Applications of MATLAB's Toolbox to Recognize Handwritten Characters Part 2: Experimental Results

[1]Pallavi Aggarwal, Yashasvi Rawal
Bharti Vidyapeeth College of Engineering, New Delhi, India

[1] iampallavi15@gmail.com, (+91)-7838907734

Abstract— Handwritten character recognition is a challenging task in the field of research on image processing, artificial intelligence as well as machine vision since the handwriting varies from person to person. Moreover, the handwriting styles, sizes and its orientation make it even more complex to interpret the text. The numerous applications of handwritten text in reading bank cheques, Zip Code recognition and in removing the problem of handling documents manually has made it necessary to acquire digitally formatted data. This paper presents the recognition of handwritten characters using either a scanned document, or direct acquisition of image using Matlab, followed by the implementation of various other Matlab toolboxes like Image Processing and Neural Network Toolbox to process the scanned or acquired image. Experimental Results are given to present the proposed model in order to recognize handwritten characters accurately.

Keywords— Image Acquisition, Image Rendering, Character Extraction, Image Processing, Edge Detection, Neural Network, Back Propagation Network, Multi Layer Perceptron Network

1. INTRODUCTION

With the advancement of technology, the interfacing between man and machine has increased the scope of research in various domains, thereby making majority of tasks automated and easier to perform. MATLAB is one such powerful machine tool where in the availability of Image Acquisition Toolbox, Image Processing Toolbox and Neural Network Toolbox simplifies the task of obtaining and understanding handwritten text.

[2] The two commonly used methods of handwritten character recognition, On-line and Off-line methods, have their own advantages and disadvantages. Where the off-line method provides more accuracy, the on-line method is superior in recognizing characters due to temporal information available to it. [10] Handwriting recognition principally entails optical character recognition (OCR). [9]The OCR systems process the image using several steps including segmentation, feature extraction, and classification. [1]It matches the images against stored bitmaps based on specific fonts. This poses a problem due to the inaccuracy in results of limited patterns available for drawing comparisons. Though, [3] it is possible that both hand written and printed characters may be recognized using OCR, the performance is directly dependent upon the quality of the input documents

The presented procedure uses MATLAB's neural network technology to overcome the challenges through analysis of hand strokes, irregularities in written characters and further matching it against multiple stored characters. [7]Other methods which use Euclidean metric distance as a form of measure have been employed earlier using neural network which have proved to be ineffective in later researches.

To recognize these characters, the first step is to acquire an image for processing. Next, it is required to use Image Processing toolbox to use various image properties in order to extract characters and later use Neural Network toolbox to train a suitable dataset. After training the network, testing is done and the performance curve is generated along with the individual required characters. The various steps for performing this task have been illustrated as follows:

- Acquiring an Image
- Image rendering
- Character extraction
- Training and testing

2. ACQUIRING AN IMAGE

To acquire an image, any one of the two methods can be followed. In the first one, image is scanned in order to make it machine editable. The image can be of any specific format like jpeg, bmp etc. A series of operations are performed after the image is taken as an input. The other method uses Image Acquisition toolbox to take the input image. [11]By opening the tool using Start → Toolboxes

→ More → Image Acquisition → Tool from Matlab window, one can directly set image acquisition parameter and preview the image acquired. This can later be exported as Image Data to Matlab for further operations.

3. IMAGE RENDERING

After the image is acquired, Image processing toolbox comes into play. Image is first converted into a greyscale image. The purpose of converting an RGB image into greyscale is due to the fact that it eliminates the hue and saturation information and retains the needed luminance.

The greyscale image is further processed into a binary image which replaces all pixels in the input image with luminance greater than level with the value 1 (white) and replaces all other pixels with the value 0 (black).

