ISSN:0975-9646

Amrita Hirwani et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (1) , 2014, 558-661

Character Recognition Using Multilayer Perceptron

Amrita Hirwani[1], Sandeep Gonnade[2]

[1]*MATS School of Engg. & IT, MATS University,*
Arang, Raipur (C.G.), India

[2]*Asst. Prof., Dept. of Computer Science,*
MATS School of Engg. & IT, MATS University,
Arang,Raipur (C.G.), India

***Abstract*- Handwritten Character recognition is an interesting area of pattern recognition. It has become the popular field of research during the last few decades. Neural network is playing an important role in handwritten character recognition. Boundary tracing along with Fourier Descriptor is used for feature extraction from the handwritten character. By analyzing the shape of handwritten character, the character is identified and by comparing its features that distinguishes each character. In this paper, we use multilayer perceptrons for the recognition of handwritten English characters. The higher level of accuracy is achieved with minimum training and classification time.**

***Keywords*- Handwritten Character Recognition, Back-propagation network, Feature Extraction, Skeletonization, Boundary Tracing, Fourier Descriptor and Multilayer Perceptron Network.**

I. INTRODUCTION

Neural network is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. Neural network is configured for specific applications like pattern recognition or data classification through a learning process. Many neural networks now being designed are statistically quite accurate. The multilayer perceptrons are used for character recognition provides excellent level of accuracy and it will take minimum time for training the network.

Handwriting recognition is categorized into two types as off-line and on-line handwriting recognition methods. Off-line handwriting recognition involves the automatic conversion of text in an image into letter codes which are usable within computer and text-processing applications. The data obtained by this form is regarded as a static representation of handwriting. On-line handwriting recognition involves the automatic conversion of text as it is written on a special digitizer or PDA. In these devices sensor picks up the pen-tip movements as well as pen-up and/or pen-down switching. The obtained signal is converted into letter codes which are usable within computer and text-processing applications.

II. CHARACTER MODELING

A. Input Characters

The English characters are taken as input.

B. Scanning

Handwritten English characters are scanned. They are resized into 1024 (32x32) binary pixels.

C. Skeletonization

The skeletonization process will be used to binary pixel image and the extra pixels which are not belonging to the backbone of the character has been deleted and the broad strokes has been reduced to thin lines [2].

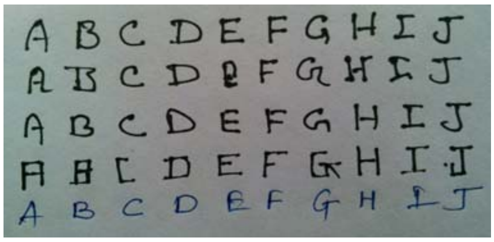

Fig.1: Handwritten English Characters

Skeletonization means transforming a component of a digital image into a subset of the original component. The two categories of skeletonization methods are: first category is based on distance transforms, and a specified subset of the transformed image is a distance skeleton. The original component can be reconstructed from the distance skeleton. Another category is defined by thinning approaches; and the result of skeletonization using thinning algorithms should be a connected set of digital curves or arcs. Motivations for is skeletonization algorithms are the need to compute a reduced amount of data or to simplify the shape of an object in order to find features for recognition algorithms and classifications [1, 3].

D. Normalization

Handwritten characters of different persons are different. In normalization of characters all characters could become in equal dimensions of matrix. We normalize the characters into 20x20 pixel size.

III. RECOGNITION SYSTEM

Steps involved in handwritten character recognition process -

- Acquiring input character by scanning
- Skeletonization and normalization operation
- Feature extraction by boundary detection
- Neural network classification
- Recognition of character

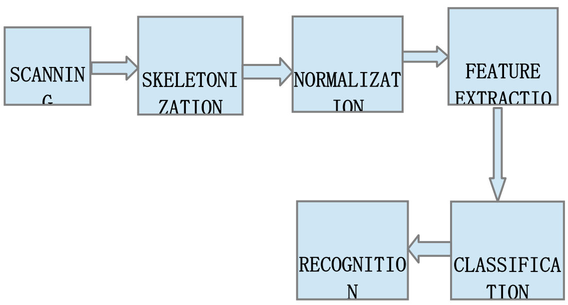

Fig.2: Block diagram of Character Recognition System

IV. Feature Extraction

The eight-neighbor adjacent method is used for extracting the information of the boundary of a handwritten character. Boundary of the character is detected by scanning the binary image. The searching follows according to the clockwise direction. Two pixels are connected if they are neighbors and their gray levels satisfy a specified criterion of similarity. The set of all foreground pixels connected to p is called connected component containing p [1]. The 8-neighbors of pixel p are shown in Figure 3.

A pixel p at coordinates (x, y) has four horizontal and vertical neighbors whose coordinates are given by

$$(x+1, y), (x-1, y), (x, y+1), (x, y-1)$$

The pixel p have four diagonal neighbors coordinates -

$$(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)$$

P1	P2	P3
P8	P	P4
P7	P6	P5

Fig. 3: Pixel P with 8-neighbors

A. Boundary Tracing

When a white pixel is detected, it checks another new white pixel and that pixel again checks for another white pixel and the process of finding white pixels continue. The tracing follows the boundary automatically [1].

