**Teach a Neural Network to Read Handwriting**

(Spriha Srivastava)

**Abstract**

HANDWRITTEN digit recognition is the ability of a computer system to recognize the handwritten inputs like digits, characters etc. from a wide variety of sources like emails, papers, images, letters etc. This has been a topic of research for decades. There are many applications for handwriting recognition are available this day. There are many techniques that have been developed to recognize the handwriting

In this internship, I have made a neural network that will read and recognise the characters based on MNIST character dataset. I have used one of the widely used algorithms, SVM (Support Vector Machine).

**Advantages of using SVM:**

1. Support Vector Machine gives a high accuracy of 97.91%, whereas other Machine Learning algorithm had lesser accuracy
2. Support vector machines are naturally resistant to overfitting and work very well when identifying boundary regions.
3. Robust to noises.
4. Easier to implement, personally I feel it is easier for me to implement SVM classifier.
5. The advantage of SVM is that once a boundary is established, most of the training data is redundant. All it needs is a core set of points which can help identify and set the boundary.

**Implementation**

1. The four MNIST data file are already downloaded and stored into a folder named dataset, these four files are divided into training set and test set. Each training and test set consist of images and labels.
2. Unzip the file
3. Navigate to: Neural Network// SVM and run the following python command >>python svm.py
4. Once the execution begins, the program loads the MNIST data which includes both training and testing data. It the prepares the classifier and also plots a confusion matrix.
5. Apart from the confusion matrix it also tells us about predicted labels for test images and accuracy of classifier on test images.
6. In the end, it shows the Test Images with Original and Predicted Labels.

