Brett Gordon

Final Report:
COVID-19 Radiography Dataset Analysis

1. Introduction

Problem Statement

Whether it was inevitable or not, it is evident that COVID-19 is endemic and a part of our everyday life, like the flu and other viral ailments. With that in mind, it is important to recognize if a patient has COVID or another illness when administered to a hospital. One of the easiest ways to primary ways to confirm the illness is with a chest X-ray as many of these illnesses affect the lungs and show physical damage. Our goal is to create a reliable model that can predict if a patient can be categorized as suffering from COVID-19, COPD, pneumonia or if their lungs are healthy.

Background

Over the past three years, the world battled the COVID-19 pandemic and despite awareness, precautions, and multiple vaccine options, COVID-19 is going nowhere. As a result it is best for us to prepare to learn how best to deal with COVID-19 and how to quickly diagnose it. Too often, and especially in countries with limited medical benefits, patients are misdiagnosed and treated incorrectly. This can lead to slower recovery times or even be fatal. If a model can be created to accurately and quickly predict what ails the patient, there can be quicker treatment and recovery. This can help lower the fatality rate of COVID-19 along with other viruses such as pneumonia and COPD (Lung Opacity).

Goal

Our goal is to use the COVID-19 Radiography Dataset to create a Convoluted Neural Network (CNN) that can predict whether a patient has COVID-19, pneumonia or COPD (or has a healthy pair of lungs) with accuracy, recall and precision all over 90%. Due to GPU limitations the plan is to use half off the dataset images (over 10,000 images) to create a training set, a validation set and finally a testing set.

2. Data Wrangling and Exploratory Data Analysis

For this project, I used the [COVID-19 Radiography Dataset] to differentiate between chest X-rays showing COVID-19, COPD, pneumonia and being healthy. There are over 21,000 images in this dataset. 3,616 images are of COVID-19 patients, 6,012 are of COPD patients, 1,345 are of pneumonia patients, and the rest are of normal, healthy lungs. Each datapoint has 6 columns. Ignoring the unique identifiers, what I learned was that each file is a PNG image ('file_format' column) and the image shape ('image_shape' column) is (299, 299). The 'image_data_grayscale' column gives the image converted into an array and in grayscale. The most important column for our project though is the 'label' column as this one has the 4 values I am looking to determine: Normal (healthy), Lung_Opacity (COPD), COVID (COVID-19), and Viral Pneumonia (Pneumonia). The label is most important as it is the dependent variable that needs to be determined for each image.

The first step with this dataset was to run an initial model to get a baseline. With how large this dataset is, I chose to only use 20% of it for the first run-through, with 3387 images in the training set and 846 in the validation set. I chose to keep the images at (299, 299) size and used a batch size of 32.

With the four classes set, the first thing I wanted to look at was the pixel intensity based on each class. There was a noticeable difference for most based on the samples chosen. With COVID-19 images, there wasn't a high frequency at lower intensities, but it strongly peaks between intensity values of 140 and 210 as shown below in Figure 1. For COPD images we see a larger frequency of 0 intensity (white) pixels and from there there's a couple more peaks, but the values are relatively spread out from 75 to 200. Normal, healthy lungs have similar intensity values to COVID lungs with a slightly narrower peak from 170 to 210 intensities. And finally, the pneumonia riddled lungs show a very high frequency of 0 intensity pixels (more than twice that of the next highest frequency). Our current observation based off this information is that the pneumonia images should be easiest to classify followed by the COPD images. However, there could be decreased accuracy when determining between COVID and healthy lungs.

