Covid-19 Detection

Submitted By: Shiv Vyas (903939075)
Instructor: Dr. Ryan White
Course: MTH 5320 Neural Networks (Project 1)

Date: October 9, 2020

Table of Contents

Objective

The objective of this project is to determine whether a person is infected by Coronavirus or not. Chest Radiography is currently one of the crucial methods for the detection of COVID- 19 in patients. Chest radiography includes Chest CT- Scans and Chest X-Ray. The project should be able to detect Covid-19 from either Chest X-ray or Chest CT-Scan or both.

Introduction

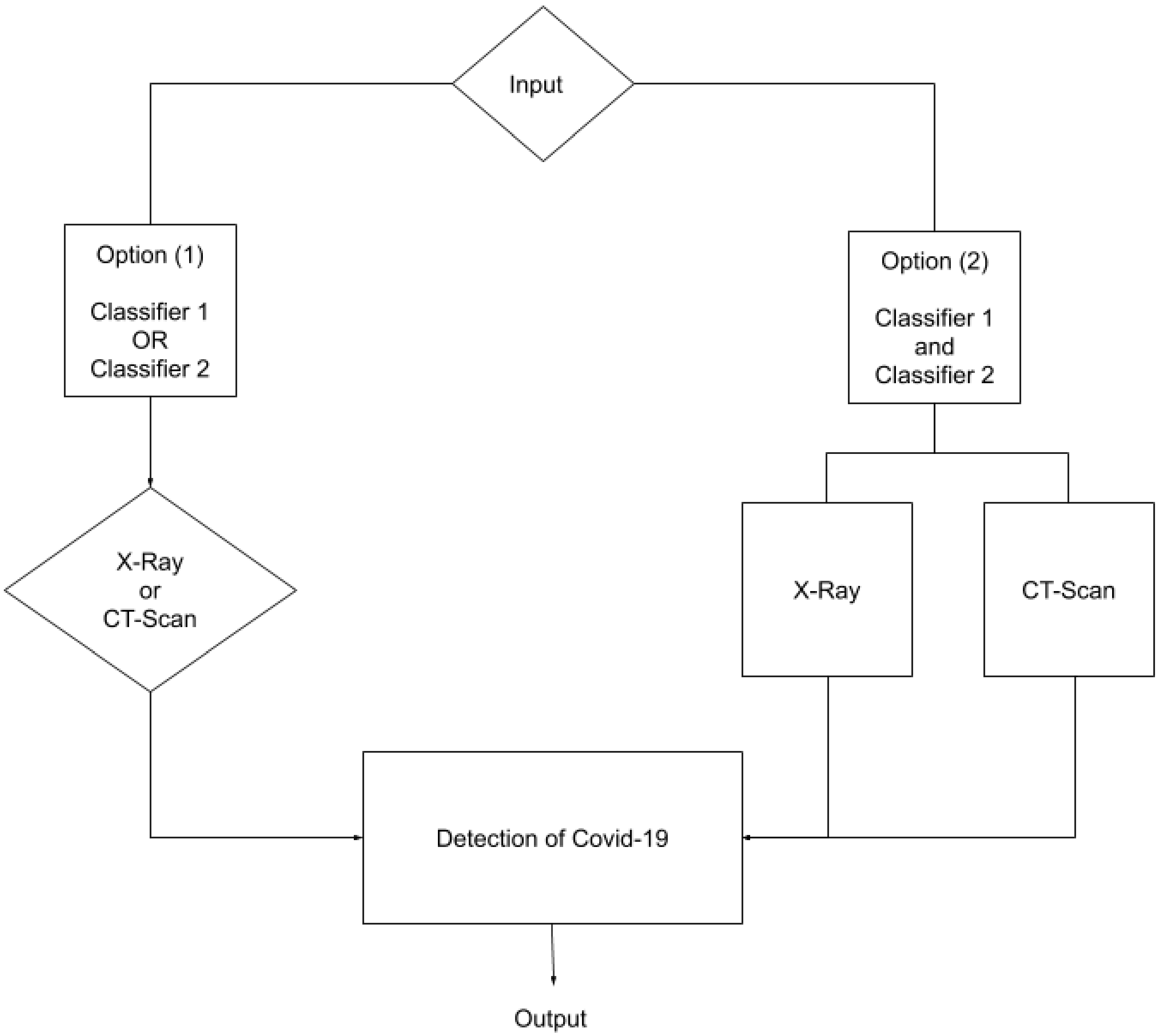
COVID-19

Covid-19 is also termed as “Coronavirus(CoV).” It is a novel virus that was not identified in humans before 2020. Cov is the large family of SARS-Cov, and it is seen to be transmitted from animals to humans. This virus causes illnesses such as common cold, respiratory problems, etc. The outbreak of this virus has resulted in a pandemic; it has affected 37.1 million people and has caused 1.07 Million fatalities worldwide till date.

