# SIIM-FISABIO-RSNA COVID-19 Classification Report

## 1. Problem Statement

Five times more deadly than the flu, COVID-19 causes significant morbidity and mortality. COVID-19 looks very similar to other viral and bacterial pneumonias on chest radiographs, which makes it difficult to diagnose. It can be diagnosed via polymerase chain reaction to detect genetic material from the virus or chest radiograph. However, it can take a few hours and sometimes days before the molecular test results are back. By contrast, chest radiographs can be obtained in minutes. We want to help radiologists diagnose millions of COVID-19 patients more confidently and quickly using the chest radiographs. This will also enable doctors to see the extent of the disease and help them make decisions regarding treatment. Therefore, we are trying to develop a computer vision model to identify the COVID-19 abnormalities in an efficient and quick way.

In this project, I tried training different supervised deep learning models to classify the images. The main tools I used are tensorflow, seaborn, pandas, numpy, matplotlib etc. The models I developed would accept the image data and output the prediction of the image class.

For the criteria of success, my goal is to achieve at least 90% accuracy and a f1-score of 0.80.

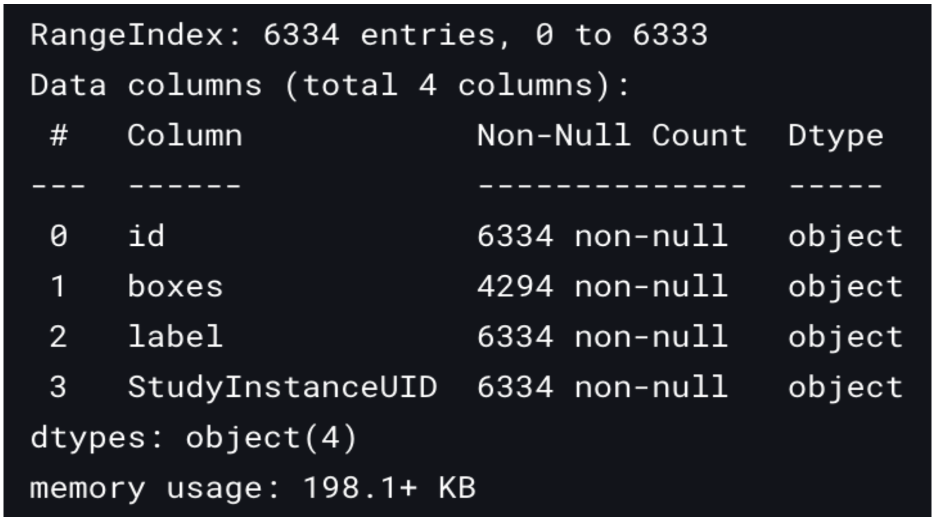
## 2. Data Wrangling

In this project, we have three data sets. There are two datasets in csv format, train_image_level.csv and train_study_level.csv. The last one is image data in dcm format.

For the file train_image_level.csv, there 6334 observations and 4 columns. (See Figure 2-1) The first column ‘ id’ is the unique id of each image in the image data. The second column, ‘boxes’ includes the coordinate of box(es) with abnormality identified by doctors. Some images have no box, while others have one or more boxes. There are 2040 None in this column. They are represented by a None or a dictionary with x,y coordinate and width and height of the box in numerical format. The ‘label’ column is the labels and the box(es) coordinates of the boxes. The ‘StudyInstanceUID’ is the study level id of the image.

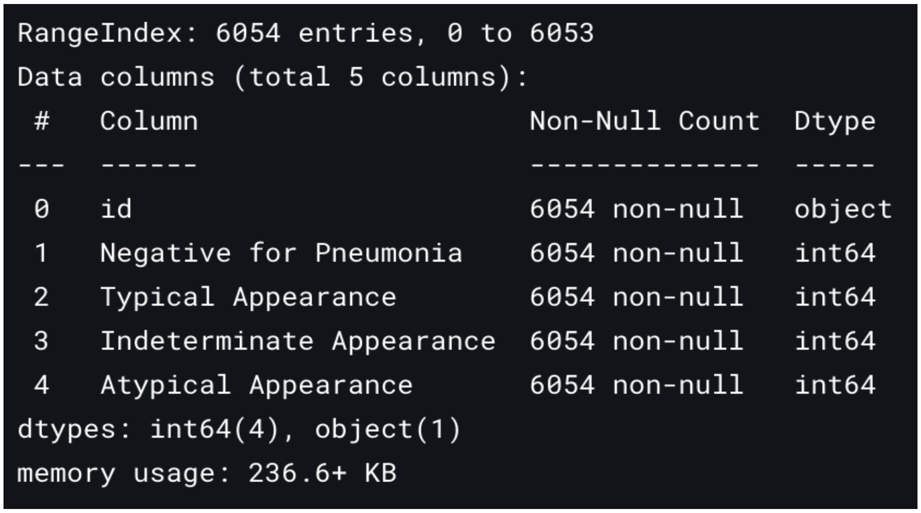
In the train_study_level.csv file, there are 6045 observations and 5 columns. (See Figure 2-2) The first column ‘id’ is the unique study level id, which is corresponding to the ‘StudyInstanceUID’ in the train_image_level.csv file. The second to fifth columns are ‘Negative for Pneumonia’, ‘Typical Appearance’, ‘Indeterminate Appearance’, ‘Atypical Appearance’, representing the class of each observation. One observation is corresponding to one class, for the class it belongs to, it is labeled 1 otherwise 0.

