

Sprint 2

Date	1 November 2022
Team ID	PNT2022TMID17749
Project Name	A novel method for handwritten digit recognition system.
Maximum Marks	4 Marks

Sprint 2:

Building the machine learning model:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Activation, Flatten, Conv2D, MaxPooling2D

model = Sequential()

model.add(Conv2D(32, (3,3), activation = 'relu', input_shape= x_trainer.shape[1:]))
# model.add(MaxPooling2D((2,2)))

model.add(Conv2D(64, (3,3), activation = 'relu'))
model.add(MaxPooling2D((2,2)))
model.add(Dropout(0.25))

# model.add(Conv2D(64, (3,3), activation = 'relu'))
# model.add(MaxPooling2D((2,2)))

model.add(Flatten())

model.add(Dense(256, activation = 'relu'))
model.add(Dropout(0.5))
model.add(Dense(10, activation = 'softmax'))
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 26, 26, 32)	320
conv2d_1 (Conv2D)	(None, 24, 24, 64)	18496
max_pooling2d (MaxPooling2D)	(None, 12, 12, 64)	0
dropout (Dropout)	(None, 12, 12, 64)	0
flatten (Flatten)	(None, 9216)	0
dense (Dense)	(None, 256)	2359552
dropout_1 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 10)	2570

=====
Total params: 2,380,938
Trainable params: 2,380,938
Non-trainable params: 0
=====

```
model.compile(optimizer = 'adam' , loss = 'sparse_categorical_crossentropy' ,  
metrics = ['accuracy'])
```

```
model.fit(x_train,y_train, epochs = 5 , validation_split = 0.3 , batch_size  
= 128,verbose=1)
```

```
Epoch 1/5  
329/329 [=====] - 136s 410ms/step - loss: 0.3023 - accuracy: 0.9078 - val_loss: 0.0832 - val_accuracy: 0.9746  
Epoch 2/5  
329/329 [=====] - 142s 432ms/step - loss: 0.0904 - accuracy: 0.9720 - val_loss: 0.0633 - val_accuracy: 0.9812  
Epoch 3/5  
329/329 [=====] - 133s 404ms/step - loss: 0.0644 - accuracy: 0.9815 - val_loss: 0.0535 - val_accuracy: 0.9840  
Epoch 4/5  
329/329 [=====] - 139s 424ms/step - loss: 0.0511 - accuracy: 0.9848 - val_loss: 0.0469 - val_accuracy: 0.9859  
Epoch 5/5  
329/329 [=====] - 134s 408ms/step - loss: 0.0418 - accuracy: 0.9865 - val_loss: 0.0438 - val_accuracy: 0.9866  
<keras.callbacks.History at 0x7fe83bd58950>
```

```
test_loss, test_acc = model.evaluate(x_tester, y_test)  
print('Test loss on 10,000 test samples' , test_loss)  
print('Validation Accuracy on 10,000 samples' , test_acc)
```

```
313/313 [=====] - 9s 28ms/step - loss: 0.0353 - accuracy: 0.9890  
Test loss on 10,000 test samples 0.03527863696217537  
Validation Accuracy on 10,000 samples 0.9890000224113464
```

```
predictions = model.predict([x_tester])  
  
print(np.argmax(predictions[54]))
```

```
plt.imshow(x_test[54])
```

```
313/313 [=====] - 9s 27ms/step
```

```
6
```

```
<matplotlib.image.AxesImage at 0x7fe83930aa50>
```

