CMPT 762

Computer Vision

Assignment 1

Gokul Mohanarangan
gma56@sfu.ca

301436162

Free-Late day: 3 late days given exclusively for Assignment 1 are intended to be used here.

Assumption: using relative paths without knowing the current working directory is impossible. i.e.) Running from SFUID/ , SFUID/ec/ or SFUID/matlab/ all lead to different relative paths as the current working directory is different in each case. The assumption here is that to run files in ec, the current working directory will be SFUID/ec/. Similarly for matlab files, SFUID/matlab/.

PART 1: Forward pass : Visualizations rendered by “test_components.m” :

Figure 1:

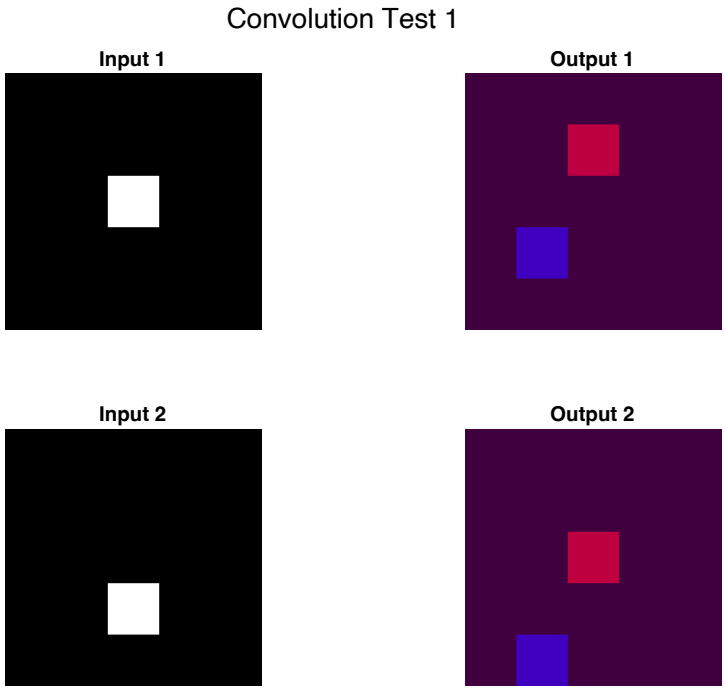


Figure 2:

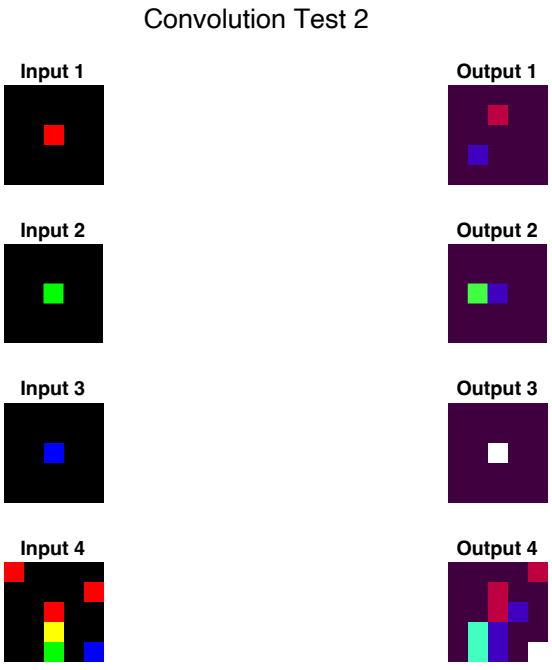


Figure 3:

