Character Recognition Using Matlab's Neural Network Toolbox

Kauleshwar Prasad, Devvrat C. Nigam, Ashmika Lakhotiya and Dheeren Umre

B.I.T Durg, India
Kauleshwarprasad2gmail.com, devnigam24@gmail.com,ashmika22@gmail.com, ud2k81@gmail.com

Abstract

Recognition of Handwritten text has been one of the active and challenging areas of research in the field of image processing and pattern recognition. It has numerous applications which include, reading aid for blind, bank cheques and conversion of any hand written document into structural text form. In this paper we focus on recognition of English alphabet in a given scanned text document with the help of Neural Networks. Using Mat labNeural Network toolbox, we tried to recognize handwritten characters by projecting them on different sized grids. The first step is image acquisition which acquires the scanned image followed by noise filtering, smoothing and normalization of scanned image, rendering image suitable for segmentation where image is decomposed into sub images. Feature Extraction improves recognition rate and misclassification. We use character extraction and edge detection algorithm for training the neural network to classify and recognize the handwritten characters.

Keywords: *Character Recognition, Neural Network, Character Extraction algorithm, Edge Detection algorithm, Image acquisition*

1. Introduction

Handwriting recognition has been one of the most fascinating and challenging research areas in field of image processing and pattern recognition in the recent years [1]. It contributes immensely to the advancement of automation process and improves the interface between man and machine in numerous applications. In general, handwriting recognition is classified into two types as off-line and on-line handwriting recognition methods. The on-line methods have been shown to be superior to their off-line counter parts in recognizing handwritten characters due to the temporal information available with the former. However, in the off-line systems, comparably high recognition accuracy levels [5].

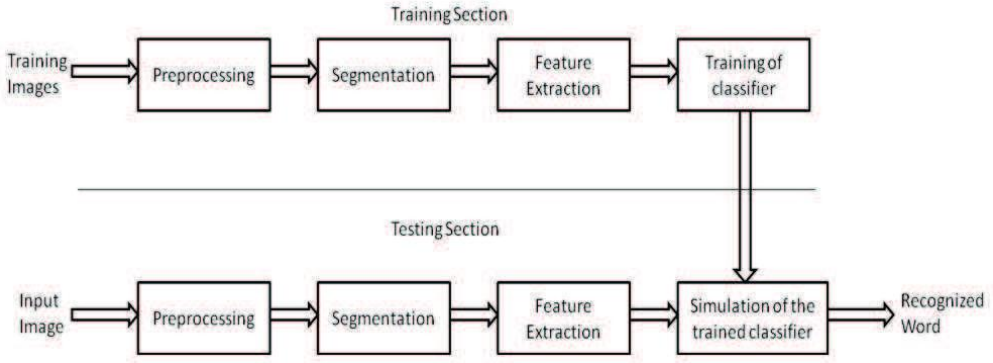

Figure 1. Generic Character Recognition Model

The recognition of characters from scanned images of documents has been a problem that has received much attention in the fields of image processing, pattern recognition and artificial intelligence. Classical methods in pattern recognition do not as such suffice for the recognition of visual characters due to the following reasons:

1. The 'same' characters differ in sizes, shapes and styles from person to person and even from time to time with the same person.
2. Like any image, visual characters are subject to spoilage due to noise.
3. There are no hard-and-fast rules that define the appearance of a visual character. Hence rules need to be heuristically deduced from samples. As such, the human system of vision is excellent in the sense of the following qualities**:**

   I. The human brain is adaptive to minor changes and errors in visual patterns. Thus we are able to read the handwritings of many people despite different styles of writing.

   II. The human vision system learns from experience, Hence we are able to grasp newer styles and scripts with amazingly high speed.

   III. The human vision system is immune to most variations of size, aspect ratio, colour, location and orientation of visual characters.

In contrast to limitations of classical computing, Artificial Neural Networks (ANNs), that were first developed in the mid 1900's serve for the emulation of human thinking in computation to a meagre, yet appreciable extent.

