Diagnosing COVID Using Chest X-RAYs

Aiden Clark - CS 4390

May 5, 2022

Contents

Abstract

Currently the coronavirus disease is spreading rapidly around the world. Diagnosing COVID-19 patients early has proven to be crucial at slowing down the spread of the disease. One of easiest options for detecting the disease is through the use of deep learning strategies. This report uses two deep learning strategies to create models with high accuracy to reliable predict COVID-19 in patients. The first strategy is a tweaked simpler convolutional neural network, while the other strategy uses the more complex VGG-16 architecture. All of the images are prepossessed which includes augmentation, normalizing and resizing the images. The images are sourced from public databases on Kaggle.

Introduction

COVID-19 (coronavirus disease 2019) is an infectious disease causing contagious and sometimes fatal respiratory illness by the (SARS-CoV-2) strain of coronavirus. As of eleventh of March 2020, COVID-19 has been recognized as a global pandemic by the World Health Organization (WHO).

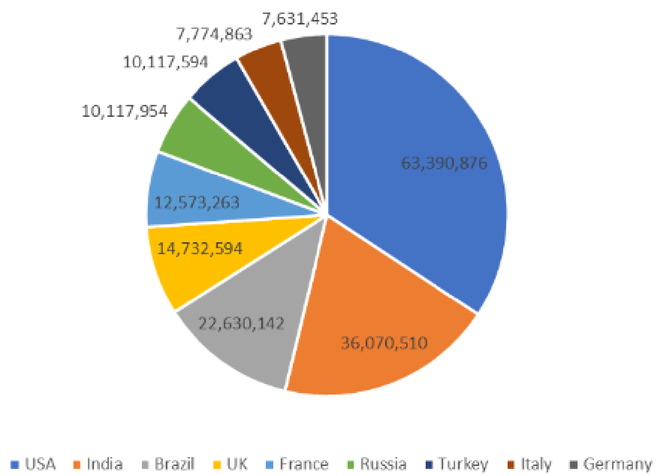

Figure 1: Confirmed COVID-19 Cases Globally [1]

Currently reverse transcription polymerase chain reactions (RT-PCR) are used for diagnoses. X-ray machines are more available and provided extremely fast imaging. They can be used as a cheaper,faster alternative to RT-PCR.

Chest x-rays are a inexpensive and fast test method to detect COVID-19. A person with COVID-19 has obvious signs of infection in their chest x-rays. However, other respiratory infections like pneumonia or a seasonal flu may show similar results in the x-ray. Taking that into account, using deep learning can create high accuracy models for detecting COVID.

The imaging is sourced using multiple data-sets from Kaggle.

GitHub Link: `https://github.com/Aidenclark/Diagnosing-COVID-Using-Chest-X-RAYS`

1 Data Preparation

1.1 Image Selection

There is a total of 5250 images of normal chest x-rays and COVID-19 infected chest x-rays. There is no image with multiple instances. Each image is unique.

1.2 Image Prepossessing

The images are classified using binary classification. Each of the images are also resized to 150 x 150 dimensions and have been cropped to show the patients chest only.

1. Normal X-ray (0)
2. COVID-19 X-ray (1)

1.2.1 Image Augmentation

The model uses image augmentation in the training process. Image augmentation is the technique of altering the image to create more data for the model to use in the training process. Some examples of augmentation used were: rotation, width shift, height shift, shear range, zoom range, and horizontal flip.

Figure 2: Example of Augmentation

1.3 Data Distribution and Visualization

The images are classified using binary classification as mentioned in section 1.2. There is a slight imbalance of 500 more images for normal chest x-rays of normal patients than COVID-19 patients. A subset of the training images can be seen below.