Why detection from Chest Radiography?

To provide Immediate results.
Currently, the Reverse transcription-polymerase chain reaction(RT PCR) test is taking 2-3 hours for the results.
Extend of spread can be detected from chest radiography.

Description



A Multi-Model Neural Network Classification
Classifier 1: Classification of Chest CT - Scan
Classifier 2: Classification of Chest X-Ray

Classifier 1 - Chest CT Scans

Dataset
Dataset Used: CT Scans for COVID - 19

Details :
Label 1 - NiCT - Images with no information (5705 Images)
Label 2 - niCT - Negative Covid-19 CT Scans (9979 Images)
Label 3 - piCT - Positive Covid-19 CT Scans (4001 Images)
This data set has some images of shape (512 x 512) and some of (1211 x 1211) pixels.

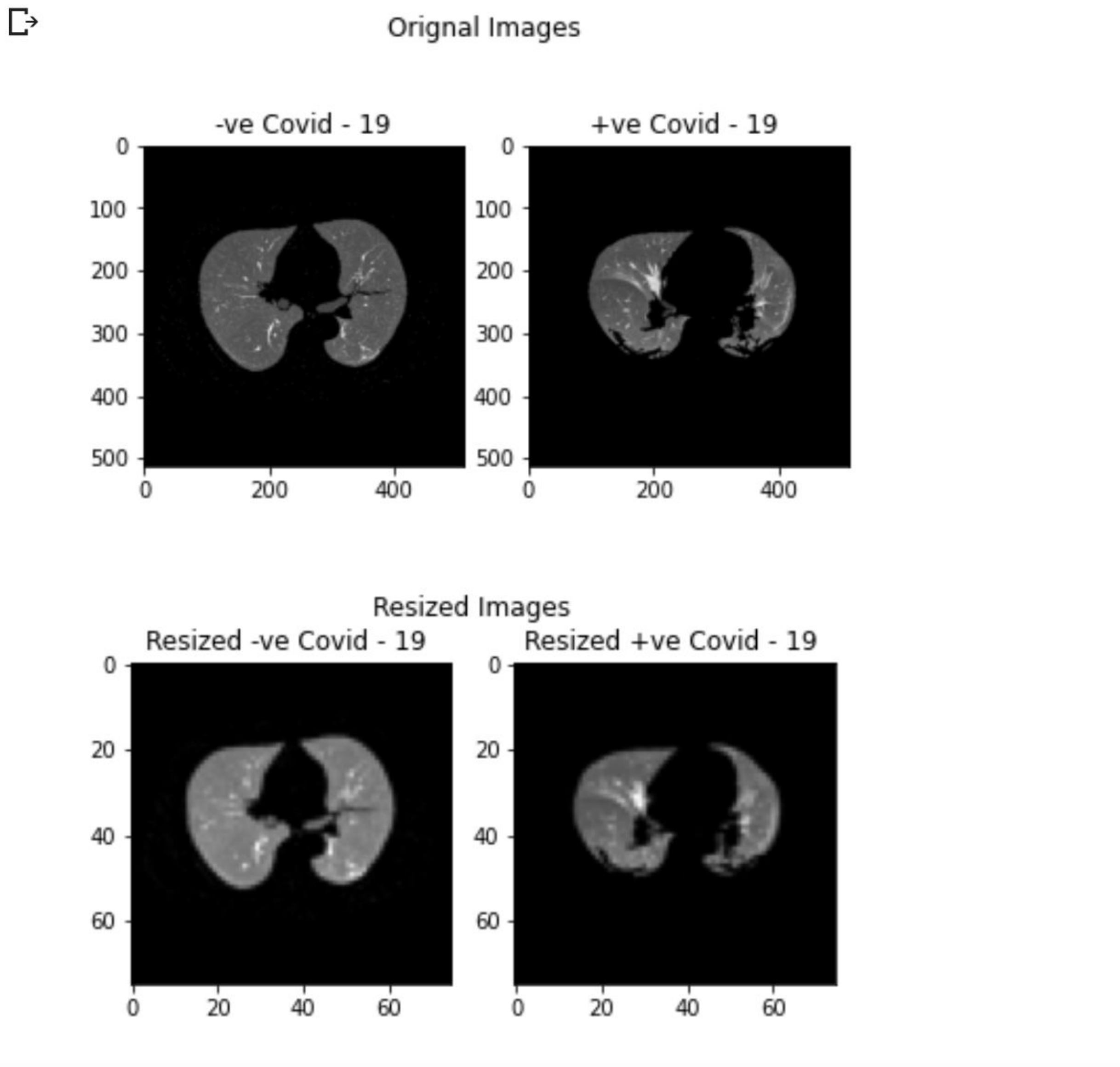
Data Pre-Processing
The Images from Label "NiCT" were removed as they had no information, which can help in classification.