2. The Proposed Model

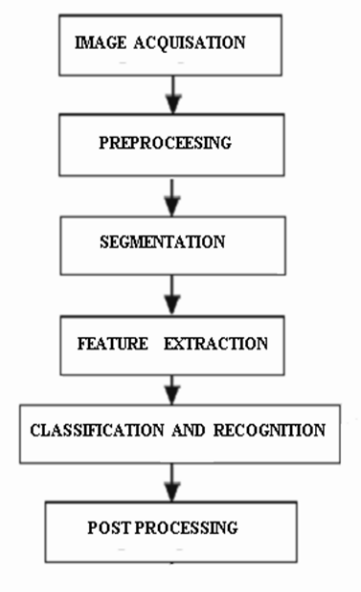

Figure 2. General Offline Character Recognition System

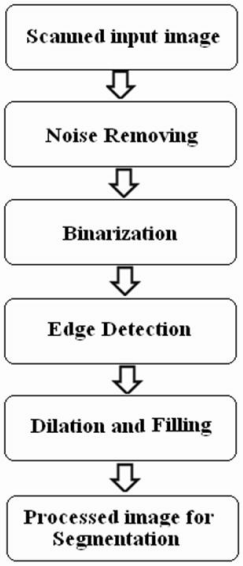

Figure 3. Pre- processing of Handwritten Character of Image

2.1 Image Acquisition

In Image acquisition, the recognition system acquires a scanned image as an input image. The image should have a specific format such as JPEG, BMT, *etc.* This image is acquired through a scanner, digital camera or any other suitable digital input device [6, 7].

2.2 Pre-Processing

The pre-processing is a series of operations performed on scanned input image. It essentially enhances the image rendering it suitable for segmentation. The role of pre-processing is to segment the interesting pattern from the background. Generally, noise filtering, smoothing and normalization should be done in this step. The pre-processing also defines a compact representation of the pattern. Binarization process converts a gray scale image into a binary image. Dilation of edges in the binarized image is done using sobel technique [10].

2.3 Segmentation

In the segmentation stage, an image of sequence of characters is decomposed into sub-images of individual character [9]. In the proposed system, the pre-processed input image is segmented into isolated characters by assigning a number to each character using a labelling process. This labelling provides information about number of characters in the image. Each individual character is uniformly resized into pixels.

2.4 Feature Extraction

In this stage, the features of the characters that are crucial for classifying them at recognition stage are extracted. This is an important stage as its effective functioning improves the recognition rate and reduces the misclassification. Diagonal feature extraction scheme for recognizing off-line handwritten characters is proposed in this work [8, 9]. Every character image is divided into equal zones, each of size 10x10 pixels. The features are extracted from each zone pixels by moving along the diagonals of its respective 10x10 pixels.

2.5 Classification and Recognition

The classification stage is the decision making part of the recognition system [3]. A feed forward back propagation neural network is used in this work for classifying and recognizing the handwritten characters. The pixels derived from the resized character in the segmentation stage form the input to the classifier. The neural classifier consists of two hidden layers besides an input layer and an output layer. The total number of neurons in the output layer is 26 as the proposed system is designed to recognize English alphabets [6].

2.6 Post- processing

Post-processing stage is the final stage of the proposed recognition system. It prints the corresponding recognized characters in the structured text form by calculating equivalent ASCII value using recognition index of the test samples.

3. Design And Implementation

Initially we are making the Algorithm of Character Extraction. We are using MATLAB as tool for implementing the algorithm. Then we design neural network, we need to have a Neural Network that would give the optimum results [11]. There is no specific way of finding the correct model of Neural Network. It could only be found by trial and error method. Take different models of Neural Net- work, train it and note the output accuracy. There are basically two main phases in our Paper: Pre-processing and Character Recognition. In first phase we have are preprocessing the given scanned document for separating the Characters from it and normalizing each characters. Initially we specify an input image file, which is opened for reading and preprocessing. The image would be in RGB format (usually) so we convert it into binary format. To do this, it converts the input image to grayscale format (if it is not already an intensity image), and then uses threshold to convert this grayscale image to binary *i.e.* all the pixels above certain threshold as 1 and below it as 0. Firstly we needed a method to extract a given character from the document. For this purpose we modified the graphics 8-way connected algorithm (which we call as Edge Detection).

