Ahsanullah University of Science and Technology (AUST)
Department of Computer Science and Engineering

Project Proposal

Course No. : CSE4238
Course Title: Soft Computing Lab
Section : B
Group No : B1
Semester : Fall 2021

SUBMITTED BY : **ID : 17.01.04.054**

SUBMITTED DATE : 11/09/2021

Section 1:

Here is my model summary discussed in bellow :

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_4 (Dense)              (None, 20)                449880
_________________________________________________________________
dense_5 (Dense)              (None, 10)                210
_________________________________________________________________
dense_6 (Dense)              (None, 1024)              11264
_________________________________________________________________
batch_normalization_1 (Batch (None, 1024)              4096
_________________________________________________________________
dropout_1 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_7 (Dense)              (None, 1)                 1025
=================================================================
Total params: 466,475
Trainable params: 464,427
Non-trainable params: 2,048
_________________________________________________________________
```

Section 2:

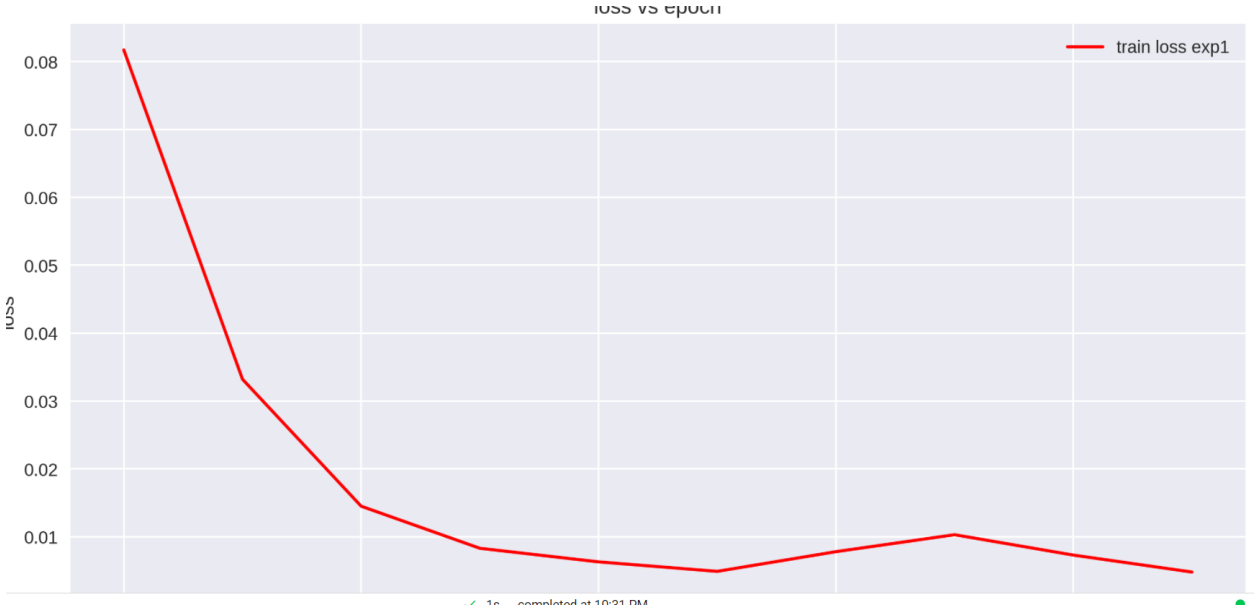
Here in the section below sectionI have shown after the number of epoch increases then what happens to its accuracy and loss. After 10 Epoch the Accuracy = .9989 and the loss = .0048.