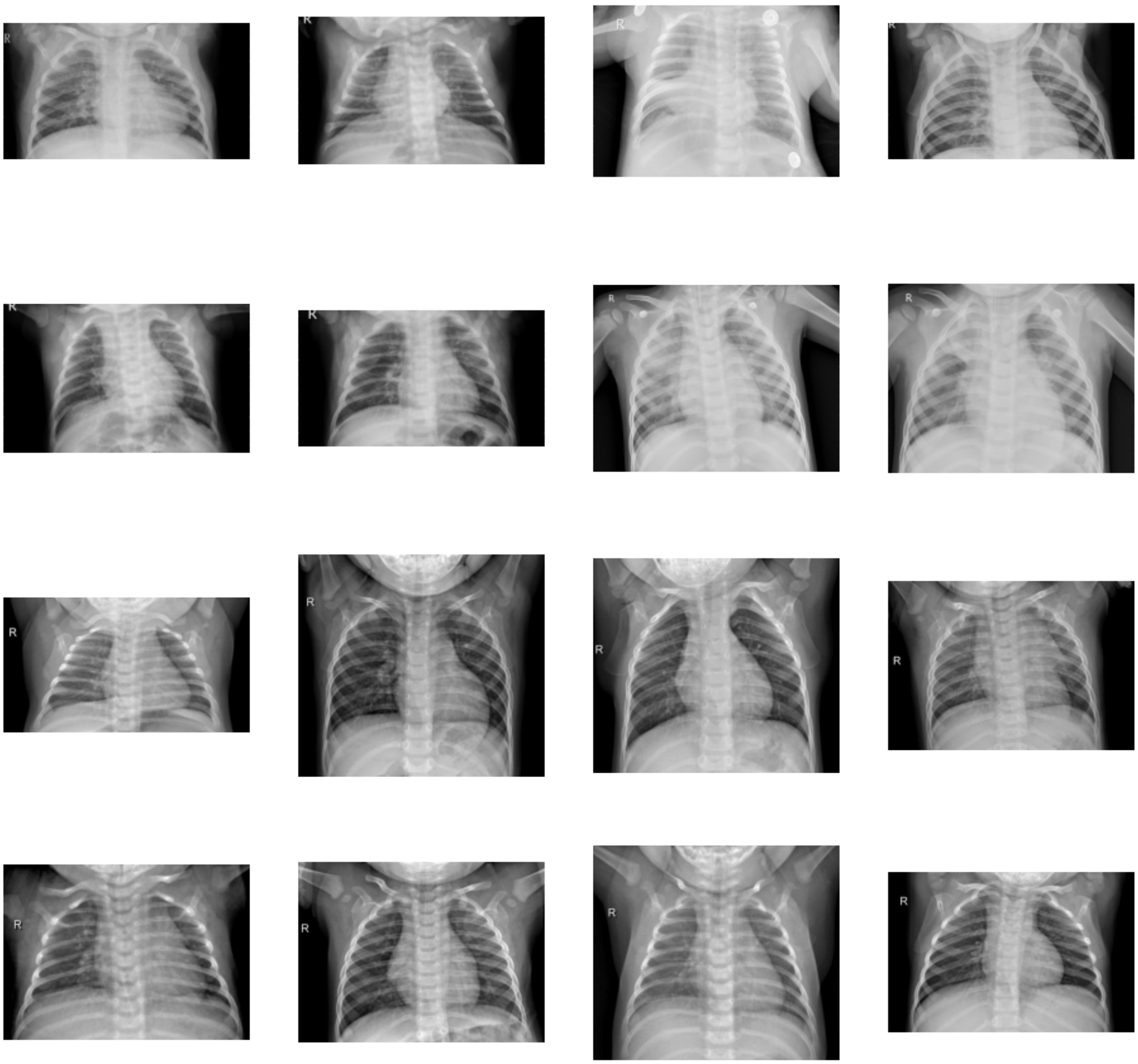

Figure 3: Visualization of Image Samples

Total	Normal X-ray	COVID-19 X-ray
Image Count	2884	2366

2 Testing Simpler Architectures

2.1 Baseline Validation Results

Accuracy	Loss	Precision	Recall
93.68	25.69	93.69	96.02

2.1.1 Baseline Model Architecture

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 148, 148, 16)      448

 conv2d_1 (Conv2D)           (None, 146, 146, 4)       580

 flatten (Flatten)           (None, 85264)             0

 dense (Dense)               (None, 10)                852650

 dense_1 (Dense)             (None, 1)                 11

=================================================================
Total params: 853,689
Trainable params: 853,689
Non-trainable params: 0
_________________________________________________________________
```

Figure 4: Baseline Model Architecture

2.2 Model 1 Validation Results

From overfit model from previous phase. Increased dense neuron count from 10 to 20.

Accuracy	Loss	Precision	Recall
94.14	14.13	93.11	97.51

2.3 Model 2 Validation Results

Adding an additional convolution layer. More layers were added, but the model is not overfit.

Accuracy	Loss	Precision	Recall
60.45	70.47	60.49	98.97

2.4 Learning Curves and Evaluation