This follows the usage of edge detection to find the edges in the image. [13] It finds the places in the image where there is a rapid change in intensity, by following one of the two given definitions:

- Looks for the intensity change at the first derivative, having larger magnitude than some other threshold
- Looks for intensity change at second derivative having a zero crossing. It gives a number of derivative estimators, and using these estimators; operational sensitivity related to horizontal edges, vertical edges or even both can be possibly specified. If the edges are found, the binary image is returned with 1's else 0's. After detecting the edges, it is necessary to use morphology for dilating and filling the image which defines the region to fill by selecting the required points.
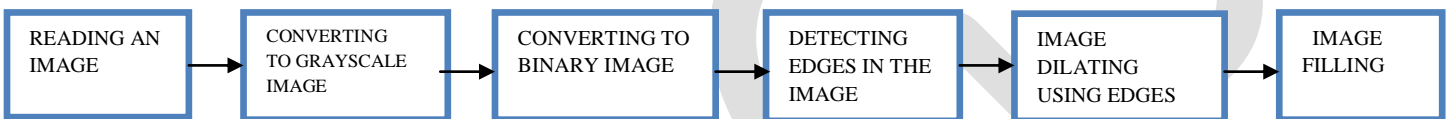
| READING AN IMAGE | → | CONVERTING TO GRAYSCALE IMAGE | → | CONVERTING TO BINARY IMAGE | → | DETECTING EDGES IN THE IMAGE | → | IMAGE DILATING USING EDGES | → | IMAGE FILLING |

Fig1: Block Diagram to represent Image Rendering

4. CHARACTER EXTRACTION

The image is then sent through connectivity test in order to check for the maximum connected components and the properties of each component, which is in the form of a box. After locating the box, the individual characters are then cropped into different sub images that is the raw data for the following feature extraction routine. The size of the sub-images is not fixed since they are exposed to noises which will affect the cropping process to vary from one to another. This will causing the input of the network become non- standard and hence, prohibit the data from feeding through the network. To avoid this, the sub-images have to be resized and then by finding the average value in each 10 by 10 blocks, the inputs for the network can be determined.

By this, extraction of the character is possible and could be passed to another stage for future classification and training purpose of the neural network.

5. TRAINING AND TESTING A NETWORK

Next, we create a training vector for the neural network in order to match the input accepted by the neural network function. The steps performed in creating and training the neural network has been illustrated below:

Type nntool in Matlab. A dialog box appears where in we are required to Import the Inputs and Targets from the MATLAB workspace. After importing, the created network appears in the network list. Open the network and select training tab. Here, we can choose the training parameters and data (inputs and targets) and finally click on Train option to train the network. I used feed forward back propagation neural network. In other words, [4] implementation based on Multi- Layer Perceptron Network (MLPN) trained with back propagation was done. [6] Other complex training methods employing Error Back Propagation Algorithm have been used earlier. There were two hidden layers used with TANSIG (tan-sigmoid) function. Further, experimental results in the next section illustrate the steps performed.
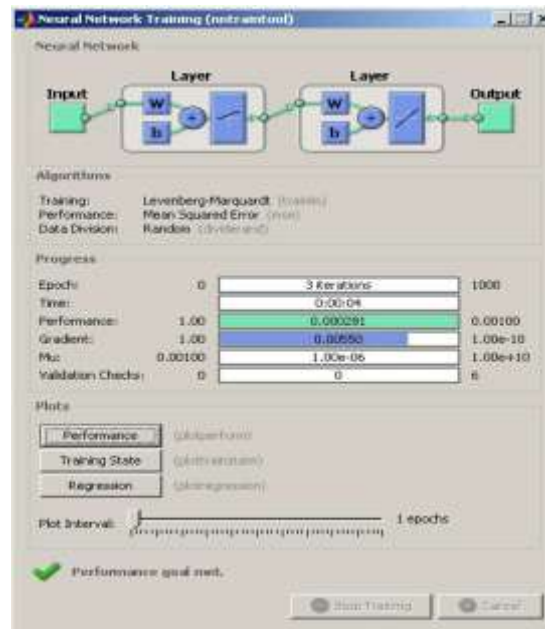
Fig2: Neural network training tool [1]

5.1 ABOUT NEURAL NETWORK

As the name suggests, Neural is related to neurons, which are an important part of biological nervous system. Like [5] human nervous system processes the information it receives from nerves, in a similar way, this Artificial Network processes information to solve specific problems. Every neural network comprises of interconnected neurons which is trained or configured for a specific application. This is used in various field of study like pattern recognition, data classification and so forth to analysis a problem and adjust its parameters accordingly.

