Digit Recognition with Convolutional Neural Networks

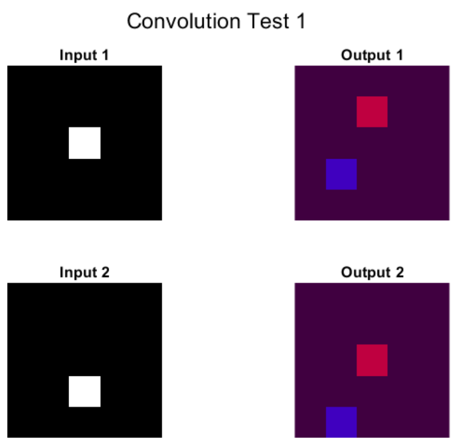
CMPT 412

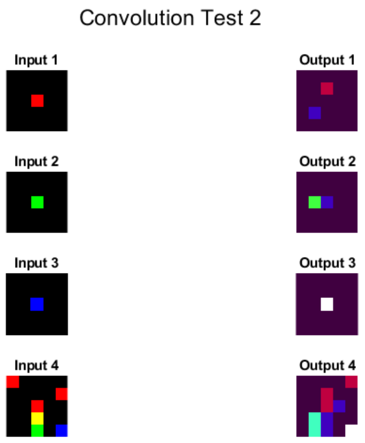
Aditya Rajvanshi – 301349754

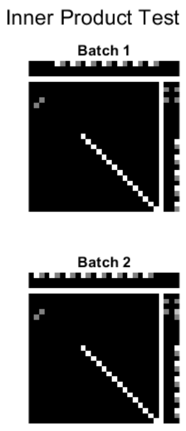
DISCLAIMER:

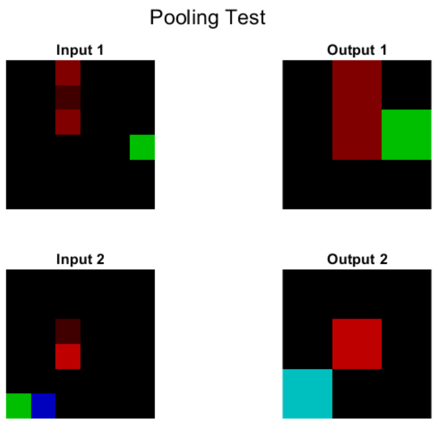
The “Computer Vision” addon for MATLAB is required to run the insertText command for part 5.