In the image dataset, there are 6334 images, and the name of the images are corresponding to the image level id. All the images are in DCM format. A DCM file is an image file saved in the Digital Imaging and Communications in Medicine (DICOM) image format. It stores a medical image, such as a CT scan or ultrasound, and may also include patient information to pair the image with the patient.



```
RangeIndex: 6334 entries, 0 to 6333
Data columns (total 4 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   id                6334 non-null   object
 1   boxes             4294 non-null   object
 2   label             6334 non-null   object
 3   StudyInstanceUID  6334 non-null   object
dtypes: object(4)
memory usage: 198.1+ KB
```

Figure 2-1



```
RangeIndex: 6054 entries, 0 to 6053
Data columns (total 5 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   id                        6054 non-null   object
 1   Negative for Pneumonia    6054 non-null   int64 
 2   Typical Appearance        6054 non-null   int64 
 3   Indeterminate Appearance  6054 non-null   int64 
 4   Atypical Appearance       6054 non-null   int64 
dtypes: int64(4), object(1)
memory usage: 236.6+ KB
```

Figure 2-2

## 3. Exploratory Data Analysis

a. The distribution of the class of the image is imbalanced. (See Figure 3-1) There are only 8% images in ‘Atypical Appearance’, and the class ‘Typical Appearance’ has about 47%.

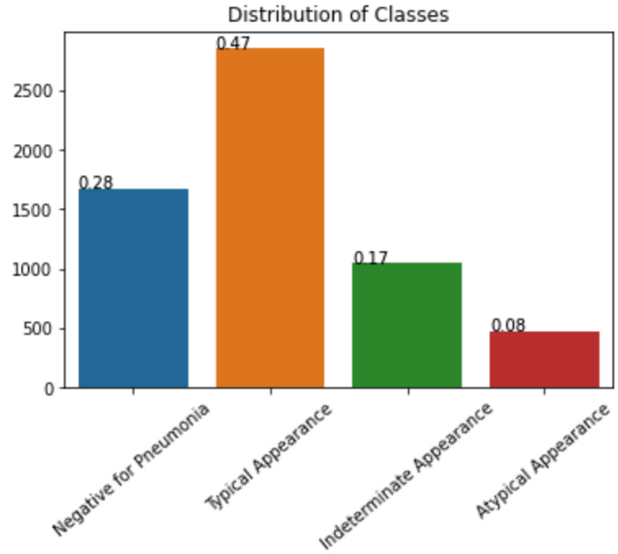


Figure 3-1

b. The distribution of the four classes with labels is the following. (See Figure 3-2) All observations in ‘Negative for Pneumonia’ don’t have a label (or box).

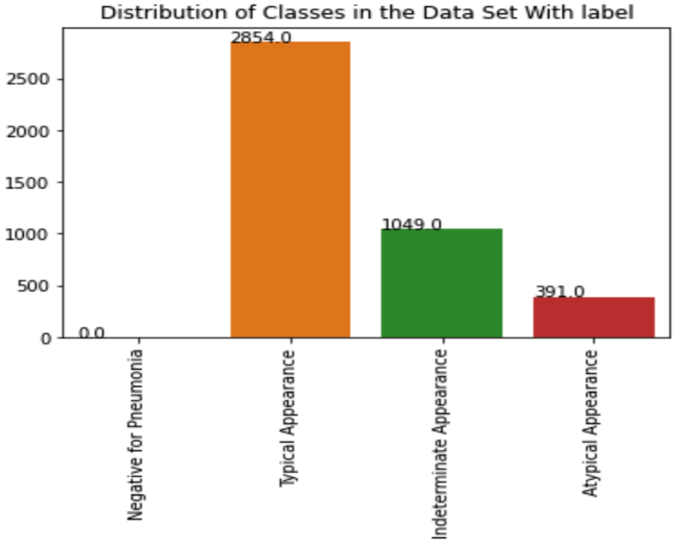


Figure 3-2

c. The distribution of the four classes without labels is shown below. (See Figure 3-3) There are some images that are not class ‘Negative for Pneumonia’(not normal) with no label (or box). Total of them is 304.

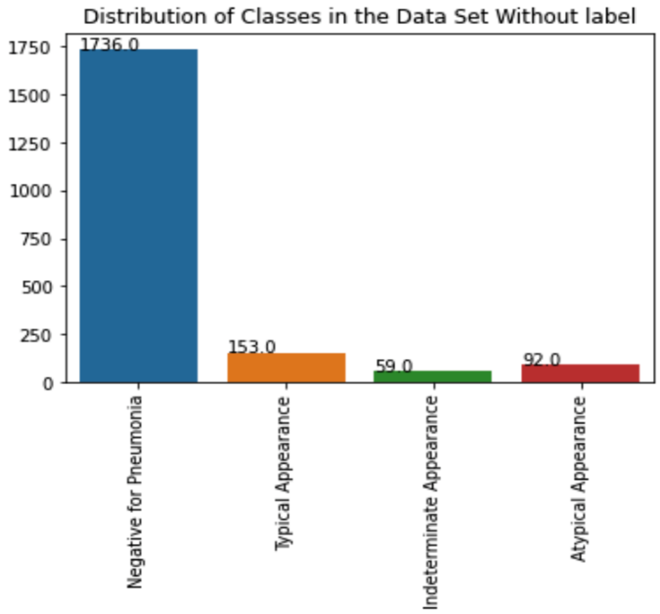
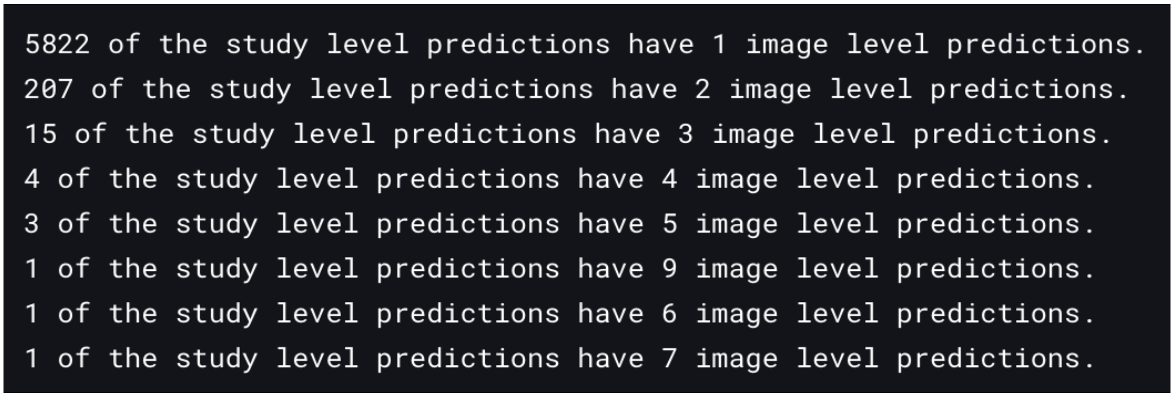