The need of neural network can be realized by comparing it with the use of conventional computers which requires an algorithm to solve a specific problem. Unlike computers, neural networks follow parallel processing architecture thereby resulting in maximum efficiency. Moreover, there are multiple network types like Perceptron, feed forward, feedback networks which present a variable ways to associate input with the output.

Neural network is not just confined to MATLAB but also suitable for real time systems. It also contributes to research in medicine such as neurology to study brain mechanism in detail. The scope of neural network is not just limited to be used alone. [8] It can be used in solving Zip Code Recognition problem. It can be integrated with other important related subjects like Fuzzy logic and Artificial Intelligence for faster response and computations.
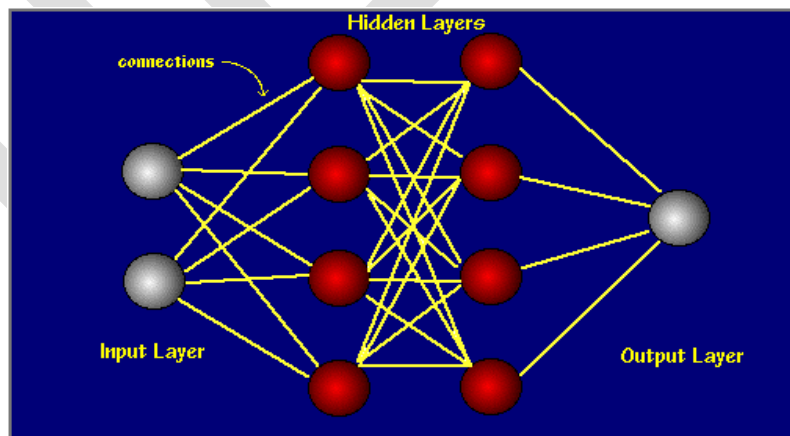

Fig3: Neural Network Architecture [12]
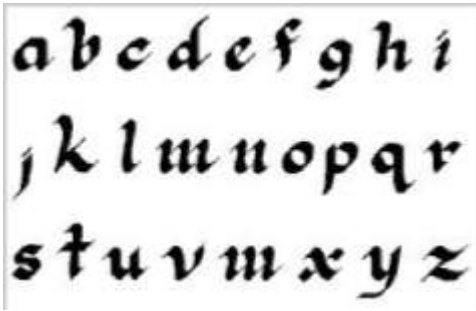
PART 2: EXPERIMENTAL RESULTS
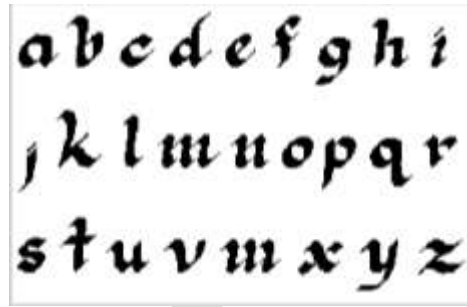

