Screenshots

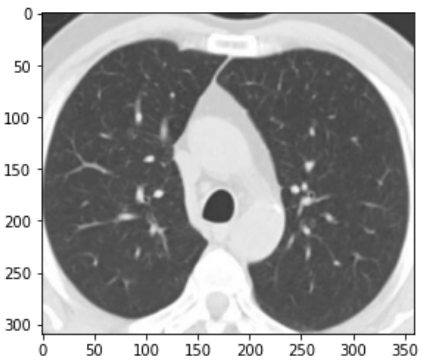
```
[16] import glob
     import matplotlib.image as mpimg
     nonCOVID_images = []
     for img_path in glob.glob(dataset + '/non-COVID/*'):
         nonCOVID_images.append(mpimg.imread(img_path))

     fig = plt.figure()
     fig.suptitle('non-COVID')
     plt.imshow(nonCOVID_images[0], cmap='gray')

     COVID_images = []
     for img_path in glob.glob(dataset + '/COVID/*'):
         COVID_images.append(mpimg.imread(img_path))

     fig = plt.figure()
     fig.suptitle('COVID')
     plt.imshow(COVID_images[0], cmap='gray')
```

```
<matplotlib.image.AxesImage at 0x7fc69a707590>
```

non-COVID


```
<matplotlib.image.AxesImage at 0x7fc69a707590>
```

non-COVID



COVID

Screenshots

```
[17] disease_types = ['COVID', 'non-COVID']

     train_dir = data_dir = '/content/gdrive/MyDrive/dataset/CapstoneProject'

     train_data = []

     for index, sp in enumerate(disease_types):
         for file in os.listdir(os.path.join(train_dir, sp)):
             train_data.append([sp + "/" + file, index, sp])

     train = pd.DataFrame(train_data, columns = ['File', 'ID','Disease Type'])
     train.head()
```

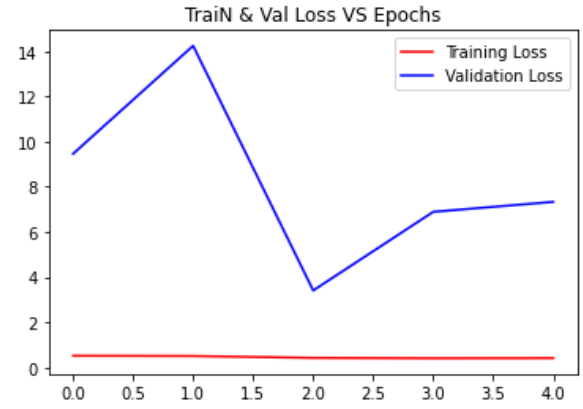
	File	ID	Disease Type
0	COVID/Covid (1).png	0	COVID
1	COVID/Covid (1005).png	0	COVID
2	COVID/Covid (1000).png	0	COVID
3	COVID/Covid (1004).png	0	COVID
4	COVID/Covid (1003).png	0	COVID

```
datagen.fit(X_train)
```

```
Model: "model_8"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_18 (InputLayer)       [(None, 64, 64, 3)]       0

 conv2d_8 (Conv2D)           (None, 64, 64, 3)         84

 resnet50 (Functional)       (None, None, None, 2048)  23587712

 global_average_pooling2d_8   (None, 2048)             0
 (GlobalAveragePooling2D)

 batch_normalization_16 (Bat  (None, 2048)             8192
 chNormalization)

 dropout_16 (Dropout)        (None, 2048)              0

 dense_8 (Dense)             (None, 256)               524544

 batch_normalization_17 (Bat  (None, 256)              1024
 chNormalization)

 dropout_17 (Dropout)        (None, 256)               0

 root (Dense)                (None, 2)                 514

=================================================================
```

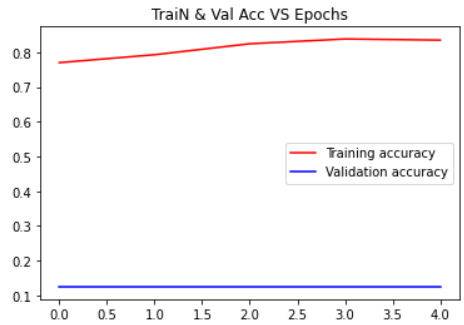
Screenshots

```
[61] epochs = range(len(acc))
     plt.plot(epochs, loss, 'r', label='Training Loss')
     plt.plot(epochs, val_loss, 'b', label='Validation Loss')
     plt.title('TraiN & Val Loss VS Epochs')
     plt.legend(loc=0)
     plt.figure()
     plt.show()
```




```
<Figure size 432x288 with 0 Axes>
```

```
[60] epochs = range(len(acc))
     plt.plot(epochs, acc, 'r', label='Training accuracy')
     plt.plot(epochs, val_acc, 'b', label='Validation accuracy')
     plt.title('TraiN & Val Acc VS Epochs')
     plt.legend(loc=0)
     plt.figure()
     plt.show()
```




```
<Figure size 432x288 with 0 Axes>
```

Screenshots

```
                        cf_matrix.flatten()/np.sum(cf_matrix)]
    labels = [f"{v1}\n{v2}\n{v3}" for v1, v2, v3 in
              zip(group_names,group_counts,group_percentages)]
    labels = np.asarray(labels).reshape(2,2)
    sns.heatmap(cf_matrix, annot = labels, fmt = '')
    plt.title("Confusion Matrix")
```

```
5/5 [==============================] - 4s 449ms/step
Text(0.5, 1.0, 'Confusion Matrix')
```