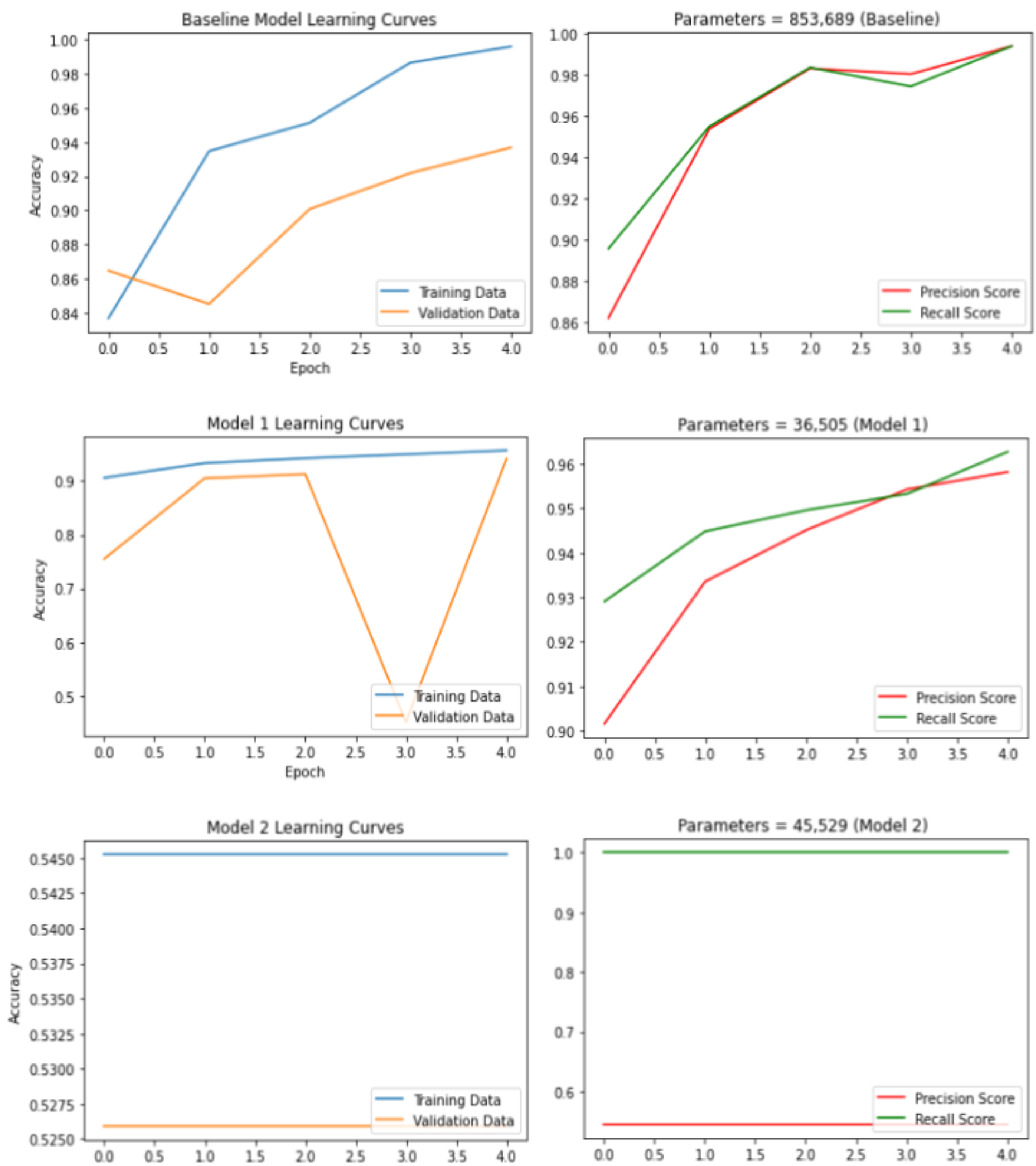

Figure 5: Learning Curves and Evaluation

3 Increasing Accuracy with Augmentation

Image augmentation can potentially increase the accuracy. The follow types of augmentation and their affect on the accuracy of the model are as follows:

3.1 Re-scale and Rotation Range (40)

Accuracy	Loss	Precision	Recall
80.60	42.33	80.27	90.05

3.2 Width (.2) and Height Shift (.2)

Accuracy	Loss	Precision	Recall
83.76	40.72	86.21	87.06

3.3 Shear Range (.2)

Shear range, although not the most augmenting, proved to have the highest accuracy on the model.

Accuracy	Loss	Precision	Recall
95.15	21.60	93.63	98.76

3.4 Shear Range (.2) and Width (.2) and Height Shift (.2)

Accuracy	Loss	Precision	Recall
80.45	44.60	81.48	87.56

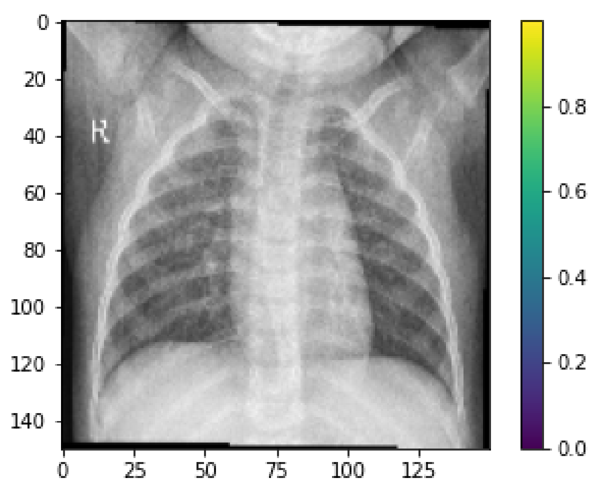

Figure 6: Example of Shear Range on the Pictures

4 Increasing Accuracy with Regularization

Regularization is the technique of attempting to increase accuracy by making slight modifications to the learning algorithm so that the model can generalize better.
This section will analyze Batch normalization, Dropout, and L2 regularization.

4.1 Batch Normalization Model

Batch normalization standardizes the inputs to a layer for each batch. This can stabilize the learning process and reduce the number of training epochs required.