1) Resizing
Original Image Size : 512 x 512 Pixels and 1211 x 1211 Pixels
Converted Image Size : 75 x 75 ,
Reason: Faster Execution, Reducing computational complexity, and time.

2) Normalization
Method 1 - Using normalize() function - It does standard normalization, $\mu=0$ and $\sigma=1$
Method 2 - Using MinMax Scaler () - In this approach, the data is scaled to a fixed range, usually 0 to 1

Processed Dataset :
The dataset was processed into two new labels :
Label: '1' → Negative COVID - 19 (9979 Images)
Label: '2' → Positive COVID - 19 (4001 Images)
Images from both the labels are converted into shape (75,75,3)
3 represents the Channel.

Data Visualization



As shown in the image, there is no major data loss in the resized image.

Architectures, Tuning - Hyperparameters, Regularization.

Architecture: Input Layer → Hidden Layers → Output Layer
As the processed images are in the size of 75 X 75 pixels and 3 channels.
The input Layer has (75*75*3) or (16875) Neurons.
We have 2 Labels so the Output layer will have (2) Neurons

Architecture: [(75 x 75 x 3) , Hidden- layers , 2]

Network Architectures Tested :

[(75 x 75 x 3) , 16 , 2]
[(75 x 75 x 3) , 64, 16 , 2]
[(75 x 75 x 3) , 128 , 16 , 2]
[(75 x 75 x 3) , 128 , 64 , 2]
[(75 x 75 x 3) , 256, 128 , 64, 2]
[(75 x 75 x 3) , 512, 256, 128, 64, 2]

Along with different Hidden layers, a total of 11 different architectures with different Combinations of Activation function, Loss function, Weight initialization methods were tested.

Parameters Tested :
Activation Function : Sigmoid , ReLU , ELU
Loss Function: cross-entropy and sum-of-square
Regularization : L1 regularization, L2 regularization, L1 & L2 regularization.
Weight Initialization Methods: Normal, Glorot, LeCun, He, Uniform

