

# Recognition of Handwritten Digits using Machine Learning Techniques

Shobhit Srivastava<sup>#1</sup>, Sanjana Kalani<sup>#2</sup>, Umme Hani<sup>#3</sup>, Sayak Chakraborty<sup>#4</sup>

Department of Computer Science and Engineering

Dayananda Sagar College of Engineering

Bangalore, Karnataka, India

**Abstract**—This paper illustrates the application of object character recognition (OCR) using template matching and machine learning techniques to solve the problem of handwritten character recognition. In this paper we perform the recognition task using Template Matching, Support Vector Machine (SVM), and Feed Forward Neural Network. Template matching is an image processing technique to break the image into smaller parts and then match to a template image. Here we use a Multi Class SVM classifier and Neural Network to classify the image. We use the dataset to train the classifier followed by feature extraction and finally applying the classifiers to recognize the digits.

**Keyword** — Support Vector Machine, Template Matching, Neural Networks, Feature Extraction

## I. INTRODUCTION

Handwritten image recognition is probably one of the most interesting and challenging applications in the field of pattern recognition. Handwritten recognition is divided into two types of techniques: on-line and off-line. Off-line techniques include reading the character using an image capture device, such as a camera, while the technique which is being dealt here is Off-line which means to convert a handwritten image into a machine readable form.

The major factor behind choosing this particular application is its numerous applications such as Automatic Number Plate Recognition, assisting blind and visually impaired people, automatic check processing for banks, and to process huge number of documents in industries like healthcare, legal, education, and finance the focus of the work described in this paper is on handwritten digits. The paper will further be covering data collection, image pre-processing, feature extraction, and finally classification.

## II. DATA ACQUISITION

The data used in this project is a set of handwritten digits from 1 to 10. The data has been divided into two categories which form the training set and the test set. Sets of data were collected which are the phone numbers, zip-codes, and address plates for testing purpose.

## III. IMAGE PREPROCESSING

The size of the image used in this project is 25 by 25 pixels. The steps used in pre-processing the image is represented in Fig 1.

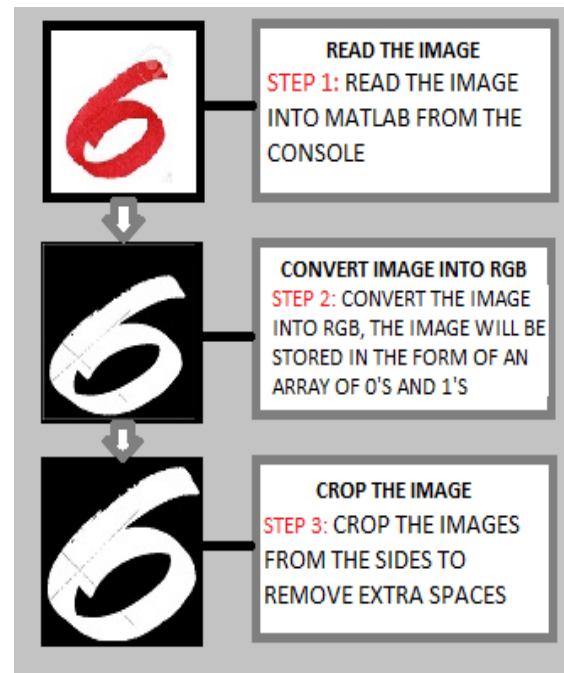


Fig. 1. Steps of Pre-Processing

Finally the image is resized to 25 by 25 pixels. The images used for testing consists of more than one digits which needs to be separated into individual digits before applying the pre-processing steps.

## IV. FEATURE EXTRACTION

**Blob Analysis:** Blob or Binary Large Object is a large image which needs to be managed, and consists of binary data. In this project we used images that consists of a sequence of digits. The image is then converted into a binary image using the *bwlabel* function in matlab, where the image is processed according to the connected components concept.

**Connected Components:** The concept is based on grouping similar pixels according to pixel connectivity. The connected components have similar levels of pixel intensity, and after grouping, each pixel is labelled according to the component it belongs to. K-connected components (here K=8), algorithm is used in this work.

The process of classification is performed using template matching, SVM and Neural Network approach.

### V. TEMPLATE MATCHING

Template Matching is a computer vision technique which is used to recognize the elements in the image by matching it with a predefined template. The process is elaborated in the flow diagram in Fig. 2.