Accuracy	Loss	Precision	Recall
75.79	51.80	71.79	98.76

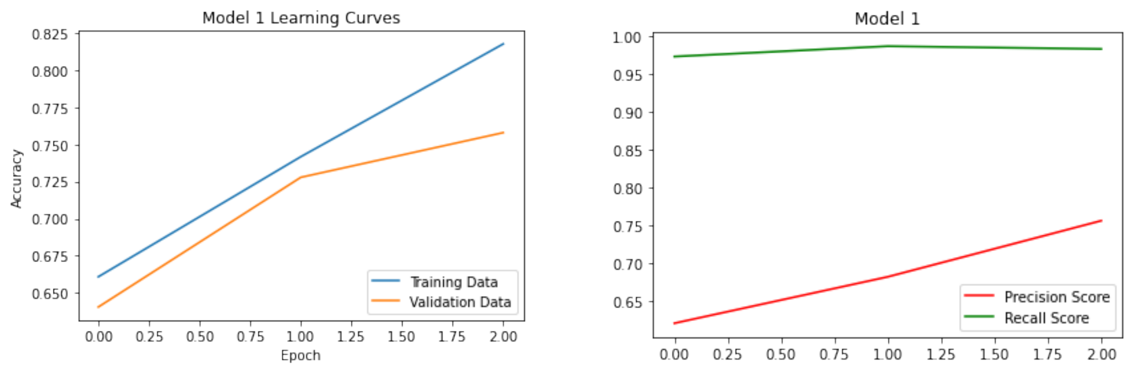

Figure 7: Learning Curves and Evaluation for Model 1

4.2 Dropout Model

Dropout is when randomly selected neurons get ignored during training- they get 'dropped-out' randomly.

Accuracy	Loss	Precision	Recall
60.45	67.87	60.45	100

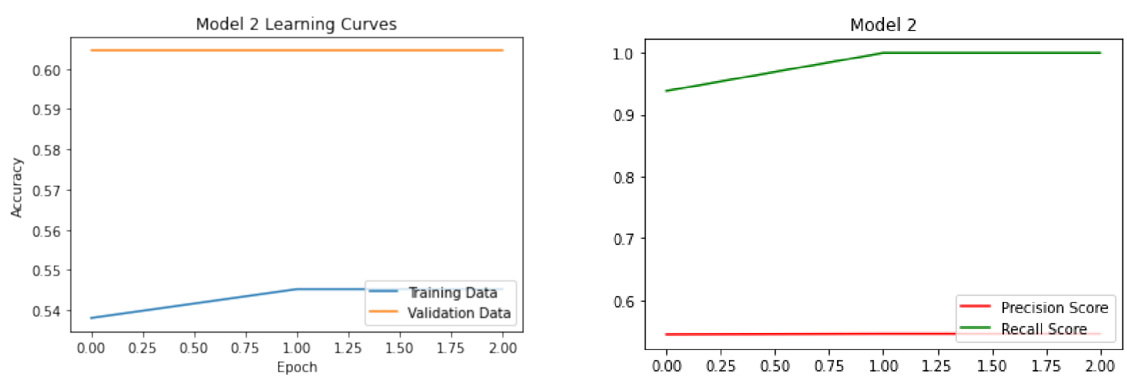

Figure 8: Learning Curves and Evaluation for Model 2

4.3 L2 Regularization Model

L2 regularization removes a small percentage of weights at each iteration.

Accuracy	Loss	Precision	Recall
91.88	36.05	89.55	98.01

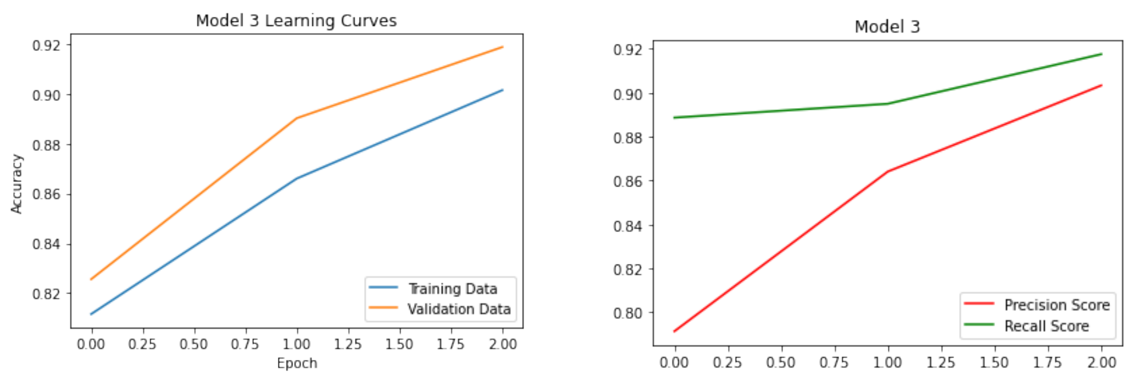

Figure 9: Learning Curves and Evaluation for Model 3

4.4 Combining L2 Regularization Batch Normalization

Accuracy	Loss	Precision	Recall
93.08	51.66	96.84	91.54

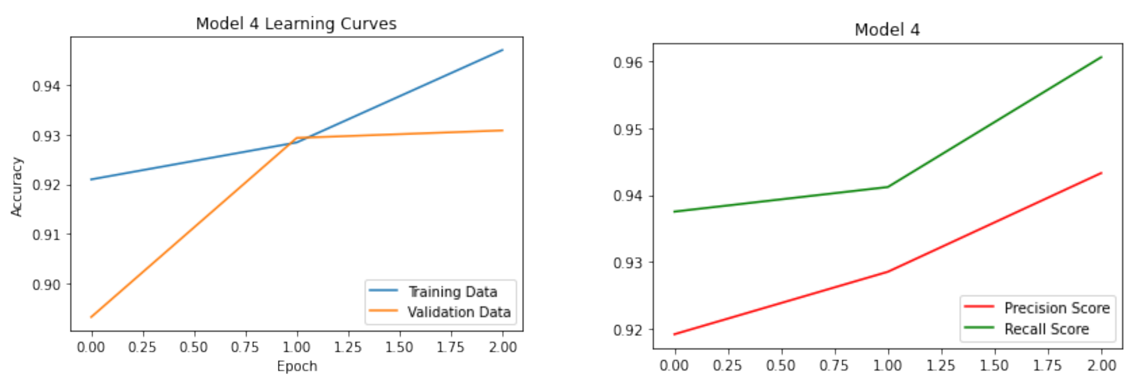

Figure 10: Learning Curves and Evaluation for Model 4

5 Increasing Accuracy Using More Powerful Architectures

There are popular convolutional neural network architectures that have proven to be effective at predicting all types of data-sets. This section will use some popular types of architecture to attempt to increase accuracy on the validation data-set. [2]