Results

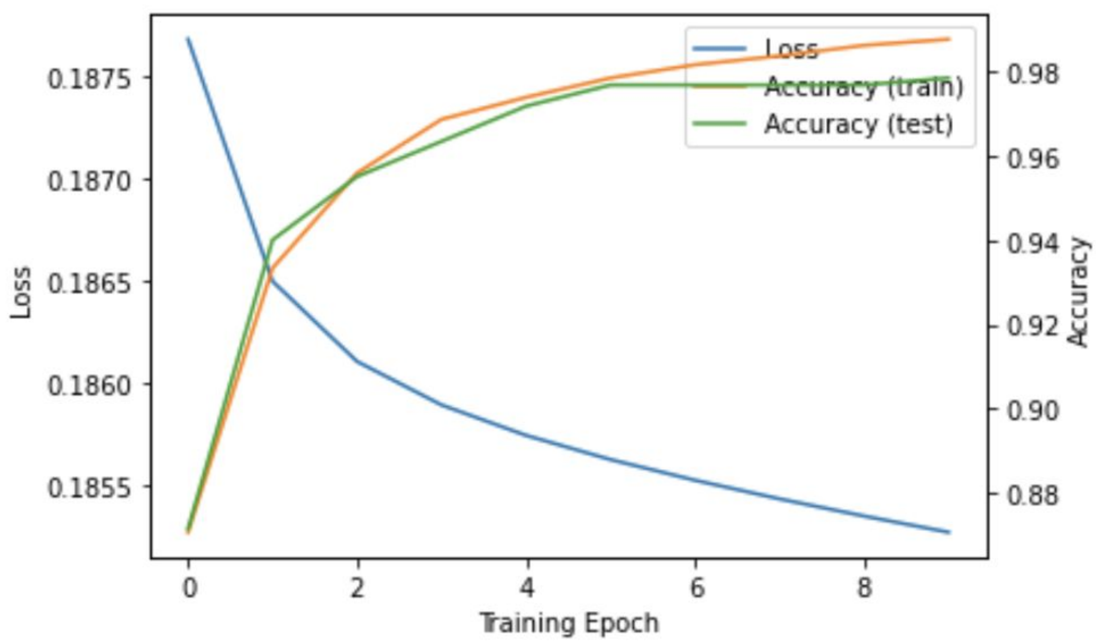
Highest Accuracy

Architecture : `[(75*75*3), 512, 256, 128, 64, 2]`

Activation Function : ReLU
Loss Function: Sum-of-Squares
Weight Initialization: Glorot
Momentum: 0
Learning Rate : 0.01
Regularization: L1 Regularization , L 1 = 0.01, L2 =0

Accuracy :

```
Training set accuracy
0.9875
Test set accuracy
0.9783333333333334
Val set accuracy
0.9833333333333333
```



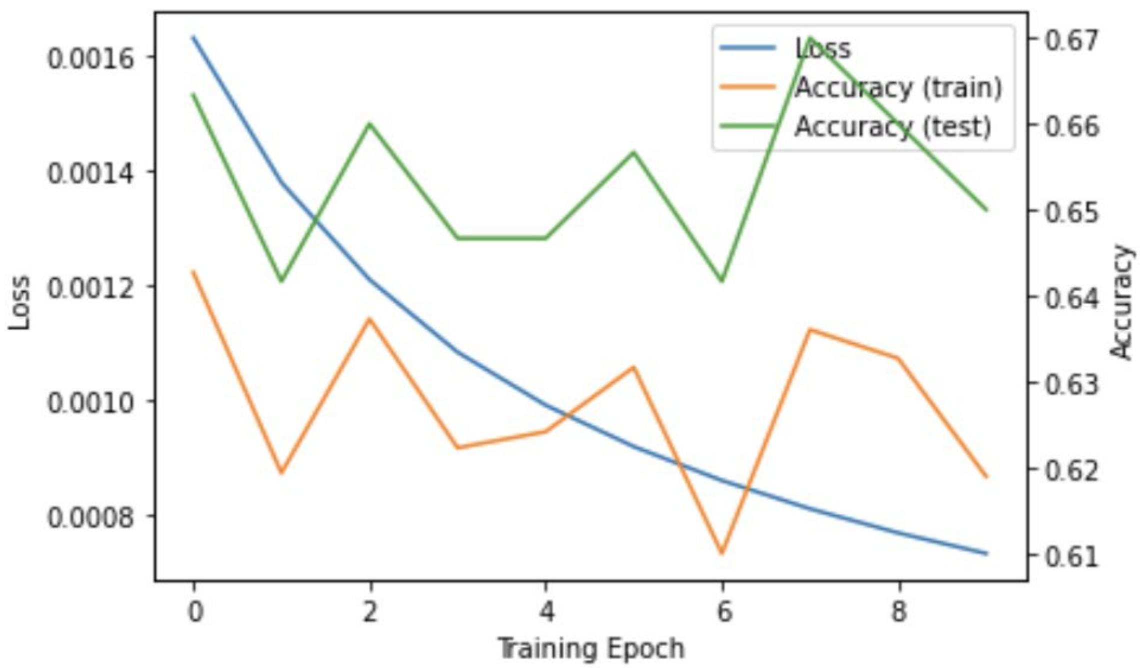
Second Highest Accuracy :

Architecture : `[(75*75*3), 512, 256, 128, 64, 2]`

Activation Function : ELU
Loss Function: Sum-of-Squares
Weight Initialization: Glorot
Momentum: 0
Learning Rate: 0.1
Regularization: L1 = 0 , L2 =0

Accuracy :

```
Training set accuracy
0.6189583333333334
Test set accuracy
0.65
Val set accuracy
0.65
```

