CMPT 412

Project 1

Digit recognition with convolutional neural networks

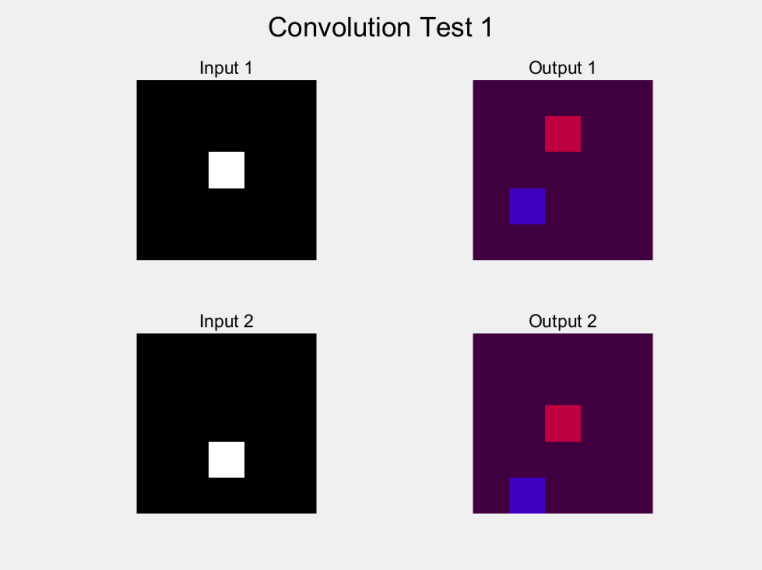
Instructor : Yasutaka Furukawa

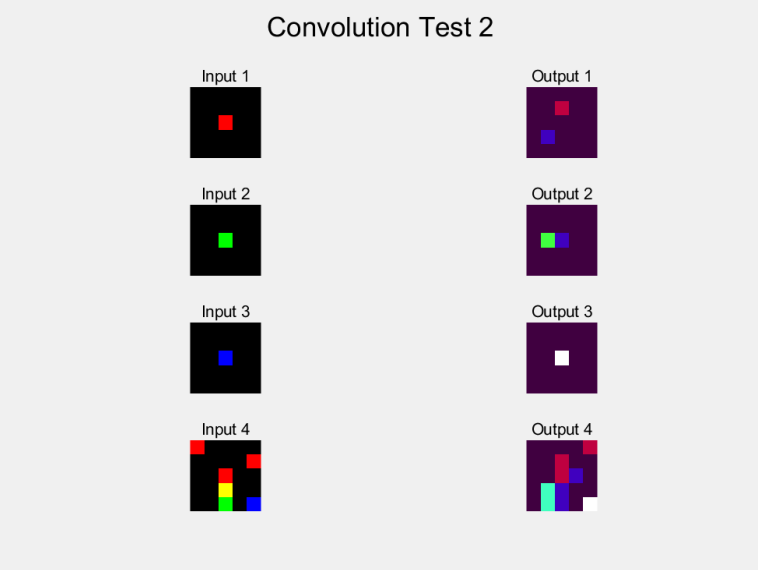
Name: Kaikun Fang

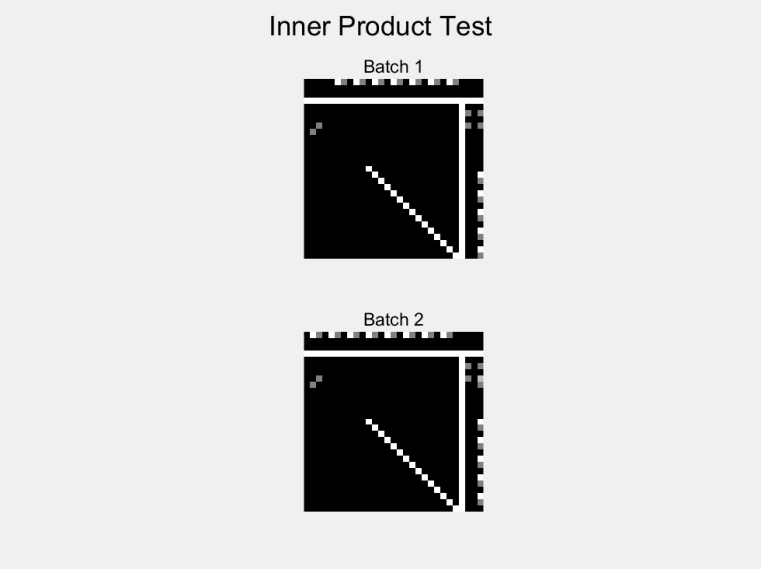
Student ID: 301416542

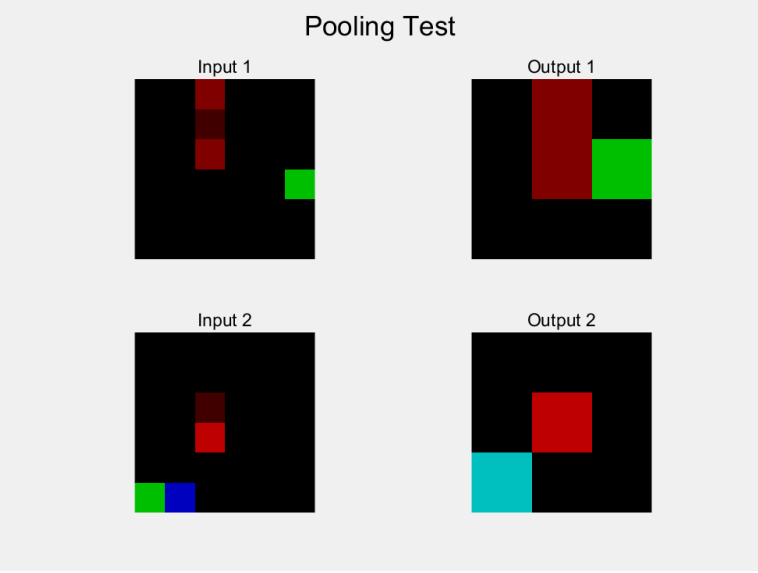
Note: The code inspiration and principles are derived from the videos and materials provided by the teacher. Discussions were also held with classmates(Cheng Hu 301435966).

Part 1: Forward Pass









Part 2: Back propagation

