Assignment 1

Wanyi Su
Student #: 301445656

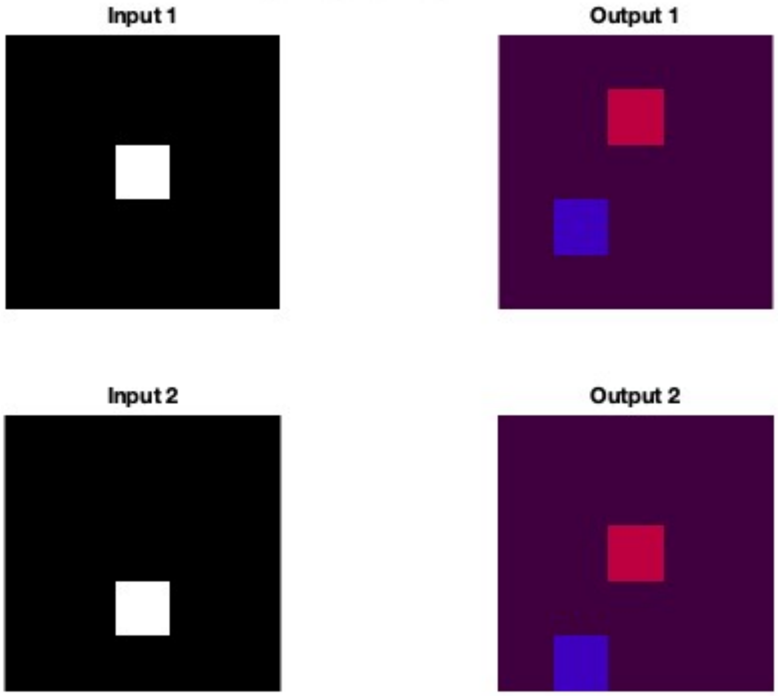
Use the extra 3 free late days.

<u>Part 1</u>

Codes are in corresponding MATLAB files.

Figures:

