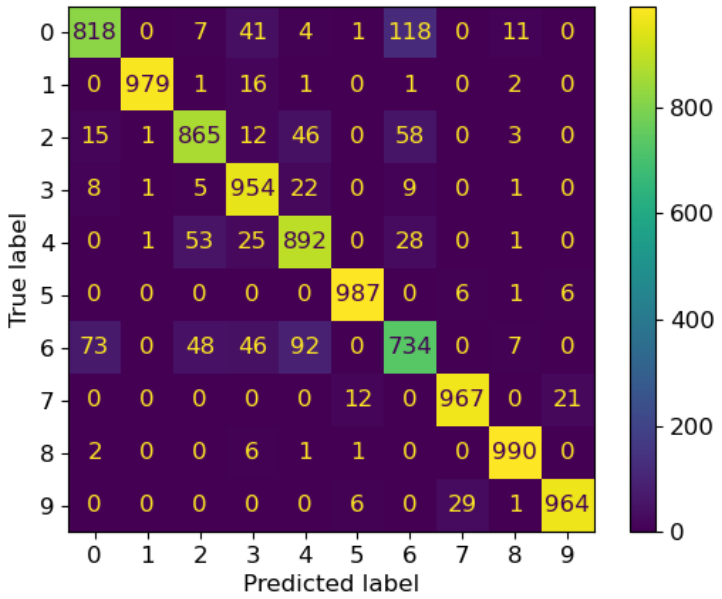
COEN 140
Lab 8 Report
Stephen Tambussi - 00001469512

Part 1 - Recognition accuracy and confusion matrix
Recognition accuracy of model on test set: **0.9150000214576721 (~91.50%)**



Part 2 - Calculation of model layers' parameters, output dimensions, and multiplications

1 - Input Layer

```
parameters = 0
print("Parameters = ", parameters)
output[0] = 28
output[1] = 28
output[2] = 1
print("Output dimensions = ", output)
num_multiplications = 0
print("Number of multiplications = ", num_multiplications)
```

Parameters = 0
Output dimensions = [28 28 1]
Number of multiplications = 0

2 - 2D Convolutional Layer

```
parameters = (3*3*1 + 1)*32 #320
print("Parameters = ", parameters)
output[0] = math.floor((28 + 2*1 - 3) / 1) + 1
output[1] = output[0]
```

```
output[2] = 32
print("Output dimensions = ", output)
num_multiplications = (3*3*1 + 1) * output[0] * output[1] * output[2]
#250,880
print("Number of multiplications = ", num_multiplications)
```

Parameters = 320
Output dimensions = [28 28 32]
Number of multiplications = 250880

3 - Max Pooling Layer

```
parameters = 0
print("Parameters = ", parameters)
output[0] = math.floor((28 + 2*0 - 2) / 2) + 1
output[1] = output[0]
output[2] = 32 #keep same depth dimension as previous layer
print("Output dimensions = ", output)
num_multiplications = 0
print("Number of multiplications = ", num_multiplications)
```

Parameters = 0
Output dimensions = [14 14 32]
Number of multiplications = 0

4 - 2D Convolutional Layer

```
parameters = (3*3*32 + 1)*64 #18496
print("Parameters = ", parameters)
output[0] = math.floor((14 + 2*1 - 3) / 1) + 1
output[1] = output[0]
output[2] = 64
print("Output dimensions = ", output)
num_multiplications = (3*3*32 + 1) * output[0] * output[1] * output[2]
print("Number of multiplications = ", num_multiplications)
```

Parameters = 18496
Output dimensions = [14 14 64]
Number of multiplications = 3625216

5 - Max Pooling Layer

```
parameters = 0
print("Parameters = ", parameters)
output[0] = math.floor((14 + 2*0 - 2) / 2) + 1
output[1] = output[0]
output[2] = 64 #keep same depth dimension as previous layer
print("Output dimensions = ", output)
num_multiplications = 0
```

```
print("Number of multiplications = ", num_multiplications)
```

Parameters = 0
Output dimensions = [7 7 64]
Number of multiplications = 0

6 - 2D Convolutional Layer

```
parameters = (3*3*64 + 1)*64 #36928
print("Parameters = ", parameters)
output[0] = math.floor((7 + 2*1 - 3) / 1) + 1
output[1] = output[0]
output[2] = 64
print("Output dimensions = ", output)
num_multiplications = (3*3*64 + 1) * output[0] * output[1] * output[2]
print("Number of multiplications = ", num_multiplications)
```

Parameters = 36928
Output dimensions = [7 7 64]
Number of multiplications = 1809472

7 - Flattening Layer

```
parameters = 0
print("Parameters = ", parameters)
output[0] = output[0] * output[0] * 64
output[1] = 1
output[2] = 1
print("Output dimensions = ", output[0])
num_multiplications = 0
print("Number of multiplications = ", num_multiplications)
```

Parameters = 0
Output dimensions = 3136
Number of multiplications = 0

8 - Fully-connected Layer

```
parameters = (output[0] + 1) * 64 #(3136 + 1) * 64
print("Parameters = ", parameters)
output[0] = 64
output[1] = 1
output[2] = 1
print("Output dimensions = ", output[0])
num_multiplications = (3136 + 1) * 64
print("Number of multiplications = ", num_multiplications)
```

Parameters = 200768
Output dimensions = 64
Number of multiplications = 200768

9 - Fully-connected Output Layer

```
parameters = (output[0] + 1) * 10 #(64 + 1) * 10
print("Parameters = ", parameters)
output[0] = 10
output[1] = 1
output[2] = 1
print("Output dimensions = ", output[0])
num_multiplications = (64 + 1) * 10
print("Number of multiplications = ", num_multiplications)
```

Parameters = 650
Output dimensions = 10
Number of multiplications = 650

model.summary()

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 28, 28, 32)	320
max_pooling2d (MaxPooling2D)	(None, 14, 14, 32)	0
conv2d_1 (Conv2D)	(None, 14, 14, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 7, 7, 64)	0
conv2d_2 (Conv2D)	(None, 7, 7, 64)	36928
flatten (Flatten)	(None, 3136)	0
dense (Dense)	(None, 64)	200768
dense_1 (Dense)	(None, 10)	650

Total params: 257,162
Trainable params: 257,162
Non-trainable params: 0

Part 3 - Comparison of CNN and Lab 7 Neural Network

CNN:

Recognition accuracy of model on test set: **~91.50%**
Total parameters: **257,162**
Total multiplications: **5,886,986**

Neural Network:

Recognition accuracy of model on test set: **~86.31%**
Total parameters: **407,050**
Total multiplications: **407,050**

The CNN has a higher recognition accuracy than the Neural Network, while also having a smaller amount of parameters compared to the Neural Network. However, the total number of multiplications for the CNN is much greater than the Neural Network. This is because the CNN has more layers than the Neural Network and has convolutional layers. The convolutional layers have a filter which shifts across the input image, horizontally and vertically computing dot products.