Figure 1: Histogram of Distribution of COVID Pixel Intensities

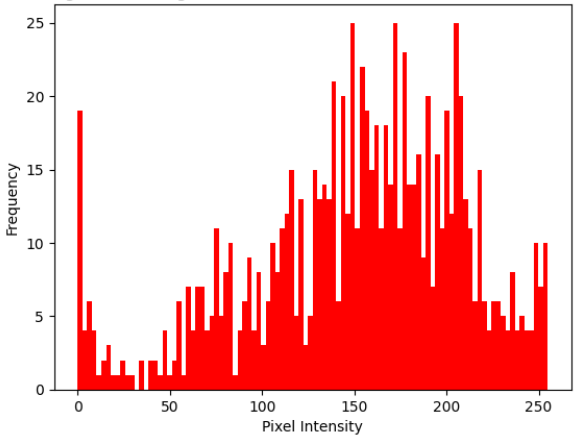
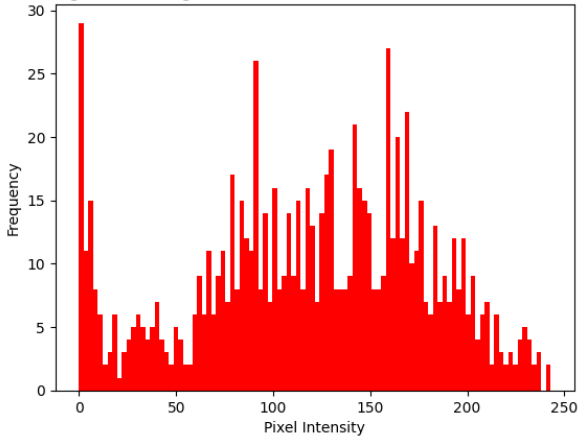
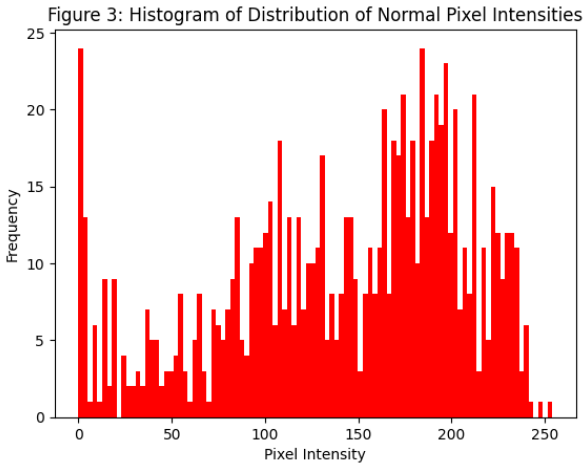
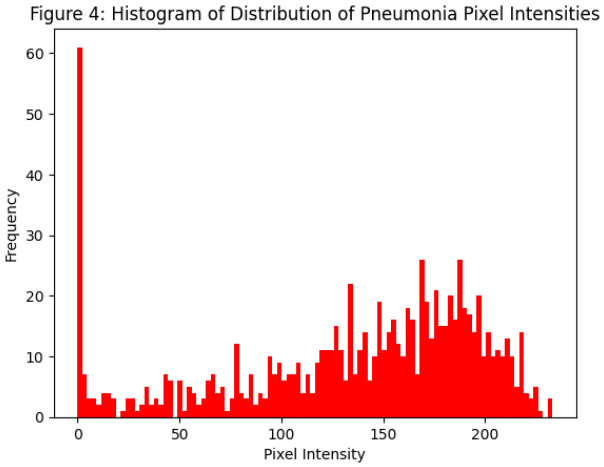


Figure 2: Histogram of Distribution of COPD Pixel Intensities







Having the training and validation datasets created, I made a single layer sequential model and then compiled it with an Adam optimizer, measuring loss and accuracy. I chose 4 epochs as this was to be a simple baseline test for future improvements. The results I got were as follows:

Training accuracy: 0.9460
Training loss: 0.1974

Validation accuracy: 0.7630
Validation loss: 0.6353

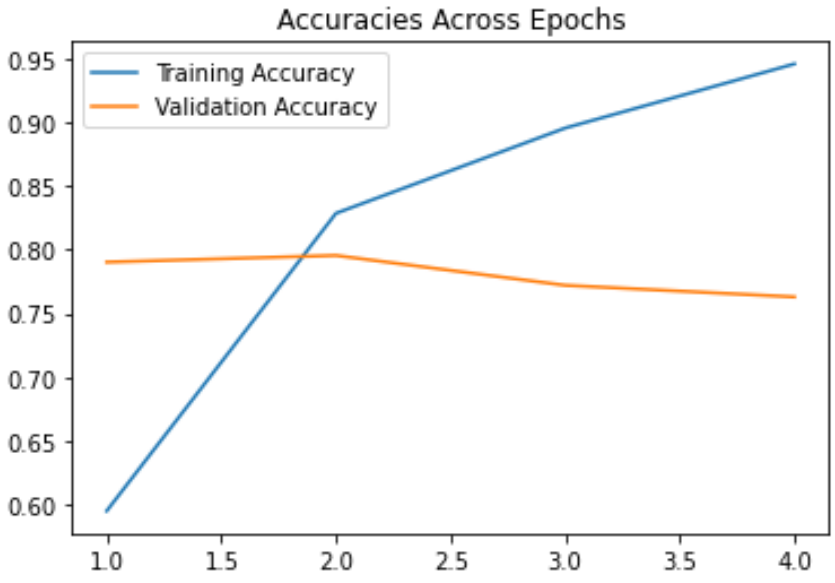


Figure 5: Accuracy across Epochs (Single Layer Model)

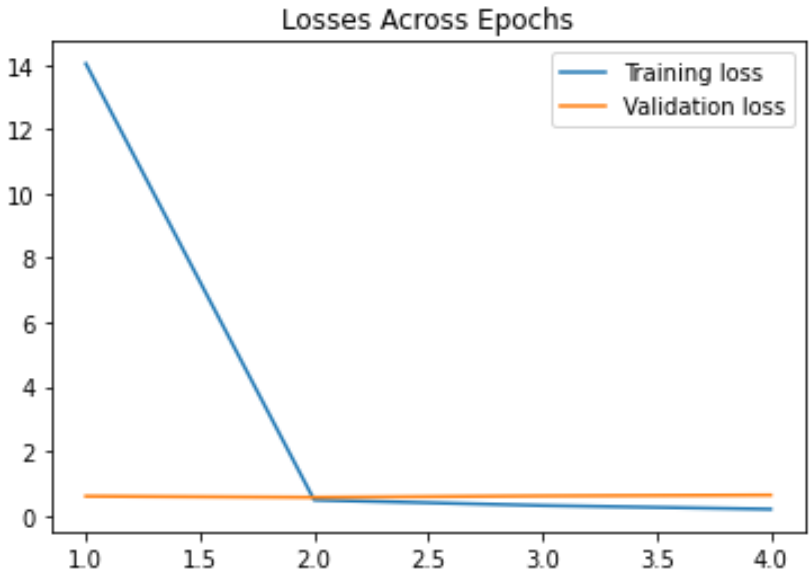


Figure 6: Loss across Epochs (Single Layer Model)

3. Pre-processing and Modeling

The next step was to determine what kind of models I was going to use for the remainder of the pre-processing and modeling steps. The first was a standard triple layered sequential model with where I started to fine tune the hyperparameters. The key hyperparameters for this project were the learning rate of the Adam optimizer, the number of nodes for each layer, and the dropout rate after each layer, if any. I also added batch normalization towards the end of this sequential model and the number of epochs ranged from 5 to 20. In the final iteration of this model, I used the following hyperparameter values:

Learning Rate: 0.001
Epochs: 15 (stopped at 13 epochs due to Early Stopping with a patience of 3)
Batch Size: 128
Dropout Rate after 2nd layer: 0.2
Dropout Rate after 3rd layer: 0.4

I also made the training and validation datasets to be larger with the training dataset having 8,466 files and the validation dataset having 2,116. With this model I got the following results:

Training Accuracy:	0.9348
Training Loss:	0.1837
Validation Accuracy:	0.8256
Validation Loss:	0.4782

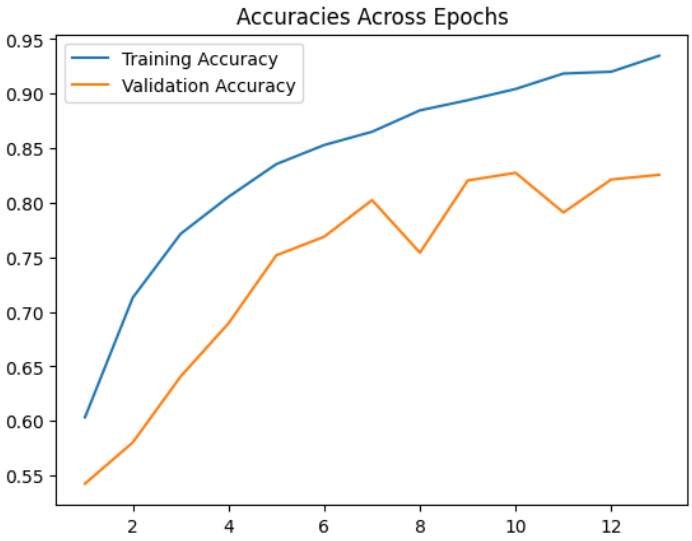


Figure 7: Accuracy across Epochs (Triple Layer Model)

Figure 8: Loss across Epochs (Triple Layer Model)

This serves as a solid baseline before determining if this model can be improved with Transfer Learning. The three transfer learning models that I implemented were VGG16, VGG19, and ResNet50. With each of these, I noticed a lot of overfitting as the model would get to the later epochs (normally starting with 10 to 15 then having 10 more after starting the fine tuning) so I changed the training/validation ration from 80/20 to 70/30 but same total of images. I also added a test dataset, taken from 20% of the validation dataset.

For VGG16, I started fine tuning at layer 14. For VGG19, I started fine tuning at layer 10. And for ResNet50, I had to start fine tuning at layer 170 because at earlier layers it

would start to overfit very quickly. I also set early stopping at 3 for each to keep a consistent stopping point.

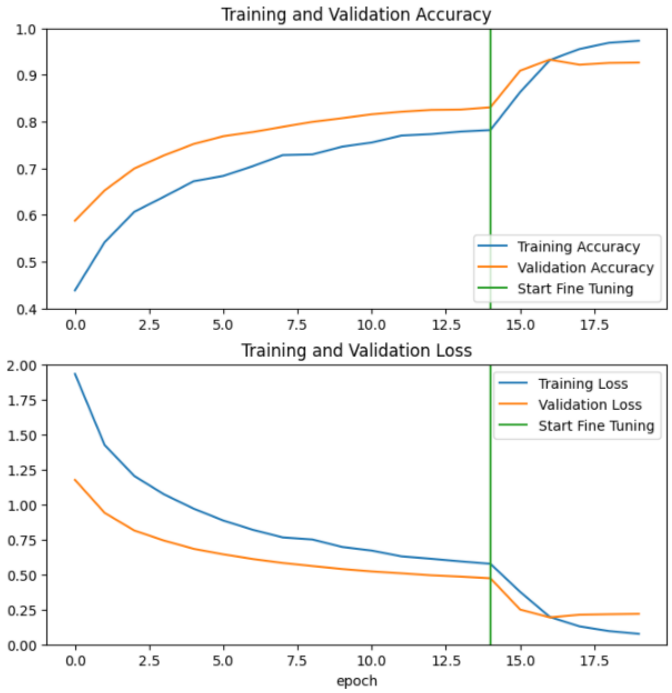


Figure 9: VGG16 Accuracy and Loss across Epochs

With VGG16 the following results after 19 Epochs due to early stopping were found:

Training Accuracy: 0.9733
Training Loss: 0.0759
Validation Accuracy: 0.9266
Validation Loss: 0.2187
Test Accuracy: 0.9141

With VGG19 the following results after 25 Epochs were found:

Training Accuracy: 0.9904
Training Loss: 0.0294
Validation Accuracy: 0.9822
Validation Loss: 0.0687
Test Accuracy: 0.9781

With VGG19 the following results after 25 Epochs were found:

Training Accuracy: 0.9333
Training Loss: 0.2039
Validation Accuracy: 0.9088

Validation Loss: 0.2531
Test Accuracy: 0.9094

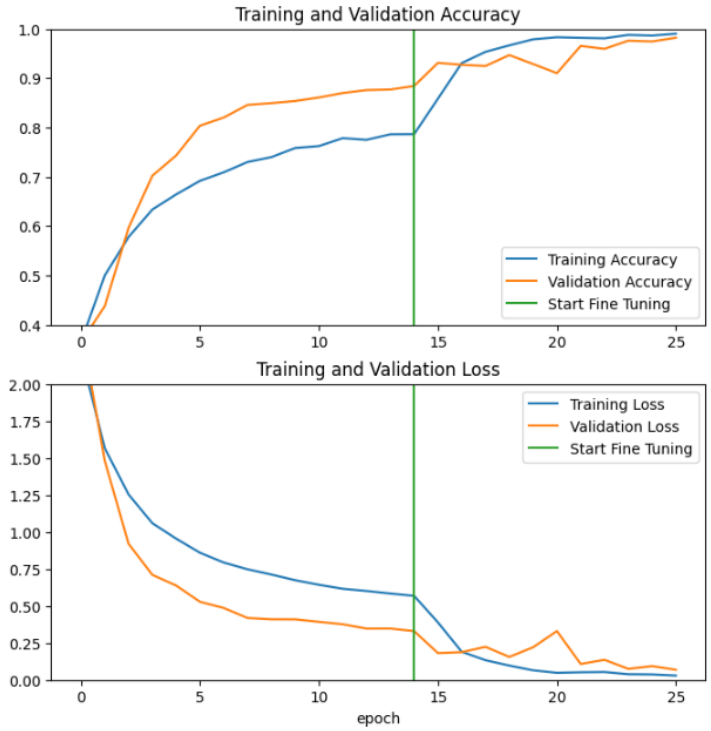


Figure 10: VGG19 Accuracy and Loss across Epochs.

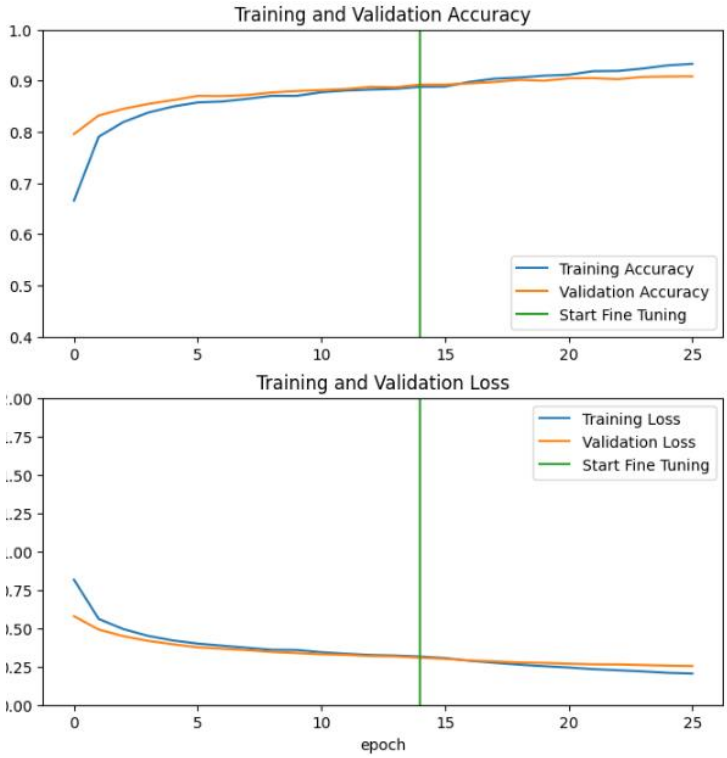


Figure 11: ResNet50 Accuracy and Loss across Epochs.

