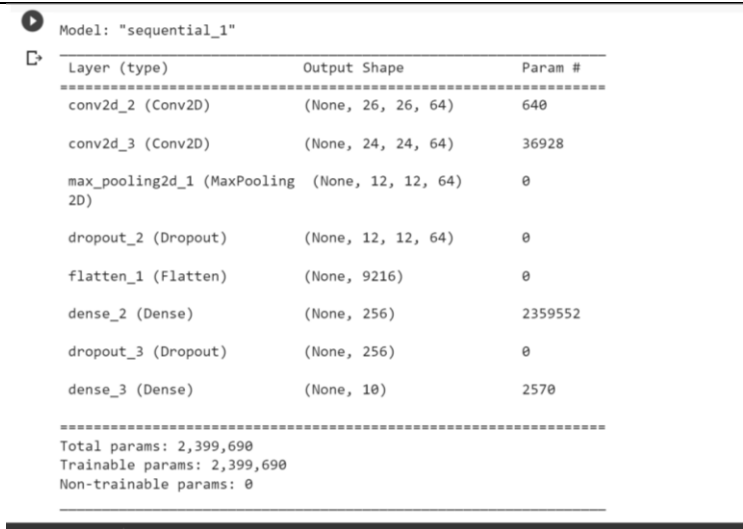
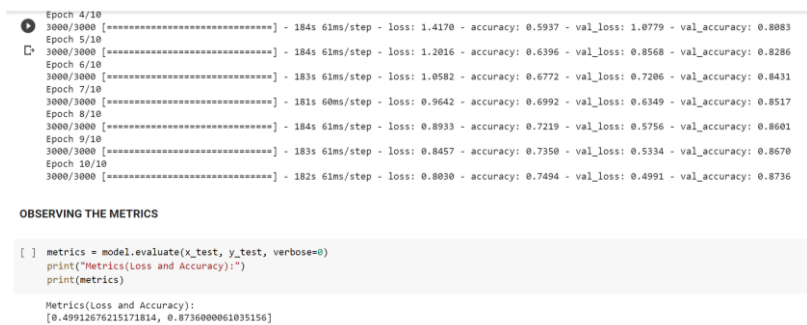


Model Performance Test

Date	14th November 2022
Team ID	PNT2022TMID28868
Project Name	A Novel Method for Handwritten Digit Recognition System
Maximum Marks	10 Marks

Model Performance Testing:

S.No	Parameter	Values	Screenshot																											
1.	Model Summary		 <p>Model: "sequential_1"</p> <table><thead><tr><th>Layer (type)</th><th>Output Shape</th><th>Param #</th></tr></thead><tbody><tr><td>conv2d_2 (Conv2D)</td><td>(None, 26, 26, 64)</td><td>640</td></tr><tr><td>conv2d_3 (Conv2D)</td><td>(None, 24, 24, 64)</td><td>36928</td></tr><tr><td>max_pooling2d_1 (MaxPooling 2D)</td><td>(None, 12, 12, 64)</td><td>0</td></tr><tr><td>dropout_2 (Dropout)</td><td>(None, 12, 12, 64)</td><td>0</td></tr><tr><td>flatten_1 (Flatten)</td><td>(None, 9216)</td><td>0</td></tr><tr><td>dense_2 (Dense)</td><td>(None, 256)</td><td>2359552</td></tr><tr><td>dropout_3 (Dropout)</td><td>(None, 256)</td><td>0</td></tr><tr><td>dense_3 (Dense)</td><td>(None, 10)</td><td>2570</td></tr></tbody></table> <p>Total params: 2,399,690 Trainable params: 2,399,690 Non-trainable params: 0</p>	Layer (type)	Output Shape	Param #	conv2d_2 (Conv2D)	(None, 26, 26, 64)	640	conv2d_3 (Conv2D)	(None, 24, 24, 64)	36928	max_pooling2d_1 (MaxPooling 2D)	(None, 12, 12, 64)	0	dropout_2 (Dropout)	(None, 12, 12, 64)	0	flatten_1 (Flatten)	(None, 9216)	0	dense_2 (Dense)	(None, 256)	2359552	dropout_3 (Dropout)	(None, 256)	0	dense_3 (Dense)	(None, 10)	2570
Layer (type)	Output Shape	Param #																												
conv2d_2 (Conv2D)	(None, 26, 26, 64)	640																												
conv2d_3 (Conv2D)	(None, 24, 24, 64)	36928																												
max_pooling2d_1 (MaxPooling 2D)	(None, 12, 12, 64)	0																												
dropout_2 (Dropout)	(None, 12, 12, 64)	0																												
flatten_1 (Flatten)	(None, 9216)	0																												
dense_2 (Dense)	(None, 256)	2359552																												
dropout_3 (Dropout)	(None, 256)	0																												
dense_3 (Dense)	(None, 10)	2570																												
2.	Accuracy	Training Accuracy – 74.94 Validation Accuracy - 87.23000	 <p>Epoch 4/10 3000/3000 [=====] - 184s 61ms/step - loss: 1.4170 - accuracy: 0.5937 - val_loss: 1.0779 - val_accuracy: 0.8083 Epoch 5/10 3000/3000 [=====] - 184s 61ms/step - loss: 1.2016 - accuracy: 0.6396 - val_loss: 0.8568 - val_accuracy: 0.8286 Epoch 6/10 3000/3000 [=====] - 183s 61ms/step - loss: 1.0582 - accuracy: 0.6772 - val_loss: 0.7206 - val_accuracy: 0.8431 Epoch 7/10 3000/3000 [=====] - 181s 60ms/step - loss: 0.9642 - accuracy: 0.6992 - val_loss: 0.6349 - val_accuracy: 0.8517 Epoch 8/10 3000/3000 [=====] - 184s 61ms/step - loss: 0.8933 - accuracy: 0.7219 - val_loss: 0.5756 - val_accuracy: 0.8601 Epoch 9/10 3000/3000 [=====] - 183s 61ms/step - loss: 0.8457 - accuracy: 0.7350 - val_loss: 0.5334 - val_accuracy: 0.8670 Epoch 10/10 3000/3000 [=====] - 182s 61ms/step - loss: 0.8030 - accuracy: 0.7494 - val_loss: 0.4991 - val_accuracy: 0.8736</p> <p>OBSERVING THE METRICS</p> <pre>[] metrics = model.evaluate(x_test, y_test, verbose=0) print("Metrics(Loss and Accuracy):") print(metrics) Metrics(Loss and Accuracy): [0.49912676215171814, 0.8736000061035156]</pre>																											