4. Algorithm Used

4.1 Character Extraction Algorithm

1. Create a Traverse List: - List of pixels which have been already traversed. This listis initially empty.
2. Scan row Pixel-by-Pixel.
3. Whenever we get a black pixel check whether the pixel is already in the traverse list, if it is simply ignore and move on else apply Edge detection Algorithm.
4. Add the List of Pixels returned by EdgedetectionAlgorithm to Traverse List.
5. Continue the steps 2 - 5 for all rows.

4.2 Edge Detection Algorithm

The Edge Detection Algorithm has a list called traverse list. It is the list of pixel already traversed by the algorithm.

EdgeDetection(x,y,T_L);

1) Add the current pixel to T_L. The current position of pixel is (x,y).

2) NewT_L= T_L + current position (x,y).

If pixel at (x-1, y-1) then
Check if it is not in T_L.
Edgedetection(x-1, y-1, NewT_L);
Endif

If pixel at (x-1, y) then
Check if it is not in T_L.
Edgedetection(x-1, y,NewT_L);
Endif

If pixel at (x-1, y) then
Check if it is not in T_L.
Edgedetection(x-1, y+1, NewT_L);
Endif

If pixel at (x,y-1) then
Check if it is not in T_L.
Edgedetection(x,y-1,NewT_L);
Endif

If pixel at (x,y+1) then
Check if it is not in T_L.
Edgedetection(x,y+1,NewT_L);
Endif

If pixel at (x+1,y-1) then
Check if it is not in T_L.
Edgedetection(x+1,y-1,NewT_L);
Endif

Edgedetection(x+1, y, NewT_L);
Endif

If pixel at (x+1,y+1) then
Check if it is not in TraverseList.
Edgedetection(x+1, y+1, NewT_L);
Endif.

3) Return;

The Edge Detection algorithm terminates when it has covered all the pixels of the character as every pixel's position would be in Traverse List so any further call to Edge Detection is prevented.

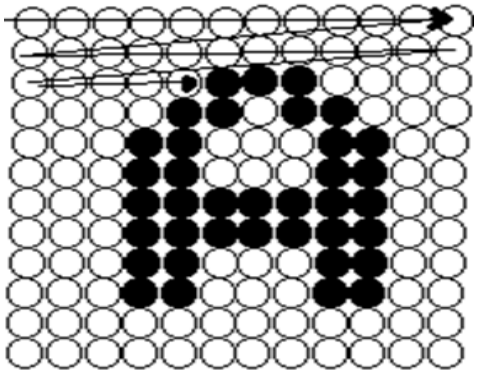

Figure 4. (a) Traversal of Each Scan Line

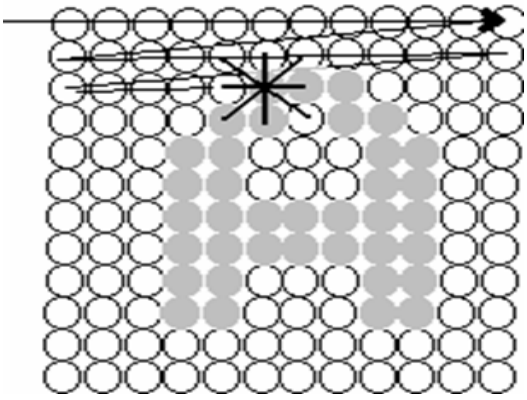

Figure 4. (b) Shows the Respective Calls Made to the all 8-neighbouring Pixels

The Edge detection is called when we hit a pixel (i.e. encounter a pixel with value 1). As per the algorithm the current position is entered in Traverse List and recursive calls are made to all 8 neighbouring pixels. Before the calls are made it is ensured that the corresponding neighbouring pixels is having value 1 and is not already encountered before *i.e.* it should not be in the Traverse List.