**Attached Screen Shots:**

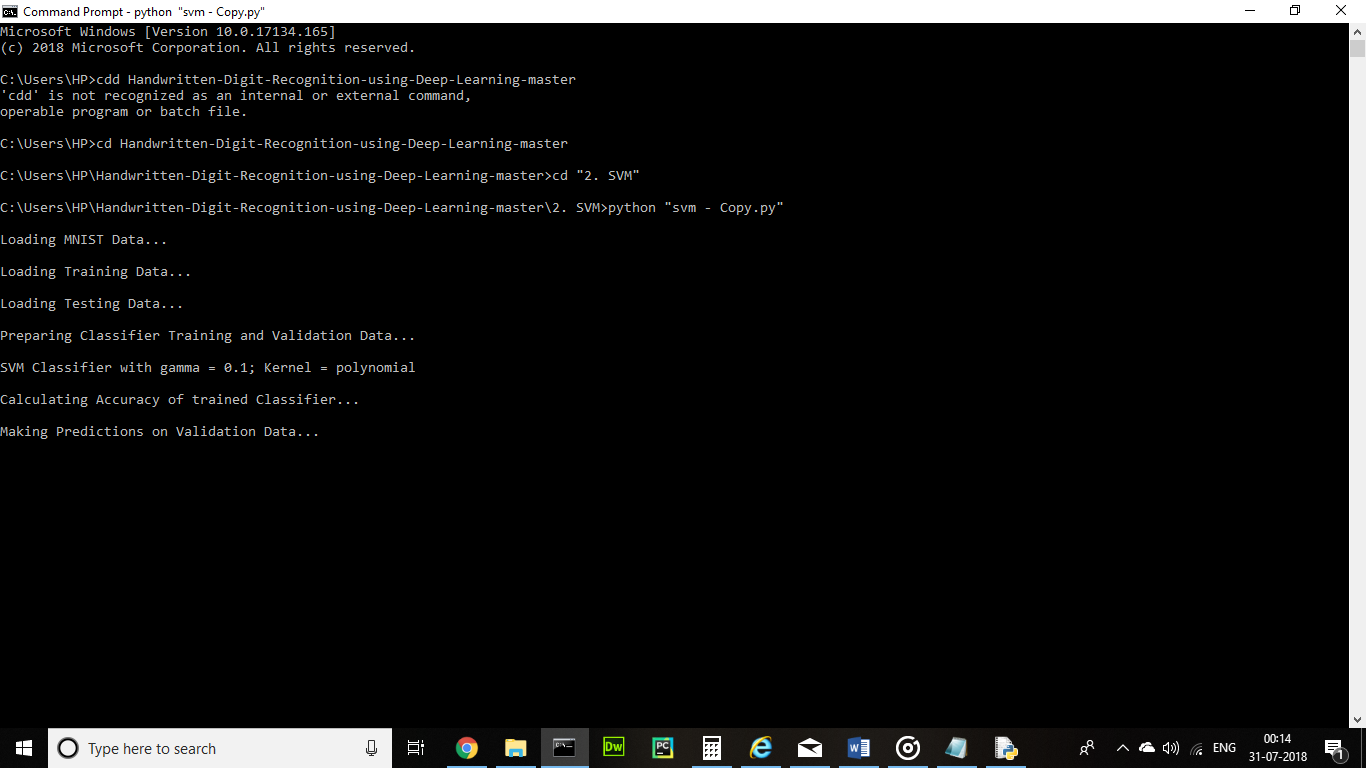


Figure 1: The execution of code

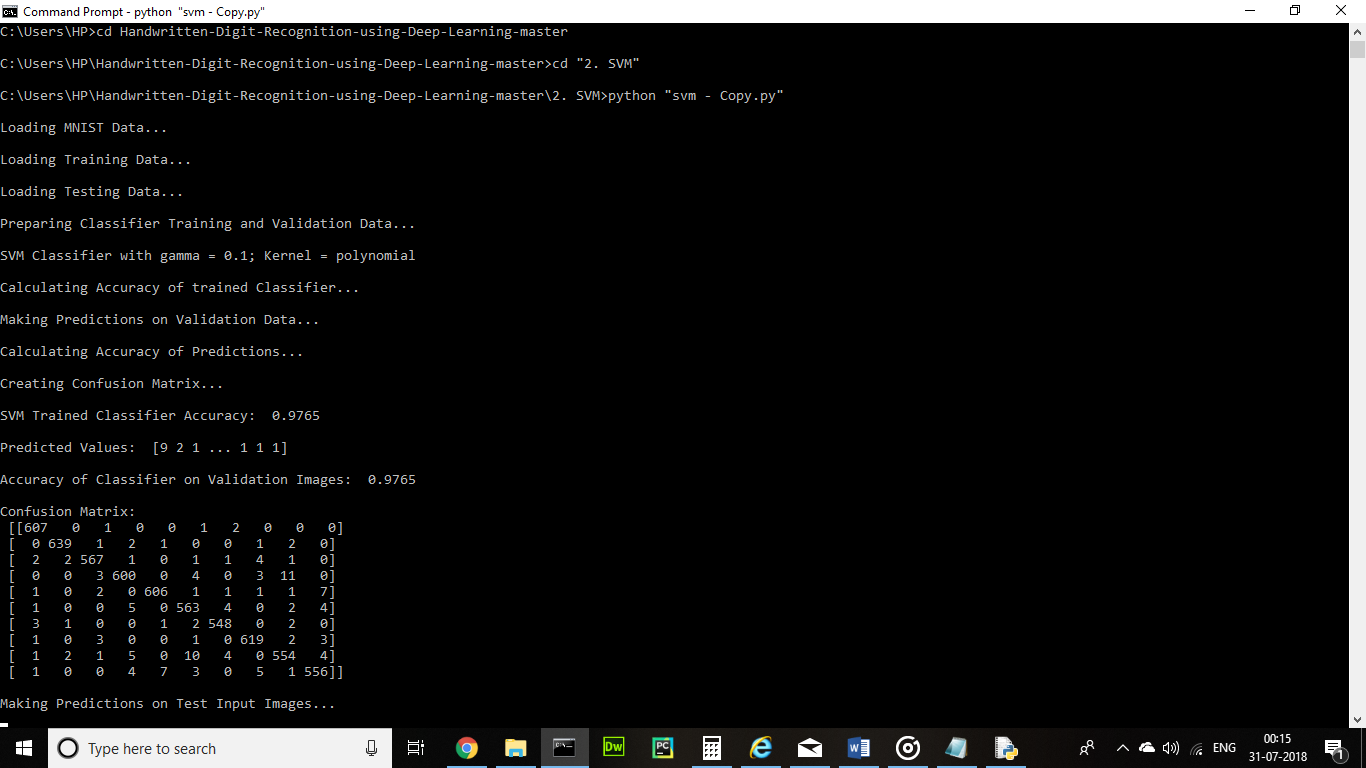