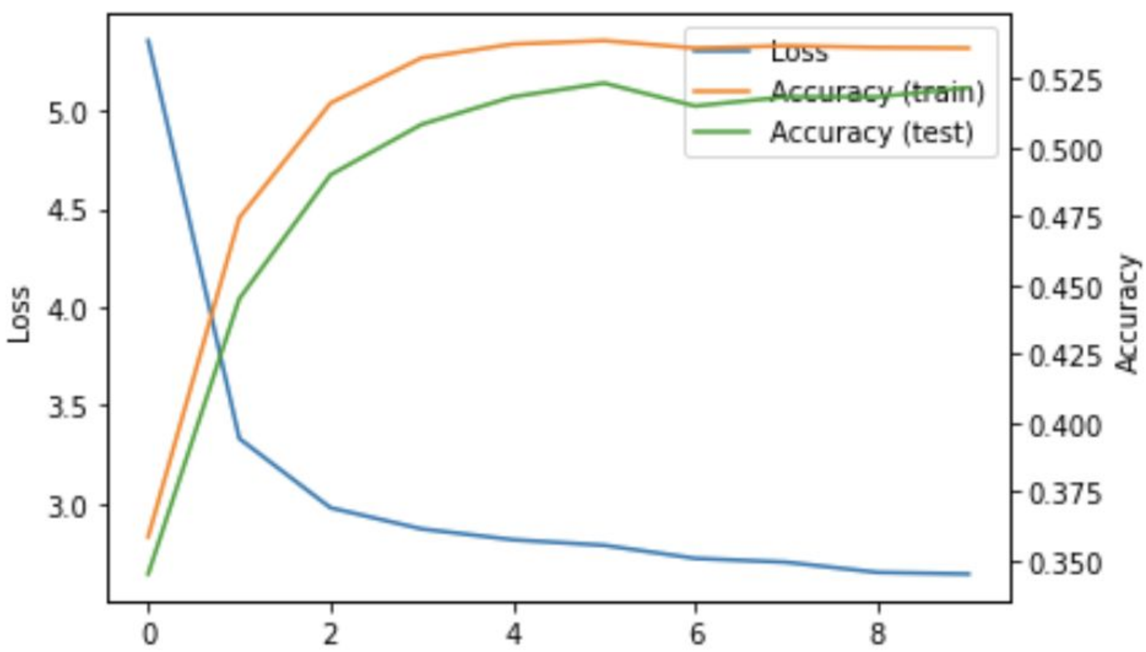


Third Highest Accuracy :

Architecture : `[(75*75*3), 512, 256, 128, 64, 2]`
Activation Function : ELU
Loss Function: Cross-Entropy
Weight Initialization: Uniform
Momentum: 0
Learning Rate : 0.01
Regularization: L1 = 0 , L2 = 0

Accuracy :

```
0.5360416666666666
Test set accuracy
0.5216666666666666
Val set accuracy
0.52
```



Classifier 2: Chest X-Ray

<u>Dataset</u>

Dataset Used

[Positive COVID-19 X-Rays]
[Negative COVID-19 X-Rays]
[Dropbox Link]

Details

Label: ‘1’ is as signed for positive COVID-19, images were downloaded from the dataset.
Label: ‘2’ is assigned for negative COVID-19, Images from folder “Normal” used.
There were only 224 Images combined together from both the labels, which is very low, so data augmentation is required.

Data Augmentation & Data Pre-Processing

Normalization
Images were normalized with a scale of 1/.255, to reduce the computational complexity and time.

Resizing
Original Image Size : 1165 x 1165 Pixels, Channel = 3
Converted Image Size : 75 x 75 pixels , Channel = 3
Data Augmentation is a technique to increase the number of Neural Network training images without changing its orientation.

Features used for Data Augmentation :
Horizontal Flip: Images are flipped Horizontally.
Shear Range: Images were slanted by nearly 20% (> 20% can change Image Orientation)
Zoom = Images were Zoomed by 20% (> 20% can change Image aspect ratio)

A total of :
Train - 4700 Images for Training, 700 Images for Testing, 700 Images for Validation were created by Augmentation.

Data Preprocessing was done along with the data augmentation.