Convolution Test 1



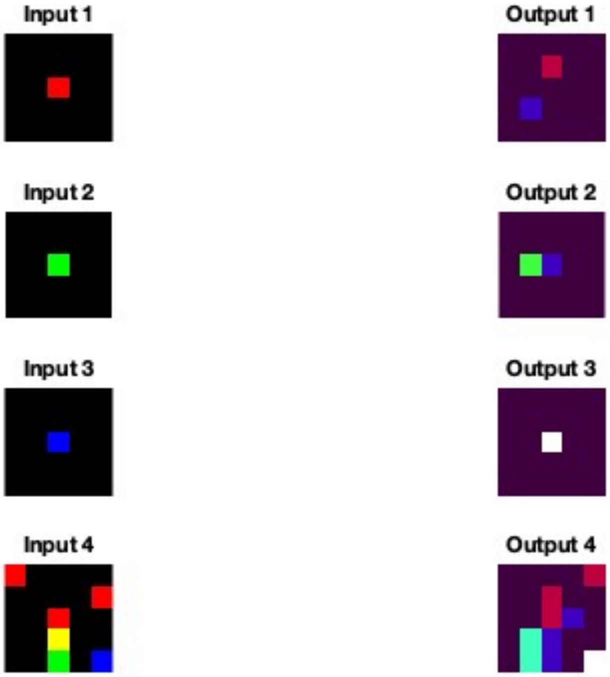
Input 1

Output 1

Input 2

Output 2

Convolution Test 2



Input 1

Output 1

Input 2

Output 2

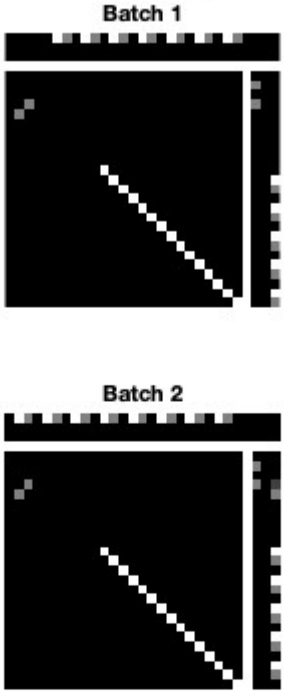
Input 3

Output 3

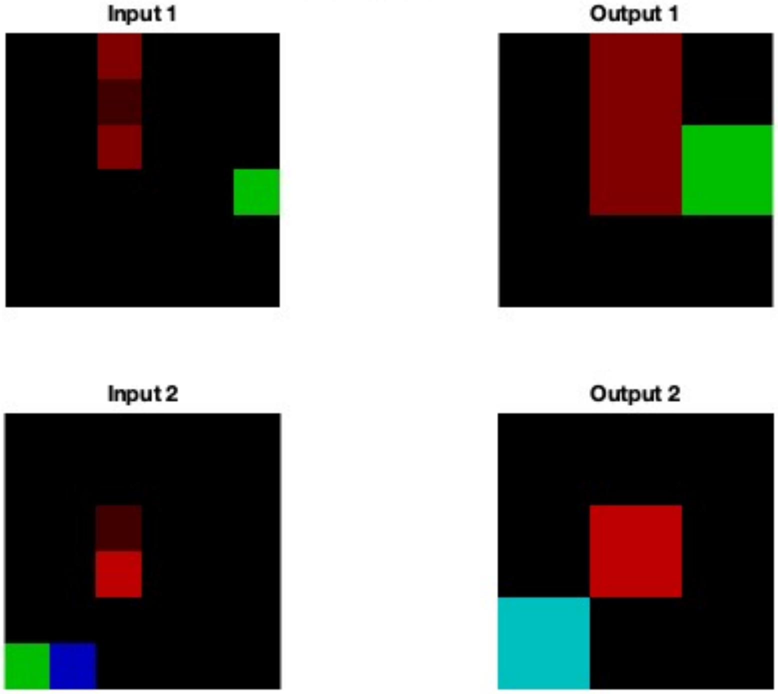
Input 4

Output 4

Inner Product Test



Pooling Test



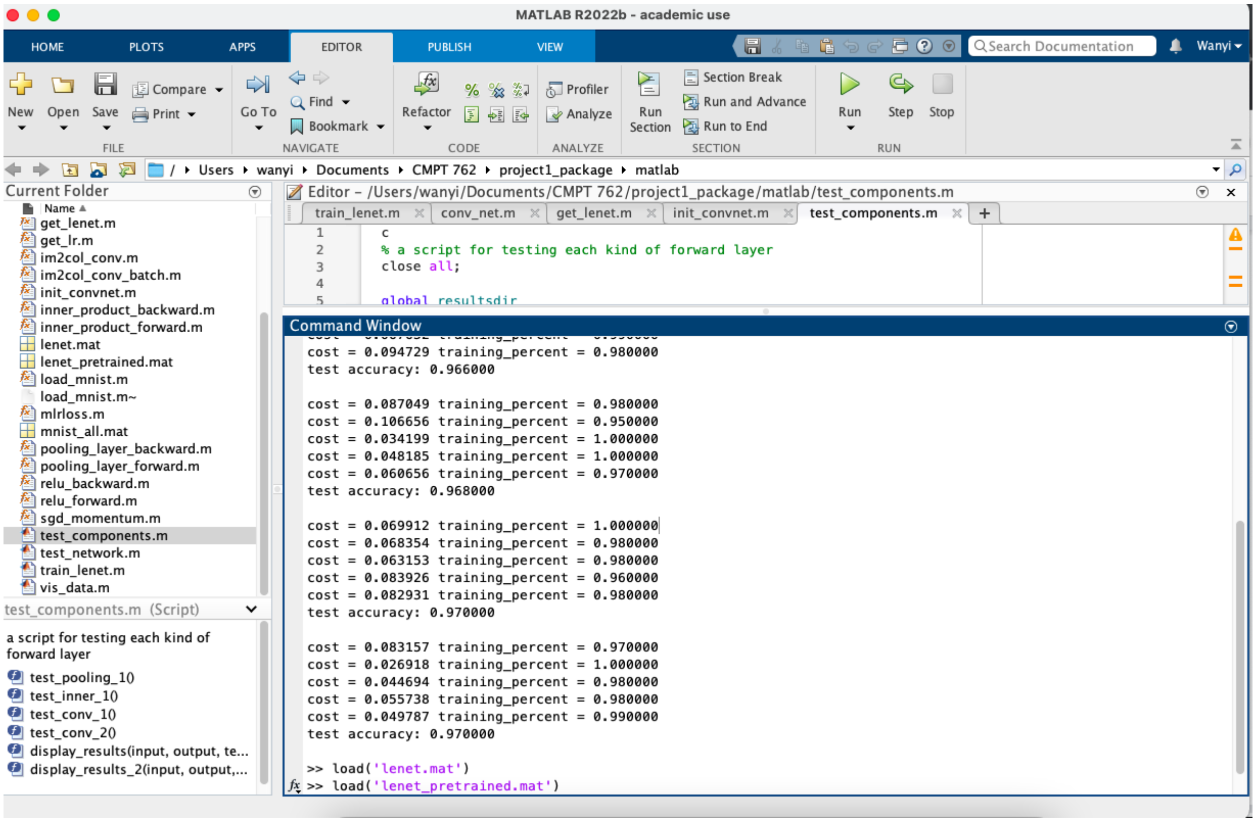
Part 2

Codes are in corresponding MATLAB files.

Part 3

Q3.1

Codes are in corresponding MATLAB files.
Test accuracy is 0.97 as below.