Figure 2: Confusion Matrix as Matrix



Figure 3: Confusion Matrix plotted

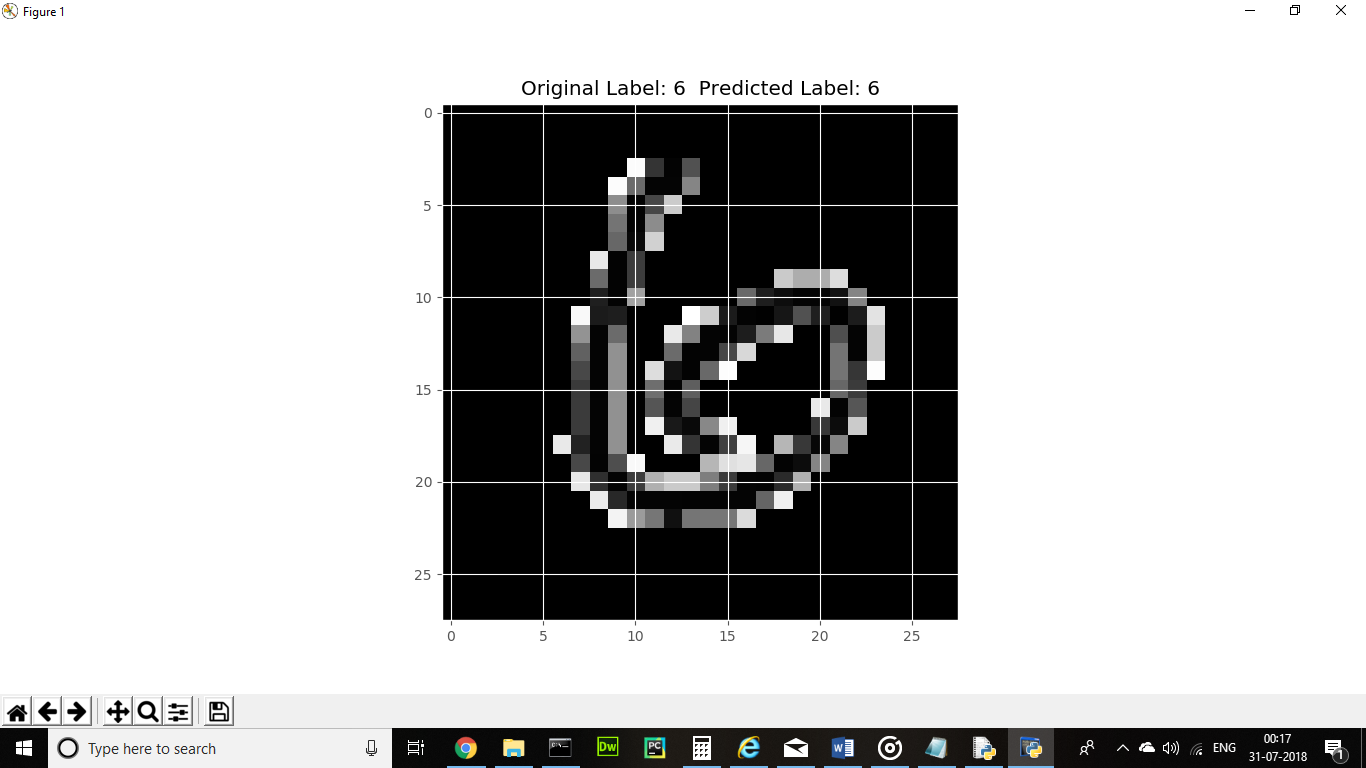
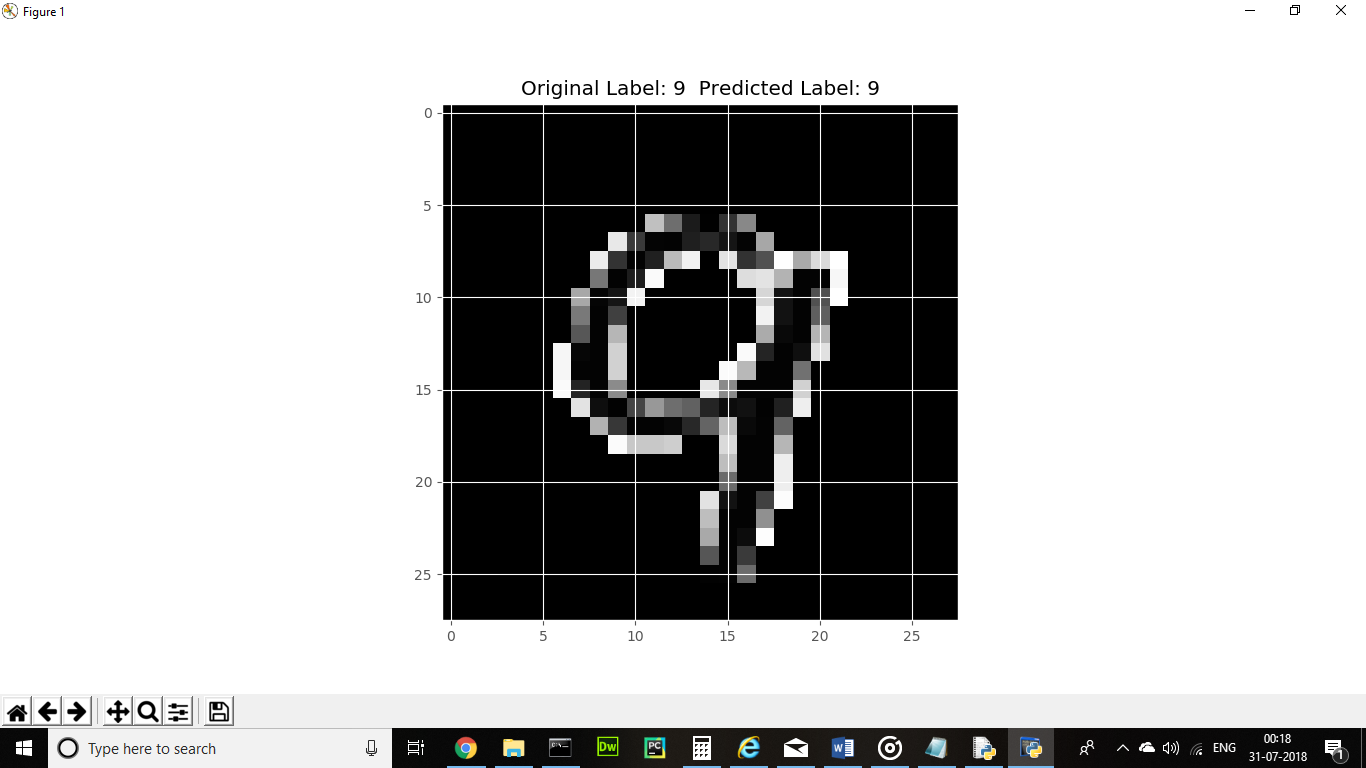