There is no visualization of the results in this section of the assignment requirements, the code has been completed and is in the corresponding file.

Part 3: Training

Q3.1

After about an hour of training, the following results were obtained. End at the test accuracy: 0.97 :

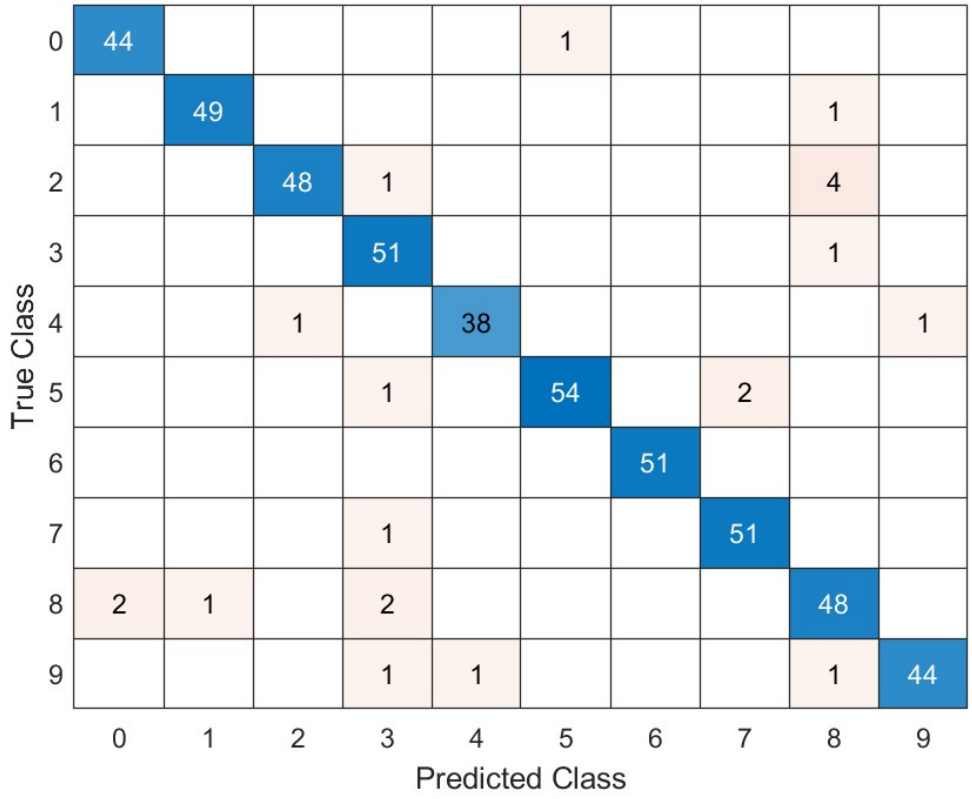
```
>> train_lenet
cost = 0.273491 training_percent = 0.910000
cost = 0.279565 training_percent = 0.910000
cost = 0.176619 training_percent = 0.920000
cost = 0.127344 training_percent = 0.950000
cost = 0.191895 training_percent = 0.960000
test accuracy: 0.944000

... ...

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
```