Figure 3-3

d. There are 6054 unique study level id and 6334 unique image level id. There are 232 study-level predictions that have more than 1 image level prediction. There 512 of the image level predictions share 232 study level ids, and the rest of 5822 image level predictions have their own unique study level id. (See Figure 3-4)



```
5822 of the study level predictions have 1 image level predictions.
207 of the study level predictions have 2 image level predictions.
15 of the study level predictions have 3 image level predictions.
4 of the study level predictions have 4 image level predictions.
3 of the study level predictions have 5 image level predictions.
1 of the study level predictions have 9 image level predictions.
1 of the study level predictions have 6 image level predictions.
1 of the study level predictions have 7 image level predictions.
```

Figure 3-4

e. For those image level studies that share the same study level id, the images are the same. We need to remove the duplicated images. The following is one example. (See Figure 3-5)

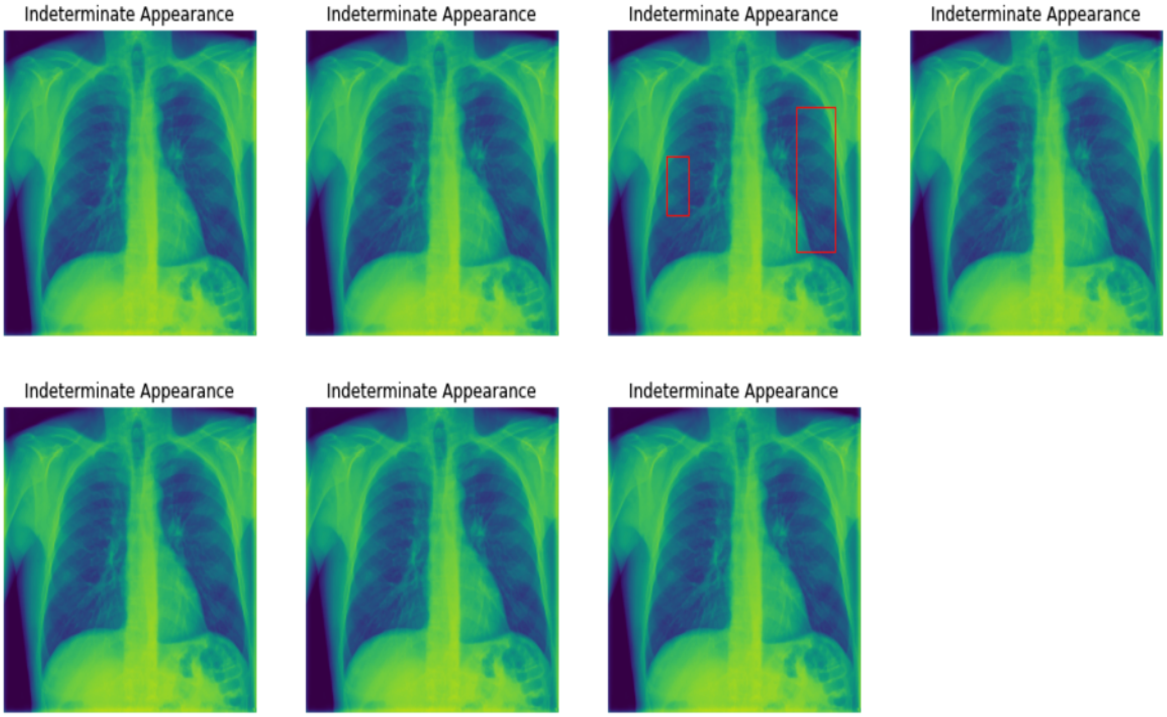
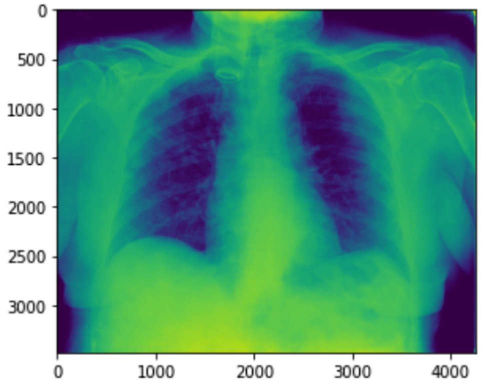


Figure 3-5

After removing the duplicated images, there are 6054 images left, which is the same as the number of study level observations.

