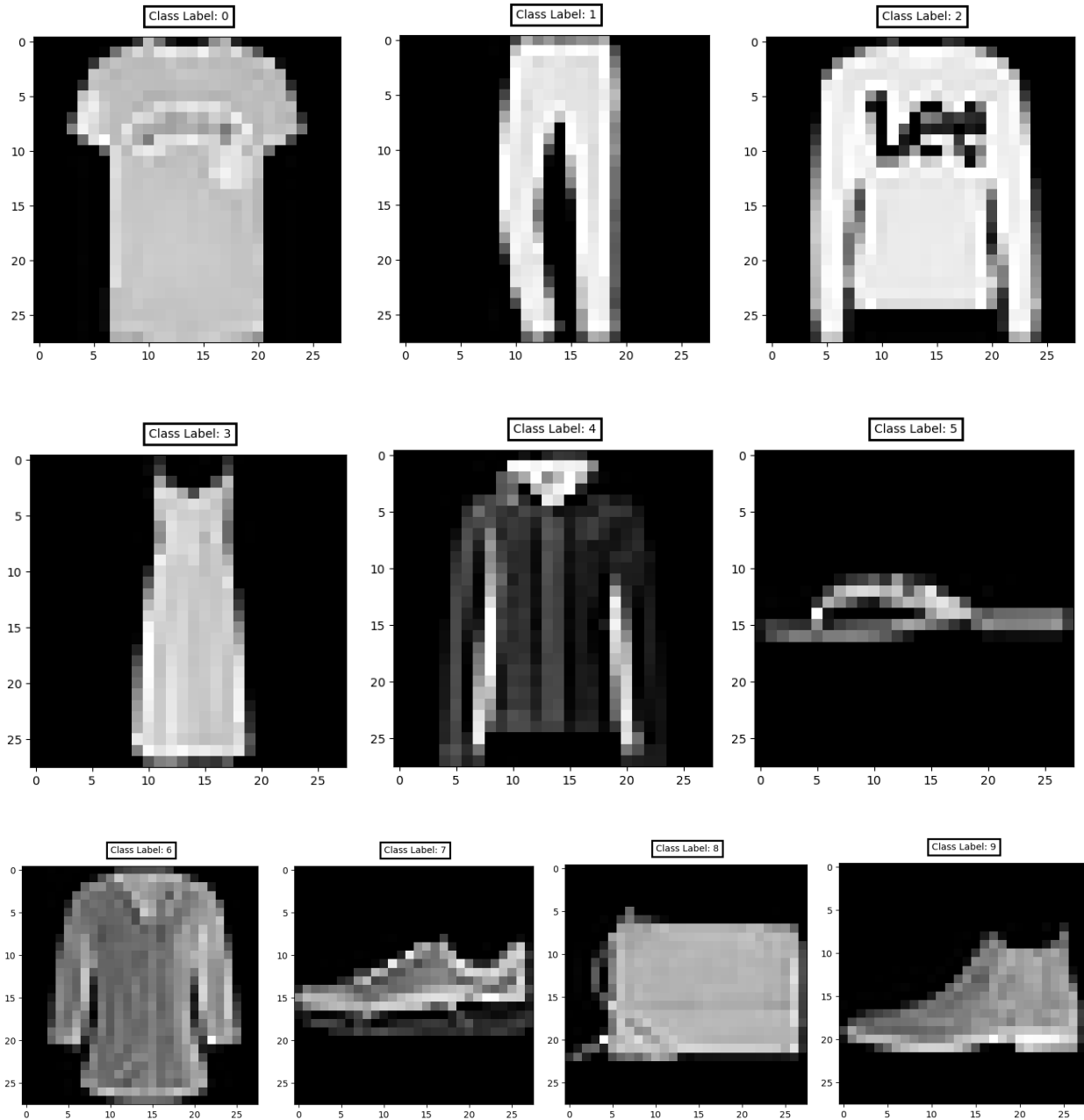


COEN 140

Lab 7 Report

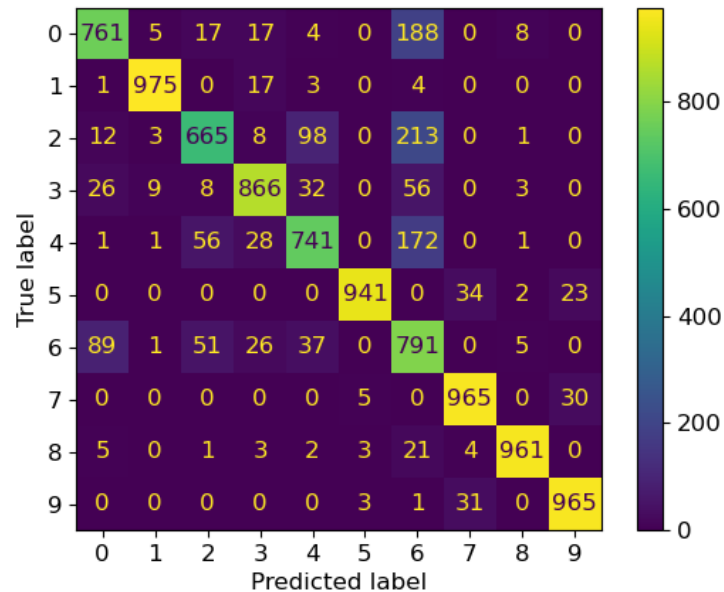
Stephen Tambussi - 00001469512

Part 1 - 10 images from test set



Part 2 - Recognition accuracy and confusion matrix

Recognition accuracy of model on testing set: **0.863099992275238 (~86.31%)**



Part 3 - Calculation of model parameters

Code

```
flattened_input = 28 * 28 #since each image is 28x28 = 784
hidden_layer = (flattened_input + 1) * 512 #1 is added for the bias
output_layer = (512 + 1) * 10 #1 is added for the bias
total_weights = hidden_layer + output_layer
```

Total number of parameters (weights with bias) in model = **407050**

Verification(model.summary())

Model: "sequential"

Layer (type)	Output Shape	Param #
flatten (Flatten)	(None, 784)	0
dense (Dense)	(None, 512)	401920
dense_1 (Dense)	(None, 10)	5130

Total params: **407,050**

Trainable params: **407,050**

Non-trainable params: 0

Part 4 - Calculation of the number of multiplications for each layer

Code

```
#Hidden layer number of multiplications = (sample dimensions + bias) *  
(output dimension of hidden layer)  
#(28*28 + 1) * 512  
hidden_layer_mult = (28 * 28 + 1) * 512  
print("Number of multiplications for hidden layer = ", hidden_layer_mult)  
#Output layer number of multiplications = (output dimension of hidden  
layer (input to this layer) + bias) * (output dimension of output layer)  
#(512*1 + 1) * 10  
output_layer_mult = (512 * 1 + 1) * 10  
print("Number of multiplications for output layer = ", output_layer_mult)  
print()  
total_mult = hidden_layer_mult + output_layer_mult  
print("Total number of multiplications to train model = ", total_mult)
```

Number of multiplications for hidden layer = **401920**

Number of multiplications for output layer = **5130**

Total number of multiplications to train model = **407050**