Q3.2



The table shows that the two numbers most likely to be misidentified are ‘2’ and ‘8’.

‘2’ is easily recognized as ‘8’, which should be because ‘2’ may be too rounded in handwriting and cause the characteristics to match ‘8’.

‘8’ is easily recognized as ‘0’ and ‘3’ because ‘8’ itself is very similar to ‘3’. is very similar to ‘3’. Sometimes a handwritten ‘8’ may have a very small circle on the top or bottom, and after processing it may only have one circle left, so it will look very much like a ‘0’.

Q3.3

I wrote a script ('test_Q3_3.m') to automatically read and test my pre-segmented ten numbers, and the results are automatically saved in the '/results/test-3.3' folder. If you run the code repeatedly please delete the results of the last run first.

Prediction: 9 1 2 3 4 5 6 7 8 9

Probably because I subjectively selected some simple and clear images and pre-processed them for the test, so the recognition rate from the results is relatively high at ninety percent.

Part4:

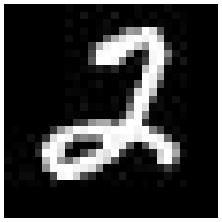


Figure 1 Origin

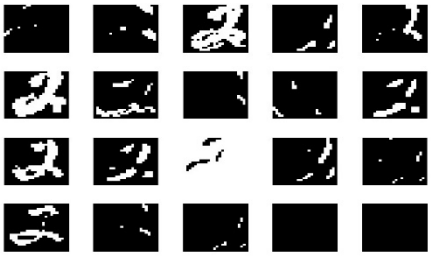


Figure 2 Different

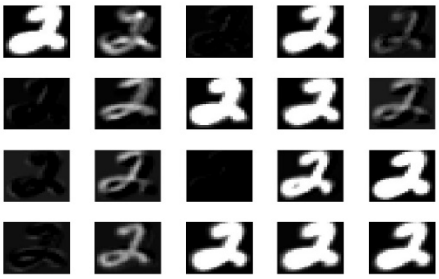


Figure 3CONV

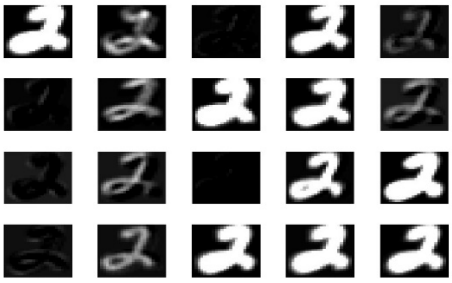


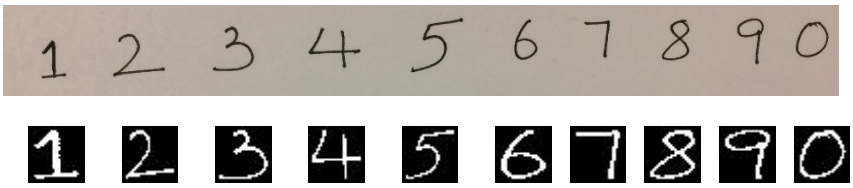
Figure 4 ReLU

I saved the generated images in the script into the '/result' folder. By observation, the results produced by both layers seem to be the same. Through research and search, it was found that this is because matlab's imshow turns negative numbers in CONV into 0 to display. And ReLU does exactly the same thing by turning the negative numbers in CONV into 0, so it causes the two plots to look the same. Since I created a new image(Figure 1) by extracting the negative numbers from CONV and converting the negative numbers in it to 1 to show it, you can see which data in CONV is modified by ReLU.