Figure 4: Test image 1 with original and predicted label

  
Figure 5: Test image 2 with original and predicted label

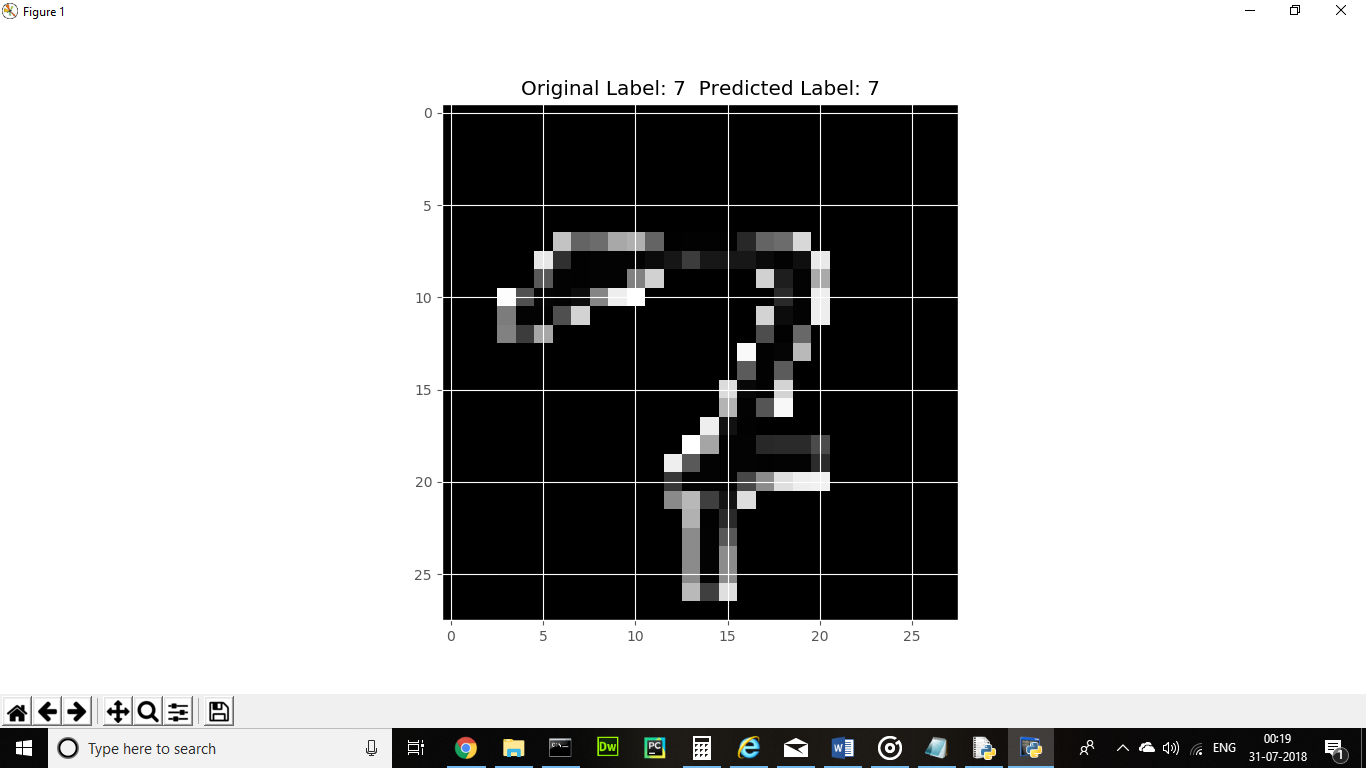


Figure 6 Test image 3 with original and predicted label