## 4. Pre-processing & Training Data Development

The original image data is in DCM format, it is about 100GB. In order to make it easier for later use, I transformed the image data from DCM into PNG format. Here is one image selected randomly from the original data set, and the pixel value distribution. (See Figure 4-1 and 4-2)



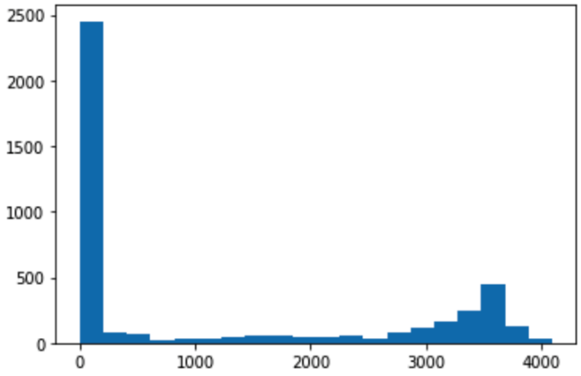
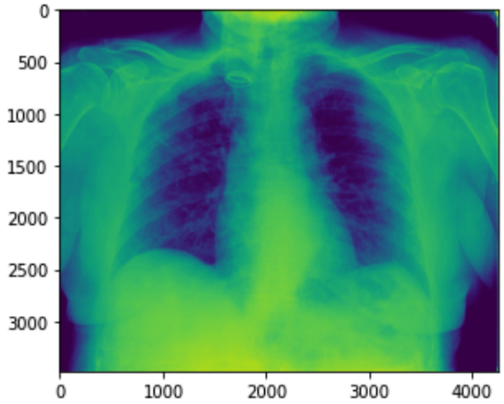


Figure 4-1 Figure 4-2

As we can see from the distribution plot above, the range of the pixel values is about 0-4000. I rescaled the pixel values of the image into 0-255. The following plots are the same observation as above after rescaling. (See Figure 4-3, 4-4)



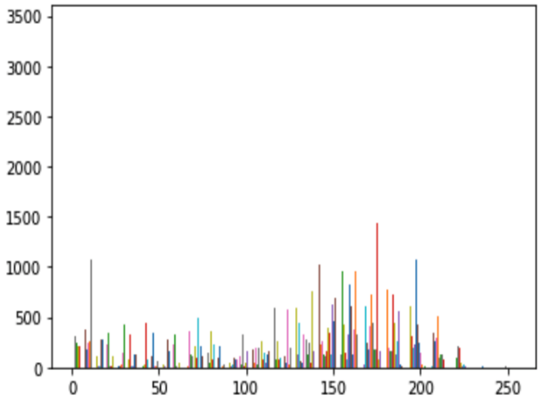


Figure 4-3 Figure 4-4

With pixel values rescaled, the image looks the same as Figure 4-1.

After the image format transformation and pixel values rescaling, I split the image data set into two groups, 80% of training data and 20% of validation data.

## 5. Models Training and Hyperparameters Tuning

Based on the coarse tuning, it is better to set the learning rate between 1e-4 and 1e-5. A bigger learning rate can make the loss and accuracy of the training and validation set very unstable. The batch size of 32 is too small, it also leads to quite large fluctuations. Therefore, in this project, I mainly set the batch size to 64. Here are some output examples of different models with different hyperparameters.

A. Vgg16, Image size: 224*224*1, lr = 5e-5, batch size =64, Epoch =50

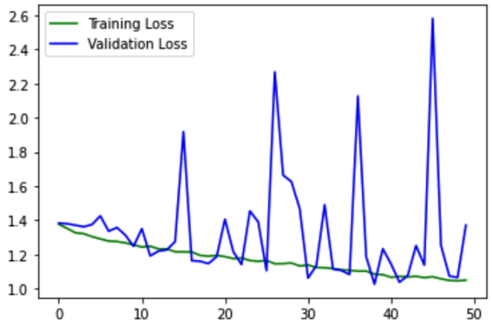


Figure 5-1

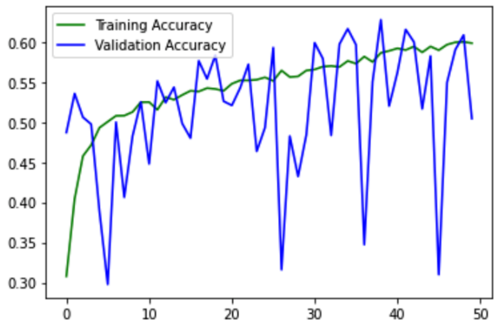


Figure 5-2

The training loss and accuracy are changing steadily, while the validation loss and accuracy are fluctuating wildly in the first 50 epochs.

B. Vgg16, Image size: 224*224*1, lr = 1e-5, batch size =64, Epoch =50

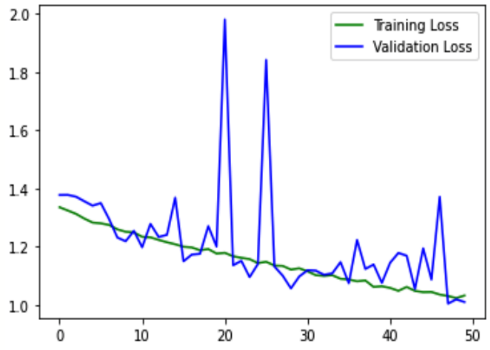


Figure 5-3

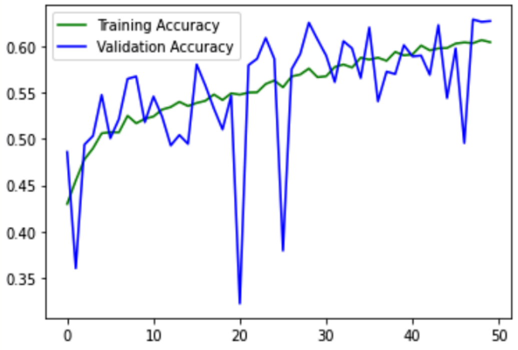


Figure 5-4

This time, the same model and same image size as the previous one, but the learning is 1e-5, which is 5 times smaller than the previous one. The training loss and accuracy are still changing steadily. Validation loss and accuracy are also fluctuating in the first 50 epochs, but it looks slightly better than the previous one.