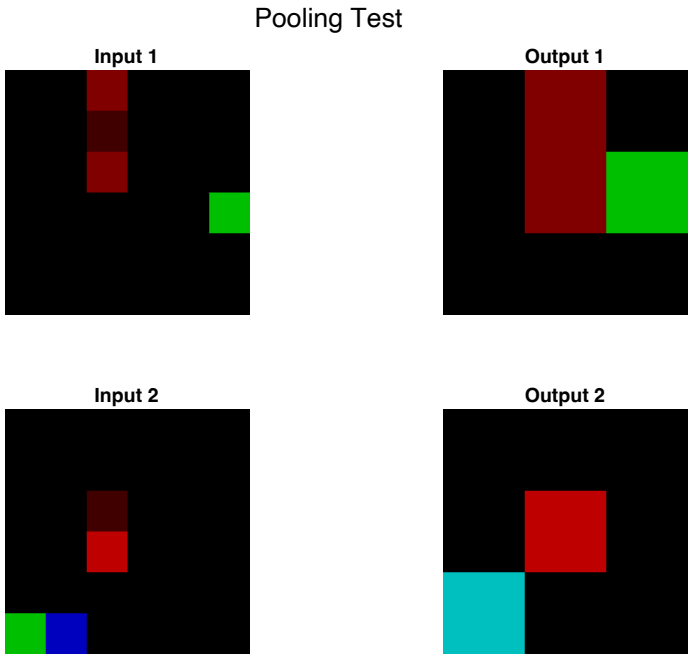
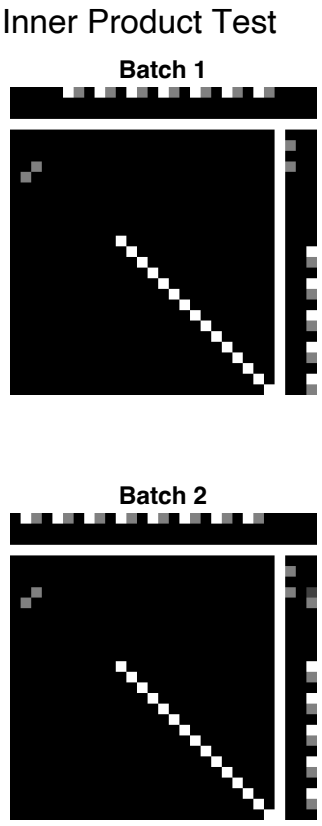


Figure 4:



PART 3 : Q 3.1 : Test accuracy:

The model was already trained for 3000 iterations and it had an initial test accuracy of around 92%. After training for 3000 more iterations, the final test accuracy was 97%.

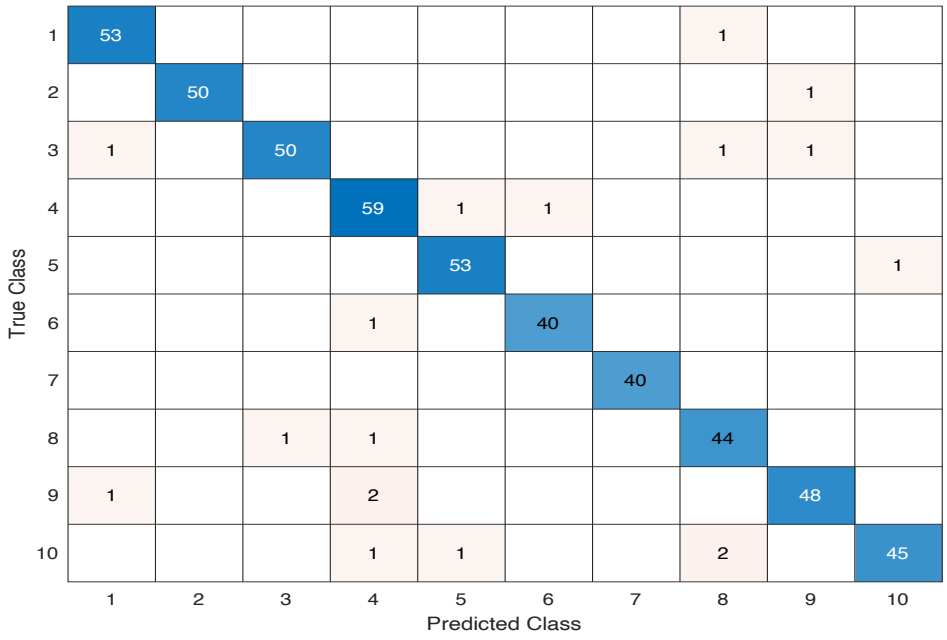
Q 3.2 Test network :

Upon running the “test_network.m” we now obtain a confusion matrix

```
>> test_network
    53     0     0     0     0     0     0     1     0     0
     0    50     0     0     0     0     0     0     1     0
     1     0    50     0     0     0     0     1     1     0
     0     0     0    59     1     1     0     0     0     0
     0     0     0     0    53     0     0     0     0     1
     0     0     0     1     0    40     0     0     0     0
     0     0     0     0     0     0    40     0     0     0
     0     0     1     1     0     0     0    44     0     0
     1     0     0     2     0     0     0     0    48     0
     0     0     0     1     1     0     0     2     0    45
```

For easy interpretation, I have also created a confusion chart that looks like this.

