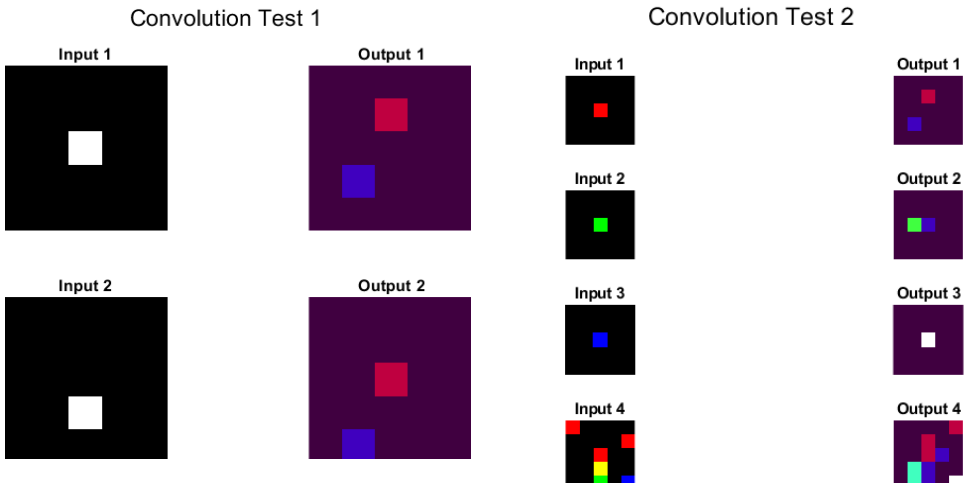
Sara Pilehroudi
StudentID: 301321656

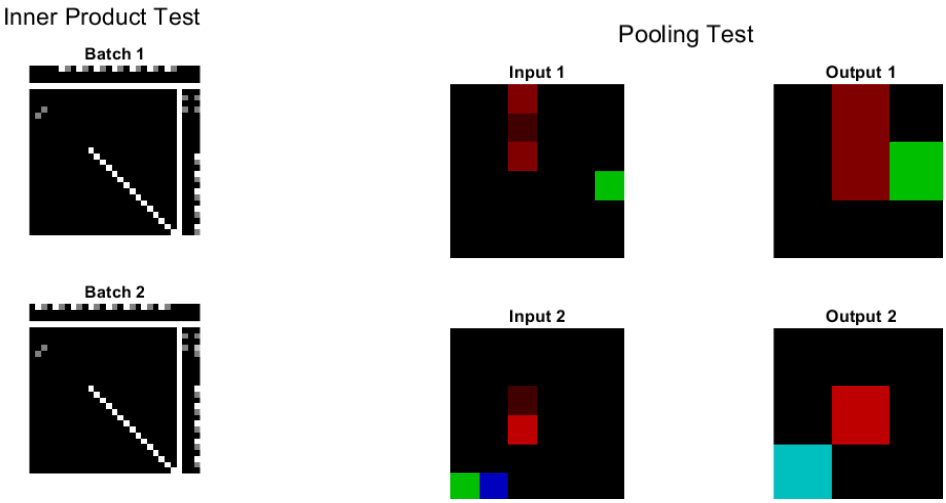
Computer Vision

Project 5 Report

---

1) Figures below show the visualization results of the convolution, max pooling and fully connected forward functions.





---

2) I trained the model for 3000 iterations and got 97% accuracy. I loaded my trained model and trained it for one iteration to show the accuracy in MATLAB console.

```
>> train_lenet
test accuracy: 0.970000
```

Sara Pilehroudi
StudentID: 301321656

3) Table below shows the confusion matrix of the test data.