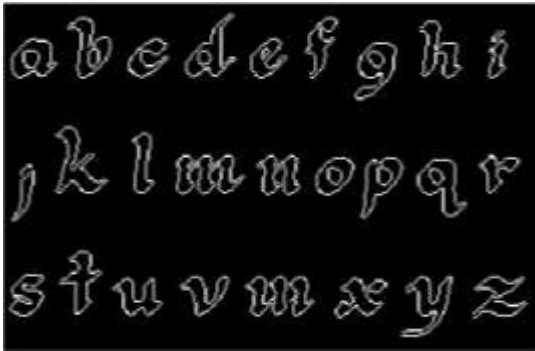**Fig 4: Reading an image**

Fig 5: Binary image

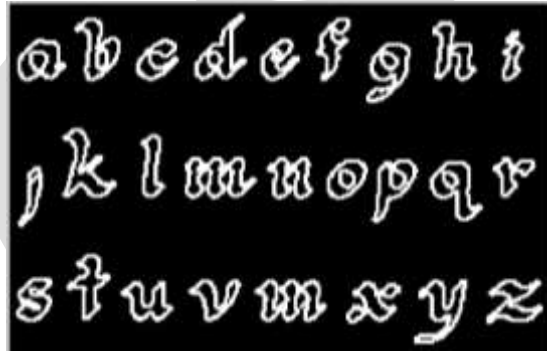**Fig 6: Character Location using Edge Detection**

Fig 7: Image Dilation

Fig 8: Image Filling
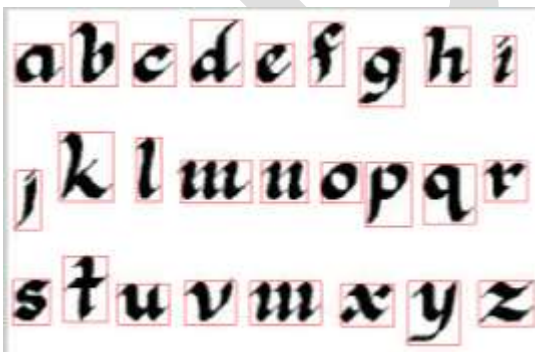

Fig 9: Character Extraction on binary image

Fig 10: Character extraction on input image

Fig 11: Extracted Characters' Blocks

ACKNOWLEDGMENT

CONCLUSION

As presented in the paper, handwritten character recognition is categorized into three main divisions: Image Acquisition, Image processing and neural network training of the dataset. The experimental results illustrate how an input image leads us to character extraction and further usage of neural network recognizes the handwritten patterns, accordingly.

The reason for choosing artificial neural networks as a part of the research to perform character recognition is due to their high noise tolerance. The designed systems have the ability to yield accurate results, provided the correct dataset is available at the time of training the network. The current stage of research depicts that the software does perform well both in terms of speed or accuracy. But the character location is not efficient since the size of every block varies. This can be taken care of by initializing the weights during training of dataset. There is a scope of improvement the current system. Hence, a simple yet effective approach for recognition of handwritten characters using artificial neural networks has been described.

REFERENCES:

[1] Žiga Zadnik " Handwritten character Recognition: Training a Simple NN for classification using MATLAB"

[2] Kauleshwar Prasad, Devvrat C. Nigam, Ashima Lokhtiya and Dheeren Umre"Character Recognition Using Matlab's Neural Network Toolbox "International Journal of u- and e- Service, Science and Technology Vol. 6, No. 1, February, 2013

[3] Sandeep Tiwari, Shivangi Mishra, Priyank Bhatia, Praveen Km. Yadav , "Optical Character Recognition using MATLAB" ,International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE) Volume 2, Issue 5, May 2013

[4] Mathias Wellner, Jessica Luan, Caleb Sylvester, "Recognition of Handwritten digits using a Neural Network", 2002

[5] http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html

[6] Vijay Patil and Sanjay Shimpi, "Handwritten English character recognition using neural network" , Elixir Comp. Sci. & Engg. 41 (2011)

[7] Sumit Saha,Tanmoy Som," Handwritten character recognition by using Neural-network and Euclidean distance metric " IJCSIC - International Journal of Computer Science and Intelligent Computing Vol. 2, No. 1, November 2010

[8] O.S. matan , R.K. Kiang,C.E. Stenard," Handwritten Character Recognition Using Neural Network Architectures", 4th USPS technology Conference, November 1990

[9] A Matlab project in OCR by Jesse Hansen : www.ele.uri.edu/~hansenj/projects/ele585/OCR/OCR.pdf

[10] http://en.wikipedia.org/wiki/Handwriting_recognition

[11] http://www.mathworks.in/products/imaq/description3.html

[12] http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html

[13] http://www.mathworks.in/help/images/detect-edges-in-images.html#f11-12512