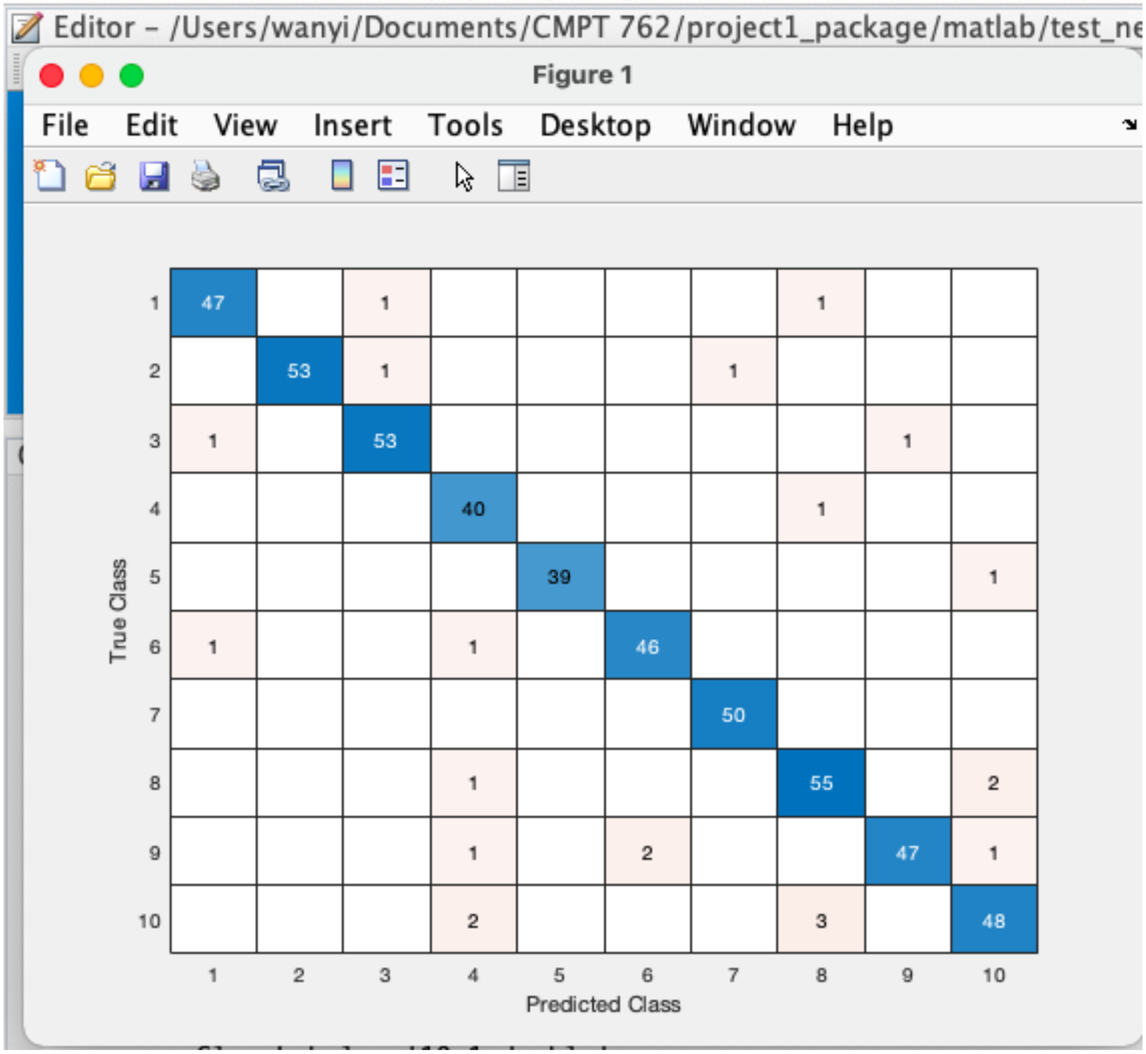
```
c
% a script for testing each kind of forward layer
close all;

global resultsdir
```

```
cost = 0.094729 training_percent = 0.980000
test accuracy: 0.966000

cost = 0.087049 training_percent = 0.980000
cost = 0.106656 training_percent = 0.950000
cost = 0.034199 training_percent = 1.000000
cost = 0.048185 training_percent = 1.000000
cost = 0.060656 training_percent = 0.970000
test accuracy: 0.968000

cost = 0.069912 training_percent = 1.000000
cost = 0.068354 training_percent = 0.980000
cost = 0.063153 training_percent = 0.980000
cost = 0.083926 training_percent = 0.960000
cost = 0.082931 training_percent = 0.980000
test accuracy: 0.970000

cost = 0.083157 training_percent = 0.970000
cost = 0.026918 training_percent = 1.000000
cost = 0.044694 training_percent = 0.980000
cost = 0.055738 training_percent = 0.980000
cost = 0.049787 training_percent = 0.990000
test accuracy: 0.970000

>> load('lenet.mat')
>> load('lenet_pretrained.mat')
```

Q3.2

We have ten classes here (“0” to “9”). The top two confusing pairs of classes are pair of class 8 (number is “7”) with class 10 (number is “9”), which has three “7” detected as “9” and two “9” detected as “7”. Another confusing pair is the pair of class 6 (number is “5”) with class 9 (number is “8”), which has two “5” detected as “8”. The main reason for confusing detection is that the pairs of numbers have similar shapes (edges or corners). When the images are input into the model after resizing, some information may lose or add because of resizing and the model can mistake one number for another when detecting numbers. The different writing styles can also result in the errors.