C. Vgg16, Image size: 224*224*1, lr = 1e-5, batch size = 64, Epoch =130

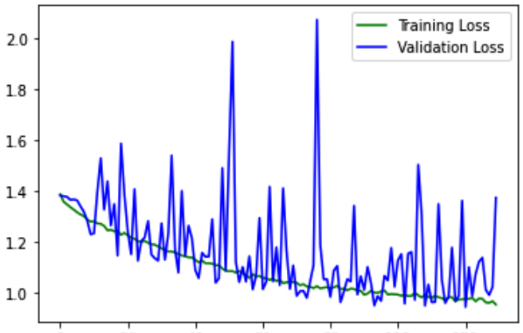


Figure 5-5

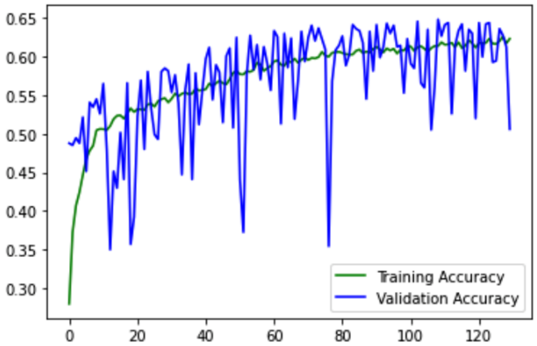


Figure 5-6

For this example, the hyperparameters are the same as the previous model, but I keep training 100 more epoches on it. The validation results are still very unstable for the later training(epoch>50). At about epoch=90, the model starts overfitting because the trend of the validation loss is starting to go up. The highest validation accuracy is about 65%.

D. Resnet50, Image size: 224*224*1, lr = 1e-5, batch size =64, Epoch=50

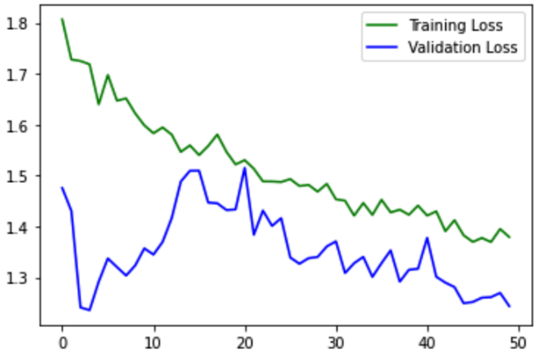


Figure 5-9

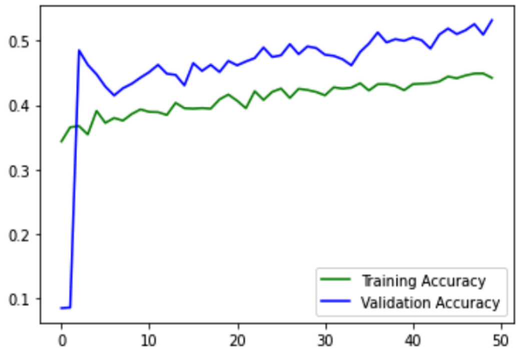


Figure 5-10

In this experiment, I trained a Resnet50 model. The hyperparameters are the same as the previous Vgg16 model. The validation results are more stable, but the learning rate 1e-5, the model is improving too slowly.

E. Resnet50, Image size: 224*224*1, lr = 5e-5, batch size =64,  Epoch=50

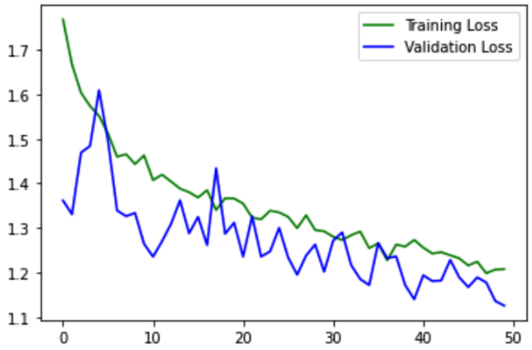


Figure 5-11

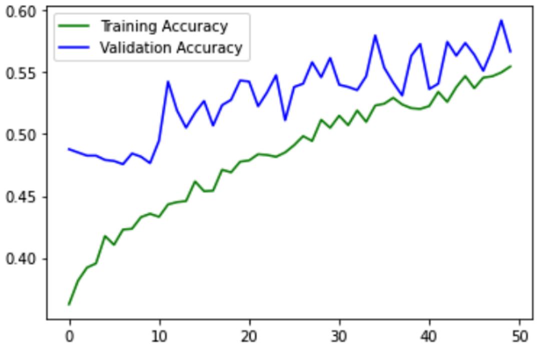


Figure 5-12

Based on the result of the previous Resnet50, I set the learning rate 5e-5 instead of 1e-5. This time, the model improves faster than the previous one, but the model is still underfitting.