```
Epoch 1/10
/usr/local/lib/python3.7/dist-packages/tensorflow/python/framework/indexed_slices.py:449: UserWarning: Converting sparse IndexedSlices(
  "shape. This may consume a large amount of memory." % value)
549/549 [==============================] - 16s 27ms/step - loss: 0.0817 - accuracy: 0.9737 - val_loss: 0.0313 - val_accuracy: 0.9942
Epoch 2/10
549/549 [==============================] - 15s 28ms/step - loss: 0.0332 - accuracy: 0.9919 - val_loss: 0.0358 - val_accuracy: 0.9951
Epoch 3/10
549/549 [==============================] - 15s 27ms/step - loss: 0.0145 - accuracy: 0.9976 - val_loss: 0.0413 - val_accuracy: 0.9947
Epoch 4/10
549/549 [==============================] - 15s 27ms/step - loss: 0.0083 - accuracy: 0.9987 - val_loss: 0.0335 - val_accuracy: 0.9956
Epoch 5/10
549/549 [==============================] - 14s 26ms/step - loss: 0.0063 - accuracy: 0.9994 - val_loss: 0.0361 - val_accuracy: 0.9966
Epoch 6/10
549/549 [==============================] - 15s 28ms/step - loss: 0.0049 - accuracy: 0.9993 - val_loss: 0.0443 - val_accuracy: 0.9951
Epoch 7/10
549/549 [==============================] - 15s 27ms/step - loss: 0.0078 - accuracy: 0.9985 - val_loss: 0.0341 - val_accuracy: 0.9956
Epoch 8/10
549/549 [==============================] - 15s 27ms/step - loss: 0.0103 - accuracy: 0.9985 - val_loss: 0.0582 - val_accuracy: 0.9947