Part 5: Image Classification

The script I wrote (ec.m) automatically reads and identifies the four images provided by the teacher and outputs the results to the corresponding "/results/image#" folder. When running the code repeatedly, please delete the results of the last run first. The results of my run are as follows:

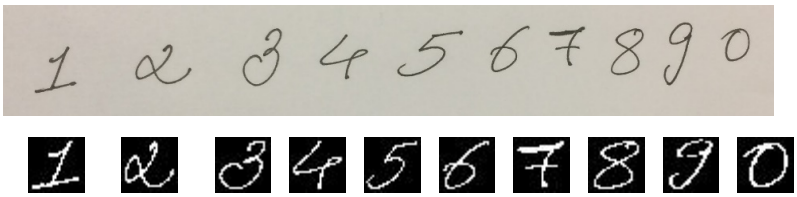
Image1:



Prediction: 1 2 3 4 7 5 3 8 7 0

Correct Rate: 6/10

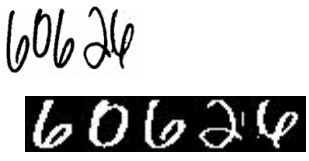
Image2:



Prediction: 1 2 3 9 5 5 3 3 7 0

Correct Rate: 5/10

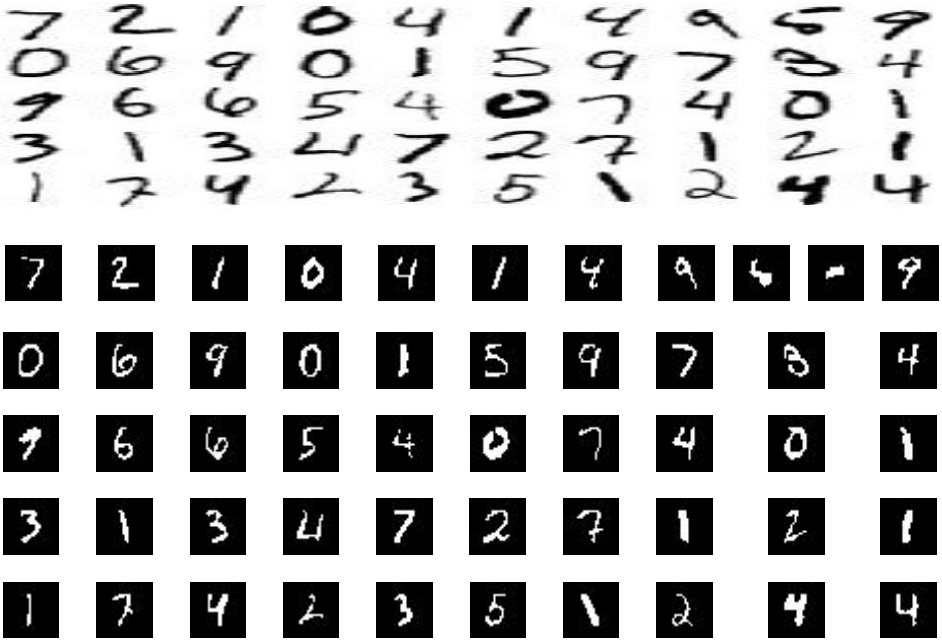
Image3:



Prediction: 6 0 6 3 6

Correct Rate: 3/5

Image4:



Prediction:	7	2	1	9	4	1	4	9	X	7
	0	6	4	9	1	5	9	7	5	4
	7	6	6	5	4	9	7	4	0	1
	7	1	3	4	7	2	7	1	2	1
	1	7	4	2	1	5	1	2	4	4

Correct Rate: 40/50

Summary:

The overall results showed that 54 out of 75 numbers were correctly identified, with an accuracy rate of about 72%. Interestingly, in image4, the number "5" in the top right of the original image was not connected together when it was written, so it was cut into two parts and recognized separately, which led to the wrong result. Secondly, the recognition rate of 0, 3 and 9 is not very accurate.