F. Resnet50, Image size: 224*224*1, lr = 3e-5, batch size =64, Epoch=100

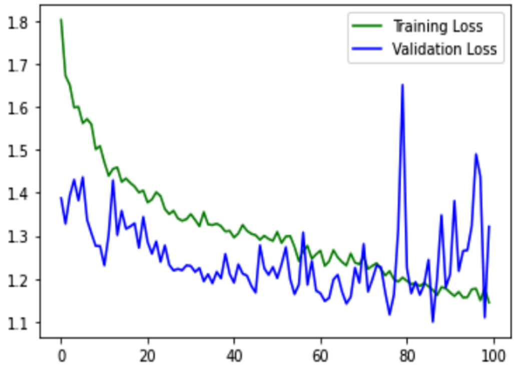


Figure 5-13

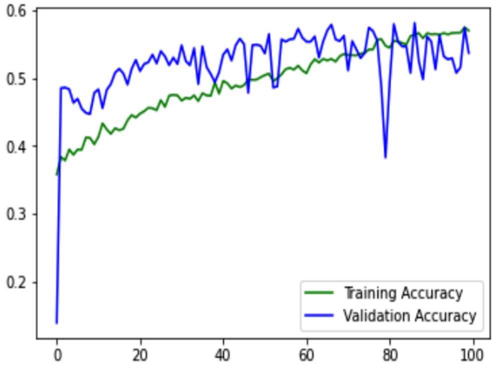


Figure 5-14

This time, I trained the Resnet50 model for 100 epochs, and set the learning rate to 3e-5, which is slightly smaller than the previous example. The training loss and accuracy are changing in a stable way. The validation loss and accuracy are also more stable than the Vgg16 compared to the  Figure 5-5 and  Figure 5-6. However, the highest accuracy is only 58%, which is about 7% less than the best Vgg16. Resnet50 is a larger model which is deeper than Vgg16. The lower accuracy may be because it extracts too many features(or too much noise is fitted in the model).Therefore, it leads to more misclassification.

G. Pre-trained Vgg16 ,Image size: 224*224*3, lr = 5e-4, batch size =64

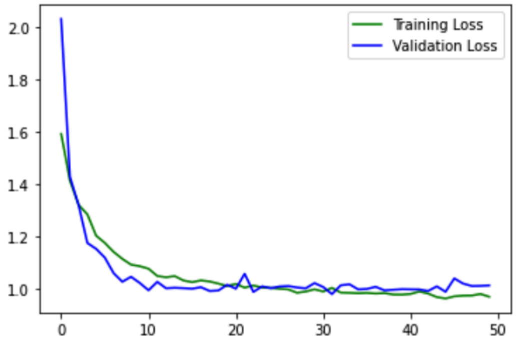


Figure 5-15

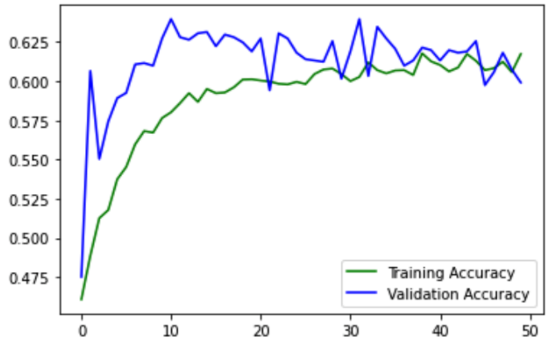


Figure 5-16

This is a Vgg16 pre-trained model. Here, I transformed each image into a 3-channel image, in which each channel is the same. Using this model required 2 times more calculations than the non pre-trained Vgg16 model in each epoch. The result of the validation loss and accuracy are more stable than the other two (non pre-trained Vgg16 and Resnet50) Besides, the highest accuracy of it is 0.64.

Finally, from the training experiments C, F, G, I saved the best models using the metric ROC-AUC. The following section will be the result about accuracy, f1-score etc. for each model.

## 6. Model Comparison and Selection

After a series of model training, I use the AUC of ORC as the metric to select the best weights of each model. The best AUC score of non pre-trained Vgg16 is 0.8455, and its corresponding accuracy is 0.6441; non pre-trained Resnet50 is 0.7832, and its corresponding accuracy is 0.5816; pre-trained Vgg16 is 0.8332, and its corresponding accuracy is 0.6397.

The following plot are the confusion matrices for each model with their best result during training. (See Figure 6-1, 6-2, 6-3) As we can see, all the models do not detect 'Atypical Appearance' and 'Indeterminate Appearance' well. For the non pretrained Vgg16 and non pretrained Resnet50 model, more than 97% of the images are classified as 'Negative for Pneumonia' and 'Typical Appearance'. In fact, 25% of images are in class 'Atypical Appearance' and 'Indeterminate Appearance'. More than 22% of images are predicted as

‘Typical Appearance’ and at least 11% of images are identified as ‘Negative for Pneumonia’ incorrectly.