Confusion matrix chart is shown as screenshot below:



	1	2	3	4	5	6	7	8	9	10
1	47		1					1		
2		53	1				1			
3	1		53						1	
4				40				1		
5					39					1
6	1			1		46				
7							50			
8				1				55		2
9				1		2			47	1
10				2				3		48

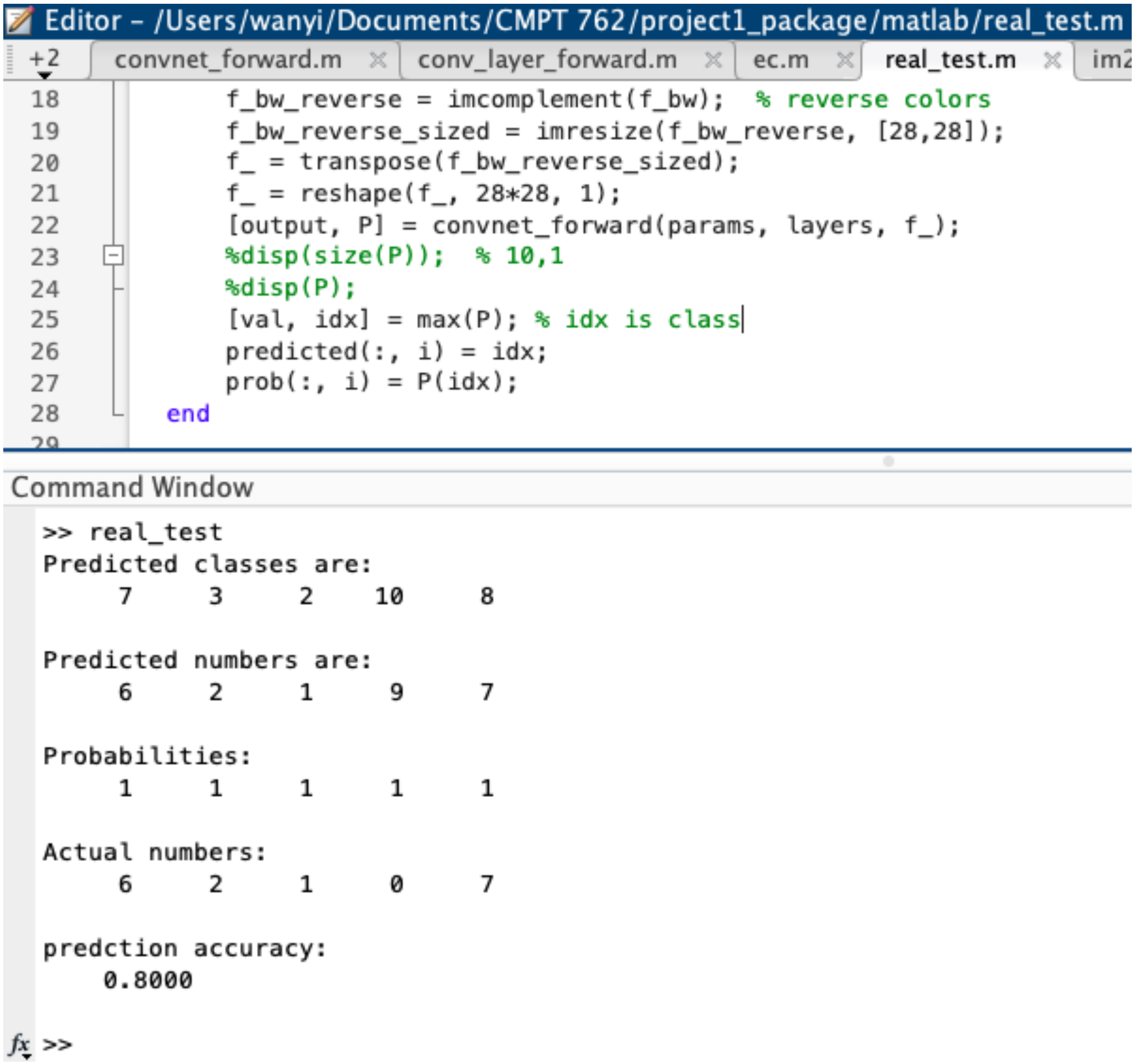
Part 3.3

Code is in **real_test.m** file.

First resize the .jpg figures and turn them into black background and white characters pictures, then do the prediction. Finally, we correctly classify 4 out of 5 numbers I wrote, and the accuracy is 0.80.

The model misclassified number 0 as “9”, which is understandable, since the two numbers are similar. The model might detect some tiny pixels in the bottom right side of “0” and recognize it as “9”.