Confusion chart:



We have to note that the confusion matrix is not always the same as the data points for training and testing are computed at random everytime, we start the network. Here, we can calculate the class with the highest prediction error from the true classes and the predicted classes. The error percentages are as follows:

0	0	0	0	0	0	0	0.0189	0	0
0	0	0	0	0	0	0	0	0.0200	0
0.0200	0	0	0	0	0	0	0.0200	0.0200	0
0	0	0	0	0.0169	0.0169	0	0	0	0
0	0	0	0	0	0	0	0	0	0.0189
0	0	0	0.0250	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0.0227	0.0227	0	0	0	0	0	0
0.0208	0	0	0.0417	0	0	0	0	0	0
0	0	0	0.0222	0.0222	0	0	0.0444	0	0

The pair of classes with the highest confusion is {7, 9} and {3, 8}.

Digits are not represented a whole image, instead they are broken down into many component features that together makeup one digit. Each feature lights up corresponding neurons. This combination of excited neurons leads to the values in the final output layer.

Between 7 and 9, some of the features seem to overlap. For example, the somewhat small straight line, and a small, curved line. The digits also look similar. We can easily convert these two digits into each other with less changes. The neurons light up on seeing these features in both 7and 9, leading to a misclassification.

Between 3 and 8, both have somewhat rounded shapes and connecting components between the rounded shapes etc. Here also, the digits look similar. We can easily convert these two digits into each other with less changes. Hence, the corresponding neurons light up together in both 3 and 8. This maybe the reason why these classes are mis predicted.

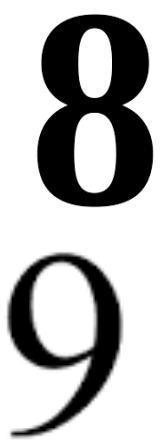
Q 3.3 :

5 images were obtained from the internet, each containing 1 digit (either full image or cropped to contain a digit) and fed to the network. The code is in file “ec/ec_real_world_testing.m”. The input images are present in “images/real_world_testing”. The probabilities of predicition are also shown here.

```
P =

    0.0000    0.0000    0.0000    0.0000    0.0022
    0.0000    0.0000    0.0000    0.0000    0.0000
    0.9479    0.0000    0.0000    0.0002    0.0002
    0.0521    1.0000    0.0000    0.0117    0.0008
    0.0000    0.0000    0.9978    0.0000    0.8467
    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.1480
    0.0000    0.0000    0.0000    0.0000    0.0002
    0.0000    0.0000    0.0000    0.9881    0.0018
    0.0000    0.0000    0.0022    0.0000    0.0001
```

INPUT	**PREDICTED CLASS (probabilities)**
	2 (0.9479)
	3 (1.0000)
	4 (0.9978)