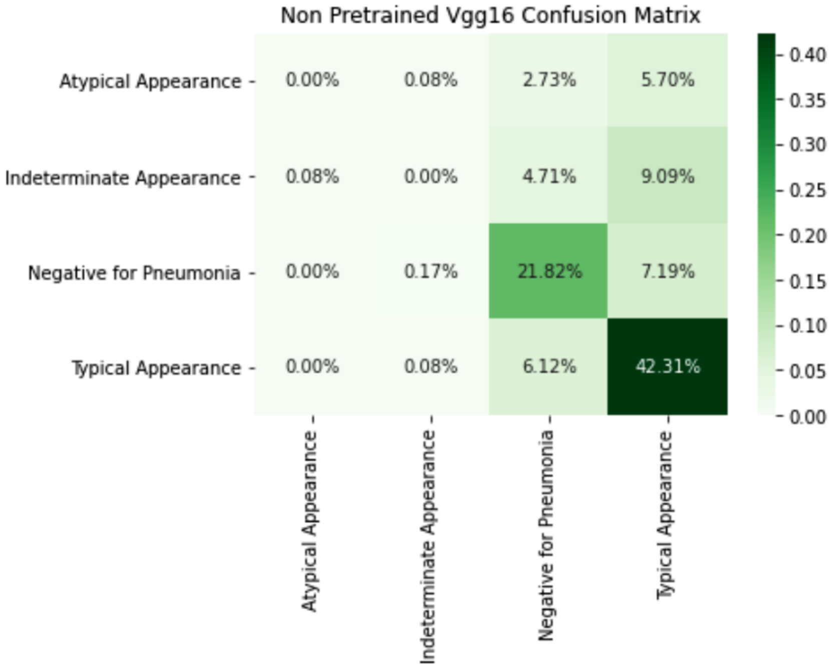


Figure 6-1

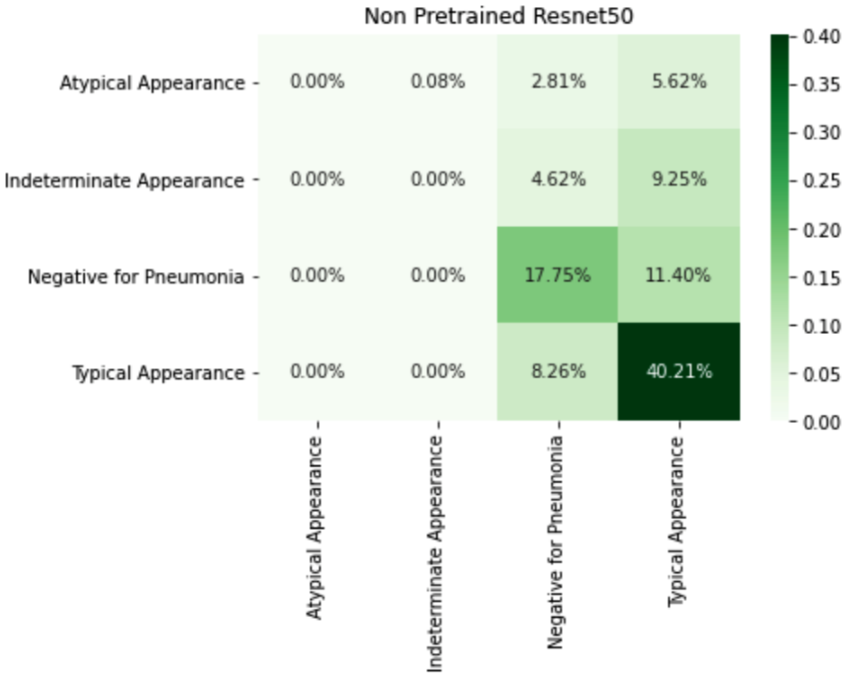


Figure 6-2

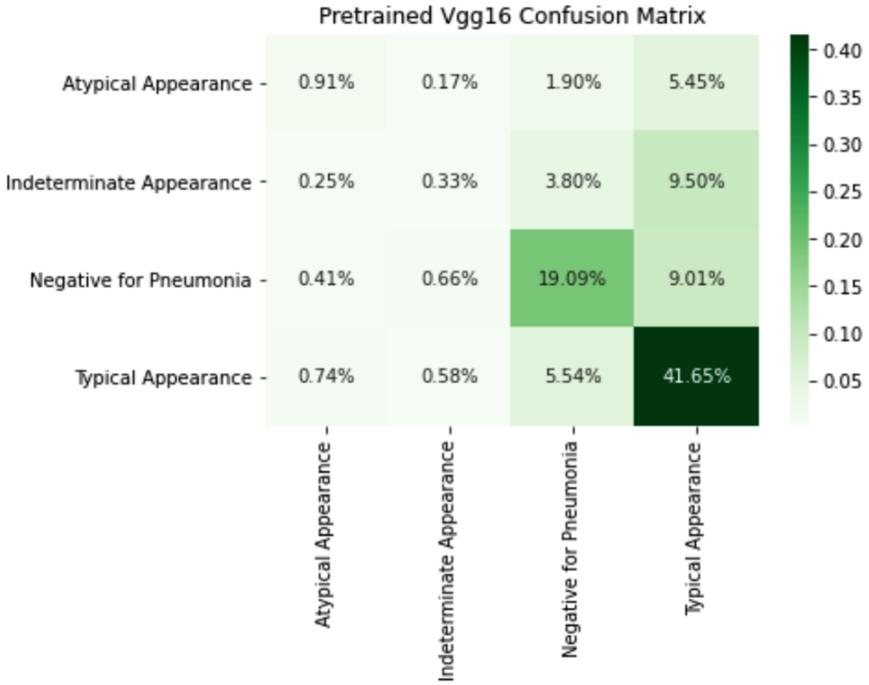
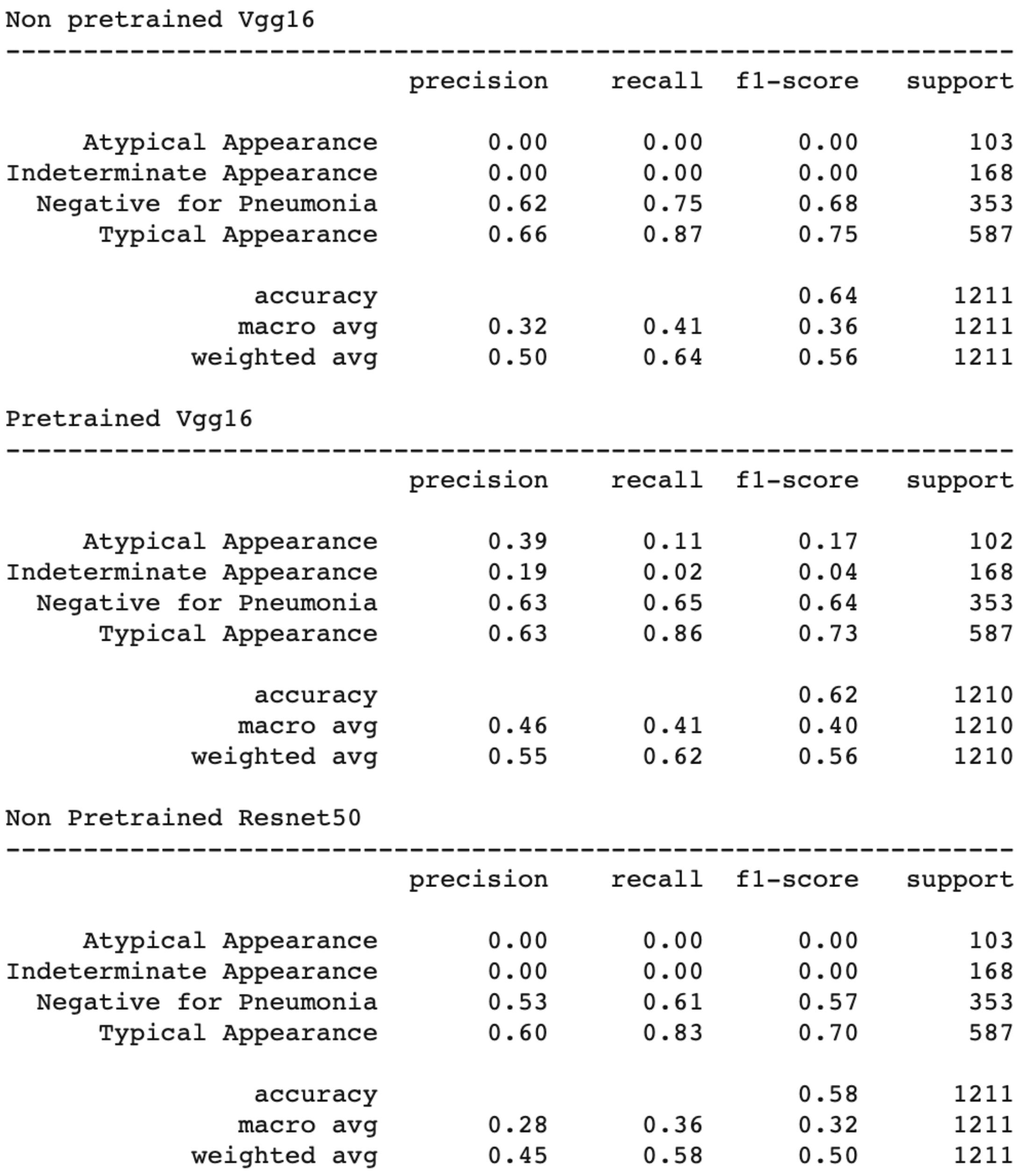


