FDA Submission

Your Name:: Kofi Ofuafor

Name of your Device:: Pneumonia detection algorithm

Algorithm Description

1. General Information

Intended Use Statement: This algorithm is intended for use in *assisting the clinician* in acute care setting with a speedy detection of pneumonia within a chest x-ray. The predicate device for this algorithm is a CADx device.

Indications for Use: This algorithm is used as a Pneumonia classifier in a chest x-ray taken in acute medical or surgical admission units, accident and emergency and acute care wards. It is intended for use in people aged 20-70 years with *no prior history* of presence of edema, pleural effusion and pulmonary infiltrations

Device Limitations: The device study was based on images from people of age 20- 70 and was significantly less accurate in the presence of concurrent disease and therefore not recommended for use in the catgory of patients were there is prior presence of disease in a coloumn num_prior_positive in the demographic data. The algorithm was not traing in patients over 80 years old.

Clinical Impact of Performance:

- Performance statistics
    - Accuracy: 81%
    - Precision: 0.16666666666666666
    - Sensitivity: 1
    - Specificity: 0

2. Algorithm Design and Function

< Insert Algorithm Flowchart >> The check_dicom function reads in a .dcm file, checks the important fields for our device, and returns a numpy array of just the imaging data.

The function, check_dicom(), is required to check the image type(MODALITY), Body Part Examined (Chest), and Image position (PA or AP) for each DICOM image and check if the input to the algorithm is valid to be predicted by our algorithm or not.

DICOM Checking Steps: There are three steps:

1. Extract the desired attributes *Modality: DX The image type DX is a digital radiography *Body Part Examined: Must be Chest *Patient Position: AP or PA The image position could be AP or PA *Study Description: label This is the label for the chest x-ray.

2. Next the check_dicom function checks each image if the input to the algorithm is valid to be predicted upon by our device.

3. Finally, valid images are included as input and a numpy array of the image data is returned.

Preprocessing Steps: The preprocess_image function used here takes in the image value returned from check_dicom, and the image mean, standard deviation as well as image size corresponding with that of VGG16 input image size of (1,224,224, 3). These inputs are used to standardize and resize the image which is returned as a processed image.

Image augmentation techniques were also applied prior to training the model especially to training data.

CNN Architecture: The pretrained model was loaded and the first 17 layers were rendered untrainable Here is the pretrained model

Loaded Model

Seven fully connected layers were added to the pretrained model as classifiers. Of the 7 layers, three were Dropout layers to prevent overfitting.

My model

3. Algorithm Training

Parameters:

- Types of augmentation used during training
  - horizontal_flip = True,
  - vertical_flip = False,
  - height_shift_range= 0.1,
  - width_shift_range=0.1,
  - rotation_range=20,
  - shear_range = 0.1,
  - zoom_range=0.1)
- Batch size
  - Training: 16
  - Validation: 32
  - Performance evaluation: 100
- Optimizer learning rate
  - Adam
  - Learning rate: 5e-6. For 5 trainable layers. Decaying rate per epoch: 0.5

Layers of pre-existing architecture that were frozen 17 Layers of pretrained model were frozen and therefore not trainable

Frozen Arhitecture

Layers of pre-existing architecture that were fine-tuned * Fully connected layers were fine-tuned.

Layers added to pre-existing architecture

One one convulated was layers were fine tune and four layres were added for training

Fine-Tuned Layers

Algorithm training performance visualization

Train and validation loss/accuracy versus epochs.

Training performance

The plot of how the model distinguih between the classes.

Roc curve

This is plot of two important parameters for the calculation of F1 Score.

Precision recall curve

Final Threshold and Explanation:

- Best_threshold = 40 The decision model for this threshold was based the calculation two thresholds, from F1 scored weighted by a precision value of 0.8 and recall value of 0.8.

The threshold value from the for recall of 0.8 was chosed because, it is important for the algorithm to pick all postive cases of pneumonia. Missing cases of pneumonia is not accepetable. Recall is more valuable under the circumstance where a negative test means you have a high chance of not having the disease.

4. Databases

(For the below, include visualizations as they are useful and relevant)

Description of Training Dataset:

Data set was split 80:20 percent in favour of training data By using the len of positive cases of pneumonia to randomly select equal number of negative cases.

Training data after split with 4:1

Sizes of train and validation data after discarding excess data

Training data after discarding excess negative data

Description of Validation Dataset:

Validation after split 80:20 with 20% of the entire data in the validation dataset By finding using random sample to add a negative data to the in the ration of 1:4 postive to negative cases

Validation data after split

Validation set must contain 20:80 ratio (positive to negative)

Validation data after split

5. Ground Truth

The dataset were labeled by qualified radiologist. In AI for medical imaging, using a radiologist's labels as our "true" label is often the standard of practice, and the algorithm performance is judged based on performance against an expert human.

6. FDA Validation Plan

Patient Population Description for FDA Validation Dataset:

Gender distribution

gender

Ags Distribution

Age

Concurrent disease

comorbidity

Ground Truth Acquisiion methodology The label was obtained by 3 radiologist by vote. The approach of using radiologist was optimal for my project becuase the data is from NIH.

Algorithm Performance Standard: Recall weighted threshold, was used because It is crucial not to miss positive cases.