5. Conclusion

This paper carries out a study of various feature based classification techniques for offline handwritten character recognition. After experimentation, it proposes an optimal character recognition technique. Artificial neural networks are commonly used to perform character recognition due to their high noise tolerance. The systems have the ability to yield excellent results. The feature extraction step of optical character recognition is the most important. A poorly chosen set of features will yield poor classification rates by any neural network. At the current stage of development, the software does perform well either in terms of speed or accuracy but not better. It is unlikely to replace existing OCR methods, especially for English text. A simplistic approach for recognition of Optical characters using artificial neural networks has been described.

References

[1] S. Mori, C. Y. Suen and K. Kamamoto, "Historical review of OCR research and development", Proc. of IEEE, vol. 80, **(1992)** July, pp. 1029-1058.
[2] N. Arica and F. Yarman-Vural, "An Overview of Character Recognition Focused on Off-line Handwriting", IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 31, no. 2, **(2001)**, pp. 216 - 233.
[3] A. Rajavelu, M. T. Musavi and M. V. Shirvaikar, "ANeural Network Approach to Character Recognition", Neural Networks, vol. 2, **(1989)**, pp. 387-393.
[4] R. Plamondon and S. N. Srihari, "On-line and off- line handwritten character recognition: A comprehensive survey", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 1, **(2000)**, pp. 63-84.
[5] U. Bhattacharya and B. B. Chaudhuri, "Handwritten numeral databases of Indian scripts and multistage recognition of mixed numerals", IEEE Transaction on Pattern analysis and machine intelligence, vol. 31, no. 3, **(2009)**, pp. 444-457.
[6] M. Hanmandlu, K. R. M. Mohan and H. Kumar, "Neural-based Handwritten character recognition", in Proceedings of Fifth IEEE International Conference on Document Analysis and Recognition, ICDAR'99, Bangalore, India, **(1999)**, pp. 241-244.
[7] T. V. Ashwin and P. S. Sastry, "A font and size-independent OCR system for printed Kannada documents using support vector machines", in Sadhana, vol. 27, Part 1, **(2002)** February, pp. 35–58.
[8] S. V. Rajashekararadhya and P. Vanajaranjan, "Efficient zone based feature extraction algorithm for handwritten numeral recognition of four popular south-Indian scripts", Journal of Theoretical and Applied Information Technology, JATIT, vol. 4, no. 12, **(2008)**, pp. 1171-1181.
[9] R. G. Casey and E. Lecolinet, "A Survey of Methods and Strategies in Character Segmentation", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 7, **(1996)** July, pp. 690-706.
[10] R. C. Gonzalez, R. E. woods and S. L. Eddins, "Digital Image Processing using MATLAB", Pearson Education, Dorling Kindersley, South Asia, **(2004)**.
[11] T. V. Ashwin and P. S. Sastry, "A font and size-independent OCR system for printed Kannada documents using support vectormachines", in Sadhana, vol. 27, Part 1, **(2002)** February, pp. 35–58.
[12] M. Blumenstein and B. Verma, "Neural-based solutions for the segmentation and recognition of difficult handwritten words from a benchmark database", in Proceedings of the Fifth International Conference on Document Analysis and Recognition, ICDAR '99, **(1999)** September, pp. 281 –284.
[13] Y. Tay, M. Khalid, R. Yusof and C. Viard-Gaudin, "Offline cursive handwriting recognition system based on hybrid markov model and neural networks", in Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation 2003, vol. 3, **(2003)** July, pp. 1190 – 1195.
[14] G. Kim, V. Govindaraju and S. Srihari, "A segmentation and recognition strategy for handwritten phrases", in Proceedings of the 13th Interna-tional Conference on Pattern Recognition 1996, vol. 4, **(1996)** August, pp. 510–514.
[15] Y. Y. Chung and M. T. Wong, "Handwritten character recognition by fourier descriptors and neural network", in Proceedings of IEEE Region 10 Annual Conference on Speech and Image Technologies for Computing and Telecommunications, TENCON '97, vol. 1, **(1997)** December, pp. 391 –394.