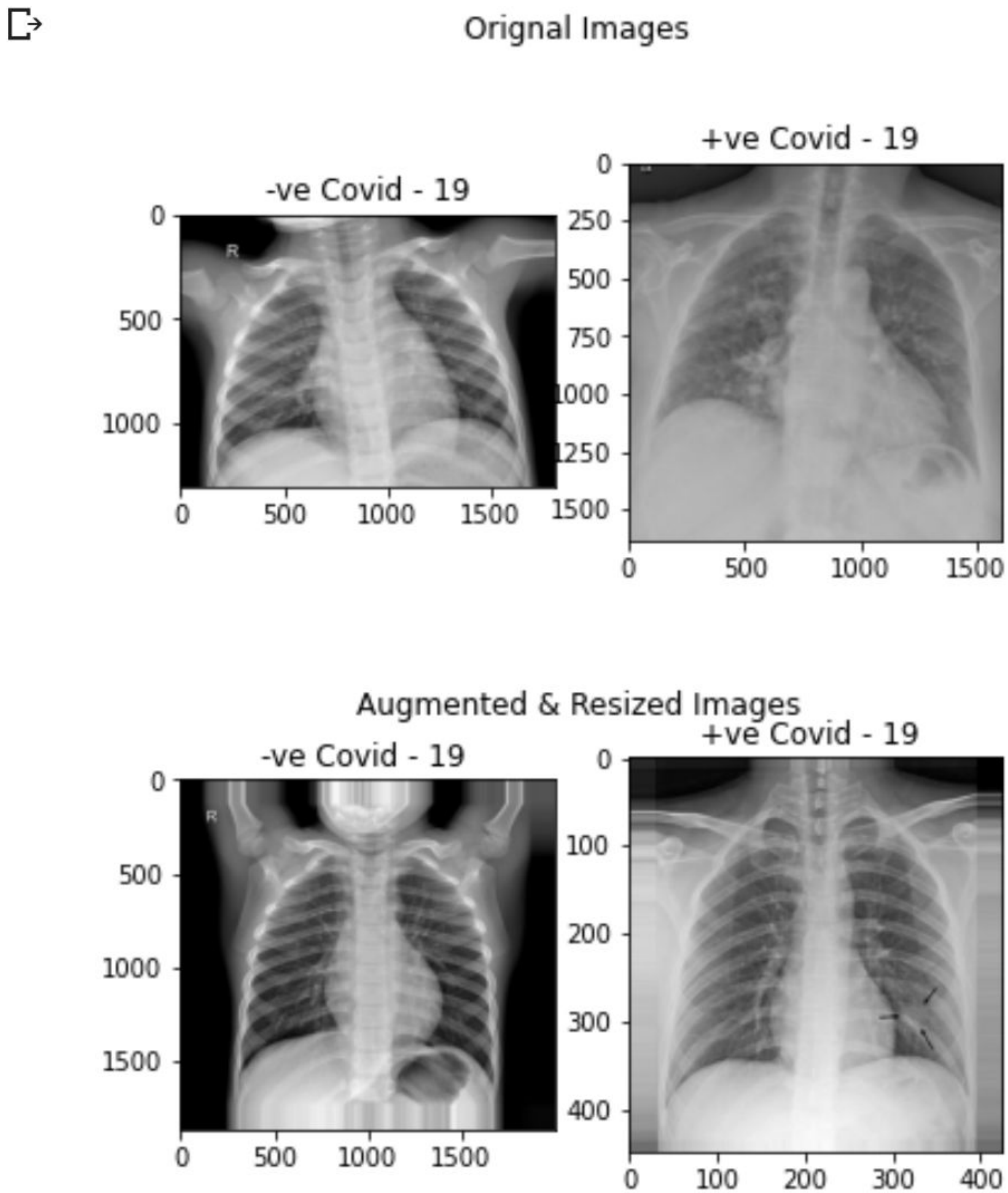
ImageDataGenerator of Keras was used for DataAugmentation:

```
#As data is low, we will perfrom augmentaion for Training Data
from keras.preprocessing.image import ImageDataGenerator, array_to_img, img_to_array, load_img
datagen = image.ImageDataGenerator(

    #Parameters for Data Augmentation
    #Rescalingn the images
    rescale = 1./255,
    #Flipping the image
    horizontal_flip = True,
    #Slanting the image
    shear_range = 0.2,
    #Zooming
    zoom_range = 0.2,

)
```

Data Visualization



As seen in figure Augmented and Resized Images are having no major loss.

Architectures, Tuning - Hyperparameters, Regularization.