Screenshot of predicted numbers, corresponding probabilities, predicted accuracy are as follows:


Editor – /Users/wanyi/Documents/CMPT 762/project1_package/matlab/real_test.m
+2
convnet_forward.m
conv_layer_forward.m
ec.m
real_test.m
im2
Command Window

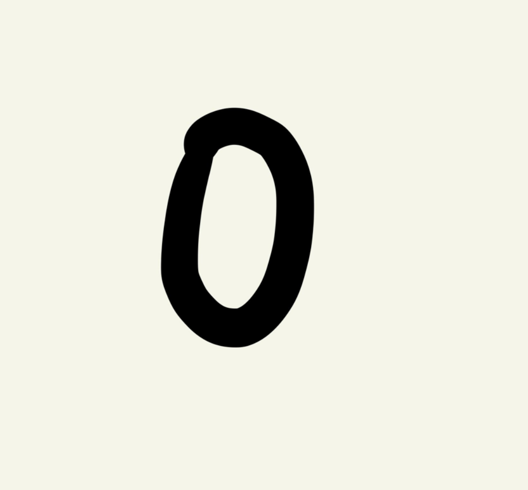
```matlab
        f_bw_reverse = imcomplement(f_bw);  % reverse colors
        f_bw_reverse_sized = imresize(f_bw_reverse, [28,28]);
        f_ = transpose(f_bw_reverse_sized);
        f_ = reshape(f_, 28*28, 1);
        [output, P] = convnet_forward(params, layers, f_);
        %disp(size(P));  % 10,1
        %disp(P);
        [val, idx] = max(P); % idx is class
        predicted(:, i) = idx;
        prob(:, i) = P(idx);
    end
```

```
>> real_test
Predicted classes are:
     7     3     2    10     8

Predicted numbers are:
     6     2     1     9     7

Probabilities:
     1     1     1     1     1

Actual numbers:
     6     2     1     0     7

predction accuracy:
    0.8000

>>
```

Input handwritten numbers are as follows:

6

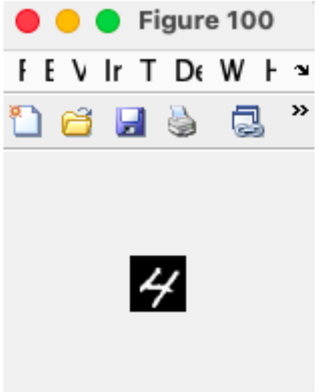
2

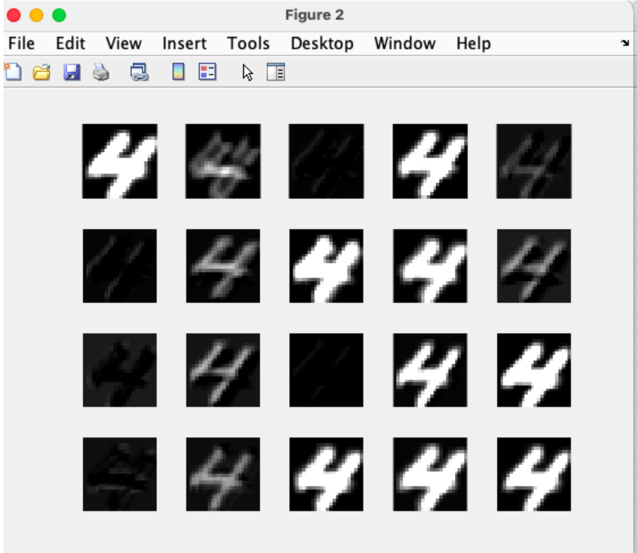


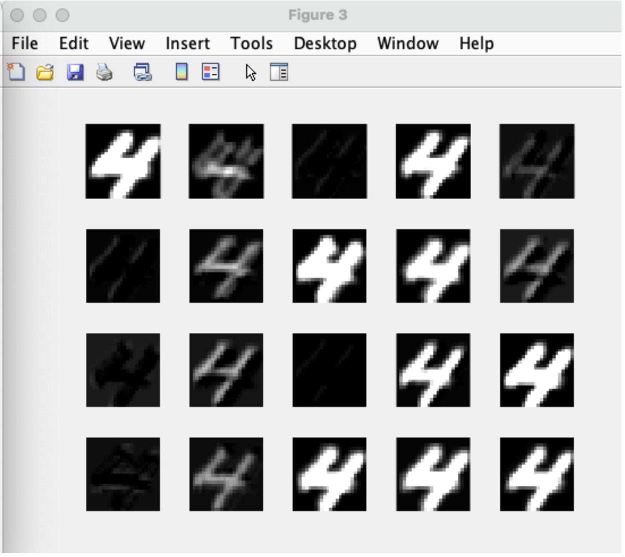
Part 4

Part 4.1

The 20 images from each of two layers are as below.







Part 4.2

The feature maps of 20 images from each of layer 2 as well as layer 3 and original image are quite similar in shape, but compared to original image, feature maps are blurred and are in different blur levels in 20 channels.