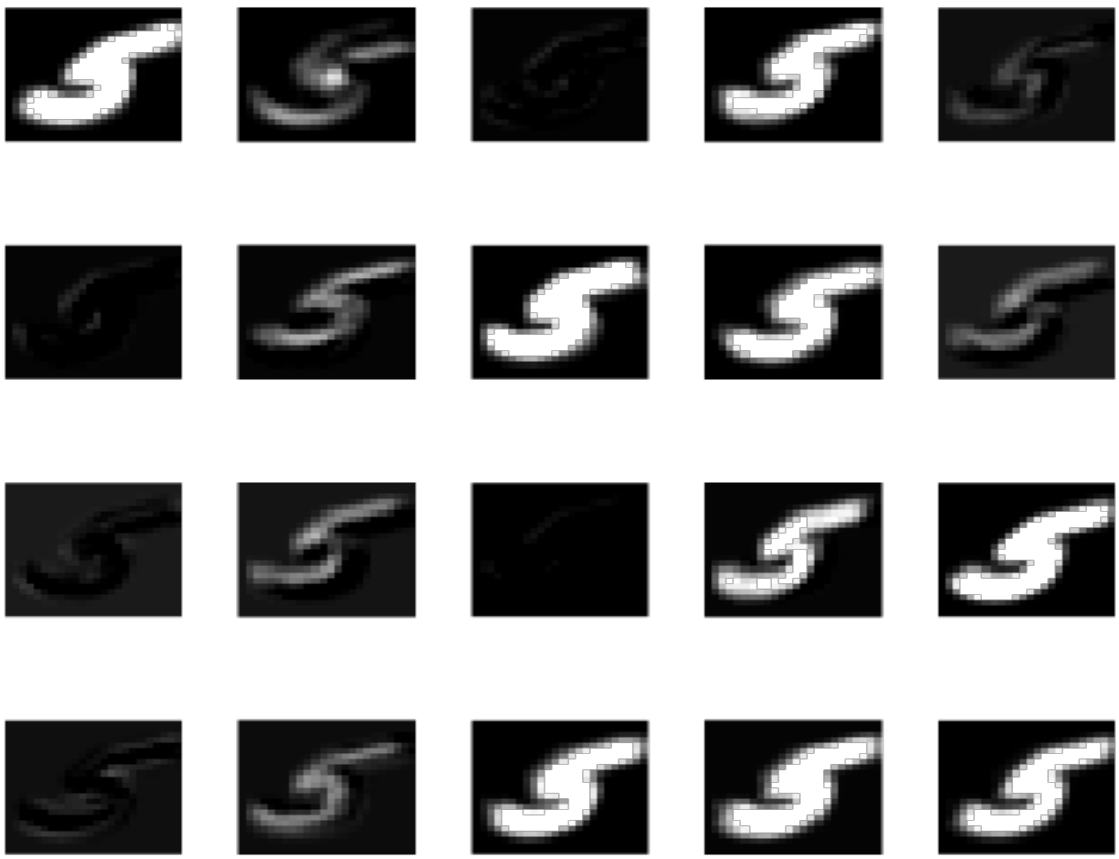


8 (0.9881)

