Assignment 5

Vishal Batvia

CMPT 762 Computer Vision

Author Note

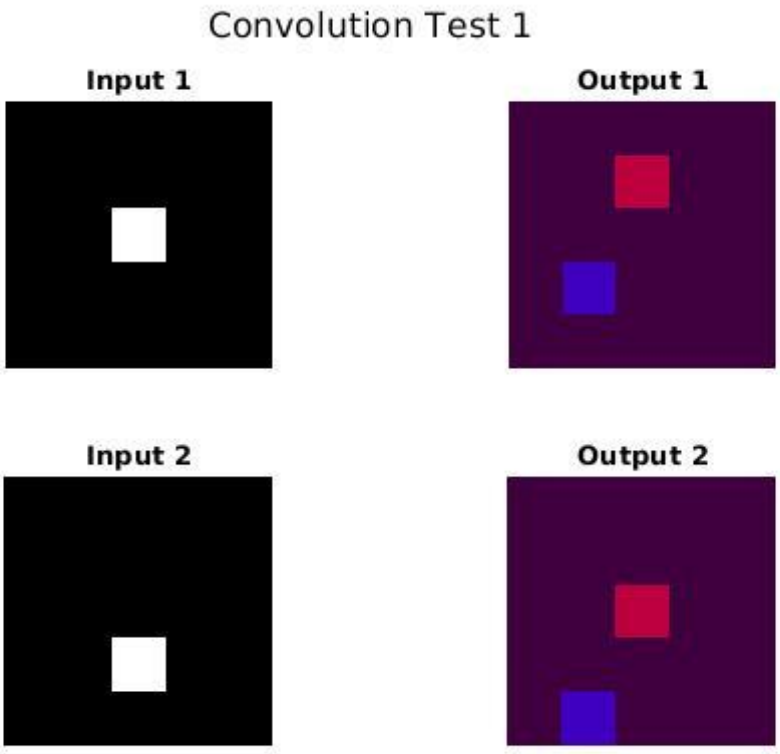
Assignment 5

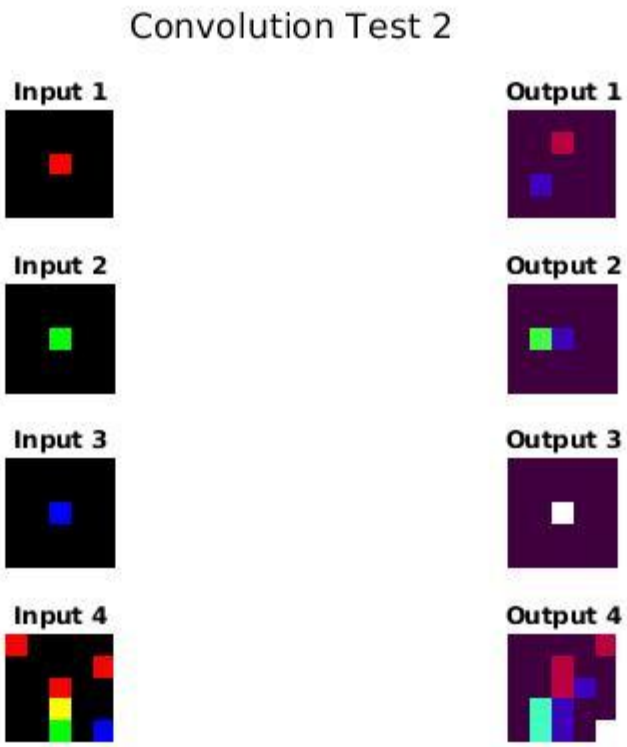
1.1. Inner Product Layer

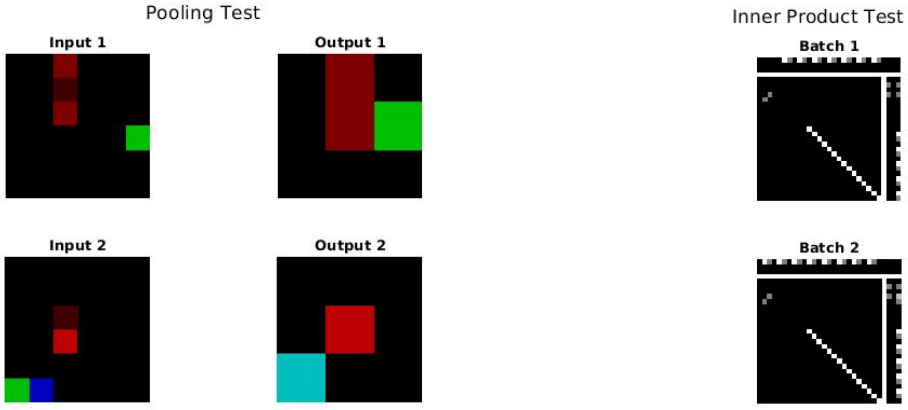
1.2. Pooling layer

1.3. Convolution Layer

1.4. Relu Layer







2.1. Relu Backpropagation

2.2. Inner Product Backpropagation

3.1. Training

Accuracy :- 97%

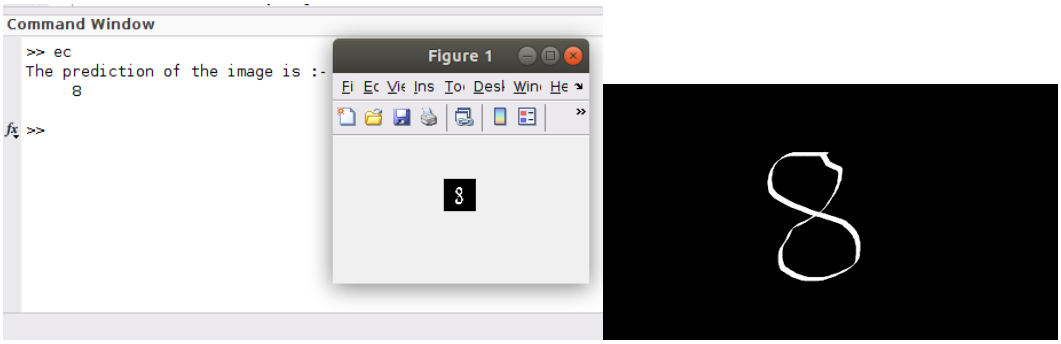
```
Command Window
>> train_lenet
cost = 0.273491 training_percent = 0.910000
cost = 0.279565 training_percent = 0.910000
cost = 0.176619 training_percent = 0.920000
cost = 0.127344 training_percent = 0.950000
cost = 0.191895 training_percent = 0.960000
test accuracy: 0.944000

cost = 0.192910 training_percent = 0.930000
cost = 0.131836 training_percent = 0.970000
cost = 0.115812 training_percent = 0.970000
cost = 0.103636 training_percent = 0.970000
cost = 0.124224 training_percent = 0.980000
test accuracy: 0.960000

cost = 0.111115 training_percent = 0.960000
cost = 0.113216 training_percent = 0.940000
cost = 0.134874 training_percent = 0.960000
cost = 0.067548 training_percent = 0.990000
cost = 0.095426 training_percent = 0.980000
test accuracy: 0.966000

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

fx >>
```