Part 1: Forward Pass



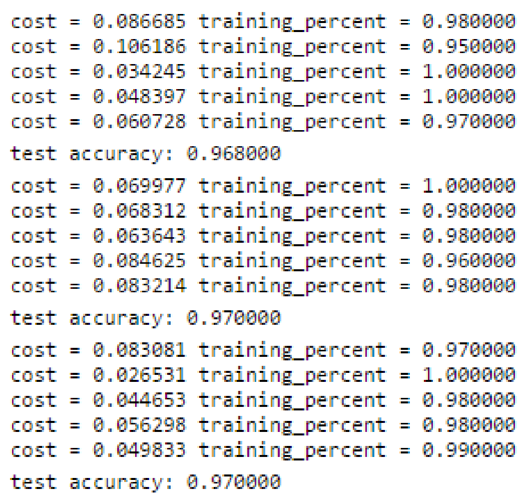






Part 3.1: Training

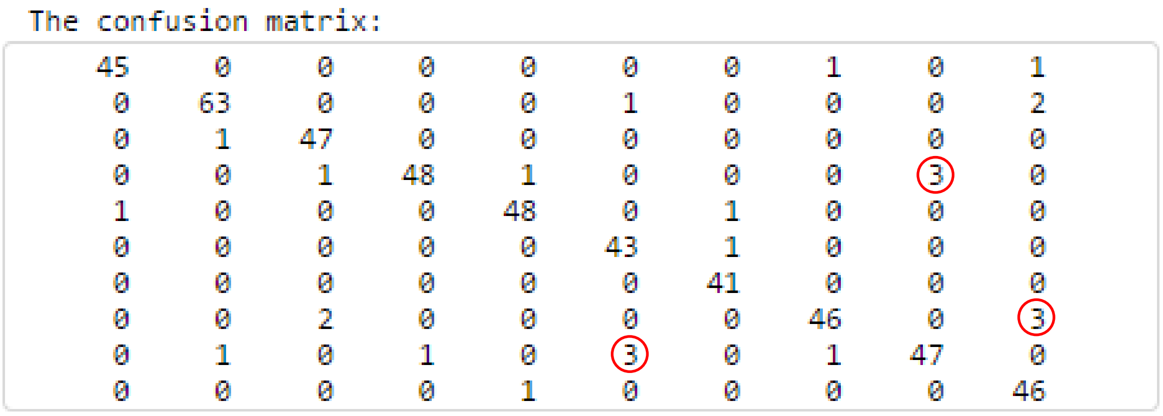
The training ran for 3000 iterations. The results for the final 1500 iterations are shown below.



```
cost = 0.086685 training_percent = 0.980000
cost = 0.106186 training_percent = 0.950000
cost = 0.034245 training_percent = 1.000000
cost = 0.048397 training_percent = 1.000000
cost = 0.060728 training_percent = 0.970000
test accuracy: 0.968000
cost = 0.069977 training_percent = 1.000000
cost = 0.068312 training_percent = 0.980000
cost = 0.063643 training_percent = 0.980000
cost = 0.084625 training_percent = 0.960000
cost = 0.083214 training_percent = 0.980000
test accuracy: 0.970000
cost = 0.083081 training_percent = 0.970000
cost = 0.026531 training_percent = 1.000000
cost = 0.044653 training_percent = 0.980000
cost = 0.056298 training_percent = 0.980000
cost = 0.049833 training_percent = 0.990000
test accuracy: 0.970000
```

As the results show, the accuracy in the trained data was 97%.

Part 3.2: Test the network



```
The confusion matrix:
    45     0     0     0     0     0     0     1     0     1
     0    63     0     0     0     1     0     0     0     2
     0     1    47     0     0     0     0     0     0     0
     0     0     1    48     1     0     0     0     3     0
     1     0     0     0    48     0     1     0     0     0
     0     0     0     0     0    43     1     0     0     0
     0     0     0     0     0     0    41     0     0     0
     0     0     2     0     0     0     0    46     0     3
     0     1     0     1     0     3     0     1    47     0
     0     0     0     0     1     0     0     0     0    46
```

There are three pairs of classes that are the most confused. The three pairs are:

Eight and five – This is understandable since 8 and 5 have similar curves and structures.

Three and eight – Similarly, 3 often looks like an 8 that has been split down the middle, which could lead to confusion for a digit recognition system.

Seven and nine – The confusion here may appear hard to understand but many 7s are written with a middle line, which could lead to confusion for being a 9. All the while, some 9s are written without curved stems, which lead them to being straight, similar to 7.

Part 3.3: Real-world testing

The script, test_real.m, was written to test the network on real-world examples of handwriting. Samples were gathered and tested on.

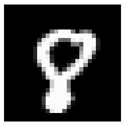
```
The true number is: 0.
The predicted number is: 0.
The true number is: 1.
The predicted number is: 1.
The true number is: 2.
The predicted number is: 2.
The true number is: 3.
The predicted number is: 3.
The true number is: 4.
The predicted number is: 4.
```