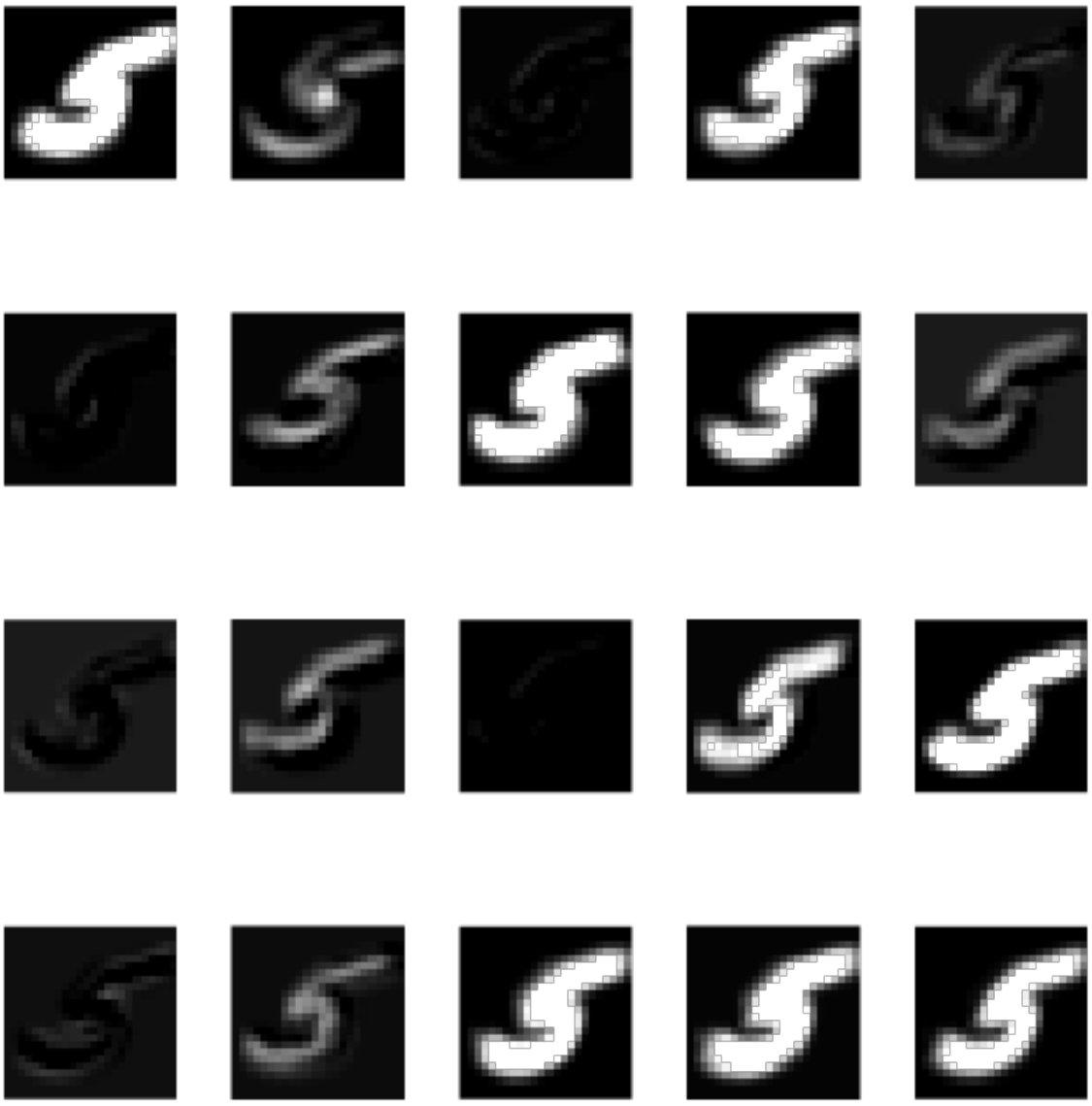
4 (0.8467)

PART 4 : Q 4.1 :

Feature visualization at the conv layer:



Feature visualization at the RELU layer :



Q 4.2:

The original image looks like this,

At the CONV layer, we have several weighted filters that slide over the image and the pixel and weights are multiplied (with a bias term) to give rise to a newly extracted feature. The number of features at layer in CNN is equal to the number of filters and the number of channels to be convoluted. As we can see, each channel in the processed image data tries to capture each feature of the input image. Feature may include, depth, edge, texture, size, shape, intensity, etc. So, each filter corresponds to one of the extracted features. There is no clear mapping of what each feature represents and which filter is mapped with which feature etc. As the number of features, increase, it becomes chaotic to make sense of it. The general idea is that each feature corresponds to some property of the input image and the model tries to “learn” it by associating them with “weights”. If similar pattern is found later, this excites the pixels correspondingly so that they influence the prediction.

At the RELU layer, only pixels beyond a certain threshold intensity are allowed to pass through. Here, the CONV and RELU layers look alike but usually CONV layers are brightened to show the features more clearly.

PART 5:

Image1

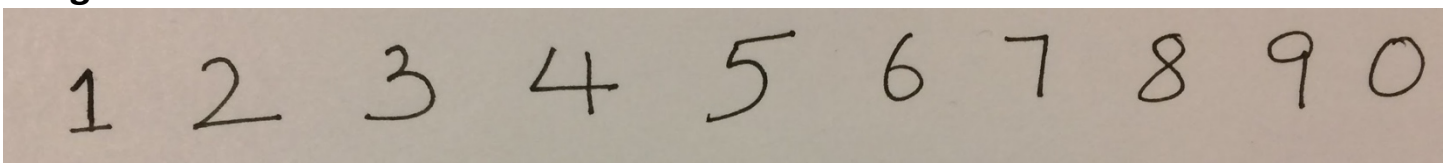


Image2

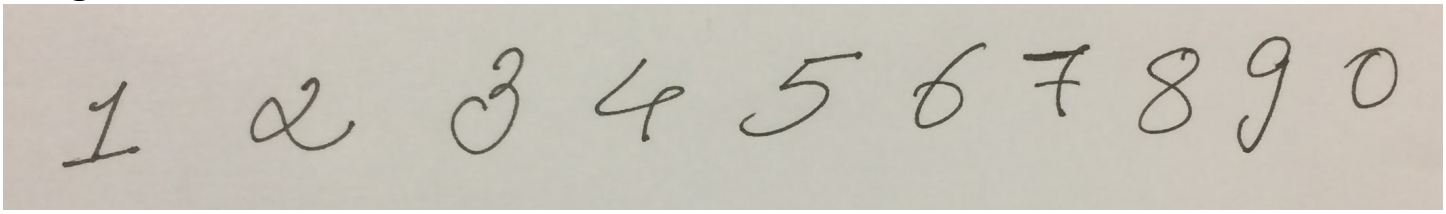


Image3

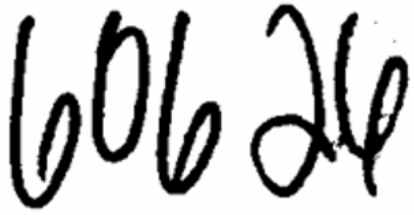
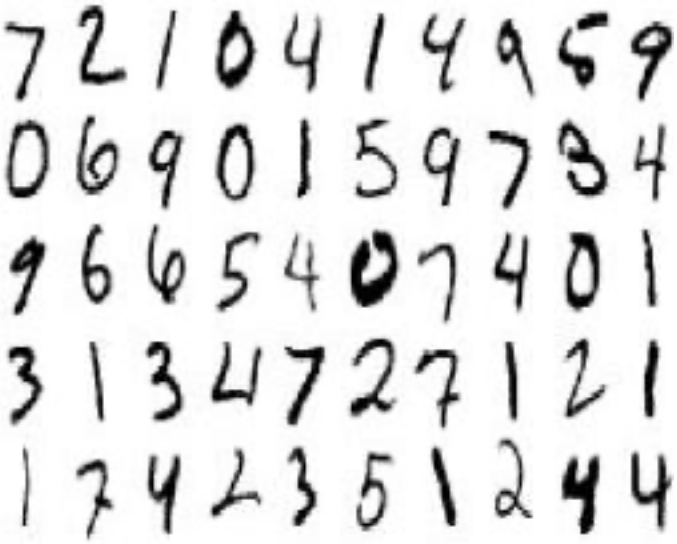