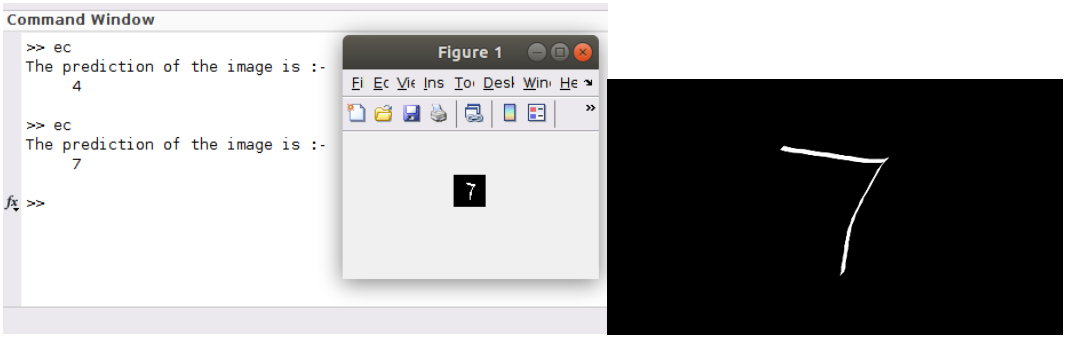
3.2. Test Network

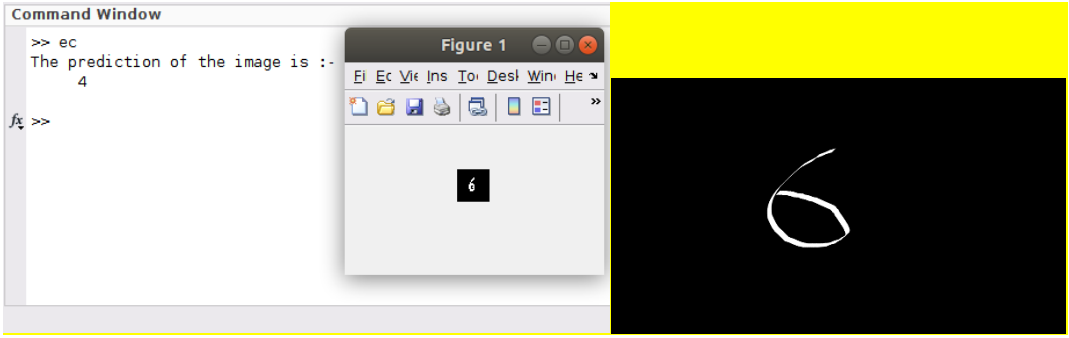
```
Command Window
>> test_network
Confusion Matrix
    39     0     0     0     0     1     1     0     0     0
     0    69     0     0     0     0     1     1     0     0
     0     0    47     0     0     0     0     1     0     0
     0     0     0    50     0     0     0     0     0     0
     0     0     0     0    51     0     1     0     0     2
     0     0     0     2     0    47     1     0     0     0
     0     0     0     0     0     0    52     0     0     0
     0     2     1     2     0     0     0    36     0     0
     1     0     1     0     0     0     1     0    36     1
     0     0     0     0     1     0     0     1     0    51
```

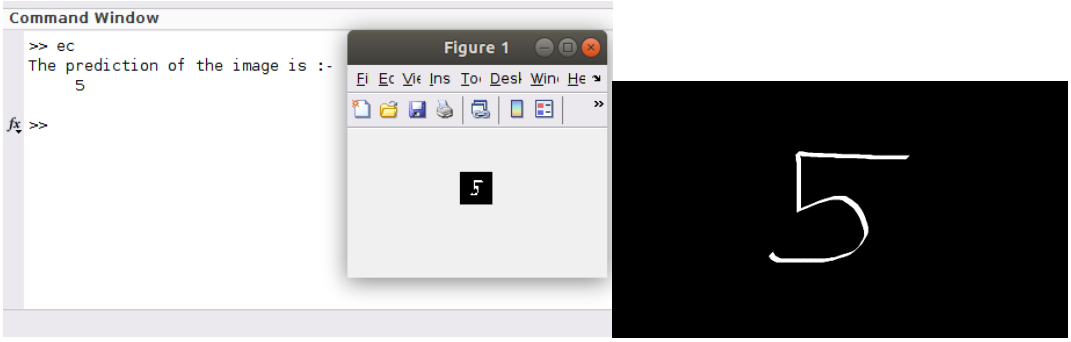
As can be seen from the image, most of the test images are correct. Majorly network is confused for 1, and 9 for being 7 and 4 resp twice while for 3 network gives the output as 5 and 7 twice. Other than that, 0 for 8, 2 for 7 and 8, 4 for 9, 5 for 0, 6 for 0,1,4,5 and 8, 7 for 1,2 and 9 and 9 for 8 has been noticed once.

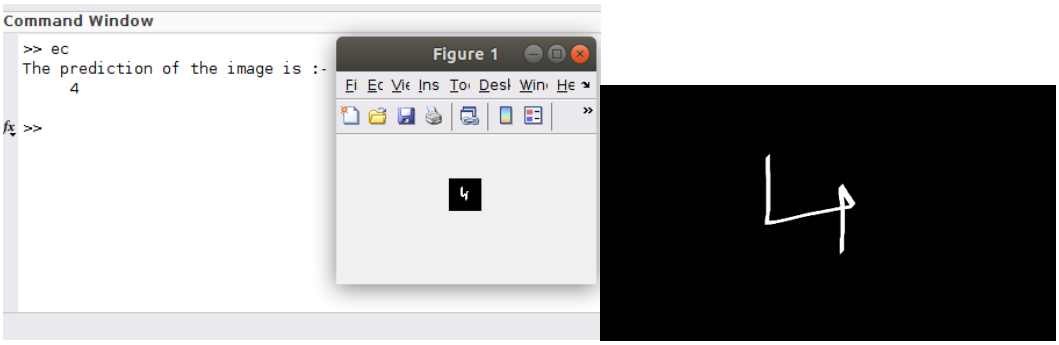
3.3. Real-World Testing (Using ec.m before adding the loops and bounding box for part 5.**)**