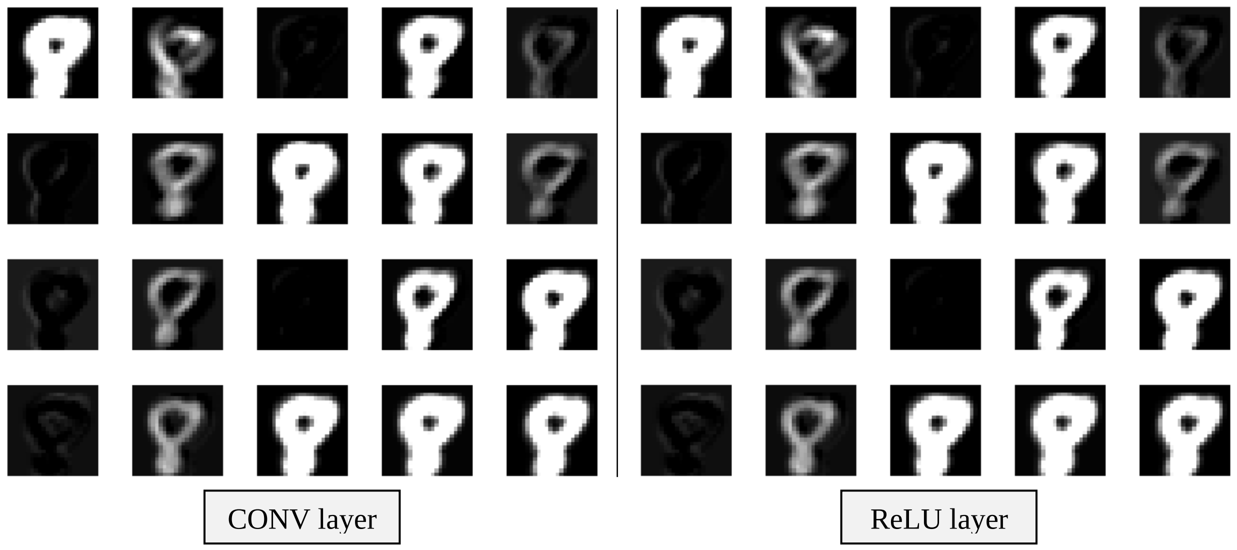
```
The true number is: 5.
The predicted number is: 5.
The true number is: 6.
The predicted number is: 6.
The true number is: 7.
The predicted number is: 7.
The true number is: 8.
The predicted number is: 8.
The true number is: 9.
The predicted number is: 4.
Correctly predicted: 9/10.
```

The output was edited only for formatting purposes. As the test demonstrates, the system accurately recognises 9/10 images, only failing to distinguish 9 from 4. Which is interesting as the confusion matrix does not indicate any confusion between 9 and 4. However, through interpretation, it is feasible that 9 and 4 look very alike based on how the 4 is drawn.

Part 4: Visualization

In this step, the original input image is shown below, along with the CONV layer on the left and the ReLU layer on the right.



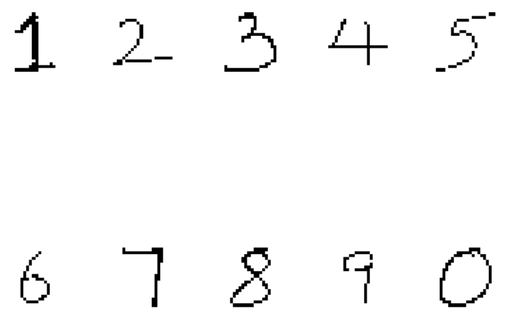


From observation, both layers appear to have the same effect on the original image. However, this is likely due to the fact that the convolutional layer cannot be visually represented due to negative activations. These negative activations are changed to 0 to be able to be represented, which is also what the ReLU layer does, thus they both have very similar representations.

When comparing the feature layers to the original image, it is evident that the feature layers dilate, over/under expose, and sharpen the image. This is all in attempt to identify key features of the image, such as the girth of strokes and the general curves. Negatives of the image also help identify things like how much emphasis is put on what parts of the image. For example, the first row and second column of the feature maps show this.

Part 5: Image Classification

Image 1



The script correctly identified each number as one connected component and split it accordingly. However, the identification of the number 7 was incorrect. The network labelled it as a 3.

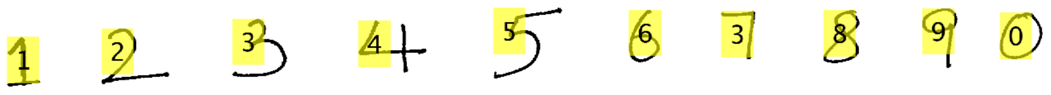
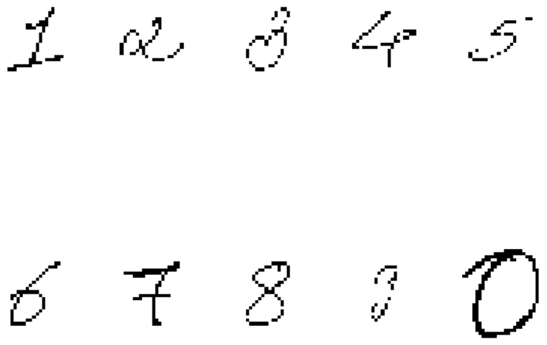


Image 2



For image 2, the network was able to correctly identify and split each number again, however, it struggled with identification even more, where 4 numbers were incorrectly labelled. I believe that this is due to how thinly split the components are in the left image. The 9 would be difficult to distinguish even for the human eye. This could be improved by dilating the pixels slightly.

