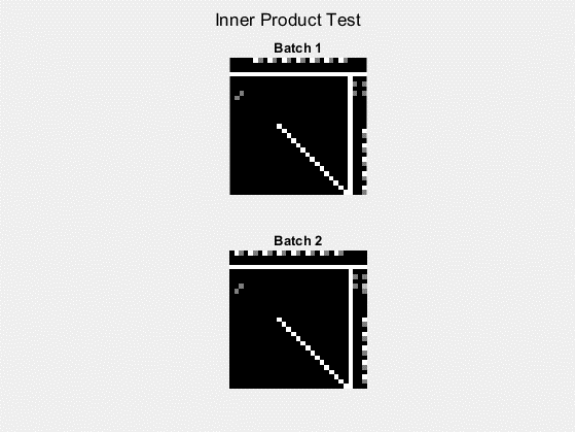
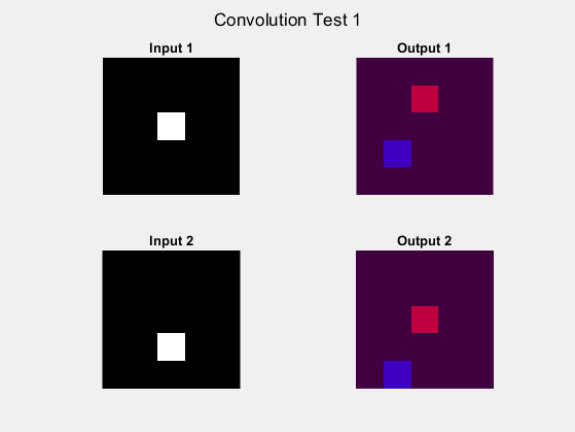
Project 1

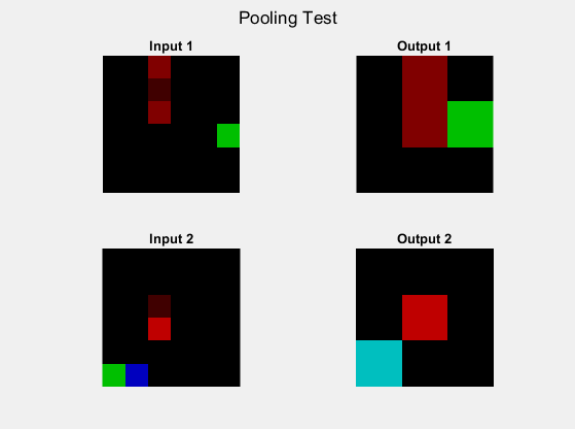
Digit Recognition with Convolutional Neural Networks

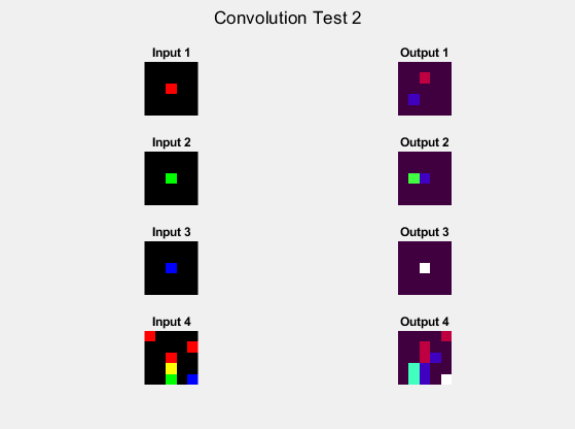
1. Forward Pass

The results obtained from the test_components.m are as follows. This matches exactly to the reference images in the provided document.









