

## RESULTS

The results for the CNN architecture are shown in figure 7. The graph shows a trade-off between epochs and accuracy. We can see from the two graphs that even though the number of epochs and the accuracy. The yellow line represents the validation accuracy while the blue line indicates the train accuracy. We can see that the accuracy is quite low at approximately 6%.

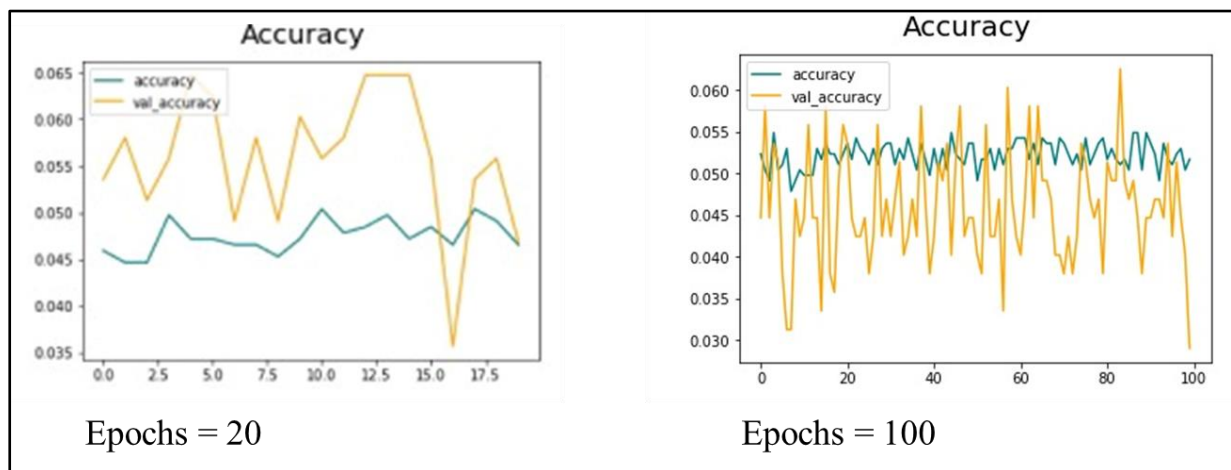


Figure 7: Accuracy graphs of CNN architecture with epochs as 20 and 100 respectively

The graph on the left-hand side with 20 epochs shows an accuracy of approximately 6%. Also, the accuracy drops drastically around the 16th epoch. On the right-hand side, the graph displays a trade-off between the number of epochs=100 and the accuracy. The accuracy keeps varying as the number of epochs increases but lowers down at the 100th epoch. The overall accuracy is 6%.

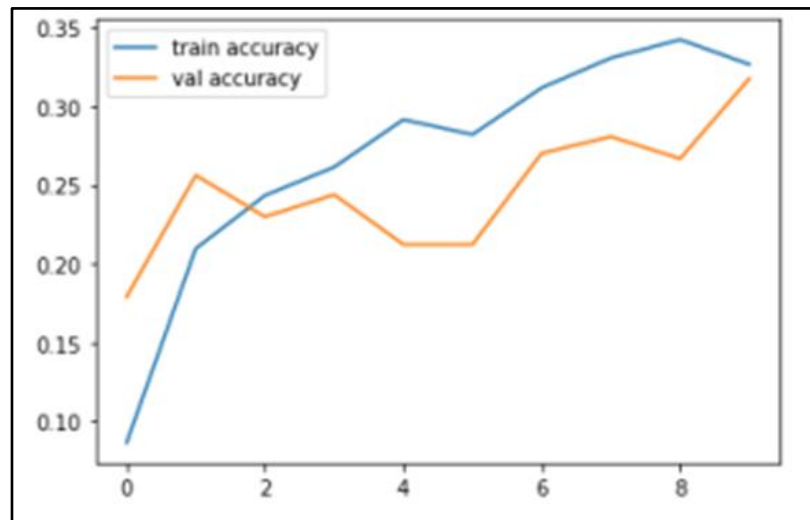


Figure 8: Accuracy graph for ResNet50

Figure 8 shows the graph of the number of epochs versus the accuracy. Here, the number of epochs was set to 10. We can observe from the graph that with the increase in the number of epochs, the accuracy displays fluctuations.

```
In [31]: model.evaluate(test_set, batch_size=512)

18/18 [=====] - 46s 3s/step - loss: 2.6761 - accuracy: 0.3140
Out[31]: [2.676133871078491, 0.3140350878238678]
```

Figure 9: ResNet50 model evaluation

The model was evaluated to obtain the test loss and test accuracy. The test loss is approximately 3 while the test accuracy is approximately 31%.

