COVID-19 CT SCAN DATA CLASSIFICATION BY DEEP LEARNING

Name : AYUSHI JAIN
Reg. Number : 202639
Roll Number : 202CS003
Mobile : 9829337176
E-mail : ayushijain.202cs003@nitk.edu.in

Submitted To.: Dr. Jenny Rajan (CS737-Deep Learning)

National Institute of Technology Karnataka, Surathkal, Mangalore

Data Preparation:

The Data is provided by the instructor which is as given below:

Train		Test	
Covid-19	Non-Covid-19	Covid-19	Non-Covid-19
700	700	302	284
Total=1400		Total=586	

Data Set preparation is a very important part in order to train the deep learning model:

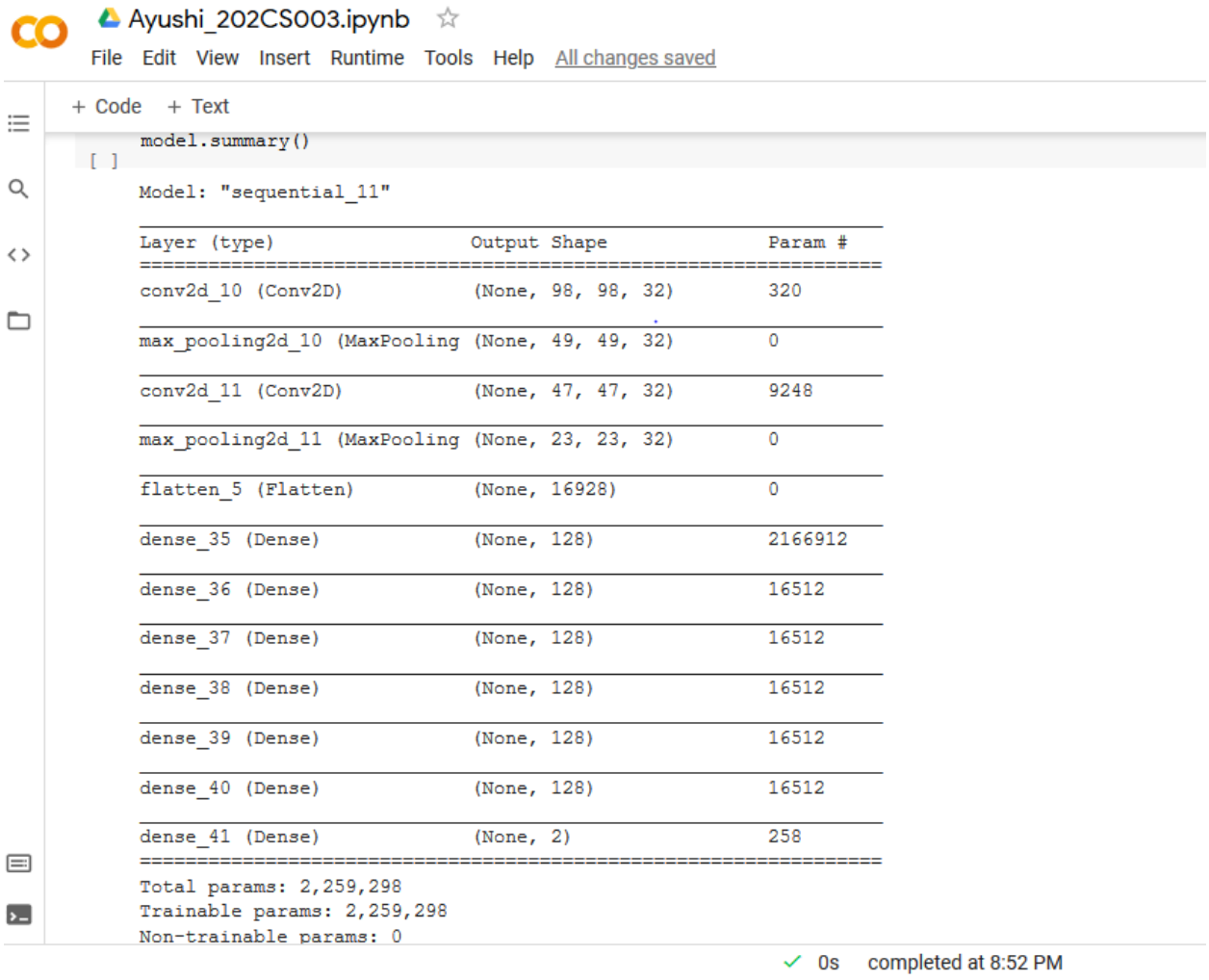
1. In this model the CT-SCAN images of both COVID and non-COVID resized into (100*100), as all the images must not be of same dimensions and fixed scaling of each image is required for which I have used OpenCV library,
2. Normalization of the data set is done by dividing it by 255 so that range becomes (0-1)
3. Data is augmented to increase the size training samples
4. I have classified the COVID images as 0 and non-COVID images as 1.
5. After this I have converted the images into feature lists