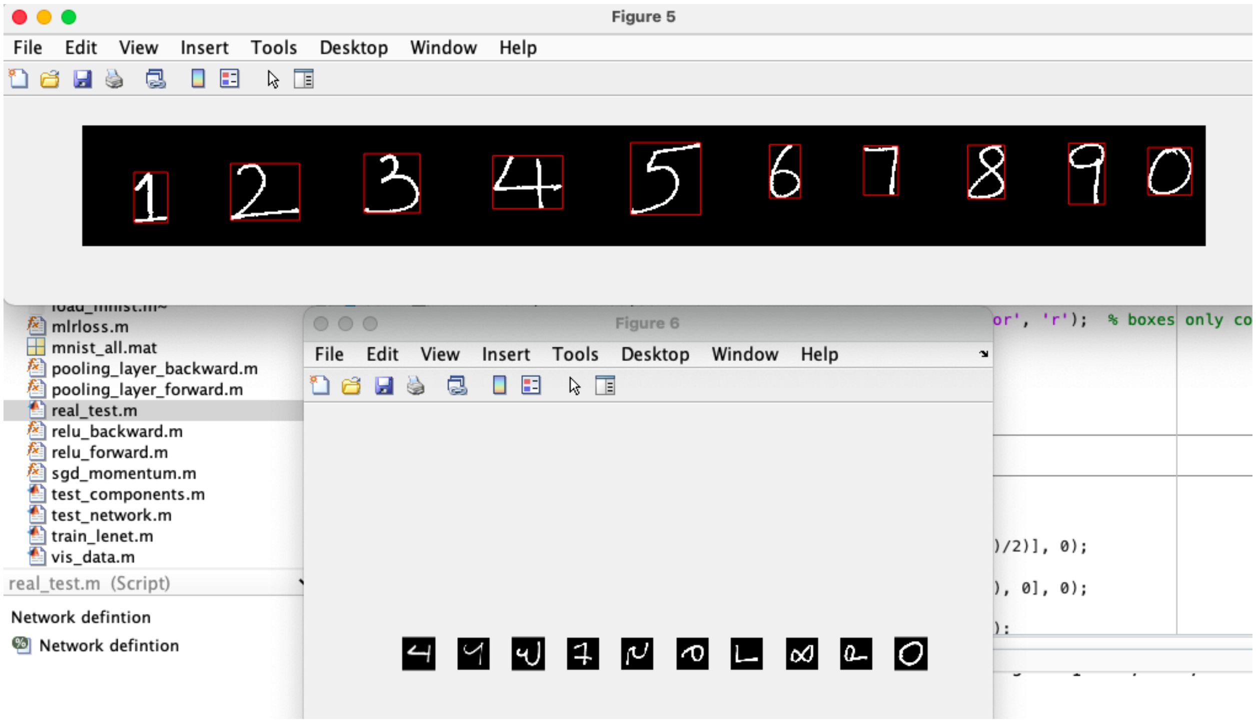
The reason can be as follows.

As we filter the original image into 20 channels in CONV and RELU layers, the feature maps in each channel can be blurred than the original image and have different light levels, while shapes remain similar since CONV layer can capture important features of the original image.

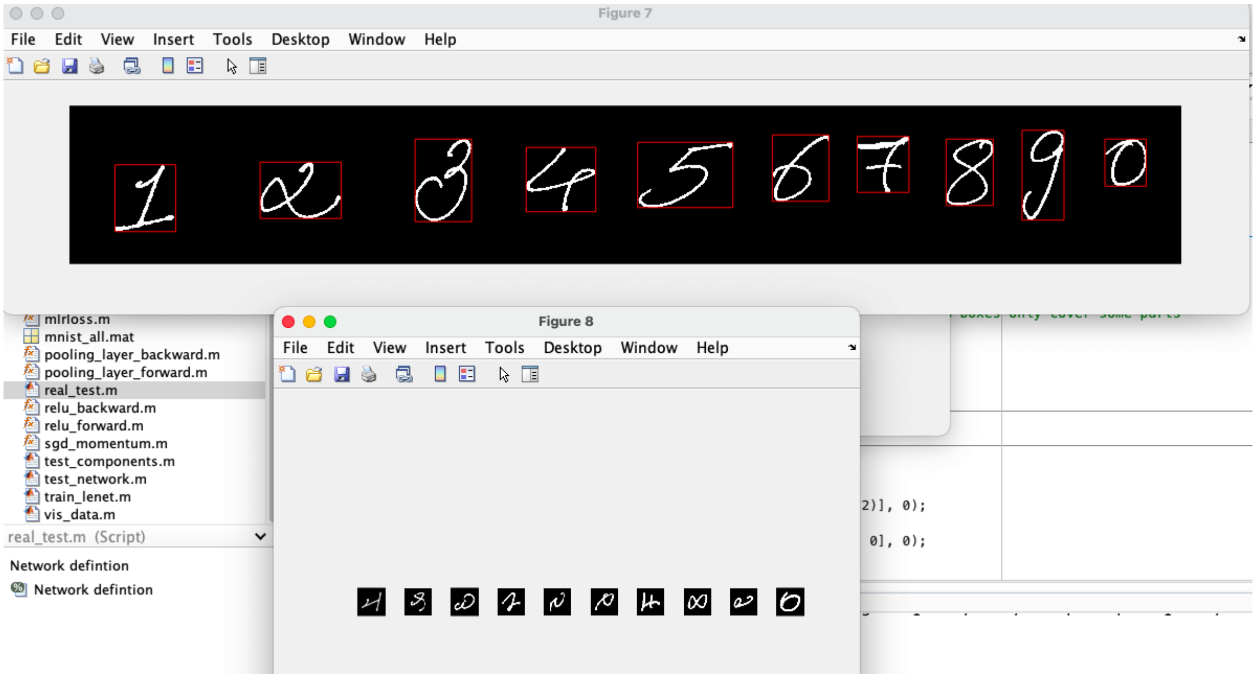
When the original image input first comes through CONV layer, there are both negative and positive features in the output. When we plot the output into 2D grayscale figures, negative numbers are treated as 0 while under image form. After CONV layer, data is input into RELU layer, where data is simply processed by MAX(0, data) function for each channel and only 0 and positive values are left. Therefore, the images of each channel from 1st layer of CONV and 2nd layer of RELU, which both have 20 channels, are similar to the original image input.