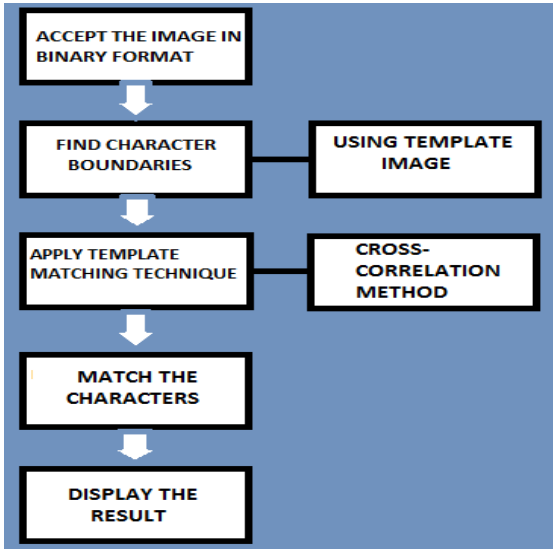


Fig.2. Flow Diagram For Template Matching Technique

**Image Correlation:** The main goal of this technique is to find similarity between images of equal dimensions. The technique used to perform this task is Cross-Correlation, and it is defined to be the sum of pairwise multiplications of corresponding pixel values. The major disadvantage of this technique is that brightening of the image will increase the cross-correlation with another image even if the pixel values of the second image are not similar, as shown in Fig. 3.

IMAGE 1	IMAGE 2	CROSS-CORRELATION
		187366474
		222556526
		250015641

Fig.3. Cross-Correlation Depending On Brightness Of The Image

### VI. SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) is one of the most popular classification algorithms used in the field of machine learning. SVM was initially built to perform binary class classification, that is one against all other classes which builds one SVM per class. Here the technique used is one-against-one which builds one SVM for each pair of classes. This method constructs  $n(n-1)/2$  classifiers where each one is trained on data from two classes. For training data from  $i^{th}$  and  $j^{th}$  classes we use the following binary classification ,

$$\min_{w^{ij}, b^{ij}, \xi_t^{ij}} \frac{1}{2} (w^{ij})^T w^{ij} + C \sum_t \xi_t^{ij}$$

$$(w^{ij})^T \phi(x_t) + b^{ij} \geq 1 - \xi_t^{ij}, \text{ if } y_t = i,$$

$$(w^{ij})^T \phi(x_t) + b^{ij} \leq -1 + \xi_t^{ij}, \text{ if } y_t = j,$$

$$\xi_t^{ij} \geq 0.$$

Fig. 4. Equation For Svm

If sign  $((w^{ij})^T \phi(x) + b^{ij})$  suggests that  $x$  is in the  $i^{th}$  class, then the votes for the  $i^{th}$  class is incremented by one. Otherwise, the votes for the  $j^{th}$  class are incremented by one. Then, we predict  $x$  is in the class with the largest number of votes. The voting approach described above is also called the “Max Wins” strategy. In case that two classes have identical number of votes we simply select the one with the smaller index (this might not be a good strategy but is adopted for simplicity). Practically we solve the dual of problem described in Fig. 4 where the number of variables is same as the number of data points in the two classes. Hence, if on average each class has  $(l/n)^{th}$  of the data points, we have to solve  $n(n-1)/2$  quadratic programming problems where each of them has about  $2/n$  variables.

### VII. NEURAL NETWORK

The approach discussed here is known as Artificial Neural Network(ANN). ANN is a model in machine learning which consists of a large number of artificial neurons connected to each other. The structure of neural network resembles axons in the human brain. The motivation behind choosing this type of architecture is to build an intelligent model with functionality similar to that of a human brain. The structure of a neural network comprises of the Input, Hidden, and the Output layers as shown in Fig. 5.

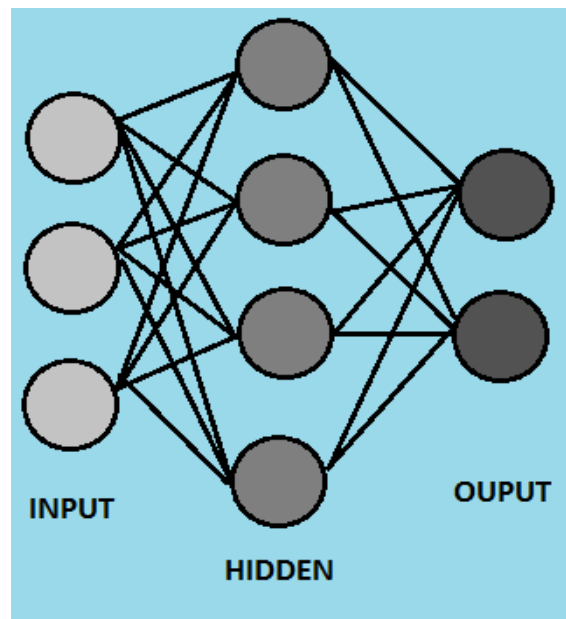


Fig.5 Structure Of Neural Network

**Multilayer Perceptron (MLP):** A MLP is a feed forward ANN which maps the input data to the corresponding output. It consists of a several layer of nodes, with each layer is connected to the following layer through a set of directed edges. Each neuron in the network is assigned an activation function which maps the weighted input to the output.