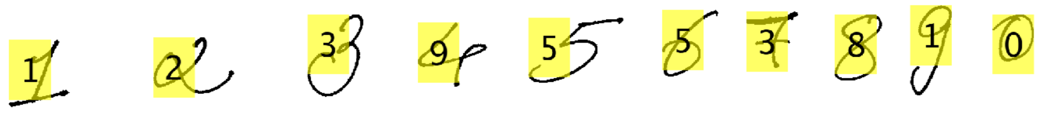
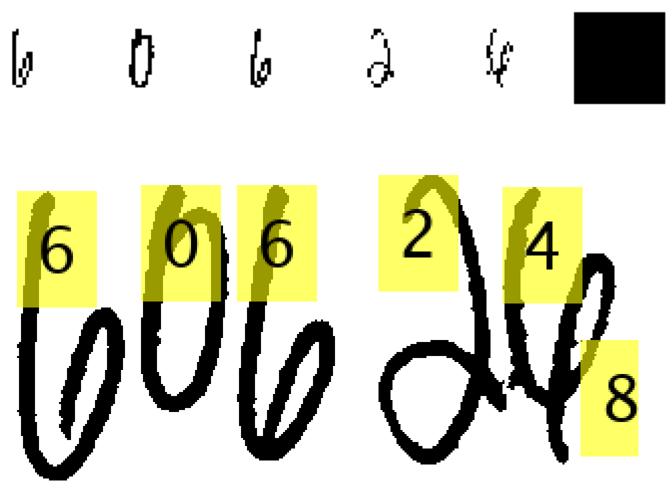
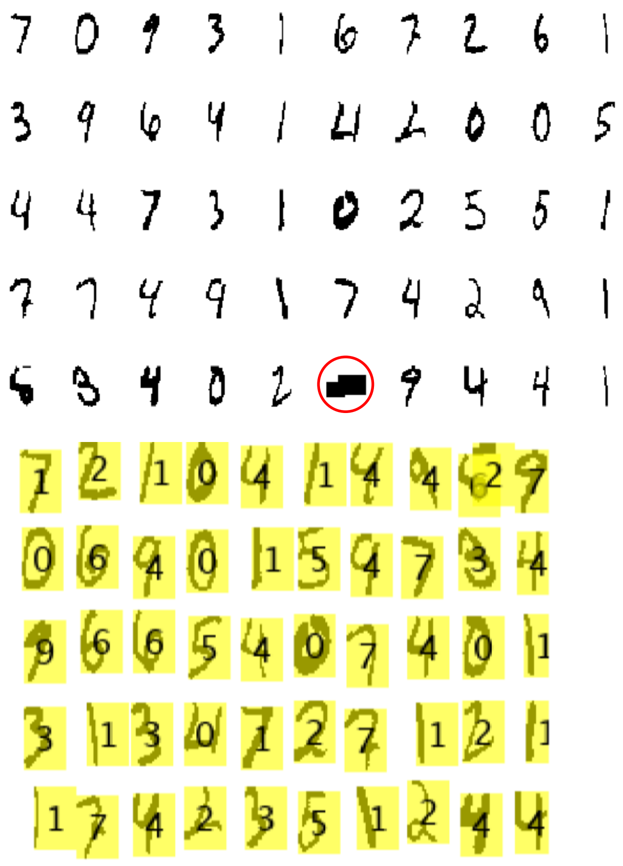


Image 3



In this example, the system correctly splits each of the numbers, however, it misidentifies some mysterious black square as it's own component. Despite this, it correctly identifies each split number, bar the incorrect split.

Image 4



The network appears to have largely correctly identified the numbers, but is splitting one number incorrectly, which is why there appears to be one random horizontal mark.

This misidentification is circled on the left picture.

In the 50 numbers, 7 were mislabelled as the bottom image shows.