Part 5

Images with bounding boxes are as below:



```
or', 'r');  % boxes only co
)/2)], 0);
), 0], 0);
);
```



```
boxes only cover some parts
2)], 0);
0], 0);
```

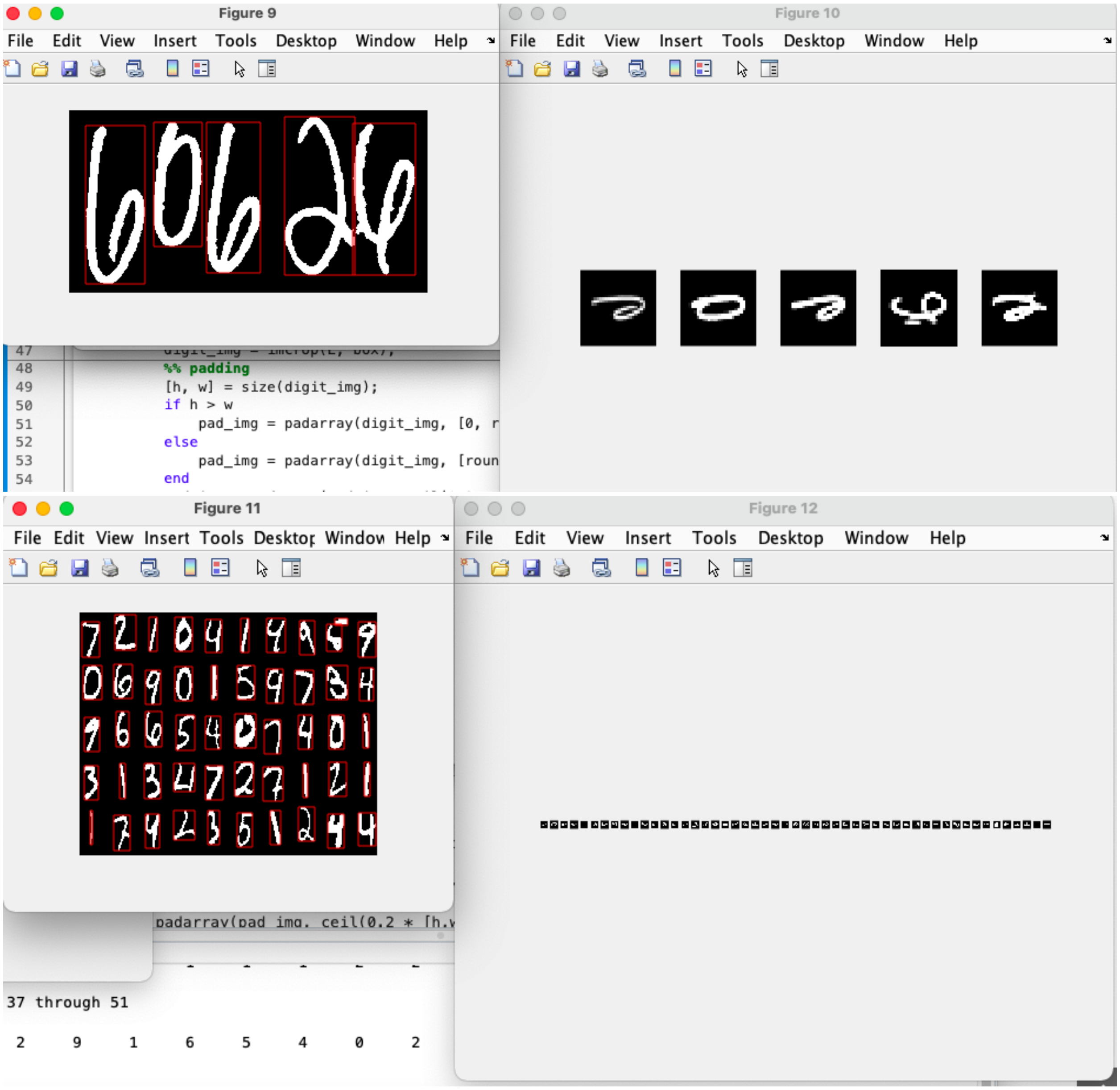


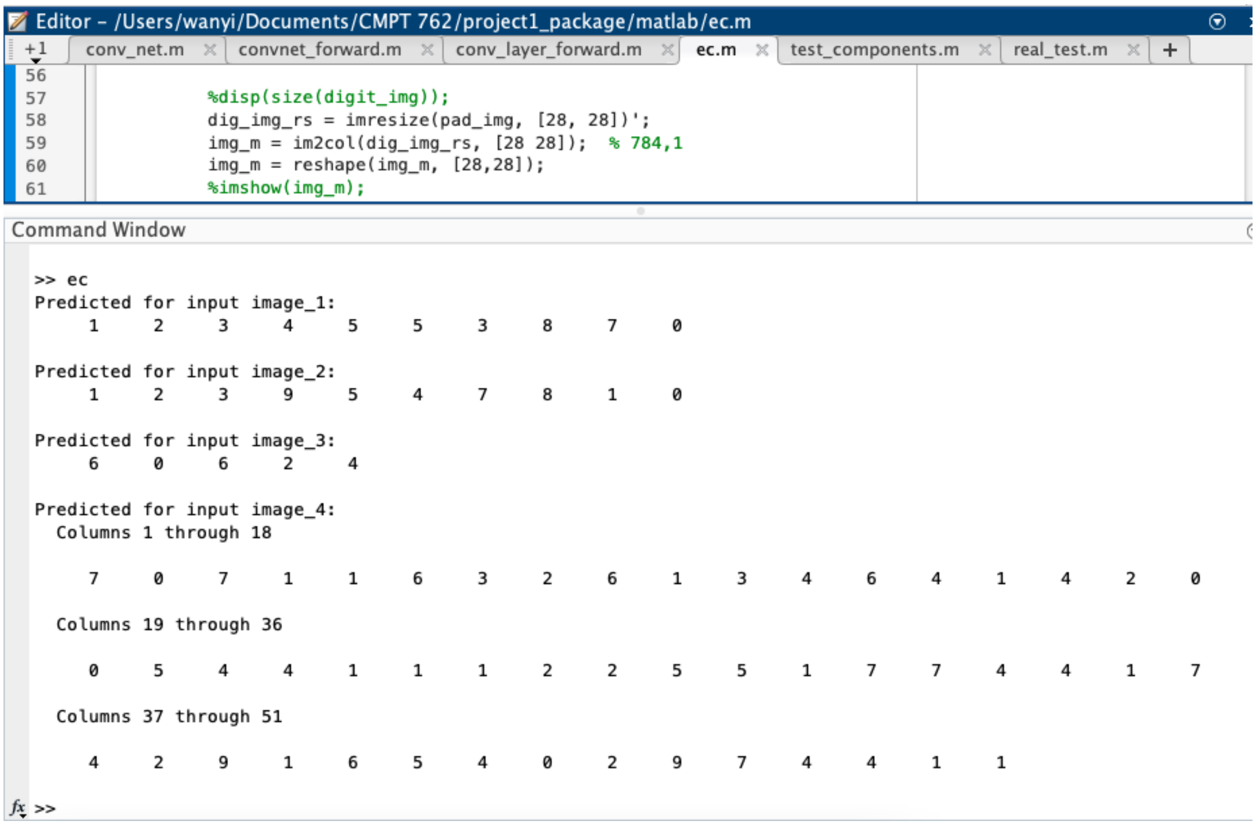
Figure 9
File Edit View Insert Tools Desktop Window Help
Figure 10
File Edit View Insert Tools Desktop Window Help
Figure 11
File Edit View Insert Tools Desktop Window Help
Figure 12
File Edit View Insert Tools Desktop Window Help

```
        digit_img = imcrop(L, box);
        %% padding
        [h, w] = size(digit_img);
        if h > w
            pad_img = padarray(digit_img, [0, r
        else
            pad_img = padarray(digit_img, [roun
        end
```

```
padarray(pad_img, ceil(0.2 * [h,w
```

```
37 through 51

 2     9     1     6     5     4     0     2
```

Predicted results are as follows:



```
        %disp(size(digit_img));
        dig_img_rs = imresize(pad_img, [28, 28])';
        img_m = im2col(dig_img_rs, [28 28]);  % 784,1
        img_m = reshape(img_m, [28,28]);
        %imshow(img_m);
```

```
>> ec
Predicted for input image_1:
     1     2     3     4     5     5     3     8     7     0

Predicted for input image_2:
     1     2     3     9     5     4     7     8     1     0

Predicted for input image_3:
     6     0     6     2     4

Predicted for input image_4:
  Columns 1 through 18

     7     0     7     1     1     6     3     2     6     1     3     4     6     4     1     4     2     0

  Columns 19 through 36

     0     5     4     4     1     1     1     2     2     5     5     1     7     7     4     4     1     7

  Columns 37 through 51

     4     2     9     1     6     5     4     0     2     9     7     4     4     1     1

fx >>
```

Predicted results for the first two images are at 70% accuracy, and 100% accuracy for the third image. For the fourth image, since “5” are separated into two obvious parts (one horizontal line on the top and the remaining part of “5”), the model detects it as a separate object. This is why we got 51 objects detected in the result.

There are multiple reasons for errors of prediction here. First, the characters are written by hand and different people have different writing styles, which can cause bias when the model tries to predict numbers. Second, the input images are cropped and resized before being put into and detected by the model. The process can add some disturbance information that causes prediction errors.