Architecture: Input Layer → Hidden Layers → Output Layer
As the processed images are in the size of 75 X 75 pixels and three channels.
The input Layer has (75*75*3) or (16875) Neurons.
We have 2 Labels, so the Output layer will have (2) Neurons.

Architecture: [(75 x 75 x 3) , Hidden- layers , 2]

Network Architecture Tested :

[(75 x 75 x 3) , 16 , 2]
[(75 x 75 x 3) , 64, 16 , 2]
[(75 x 75 x 3) , 128 , 16 , 2]
[(75 x 75 x 3) , 128 , 16 , 2]
[(75 x 75 x 3) , 256, 128 , 16, 2]

Along with different Hidden layers, 13 different architectures with different Combinations of Activation function, Loss function, Weight initialization methods were tested.

Parameters Tested :
Activation Function : Sigmoid , ReLU , ELU
Loss Function: cross-entropy and sum-of-square
Regularization : L1 regularization, L2 regularization, L1 & L2 regularization.
Weight Initialization Methods: Normal, Glorot, LeCun, He, Uniform

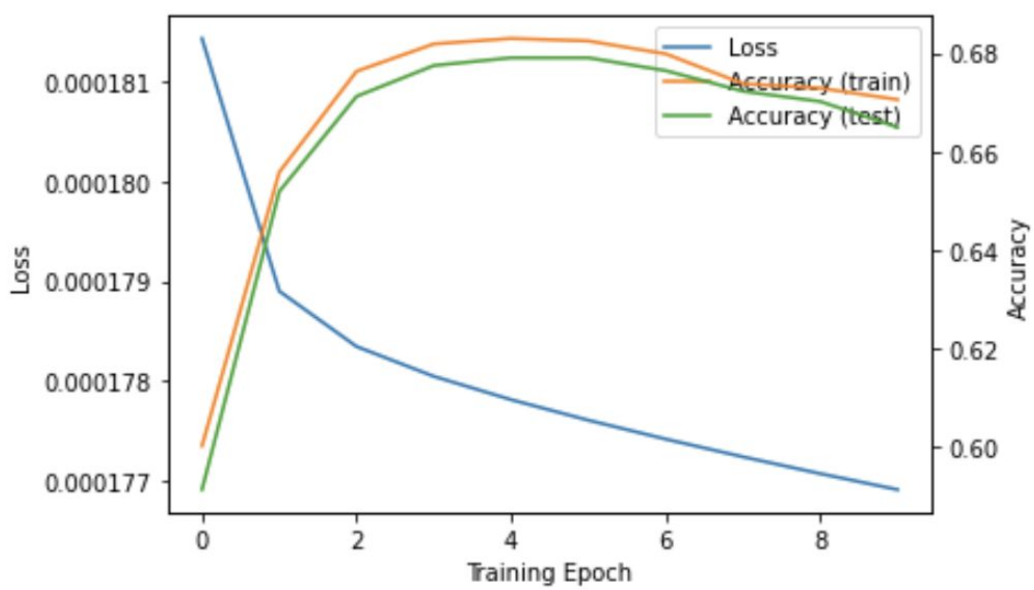
Results

Highest Accuracy

Architecture : `[(75*75*3),  128, 16, 2]`
Activation Function : ELU
Loss Function: Sum-of-Squares
Weight Initialization: normal
Momentum: 0
Learning Rate : 0.01
Regularization: L1 & L2 Regularization , L1 = 0.001 , L2 = 0.001

Accuracy:

```
Training Time :  295.3686304092407 seconds
Training set accuracy
0.670534813319879
Test set accuracy
0.6649269311064718
Val set accuracy
0.6388744137571651
```