VGG19 had by far the best results with a test accuracy of 0.9781. With this information I added Recall and Precision to the observed metrics along with a classification report and confusion matrix. Due to these additions however, I had to limit the number of epochs to 10 on the first part before the fine tuning with the limited GPU space I had. In the final run of the model, it seemed the accuracy dropped but that can be attributed to early stopping and limited GPU availability. Below are the final values I got after 17 epochs:

Training Accuracy:	0.9852
Training Loss:	0.0397
Training Precision:	0.9853
Training Recall:	0.9846
Validation Accuracy:	0.9250
Validation Loss:	0.2490
Validation Precision:	0.9272
Validation Recall:	0.9246
F1 Score:	0.9259
Test Accuracy:	0.9203

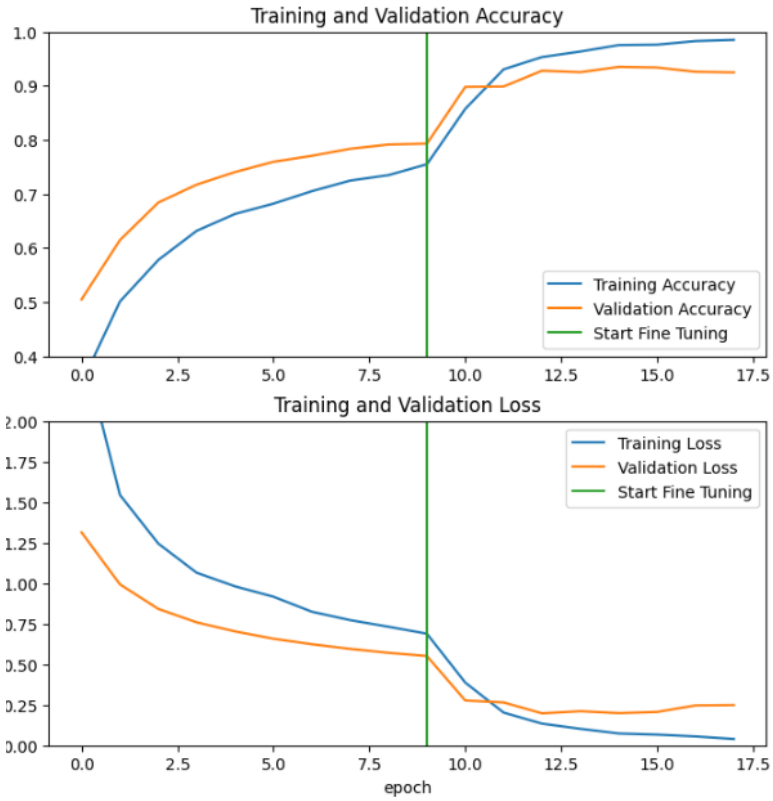


Figure 12: VGG19 Accuracy and Loss across Epochs (Final Model).

	precision	recall	f1-score	support
0	0.95	0.97	0.96	427
1	0.85	0.95	0.90	712
2	0.97	0.89	0.92	1223
3	0.94	0.97	0.96	172
accuracy			0.93	2534
macro avg	0.93	0.95	0.94	2534
weighted avg	0.93	0.93	0.93	2534