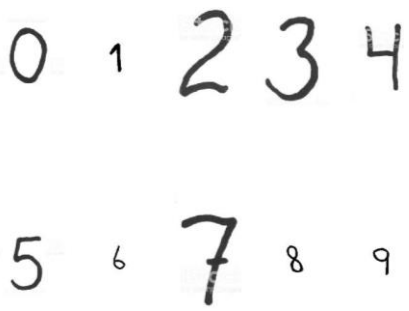
Actual/Predicted	0	1	2	3	4	5	6	7	8	9
0	53	0	0	0	0	0	0	0	1	0
1	0	55	2	0	0	0	0	0	2	0
2	1	0	44	0	0	0	0	0	0	0
3	0	0	0	49	1	2	0	0	0	0
4	0	0	0	1	52	0	1	0	0	1
5	0	1	0	2	0	40	1	0	0	0
6	0	0	0	0	0	0	55	0	0	0
7	0	0	1	0	0	0	0	33	0	0
8	0	0	1	0	0	0	0	0	52	0
9	0	0	0	0	2	0	0	0	1	46

As we can see from the table the network top confused classes are:

- 1 with 2: The network confused 1 with 2 because in some handwritings people draw a long horizontal line under one. This can create features that make the network confuse It with 2.
- 5 with 3: Both 5 and 3 has the same curve at the lower part of the digit. These curves will create similar features which makes the network confused.
- 9 with 4: 9 and 4 have similar shapes. Some handwritings also tend to have a more rounded edge four. This can confuse the network.

---

4) These are the pictures that I use to test the network for part 3.3 of the assignment.

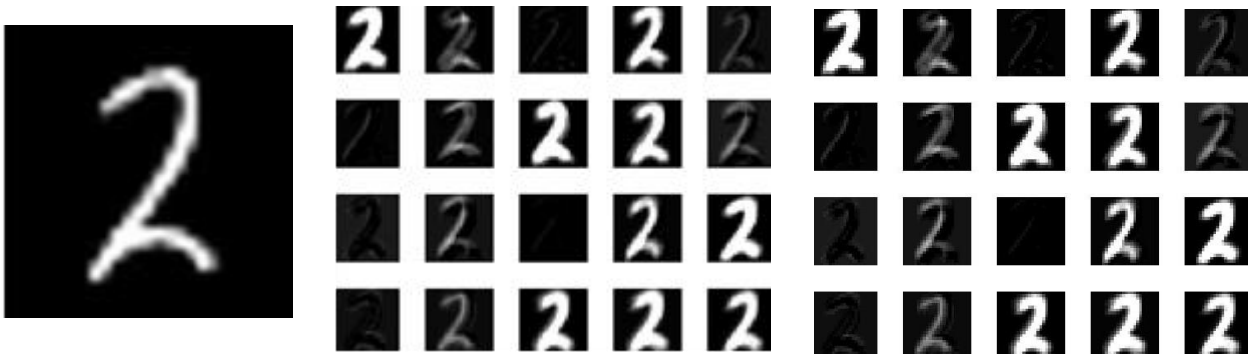


The model is able to predict all the images correctly.