The MLP network is trained using the Backpropagation algorithm.

**Backpropagation Algorithm:** Below are the equation which explains the backpropagation algorithm.

**Phase 1: Compute the error in output layer  $\delta L$ :** The components of  $\delta L$  are given by  $\delta L_j = \partial C / \partial a L_j \sigma'(z L_j)$ . The term  $\partial C / \partial a L_j$  is the rate of change of the cost function with respect to the output activation function.

**Phase 2: Compute the error  $\delta l$  in terms of the error in the next layer,  $\delta l+1$ :**

$$\delta l = ((w_{l+1})^T \delta l+1) \odot \sigma'(z l)$$

Suppose we know the error  $\delta l+1$  at the  $l+1$ th layer. When we apply the transpose weight matrix,  $(w_{l+1})^T$ , we can think of this as moving the error *backwards* through the network, giving us a measure of the error at the output of the  $l$ th layer. We then take the Hadamard product  $\odot \sigma'(z l)$ . This moves the error backward through the activation function in layer  $l$ , giving us the error  $\delta l$  in the weighted input to layer  $l$ .

Combining phase 1 and phase 2 we can compute error  $\delta l$  for any layer.

### VIII. RESULT ANALYSIS Template Matching

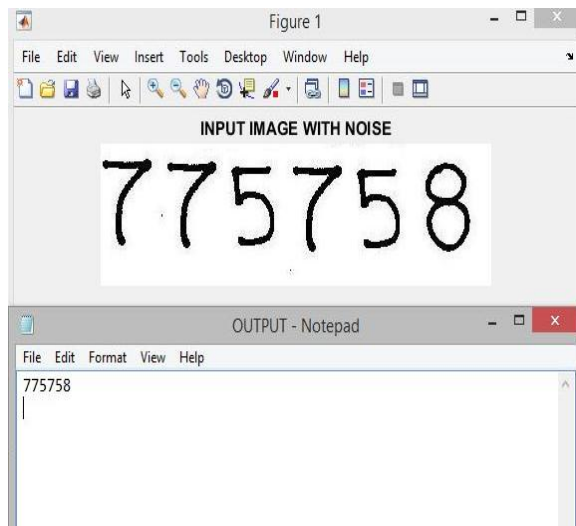


Fig. 6 Output From Template Matching Algorithm

### Support Vector Machine

The results from the SVM algorithm are the numbers contained in the input image. The accuracy obtained on the training data set was 76.7%.

Below are the ROC and Confusion Matrix for SVM.

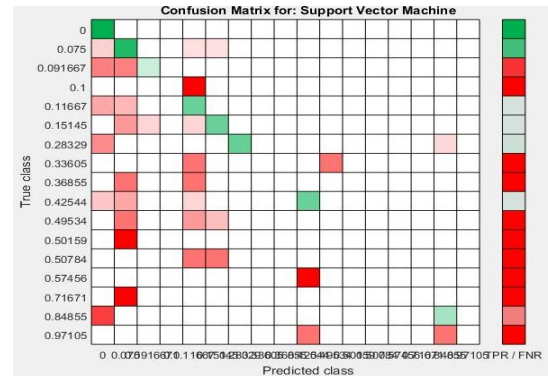


Fig. 7 Confusion Matrix

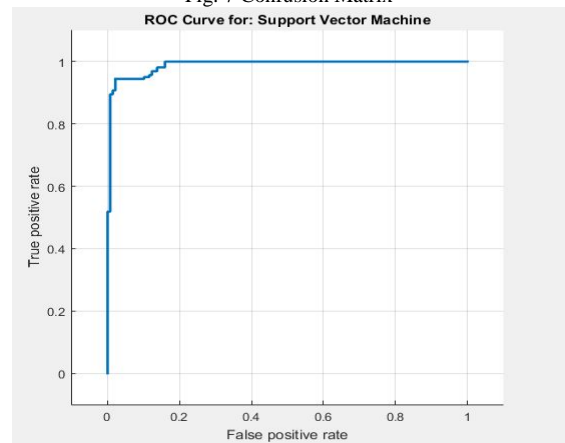


Fig. 8 Roc Curve

### Neural Network

The input is explained in the following GUI.