Figure 13: Classification Report for VGG19 Model

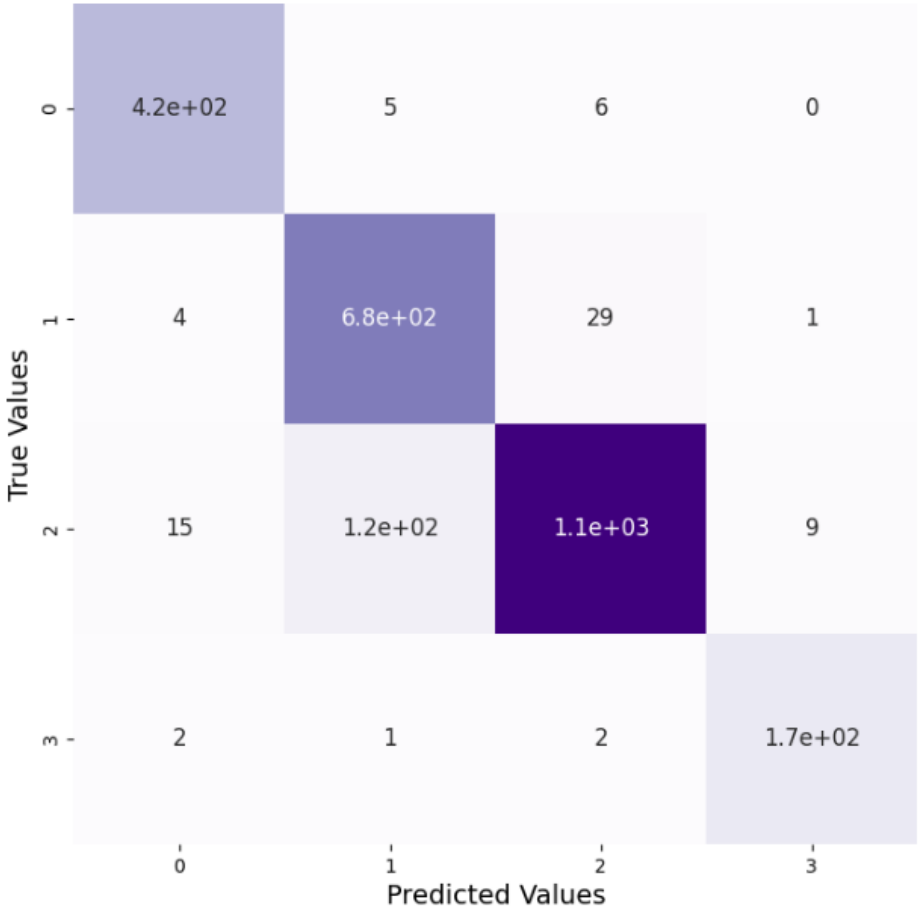


Figure 14: Confusion Matrix for VGG19 Model

What we can determine from these figures above is that there is often an assumption from the model that the predicted values 1 and 2 are confused at times. Due to the limitations of our model I couldn't determine what each value is (0,1,2,3 vs COVID, COPD, Pneumonia, Healthy) but with the sizes of each value in the confusion matrix I can assume that 2 is Healthy lungs and 1 is likely either COPD or COVID. In this case it is a little concerning that the recall for healthy lungs is as low as it is since we want to

make sure we know if a patient is ill or not. However, if we know a patient is ill, the model is excellent at determining what their ailment is.

4. Future Research

What we can take away is that a VGG19 Transfer Learning model is likely the best option for determining what ailment a patient has based off the given chest X-rays when the options are COVID-19, COPD, viral pneumonia, or normal, healthy lungs. With this information it seems critical to reduce the confusion between healthy lungs and non-healthy lungs, improving the precision of the sick lungs and recall of the healthy lungs.

To improve the numbers, I will have to use the entire dataset further instead of half the images and maybe add separate datasets with similar labels for more variation. This is a critical first step as we never want to treat a healthy patient for an illness they don't have, which would be the case for a considerable number of patients in the final test for the VGG19 model.