Figure 6-3

The table below shows the information of the f1-score, accuracy and other information of each model.(See Figure 6-4) Both the non pretrained Vgg16 and non pretrained Resnet50 have f1-score on class ‘Atypical Appearance’ and ‘Indeterminate Appearance’ 0, which means that they can not detect any image in those two classes. The f1-score for those two classes predicted by pretrained Vgg16 are 0.17 and 0.04. For class ‘Negative for Pneumonia’ and ‘Typical Appearance’, the non pretrained Vgg16 has the highest f1-score on them, they are 0.68 and 0.75. The non pretrained Vgg16 also has the highest accuracy on predicting the image.

According to these data, the best model for this data set would be Vgg16 with its pretrained weights. Because of data imbalance, I used f1-score as the main metric to evaluate the model. Even though the pretrained Vgg16 has a slightly lower accuracy compared to the non pretrained Vgg16, its macro average f1-score is higher than the other two.



```
Non pretrained Vgg16
-----------------------------------------------------------------
                          precision    recall  f1-score   support

     Atypical Appearance       0.00      0.00      0.00       103
Indeterminate Appearance       0.00      0.00      0.00       168
  Negative for Pneumonia       0.62      0.75      0.68       353
    Typical Appearance       0.66      0.87      0.75       587

              accuracy                           0.64      1211
             macro avg       0.32      0.41      0.36      1211
          weighted avg       0.50      0.64      0.56      1211

Pretrained Vgg16
-----------------------------------------------------------------
                          precision    recall  f1-score   support

     Atypical Appearance       0.39      0.11      0.17       102
Indeterminate Appearance       0.19      0.02      0.04       168
  Negative for Pneumonia       0.63      0.65      0.64       353
    Typical Appearance       0.63      0.86      0.73       587

              accuracy                           0.62      1210
             macro avg       0.46      0.41      0.40      1210
          weighted avg       0.55      0.62      0.56      1210

Non Pretrained Resnet50
-----------------------------------------------------------------
                          precision    recall  f1-score   support

     Atypical Appearance       0.00      0.00      0.00       103
Indeterminate Appearance       0.00      0.00      0.00       168
  Negative for Pneumonia       0.53      0.61      0.57       353
    Typical Appearance       0.60      0.83      0.70       587

              accuracy                           0.58      1211
             macro avg       0.28      0.36      0.32      1211
          weighted avg       0.45      0.58      0.50      1211
```

Figure 6-4

## 7. Conclusion and Final Thought

- None of these models can classify the image data set well. The highest accuracy is lower than 65%. 'Atypical Appearance' and 'Indeterminate Appearance' images are mainly identified as 'Typical Appearance'. One main reason may be because the image data set is too small. Another reason may be that the images themselves are very similar to each even though they are in different classes.

- Based on the criteria of success at the beginning of the project, the best model failed to achieve the 90% accuracy and 0.80 f1-score.
- The reason for failure may be because of the lack of image data or the images in different classes are too similar to each other, but I need further research for these guesses.
- For further research, the data provider Foundation for the Promotion of Health and Biomedical Research of Valencia Region should provide more image data, specially the images in classes ‘Atypical Appearance’ and ‘Indeterminate Appearance’ so that we can make sure how to improve the model, or why the Vgg and Resnet models can not classify the images well.