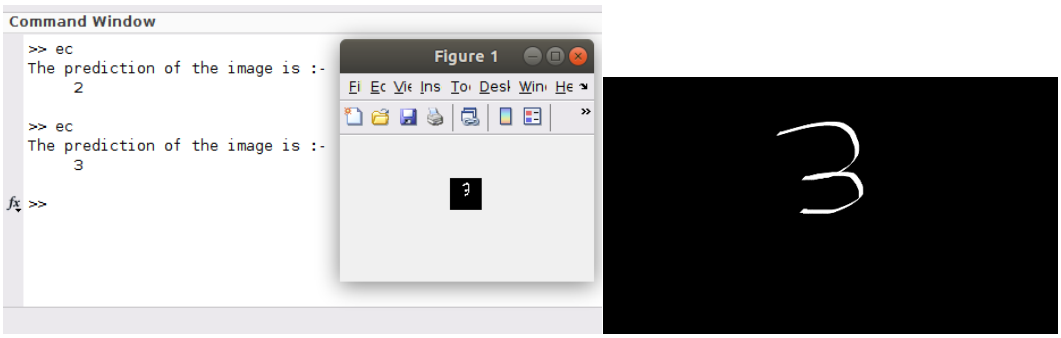


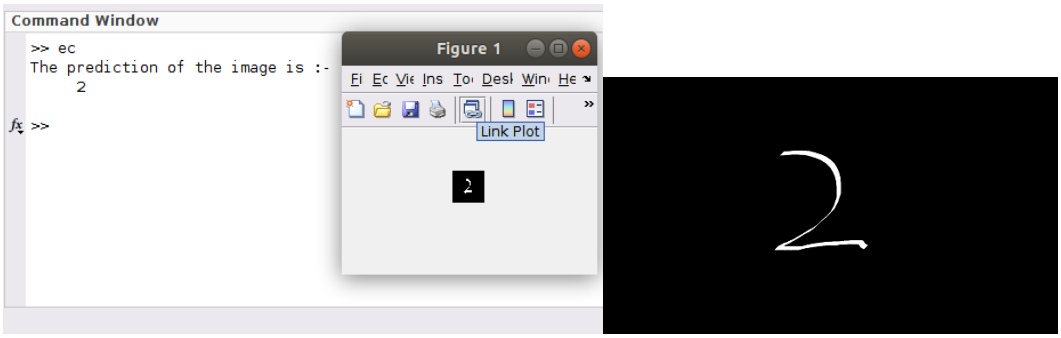


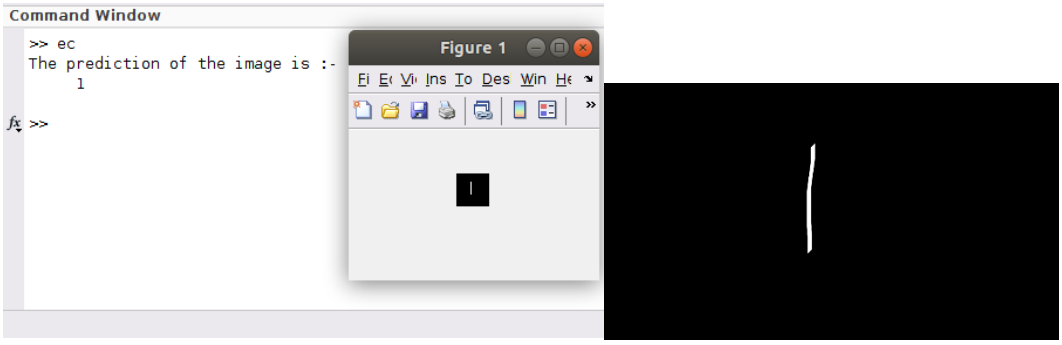


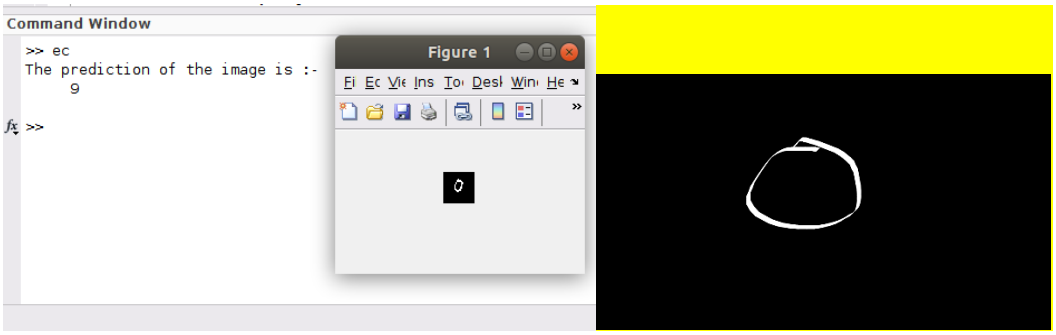


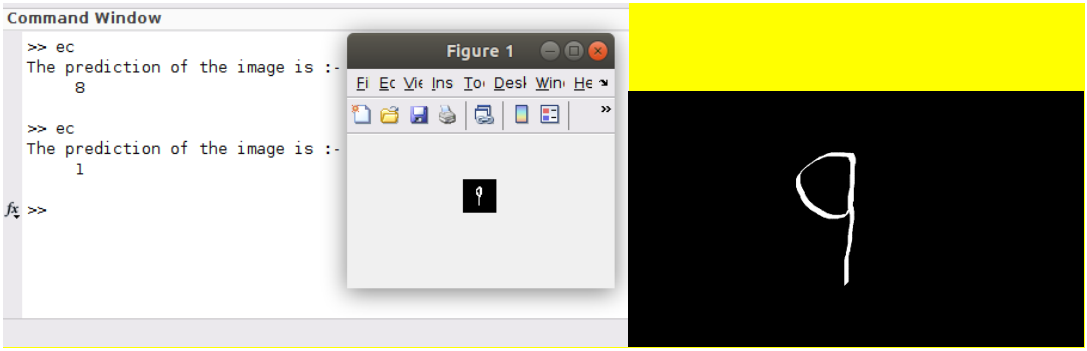

Command Window
>> ec
The prediction of the image is :-
4
>>
Figure 1

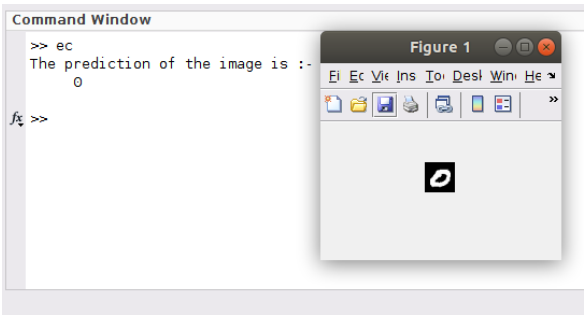

Command Window
>> ec
The prediction of the image is :-
2
>> ec
The prediction of the image is :-
3
>>
Figure 1