Architecture Used:

I have selected a sequential model for covid and non-covid classification. Model have 14 layers :
1. Convolution layer that will be having 32 filters and his kernel size will be 3*3 and the activation function used is RELU which is good as compared to other activation functions in a hidden layer.
2. dropout layer with dropout rate 0.5.
3. max-pooling of size 2*2 .
4. Convolution layer which will be having filters of size 32 and kernel size are having 3*3 and the activation function is RELU.
5. dropout layer with dropout rate 0.5.
6. max-pooling of size 2*2 .
7. a flattering layer that helps to convert the 2D size into a 1D size which will flatter the image.
8. 5 dense layers with RELU as activation function
9. the last layer is a dense layer which is having a activation as SIGMOID.

I have used an ADAM optimizer with a learning rate of 0.001 and have a loss function as a categorical cross-entropy and having a batch size of 32 and run for 30 epochs with validation split as 0.1

In the above screenshot image the number of trainable parameters and size is mentioned.



Novelty Adopted: I

While designing the architecture for the model, I have applied several standard architectures like VGG, AlexNet and GoogLeNet and found that accuracy of the model was not satisfactory hence I designed my own model. Other than that, for optimizers I have used Adam Optimizer. Also for the outer layer earlier I have used SoftMax activation function and then after experimenting switched to Sigmoid activation function. Lastly the loss function which is used is Categorical Cross Entropy loss function.

Results:

Test Results

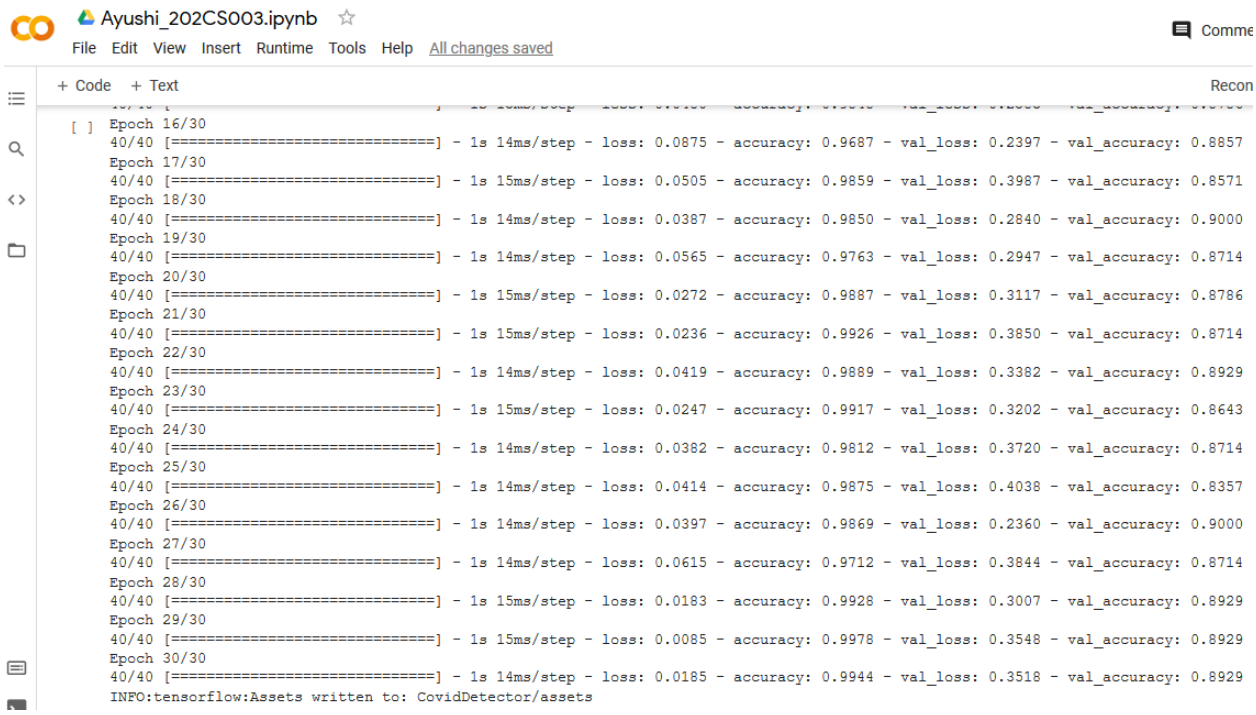
I ran the model for 30 epoch and got

Training accuracy = 99.44%.

Validation accuracy = 89.29%.

Training loss = 0.0185.

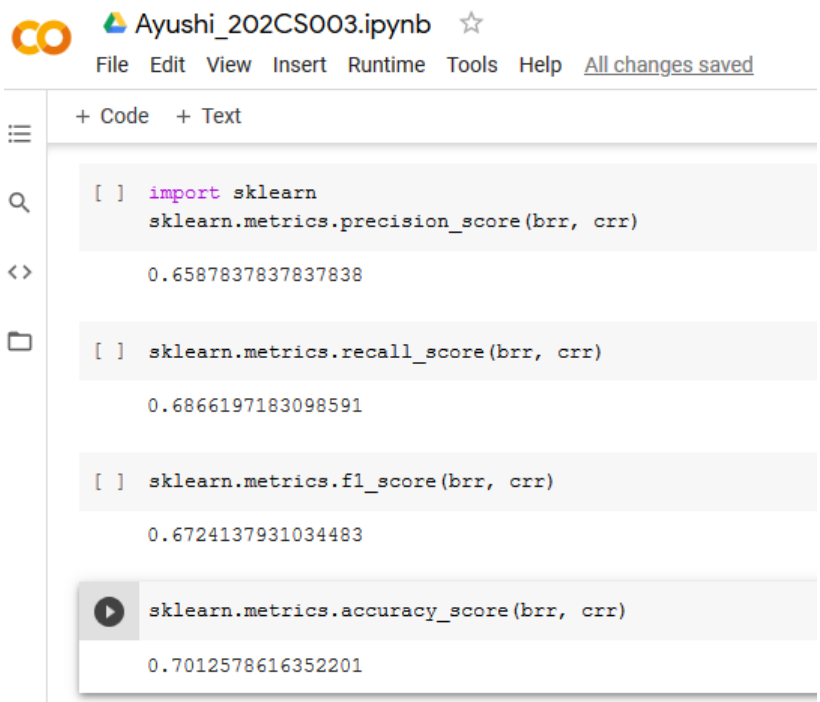
screenshots of the results:



Testing Results

Parameter	Value
Accuracy	**0.7012578616352201**
Precision	**0.6587537837837838**
Recall	**0.6866197183098591**
F1 score	**0.6724137931034483**
Confusion Matrix	**[[251 101]** **[89 195]]**

screenshots of the results.



Ayushi_202CS003.ipynb

File Edit View Insert Runtime Tools He

+ Code + Text

```
[ ] cm=confusion_matrix(brr, crr)
    print(cm)
```

```
[[251 101]
 [ 89 195]]
```

```
import seaborn as sn
import pandas as pd
import matplotlib.pyplot as plt
sn.set(font_scale=1.2) # for label size
sn.heatmap(cm, annot=True, annot_kws={"size": 24})
plt.show()
#categories=['COVID','non-COVID']
#          =[ 0     , 1         ]
```