5.1 ResNet50 Architecture

ResNet50 is a convolutional neural network that is a variant of the ResNet model which has 48 convolution layers with 1 MaxPool and 1 average pool layer ResNet is so acclaimed because of how deep it is with little percentages of errors.

Accuracy	Loss	Precision	Recall
83.13	80.91	80.55	81.64

5.2 DenseNet121 Architecture

DenseNet architecture focuses on making deep learning networks go as deep as possible while making them more time efficient by utilizing shorter connections between each of the layers.

Accuracy	Loss	Precision	Recall
90.88	28.52	87.87	88.43

5.3 EfficientNetB7 Architecture

EfficientNet is a convolutional neural network architexture that uniformly scales all dimensions of depth, width and resolution using a compound coefficient.

Accuracy	Loss	Precision	Recall
92.40	23.52	89.40	89.31

6 Modifying the VGG-16 Architecture

A popular architeture is the VGG-16 model. VGG-16 is a convolution neural network which was used to win the ILSVR(Imagenet) competition in 2014. It is considered to be one of the best model architectures to date. [3]

6.1 Using VGG-16 to Predict Class Probability

Unfortunaetly, the pre-trained VGG-16 model was failing to recognize the X-Ray image when used to predict the class. The model recogized the images as a 'conch' shell. Modifications can be made to use the architecture.

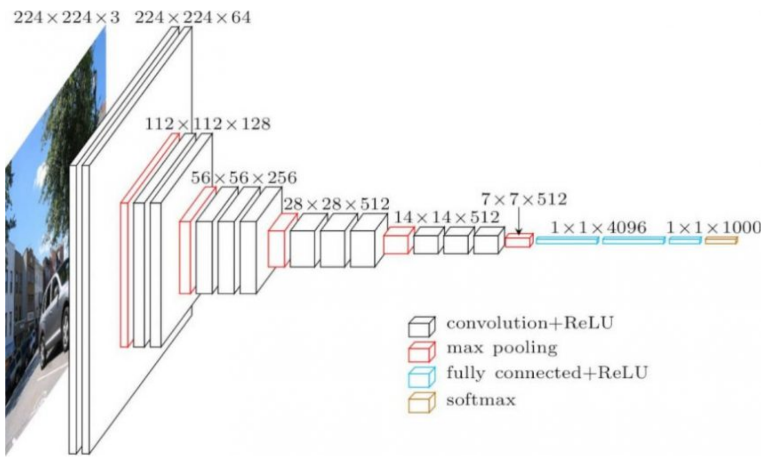

Figure 11: VGG-16 Architecture [3]

6.2 Results

The VGG-16 architecture was modified to fit the data-set and tested against the best model the was previously developed from the later Phases. The results are as follows:

	Accuracy	Loss	Precision	Recall
First Model	94.29	62.08	95.27	95.27
VGG-16	95.34	46.58	96.96	95.27

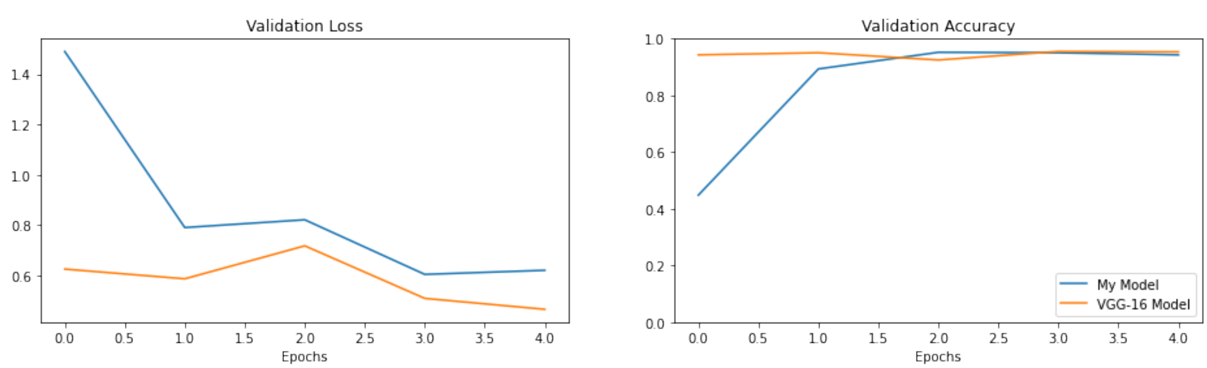

Figure 12: Loss and Accuracy Comparisons

7 Conclusion and Future Work

There are a few places that this project can go for improvements. First lets analyze the project itself. The simple models that were created first proved to have decent accuracy, but was trumped by the more powerful architectures. There is also some worry that the baseline model may be over fit. The baseline model did not include max pooling, and when pooling layers were added it decreased the accuracy. It can also be concluded that too much augmentation can be a bad thing. In the case of X-Rays, if the image is distorted too much then the complexity of finding COVID-19 in the image is lost. There were also some challenges involved when creating this project. The length of time the models took to compile was difficult to work with. Five epochs would take around 30 minutes, which is extremely inconvenient. Hence, the small number of data points in the graphs. The models could have also improved accuracy if they were allowed to fun longer.

In the future, adding a pneumonia data set would improve the durability of the models. Currently the models are just binary classification, either COVID-19 or normal patient. Pneumonia looks similar to COVID-19 in an X-Ray, and building a model that can recognize that will be more durable then the current models. More training would also be helpful in the future. Five or three epochs will not cut it. Another route could be exploring different architectures and their effectiveness.

References

[1] Gouda, W.; Almurafeh, M.;Humayun, M.; Jhanjhi, N.Z.Detection of COVID-19 Based onChest X-rays Using Deep Learning.Healthcare2022,10, 343

[2] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenetclassification with deep convolutional neural networks.InNIPS, pp. 1097–1105, 2012.

[3] Thakur, R. (2020, November 24). Step by step VGG16 implementation in Keras for beginners. Medium. Retrieved April 19, 2022, from https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c