of false positives' real-world impact will be integral to the evaluation process.