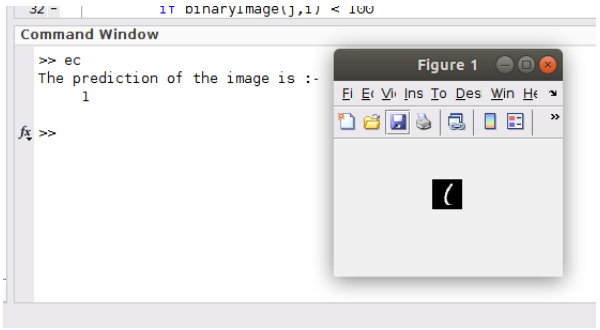

Command Window
>> ec
The prediction of the image is :-
2
>>
Figure 1
Link Plot

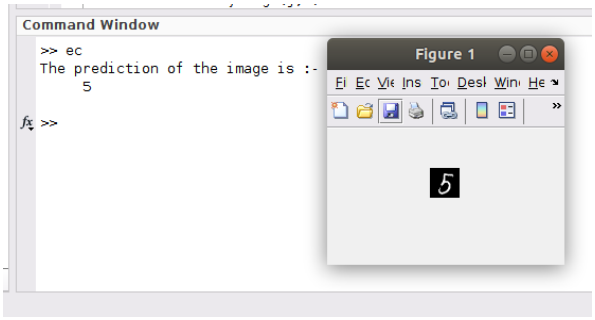

Command Window
>> ec
The prediction of the image is :-
1
>>
Figure 1

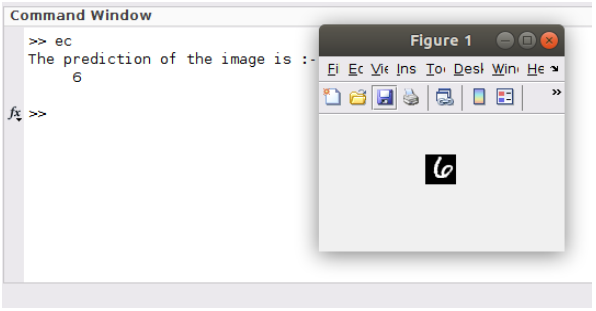

Command Window
>> ec
The prediction of the image is :-
9
>>
Figure 1

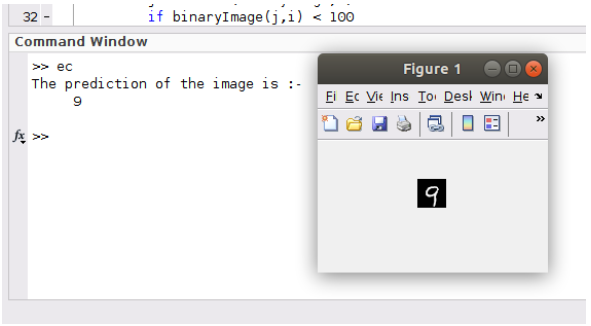

Command Window
>> ec
The prediction of the image is :-
8
>> ec
The prediction of the image is :-
1
>>
Figure 1


Command Window
>> ec
The prediction of the image is :-
0
>>
Figure 1


Command Window
>> ec
The prediction of the image is :-
1
>>
Figure 1

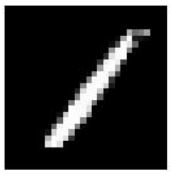


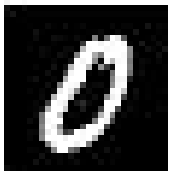




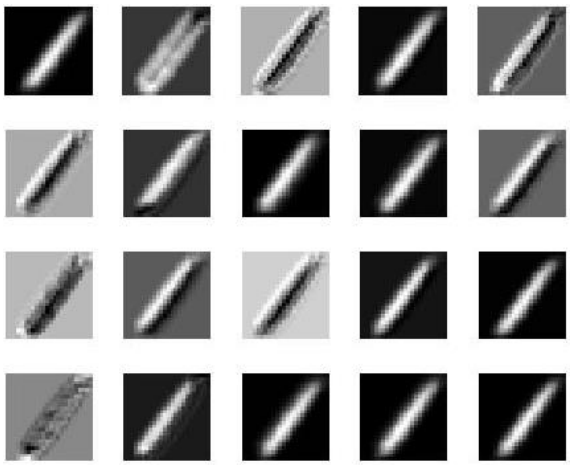
Errors in detection are highlighted.