3. Training Loop

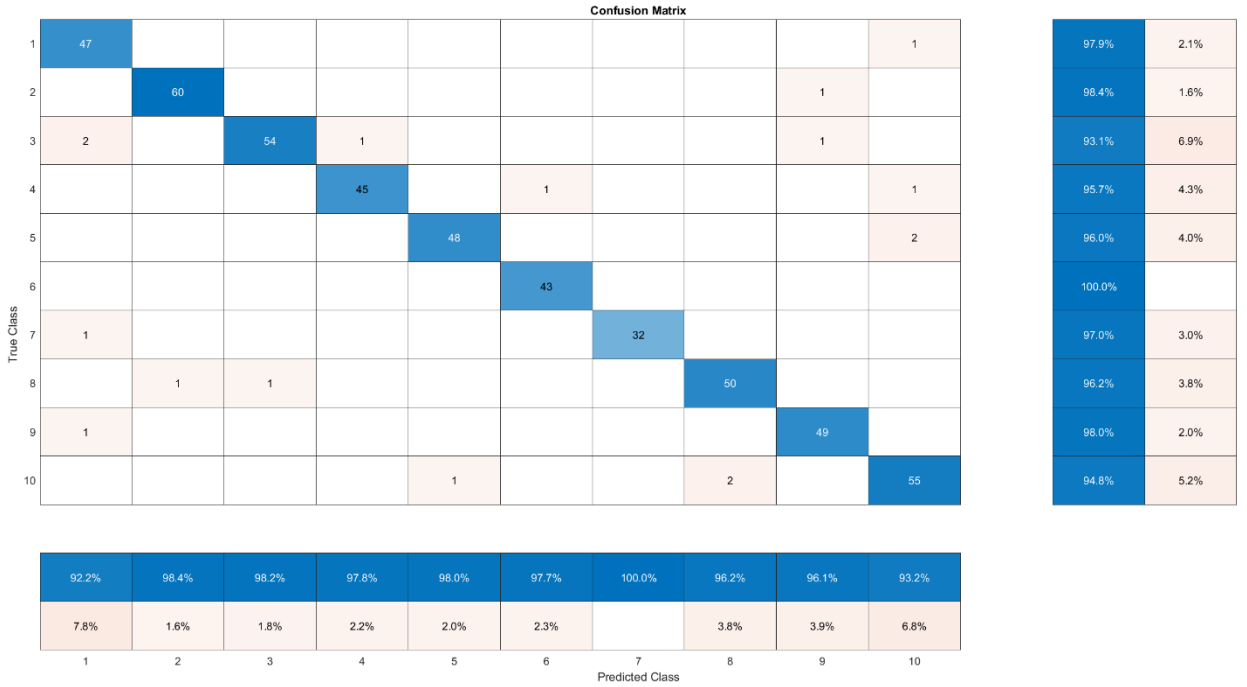
3.1. Training

After 3000 iterations and the model being "tested" 6 times (after every 500 iterations) the testing accuracy plateaued at around 96.670 % and the training accuracy at 100 iterations before was around 99.0% with cost being 0.047235.

cost = 0.047235 training_percentage = 0.990000
test_accuracy: 0.966700

3.2. Testing

Here is the output of the confusion matrix. In this image it is quite visible that the 1st class (i.e., digit 0), 8th class (digit 7) and the 10th class (i.e., digit 9) were most often confused during the testing phase. The digit 0, digit 7 and digit 9 were predicted as digit 2, digit 9 and digit 4 respectively. This is because the digit 4, 7 and 9 all have extended stalk on the right which is why the model confuses the three digits and probably the digit 0 and digit 2 both have curved shapes on the top right which makes the model indecisive between the two digits.



3.3. Real world Testing

Here is the list of digits that I myself scribbled on a piece of paper using a pen (file format 0d) and a pencil (file format d):

I used a pen and a pencil to get a change in distribution (both set of images have different filters applied to them with contrast set to 100 in one of the sets[1]). I manually cropped the images and applied image thresholding, image inversion and resizing before passing it through the network. These are the results of the network:

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

[1] Filters and the contrast changes were applied using Microsoft Photos.

4. Visualization

4.1. Filters

Here's the output of the vis_data.m script. These are the feature maps when the data has passed through the second (convolution layer [shape: 24, 24, 20]) and the third layer (ReLU layer [shape: 24, 24, 20]).

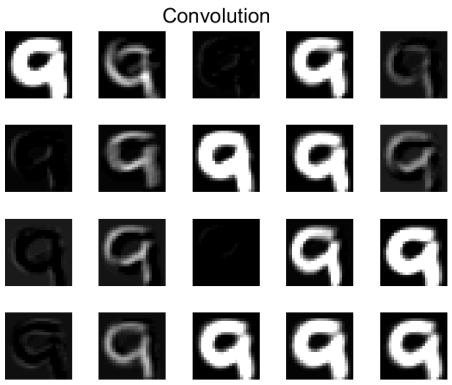


Figure 7: Feature Map of Convolution Layer (Layer 2)

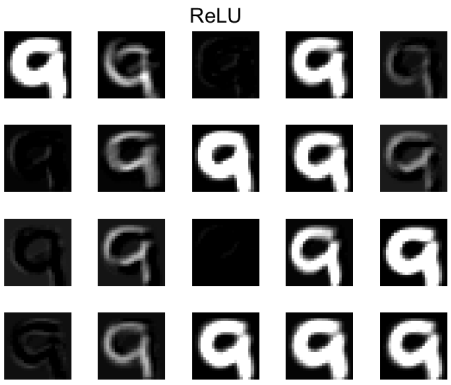


Figure 8: Feature Map of ReLU Layer (Layer 3)