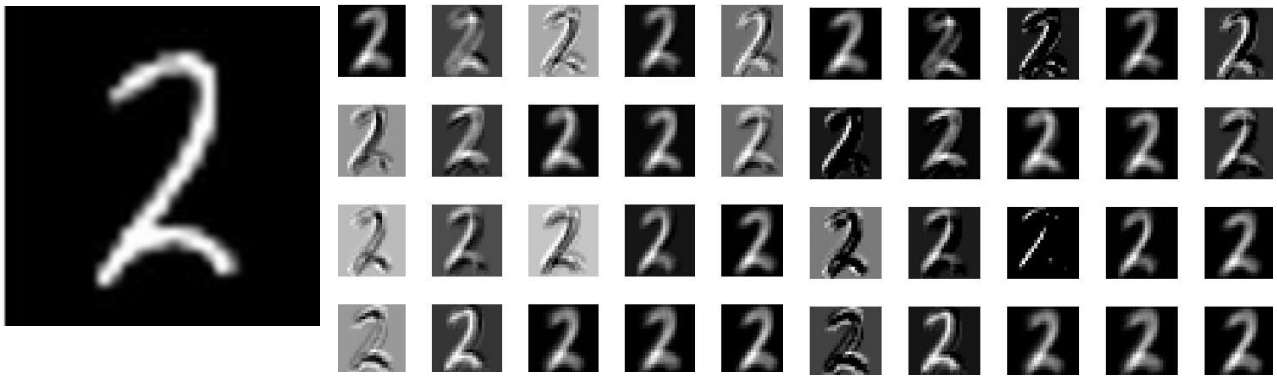
```
>> test_Real
test accuracy: 100
```

Sara Pilehroudi
StudentID: 301321656

5) The figures below show the output of second (CONV) and third (RELU) layer of the network. (Left to right). And the original image.



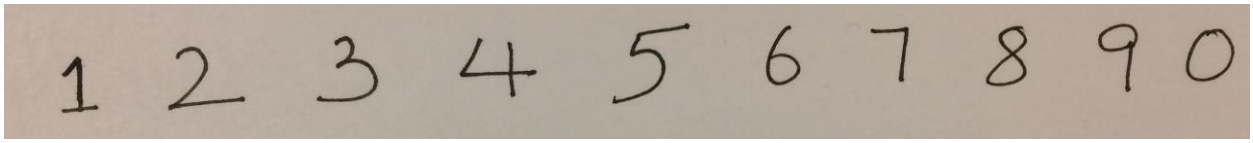
The figures below show the normalized output of second and third layer of the network in the same order.



As we can see in the figures the feature maps enhance the features of the image. For example, they made the number thicket and also extracted both horizontal and diagonal edges. They have also extracted the diagonal lines.

---

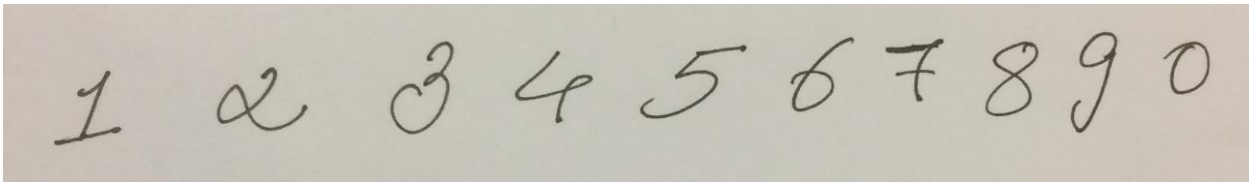
6) The figures below show the numbers extracted from each image provided in images folder. Original Image:



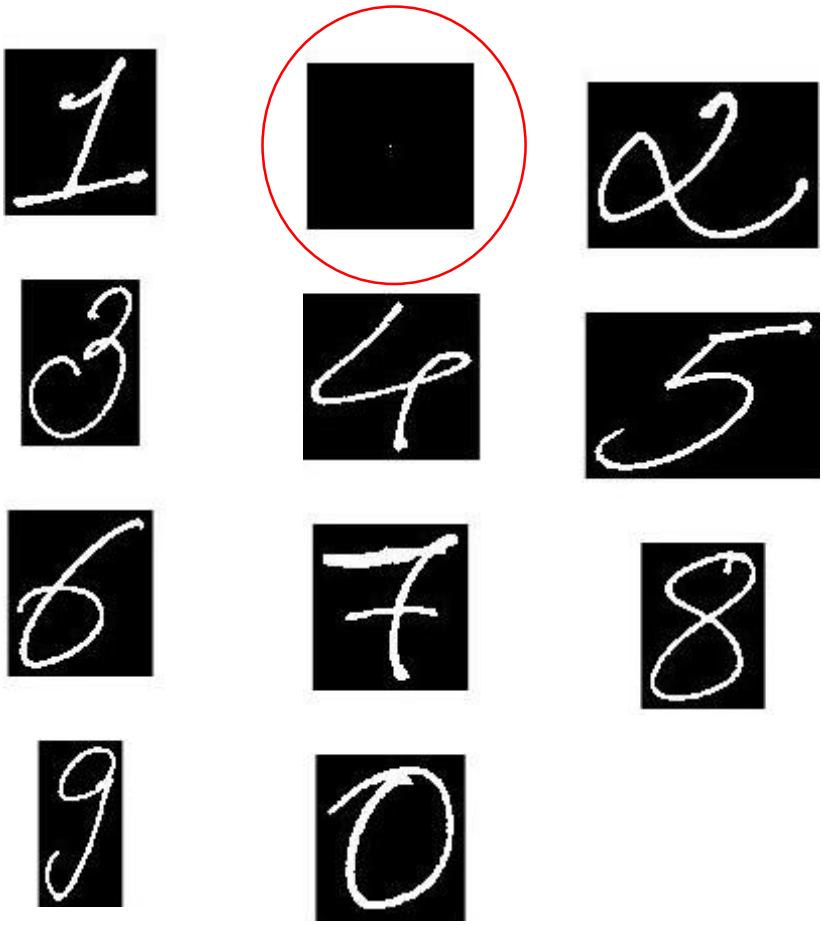
Extracted Numbers:

Sara Pilehroudi
StudentID: 301321656

Original Image:

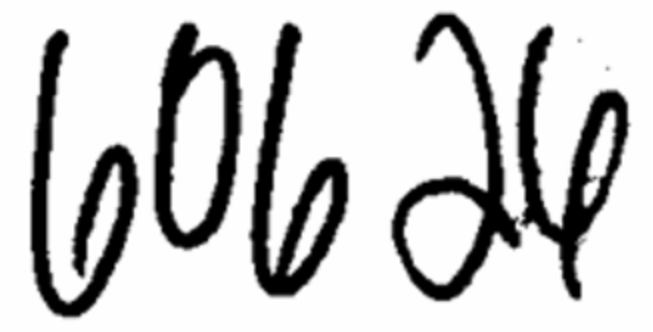


Extracted Numbers:

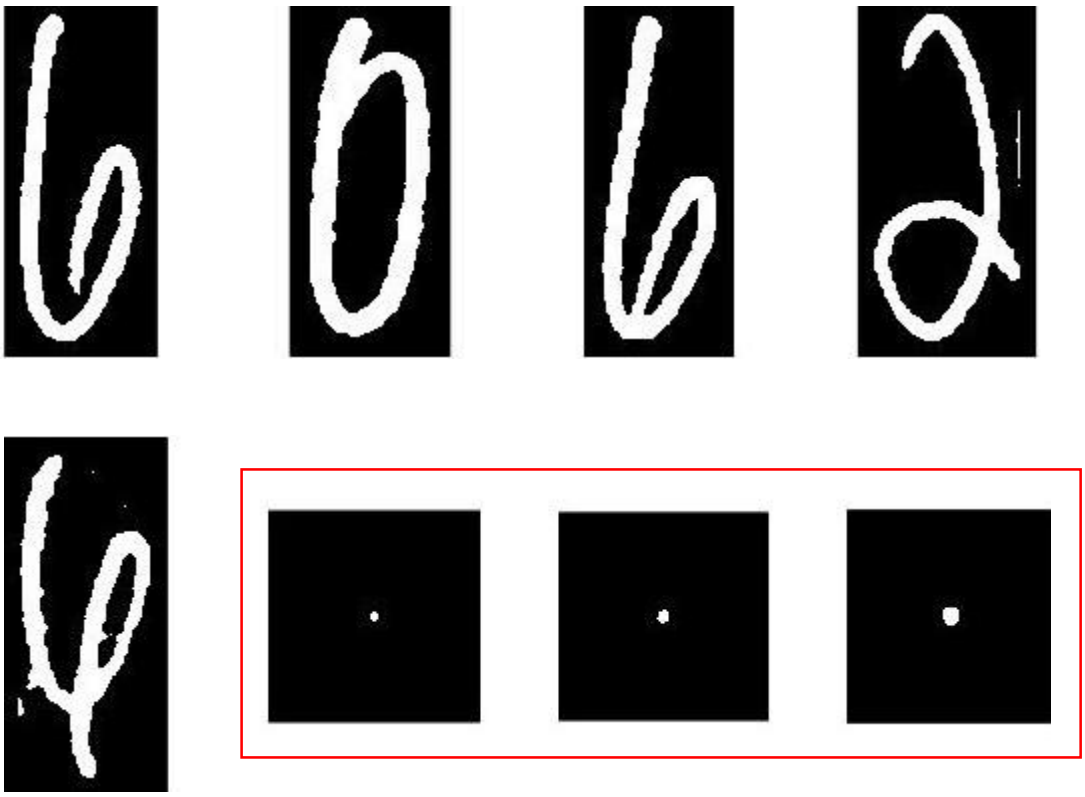


Sara Pilehroudi
StudentID: 301321656

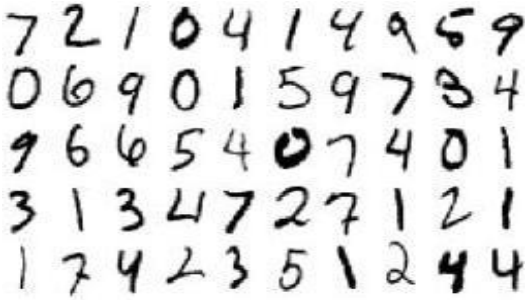
Original Image:



Extracted Numbers:

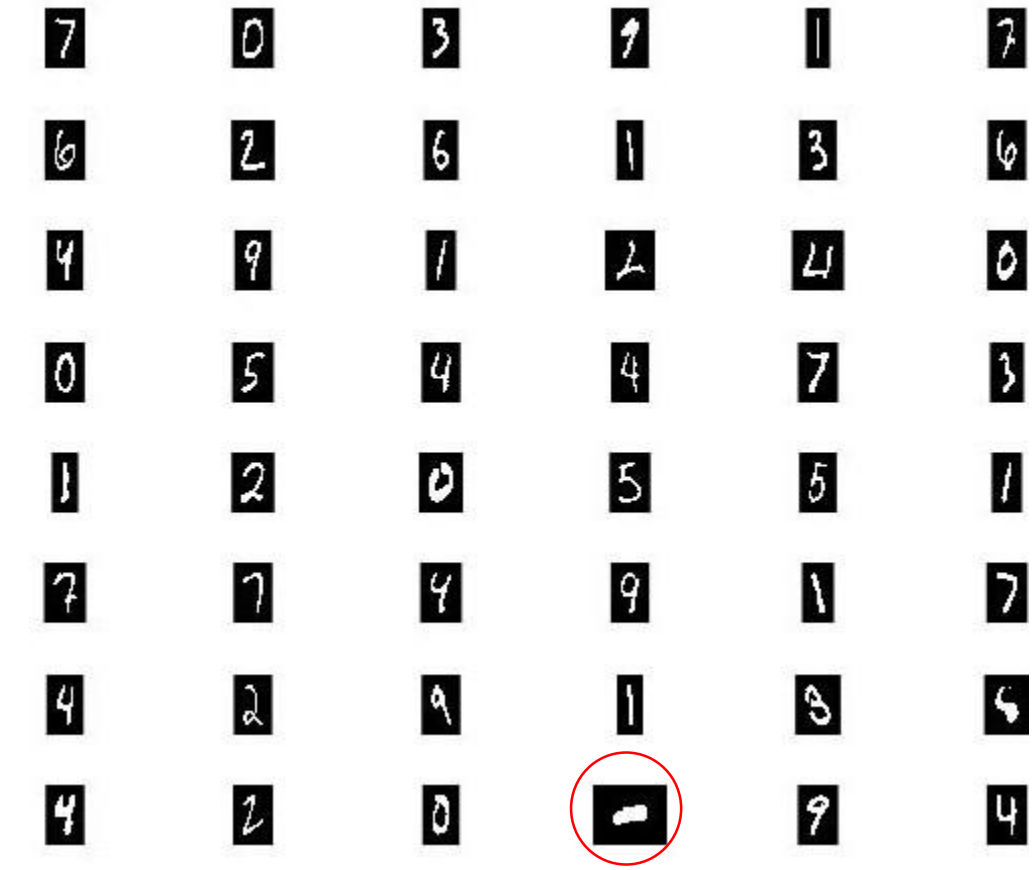


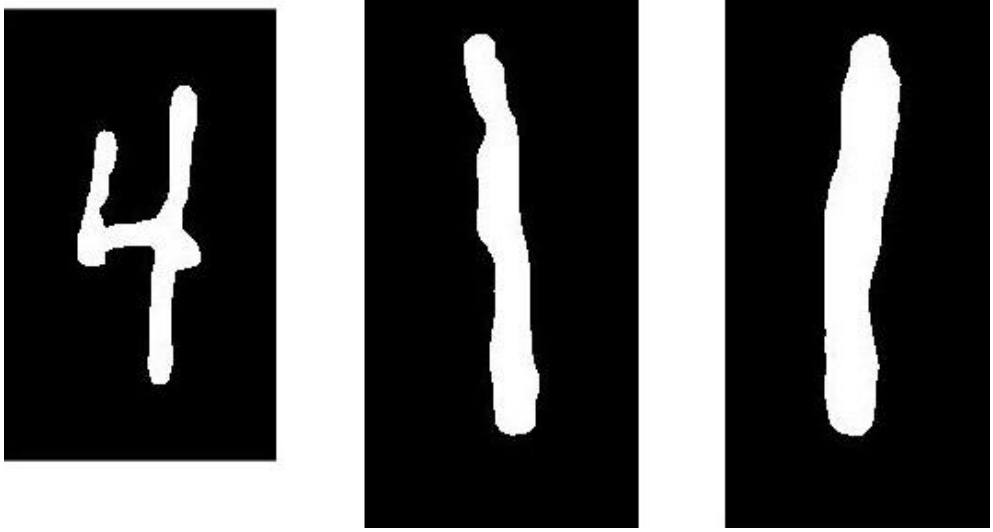
Original Image:



Sara Pilehroudi
StudentID: 301321656

Extracted Numbers:





Sara Pilehroudi
StudentID: 301321656

I used Matlab's built in adaptive threshold function to make the extraction process more robust. In order to have more precise bounding boxes, I first upscale each image 8 times bigger. This helps making the bounding boxes larger and create more accurate number patches specially in the last image where the numbers are rather small and close together. I also blur the images to reduce any probable noise. Then I invert the patches in order to obtain white numbers on black background. Finally, after resizing the images to 28x28 I also pad them to avoid numbers overlap with the patch borders.

As it can be observed from the extracted numbers, Matlab's *"bwconncomp"* function extract bounding boxes which necessarily do not include any meaningful data (they have been highlighted with red circle and rectangle around them in the pictures). I skip these not applicable data during the recognition process.

The final accuracy of the system is:

```
test accuracy: 77.333333 >>
```

The confusion matrix is the following:

Actual/Predicted	0	1	2	3	4	5	6	7	8	9
0	7	0	1	0	0	0	0	0	0	0
1	0	10	0	0	0	0	0	1	1	0
2	0	0	7	0	0	0	0	0	0	0
3	0	0	0	5	0	1	0	0	0	0
4	0	0	0	0	10	0	0	0	0	1
5	0	0	0	0	0	4	1	1	0	0
6	0	0	0	1	0	2	6	0	0	0
7	0	1	1	1	0	0	0	5	0	0
8	0	0	0	1	0	0	0	0	1	0
9	0	0	0	0	0	0	0	2	1	3

As we can see the top most confused groups are 6 with 5 and 9 with 7.