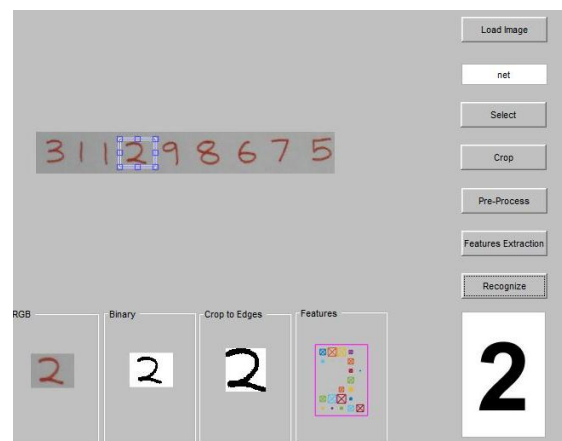


Fig. 9 Neural Network Gui

Below are the confusion matrix and training performance for the neural network.

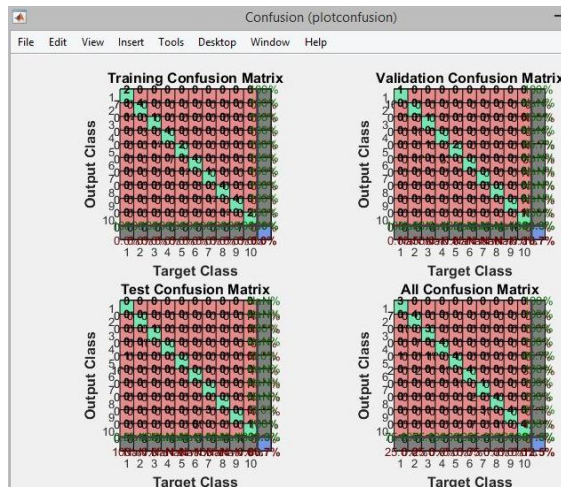


Fig. 10 Confusion Matrix For Neural Networks

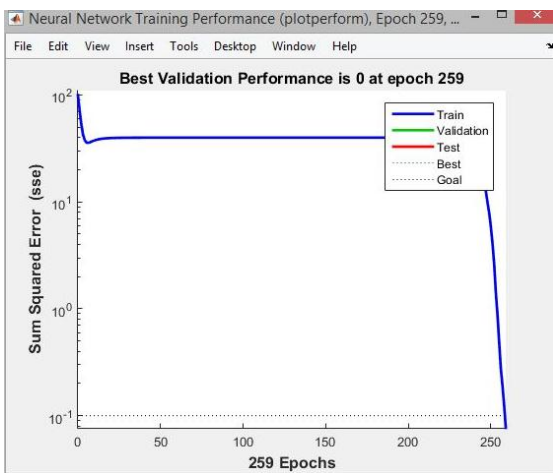


Fig. 11 Training Performance For Neural Networks

## IX. CONCLUSION

In this project we used Template Matching, Support Vector Machine, and Artificial Neural Network for digit recognition. It turned out that all the three methods were very promising but Neural Networks was very challenging to apply and yielded very good results, followed by SVM and Template Matching. Due to time constraints our project was restricted to digits, and for future work it will be interesting to investigate characters and more advanced applications could involve facial or handwriting recognition.

## X. ACKNOWLEDGEMENT

We are very thankful to our professors and specially to our guide Dr. Selvam Venkatesan for his significant help in completing this project.

## XI. REFERENCES

- [1] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012
- [2] Qiao Tan, Yuanlin Wen, Chenyue Meng. Learning of Visualization of Object Recognition Features and Image Reconstruction
- [3] Žiga Zadnik, Handwritten character Recognition: Training a Simple NN for classification using MATLAB)
- [4] J.Pradeep, E.Srinivasan, and S.Himavathi, Diagonal Based Feature Extraction For Handwritten Character Recognition System Using Neural Network
- [5] O. Matan, J. Bromley, C. J. Burges, J. S. Denker, L. D. Jackel, Y. LeCun, E. P. Pednault, W. D. Satterfield, C. E. Stenard, T. J. Thompson, Reading Handwritten Digits: A Zip Code Recognition System
- [6] C-L. Liu and K. Marukawa, "Normalization Ensemble for Handwritten Character Recognition", *The Ninth International Workshop on Frontiers in Handwriting Recognition (IWFHR 9)*, Tokyo, Japan, pp. 69-74, 2004
- [7] <https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es1999-461>
- [8] Vanderbrug, G.J, Rosenfeld, —Two-Stage Template Matching, *IEEE Transactions on Computers*, Vol. 60, Issue 11, 1977
- [9] P. Weinzaepfel, H. Jegou, and P. Perez. Reconstructing an image from its local descriptors. In *Computer Vision and Pattern Recognition, 2011*, pages 337 – 344. IEEE.
- [10] D. M. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Computer Vision and Pattern Recognition, 2013*.
- [11] M. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *arXiv:1311.2901*, 2013.
- [12] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Computer Vision*