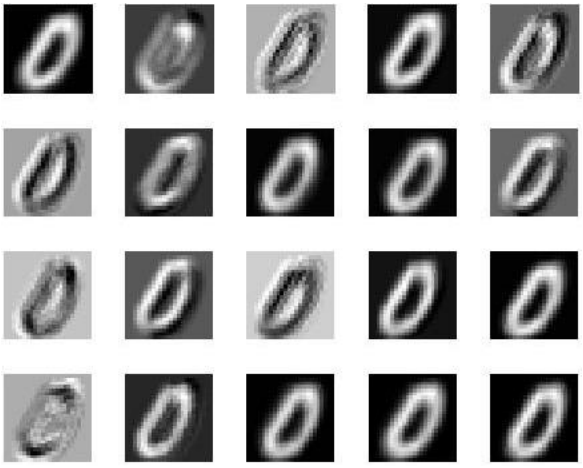
4.1. Visualization



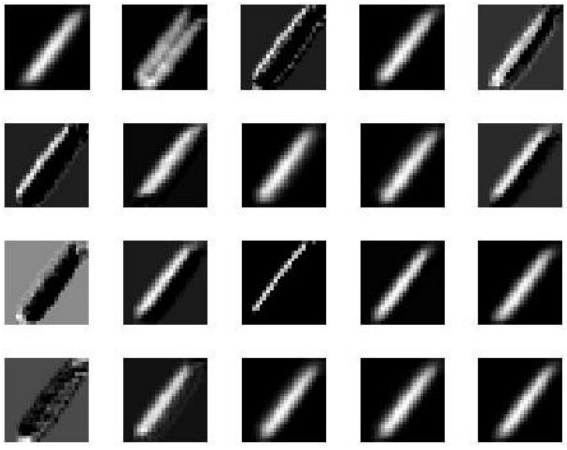


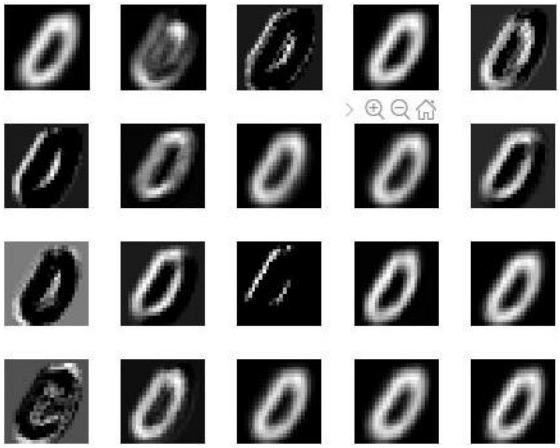
CONV





RELU





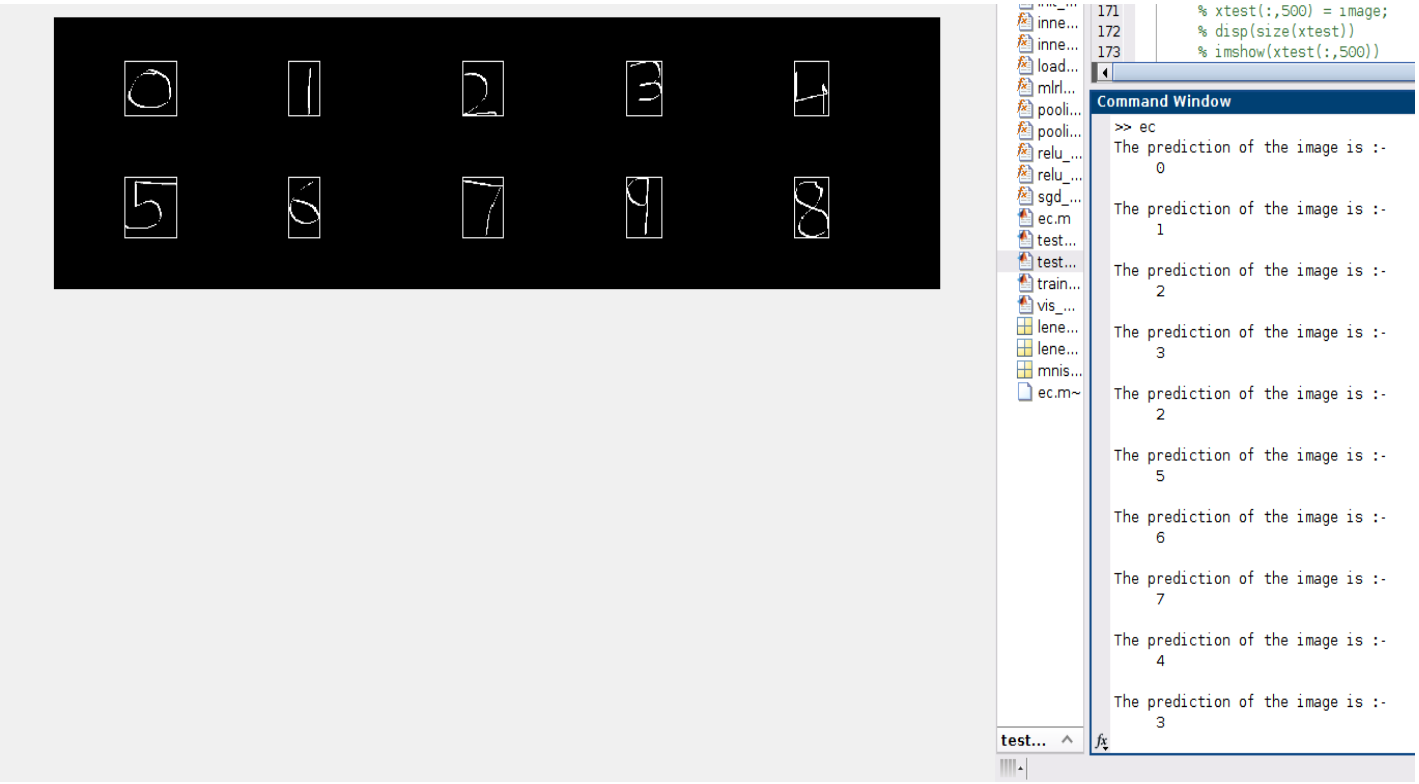
4.2. Comparison of Visualization

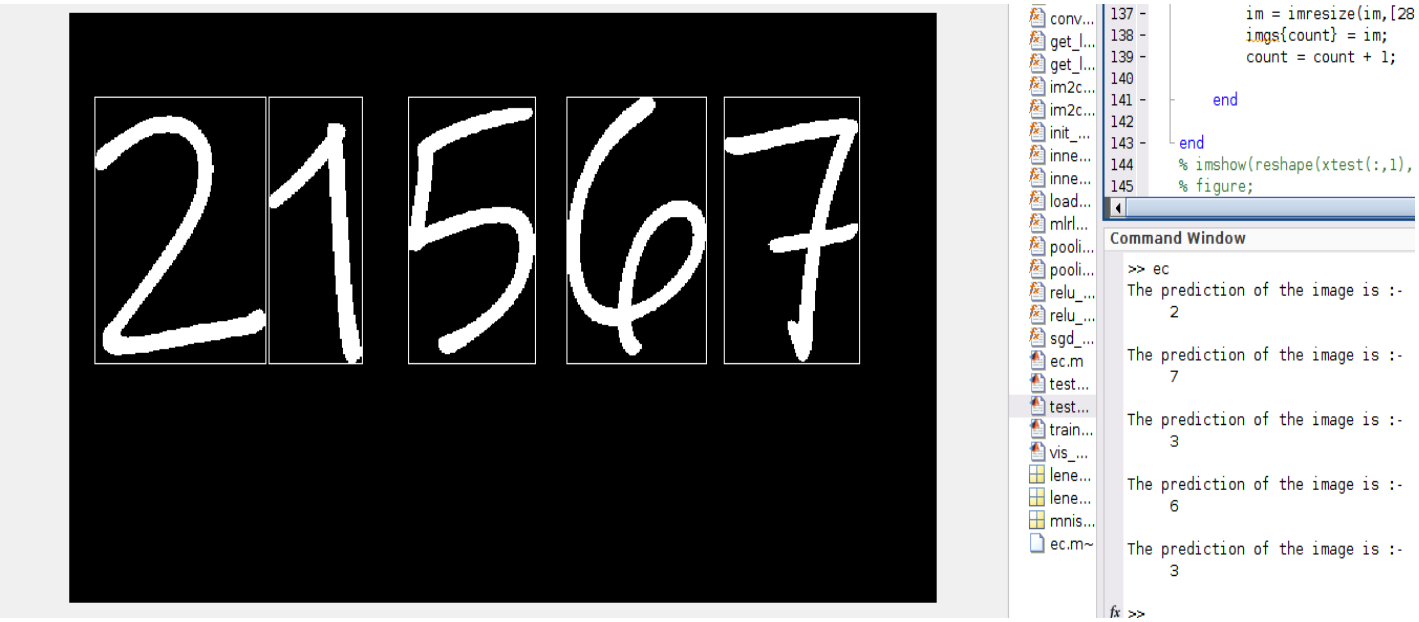
In the images of the Conv layer, it can be seen that each image represents some features like the edge of the digit, illumination at each pixel, and so on. It represents features at each pixel when studied. Some of the images depict vertical lines, horizontal lines, blur, etc.

In the images from the Relu layer, it clearly shows suppressed images. It is because Relu makes pixel values less than zero to be zero, and that is why we see a lot of black in the images. It shows more clear features coming from a conv layer like a bright vertical line, precise edges etc.

In all after these layers, we can still identify that the images are 1 and 0, but as the layer goes deeper, it becomes more pixel level and unrecognizable.

5. Image Classification





We can see some error in the prediction like in the second image 5 and 7 is detected as 3. In the first image, 4,9 and 8 are detected as 2,4,3, respectively. There are many reasons for the

inaccuracies. It can be because of the bounding box if it is too tight or if it crops it halfway. Also, if the writing style is not in a way that is generalizable by the network, then it is also one of the reasons. Also, if the background is different than the training images, then also the network is prone to errors.