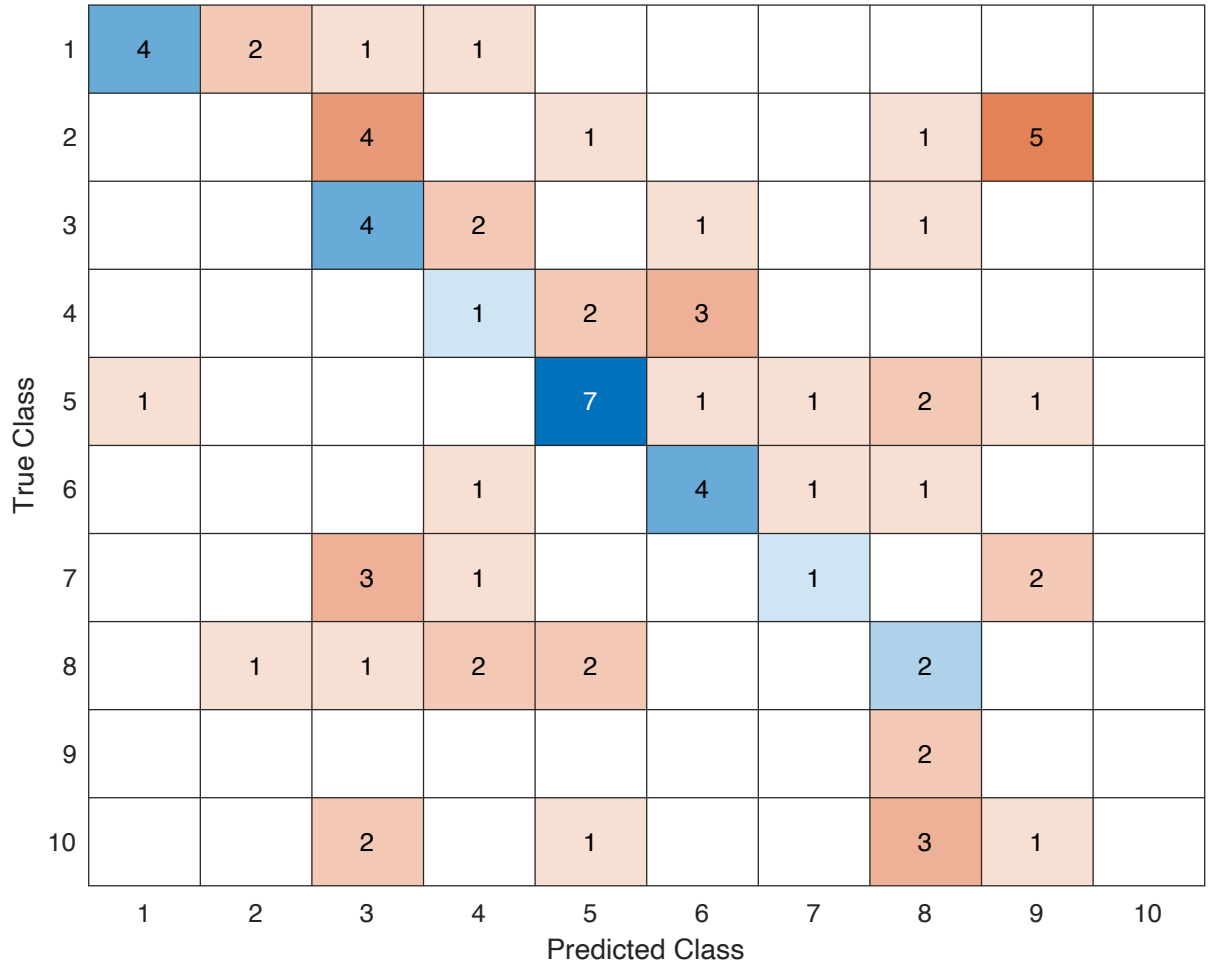
Image4



Statistics:

Image Name	Actual no. of digits	Digits found by bounding boxes script	Valid bounding boxes	Number of errors
Image1.jpg	10	10	10	-
Image2.jpg	10	10	10	-
Image3.png	5	6	5	1
Image4.jpg	50	51	50	1

Confusion chart for the predictions:



Expected Label	Predicted Label	Probability of prediction
1	2	0.6232
2	3	0.967
3	8	0.6952
4	5	0.6399
5	8	0.9572
6	6	0.7489
7	3	0.6026
8	3	0.6003
9	8	0.907
10	3	0.5693
1	2	0.7959
2	3	0.9485
3	4	0.9935
4	5	0.3845
5	6	0.7567
6	6	0.8855
7	4	0.9019
8	5	0.5926
9	8	0.9988
10	3	0.3206
7	3	0.5239
1	1	0.7403
7	7	0.7434
3	4	0.8738
5	7	0.9321
5	8	0.994
8	8	0.632
1	4	0.9935
10	8	0.9129
4	4	1
2	3	0.7562
7	9	0.8353
8	8	0.7426
3	6	0.9972
7	3	0.9721
2	8	0.5242
4	6	0.7028
10	8	0.9574
7	9	0.6137

5	5	0.9856
2	5	0.5003
5	1	0.81
3	3	0.9996
1	1	0.8378
1	1	0.9848
6	6	0.7051
5	5	1
5	5	0.9963
8	2	0.751
4	6	0.8968
2	9	0.9997
1	3	0.6914
3	3	0.9965
6	6	0.8673
6	8	0.8725
2	9	0.888
8	5	0.6732
8	4	0.4504
5	5	0.8452
10	5	0.9888
2	3	0.8221
8	4	0.4259
5	5	0.9993
3	3	0.9999
10	8	0.8152
2	9	0.9999
6	7	0.9999
4	6	0.814
5	9	0.6435
1	1	0.8054
3	3	0.9953
6	4	0.6056
10	9	0.9988
5	5	0.9949
5	5	0.9212
2	9	0.9996
2	9	1