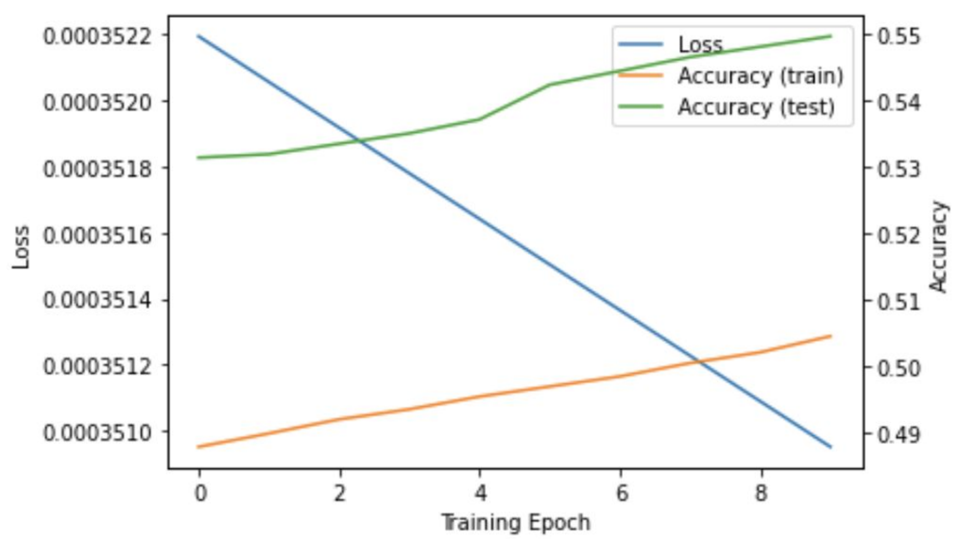


Second Highest Accuracy

Architecture : `[(75*75*3),  256,128, 16, 2]`
Activation Function : ReLU
Loss Function: Sum-of-Squares
Weight Initialization: normal
Momentum: 0
Learning Rate : 0.01
Regularization: L1 & L2 Regularization , L1 = 0.001 , L2 = 0.001

Accuracy:

```
Training set accuracy
0.5044399596367306
Test set accuracy
0.5495824634655533
Val set accuracy
0.5471599791558103
```

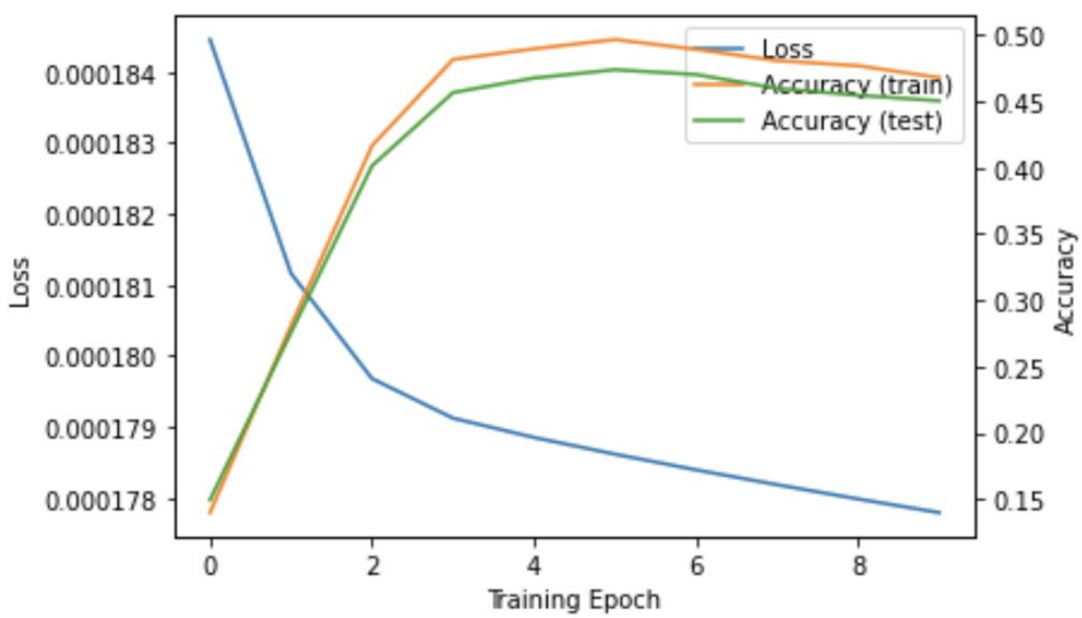


Third Highest Accuracy

Architecture : `[(75*75*3),  128, 16, 2]`
Activation Function : ELU
Loss Function: Sum-of-Squares
Weight Initialization: Normal
Momentum: 0
Learning Rate : 0.01
Regularization: L1 = 0 , L2 = 0

Accuracy:

```
Training set accuracy
0.4679112008072654
Test set accuracy
0.45041753653444677
Val set accuracy
0.4585721730067744
```



Conclusion

It is seen that Learning rate , Weight Initialization and Loss Function had the major impact on accuracy.

Future Work

More Detailed Feature Extraction using higher end models like CNN can provide better accuracy.
Images will be augmented in more numbers and to be tested on CNN Model.