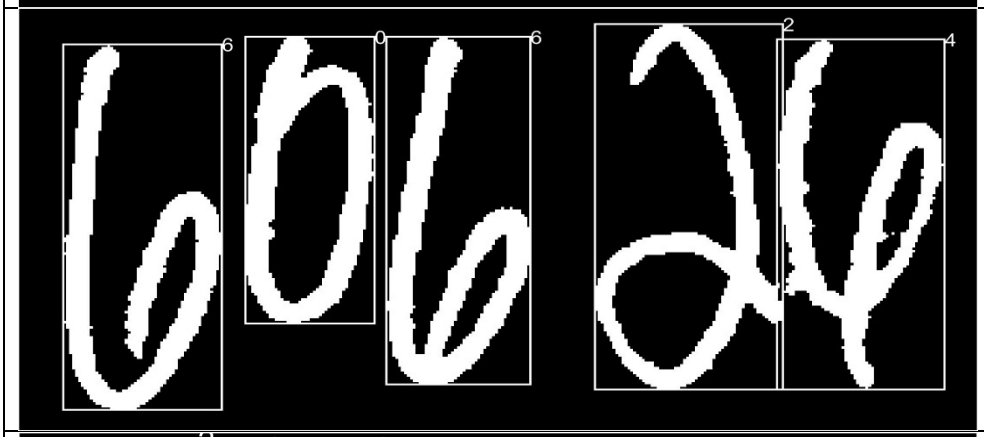
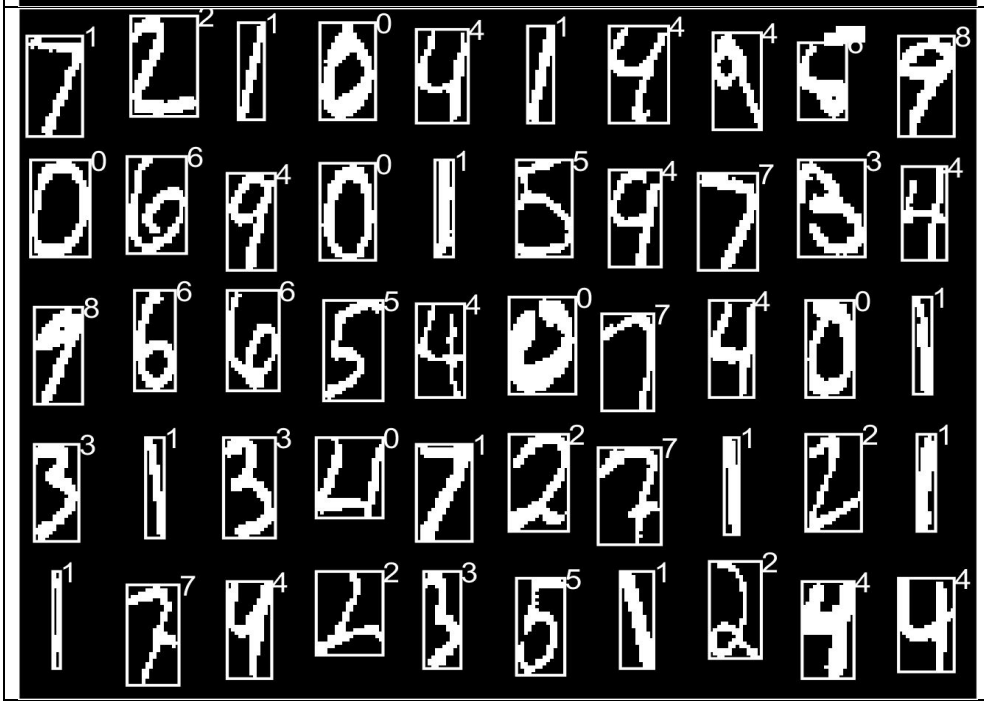
4.2. Explanation

It is quite evident from the two images (Figure 7 and Figure 8), which show the feature maps of the two layers, The convolution layer gathers features form the input image, which appears to be the digit 9. The Convolution layer detects different contours in the digit. The left bottom three are looking for the silhouette of the digit 9, whereas other feature maps try to get information that is of varying brightness and some of the feature maps show the edge detection from different angles.

Similarly, the ReLU layer shows a very similar feature map to what is evident in the second layer (Convolution Layer). This is because ReL Unit just takes the maximum of 0 or x, which means it eliminates the negative numbers, this is depicted as the same because the MATLAB function imshow already filters out the negative numbers while viewing the image, so the two images are exactly the same.

5. Image Classification

The Image Classification script ec.m produces above average results. For this task I employed image binarization, inversion, bounding box, image dilation, padding and reshaping before sending it to the convolution neural network. The average accuracy across the four images is around 73.33%.

Image	Correct Predictions[2]
	3/10
	7/10
	4/5
	41/50

[2] Open to interpretation