Epoch 9/10
549/549 [==============================] - 14s 26ms/step - loss: 0.0073 - accuracy: 0.9984 - val_loss: 0.0594 - val_accuracy: 0.9951
Epoch 10/10
549/549 [==============================] - 16s 28ms/step - loss: 0.0048 - accuracy: 0.9989 - val_loss: 0.0584 - val_accuracy: 0.9951
```

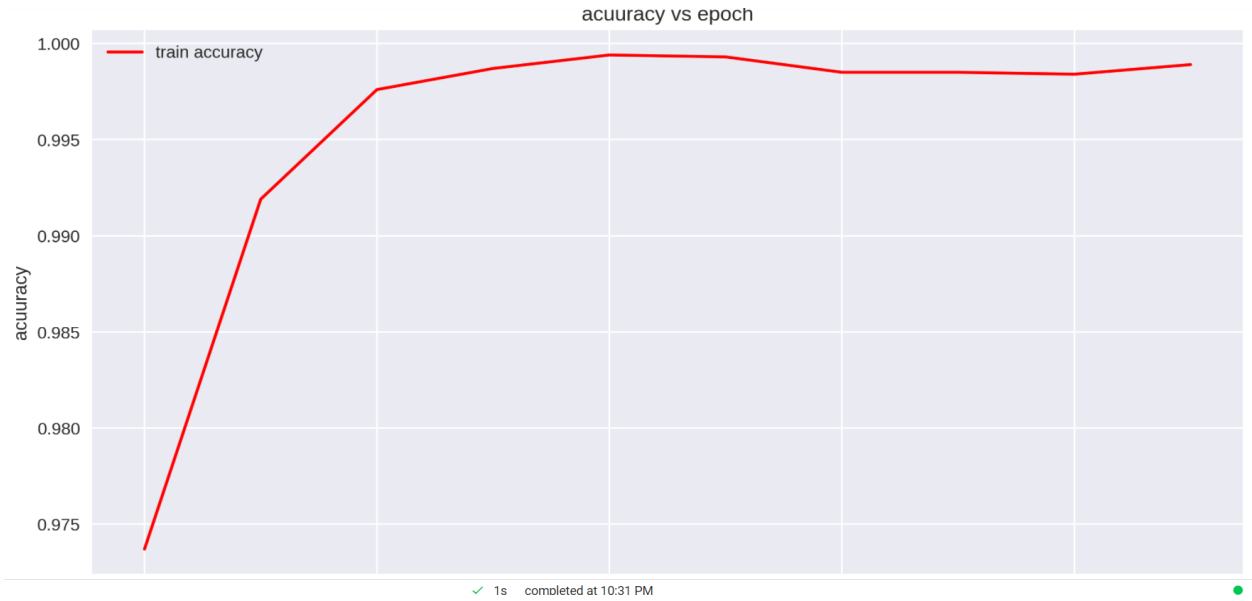
Section 3

We will see how our loss and accuracy behaves after the epoch increases.

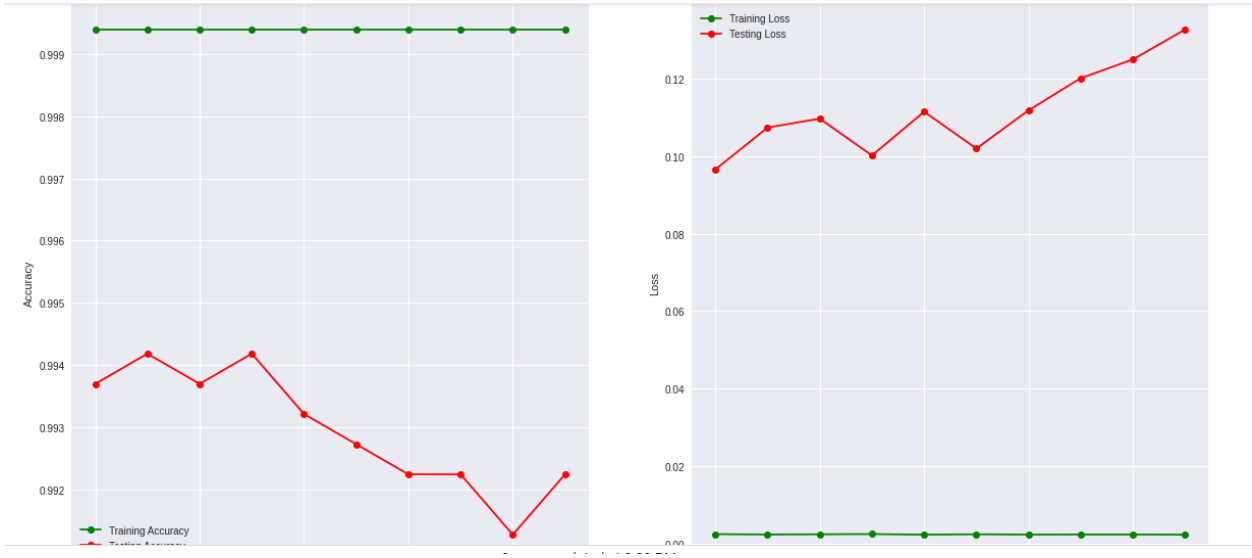
Loss vs Epoch



Accuracy vs Epoch



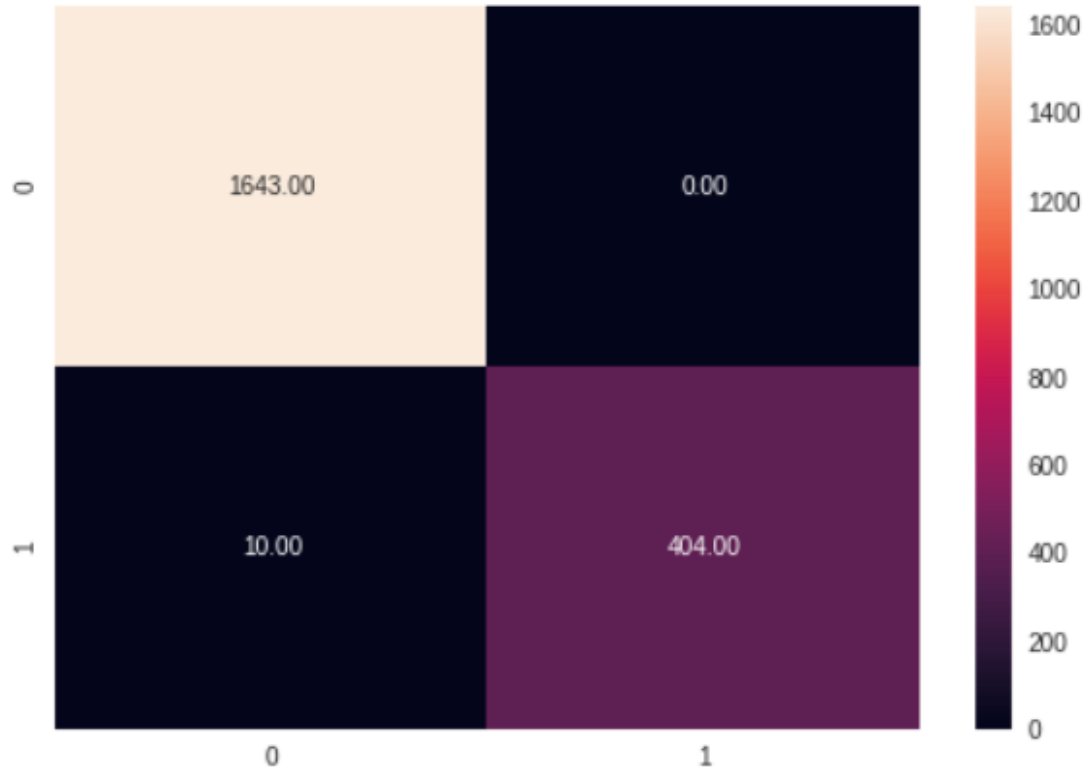
Now we will see the testing and training accuracy and loss.



Section 4

As my accuracy is higher that why in my confusion matrix we can see that the True positive and True Negative are very high**section1:**

```
For CNN
<matplotlib.axes._subplots.AxesSubplot at 0x7f75b7ce5a50>
```




```
For CNN :
              precision    recall  f1-score   support

           0       0.99      1.00      1.00      1643
           1       1.00      0.98      0.99       414

    accuracy                           1.00      2057
   macro avg       1.00      0.99      0.99      2057
weighted avg       1.00      1.00      1.00      2057
```

Github link: https://github.com/Jabed27/SoftComputingAssignment