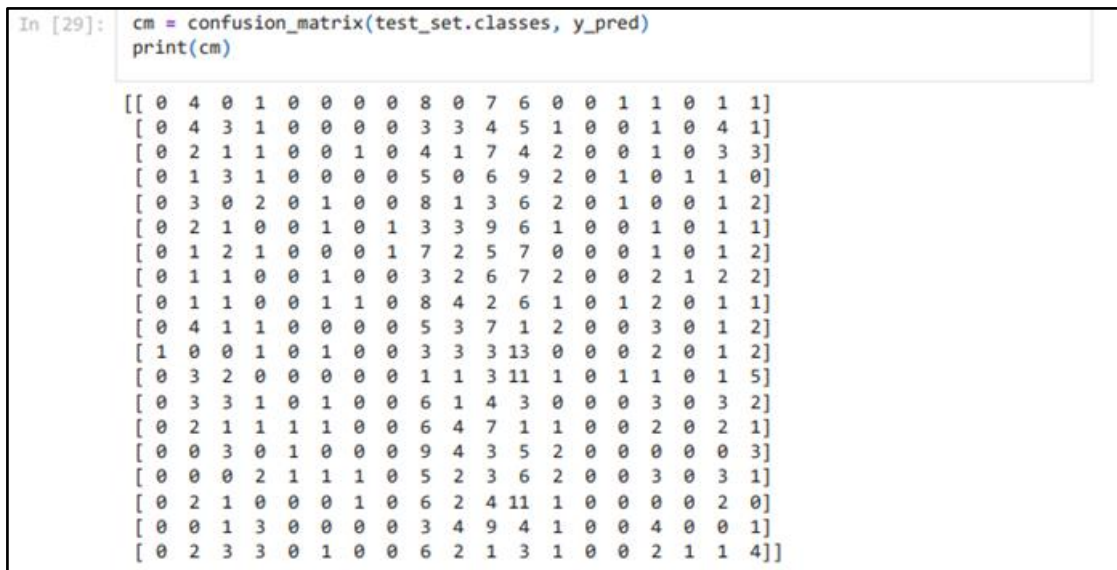


Figure 9: Confusion matrix for ResNet50

Prediction on test data is done using the trained model and a confusion matrix is plotted. A confusion matrix is basically a visualization of the prediction of classes in an image dataset. This confusion matrix consists of values from 0 to 18 which indicates the class number that is 19 classes.

While the development of the above-mentioned models and codes, we faced two major issues which proved to be an obstacle in the execution of the model.

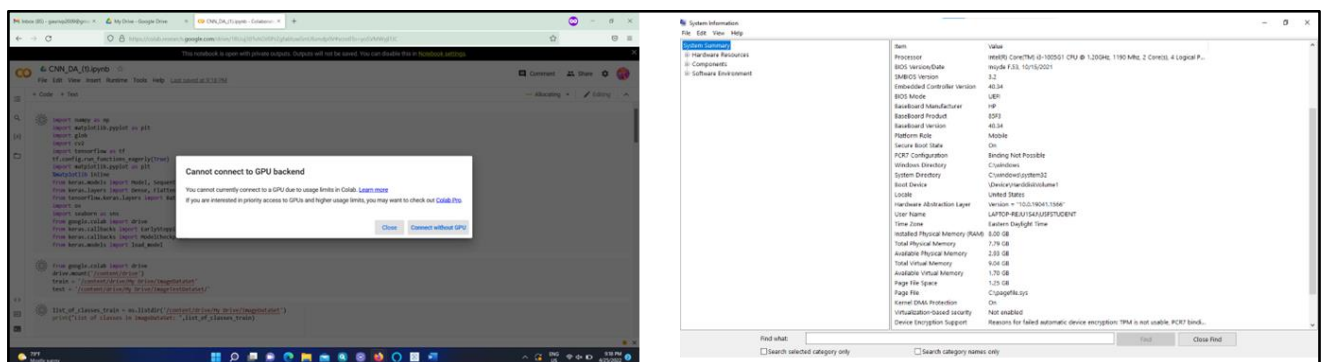


Figure 10: Issue of google colab and system configuration

The snapshot on the left-hand side displays an issue faced while executing the code in google colab. The google colab was not able to connect to the GPU due to which we had to execute the code for long hours thus increasing the response time of the system.

Also, the system configuration was i3 and other specifications were as shown in figure 10 (right-hand side) due to which the session crashed frequently.