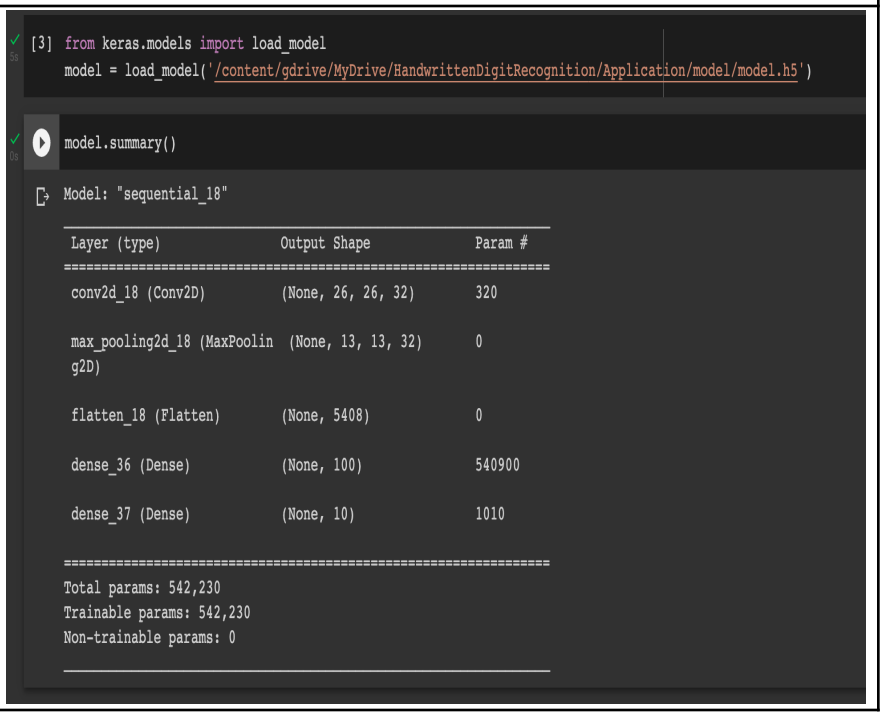
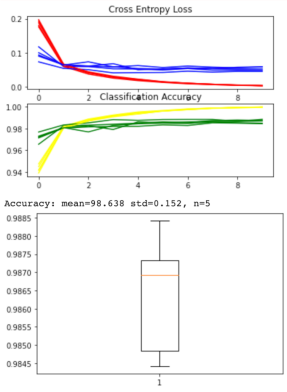
Project Development
Phase
Model Performance Test

Date	10 November 2022
Team ID	PNT2022TMID00805
Project Name	Project - A Novel Method for Handwritten Digit Application
Maximum Marks	10 Marks

Model Performance Testing:

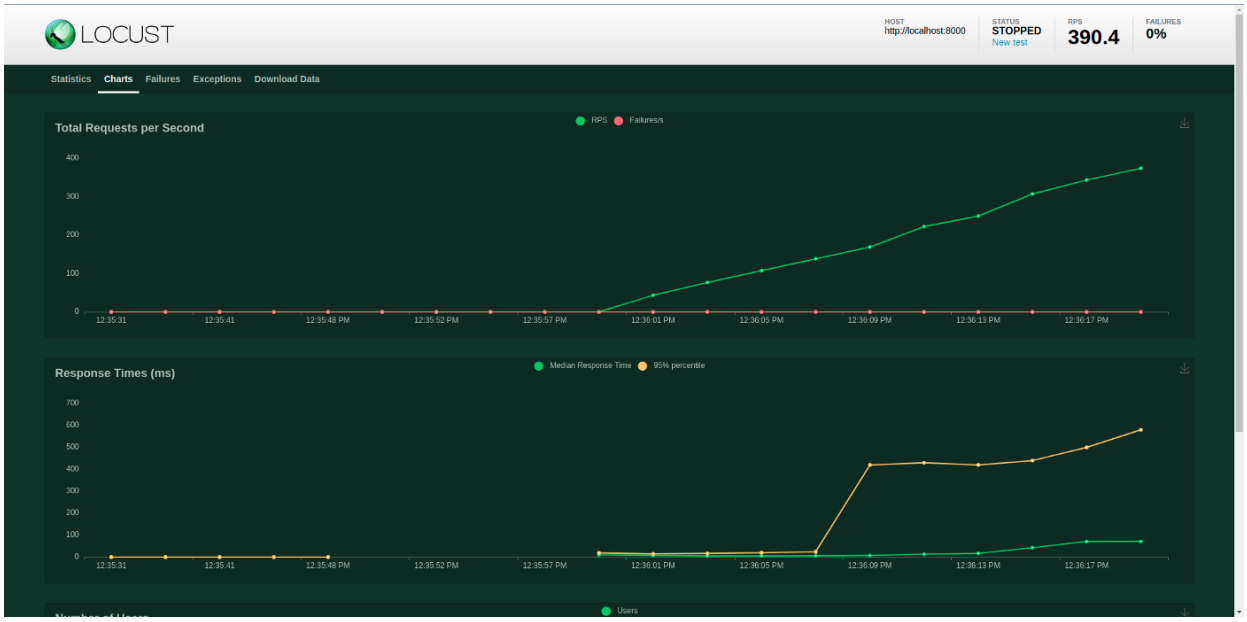
S.No.	Parameter	Values	Screenshot
1.	Model Summary	**model.summary()**	

2.	Accuracy	Training Accuracy -
		
		Validation Accuracy -
			98.638
			
			Learning Curve
			
			In [84]:
			`def show_learning_curve(histories):`
			`  for i in range(len(histories)):`
			`  """ Plot loss"""`
			`  plt.subplot(2, 1, 1)`
			`  plt.title('Cross Entropy Loss')`
			`  plt.plot(histories[i].history['loss'], color='red', label='train')`
			`  plt.plot(histories[i].history['val_loss'], color='blue', label='test')`
			`  """ Plot accuracy"""`
			`  plt.subplot(2, 1, 2)`
			`  plt.title('Classification Accuracy')`
			`  plt.plot(histories[i].history['accuracy'], color='yellow', label='train')`
			`  plt.plot(histories[i].history['val_accuracy'], color='green', label='test')`
			`  plt.show()`
			
			> 98.483
			> 98.733
			> 98.442
			> 98.842
			> 98.692
			
			/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:4: MatplotlibDeprecationWarning: Adding an axes using the same arguments as a previous axes currently reuses the earlier instance. In a future version, a new instance will always be created and returned. Meanwhile, this warning can be suppressed, and the future behavior ensured, by passing a unique label to each axes instance.
			after removing the cwd from sys.path.
			/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:9: MatplotlibDeprecationWarning: Adding an axes using the same arguments as a previous axes currently reuses the earlier instance. In a future version, a new instance will always be created and returned. Meanwhile, this warning can be suppressed, and the future behavior ensured, by passing a unique label to each axes instance.
			if __name__ == '__main__':
			 
			Accuracy: mean=98.638 std=0.152, n=5

Performance Testing

It was done with locust.py

locust.py

```
from locust import HttpUser, task, between
import os
```

```python
def getImg(filePath):
     _fileName = os.path.basename(filePath)
     _fileContent = open(filePath,'rb')
     return _fileName, _fileContent, 'file'

_files= {
   "Images":getImg("../Test Images/2.jpeg"),
   "Images":getImg("../Test Images/3.jpeg"),
}
headers = {'content-type': 'multipart/form-data'}

class PerformanceTest(HttpUser):

   @task
   def home(self):
       r = self.client.post("", files=_files, headers = headers)
       print(r)
```