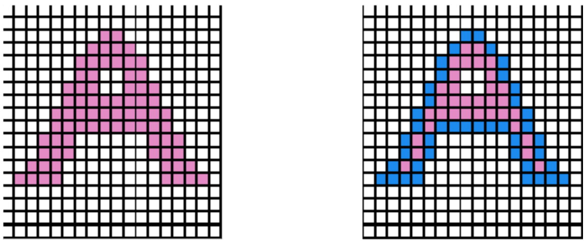

Fig 4: Images showing boundary

1) Inner Boundary Tracing Algorithm

Step I. Search the image form top left until a pixel of a new region is found; this pixel P1 then has the minimum column value of all pixels of that region having the minimum row value. Pixel P1 is a starting pixel of the region border. Take a variable "d" to store the previous move along the border from the previous border element to the current border element and assign -

a. d = 0 if the border is detected in 4-connectivity (Fig 5 a)

b. d = 7 if the border is detected in 8-connectivity (Fig 5 b)

Step II. Search the 3x3 neighborhood of the current pixel (in an anti-clockwise direction), beginning the neighborhood search in the pixel positioned in the direction

a. (d + 3) mod 4 (Fig 5 c)

b. (d + 7) mod 8 if d is even (Fig 5 d)

(d + 6) mod 8 if d is odd (Fig 5 e)

The first pixel found with the same value as the current pixel is a new boundary element Pn. Update the d value.

Step III. If the current boundary element P1 is equal to the second boundary element P2, and if the previous border element Pn-1 is equal to P1, stop. Otherwise repeat *Step II*.

Step IV. The detected inner border is represented by pixels P1...Pn-2.

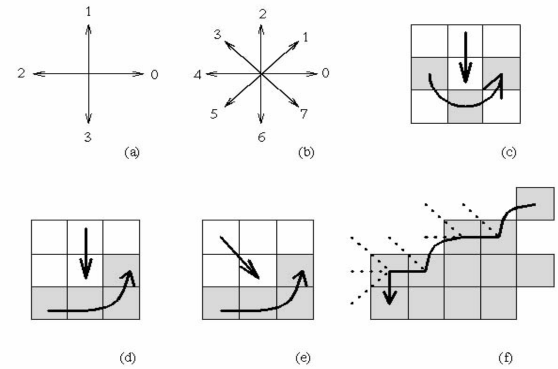

Fig 5: Inner boundary tracing: (a) Direction notation, 4-connectivity, (b) 8-connectivity, (c) pixel neighborhood search sequence in 4-connectivity, (d),(e) search sequence in 8-connectivity, (f) boundary tracing in 8-connectivity (dashed lines show pixels tested during the border tracing)

2) Outer Boundary Tracing Algorithm

Step I. Trace the inner region boundary in 4-connectivity until done.

Step II. All non-region pixels form outer boundary that were tested during the search process; if some pixels were tested more than once, they are listed more than once in the outer boundary list

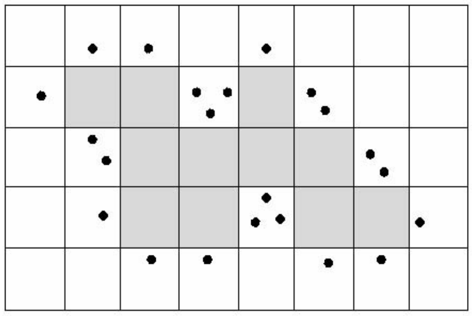

Fig 6: Outer boundary tracing: dot denotes outer border elements

B. Fourier Descriptors

Fourier Descriptors are involved in finding the Discrete Fourier coefficients a[k] and b[k] for $0 \leq k \leq L - 1$,

$$a[k] = 1/L \, \Sigma \, x[m]e^{-jk(2\pi/L)m} \quad (1)$$

$$b[k] = 1/L \, \Sigma \, y[m]e^{-jk(2\pi/L)m} \quad (2)$$

From equations (1) and (2) Fourier coefficients derived and are not rotational or shift invariant but Fourier Descriptors that have the invariant property with respect to rotation and shift the following operations are defined. Compute a set of invariant descriptors r(n) for each n as follows

$$r(n) = [|a(n)|^2+|b(n)|^2]^{1/2} \quad (3)$$

Computing a new set of descriptors s(n) by eliminating the size of character from r(n)

$$s(n) = r(n)/r(1) \quad (4)$$

a(n), b(n) and invariant descriptors s(n), n = 1,2,.....(L 1) were derived for all of the characters [1].

V. RECOGNITION

Recognition of handwritten characters is a very difficult task because different people have different writing style. Characters written by different people could be different in various manners such as size, orientation, thickness, and dimension. This yields infinite variations. Neural network is used for recognizing handwritten characters. In this paper, for English handwritten character recognition Feed Forward Multi-Layer Perceptron neural network (MLPN) has been used and back-propagation algorithm has been used for training.

A. Multilayer Perceptron Network

The multilayer perceptron neural networks have been applied successfully to solve some difficult and diverse problems by training them in supervised manner with a highly popular algorithm known as error back-propagation algorithm. The error back-propagation algorithm is based on error-correction learning rule. In this paper, two-layer perceptron network is used. In two-layer perceptron there are one hidden layer and one output layer presents. Fig.7. shows multi-layer perceptron network.

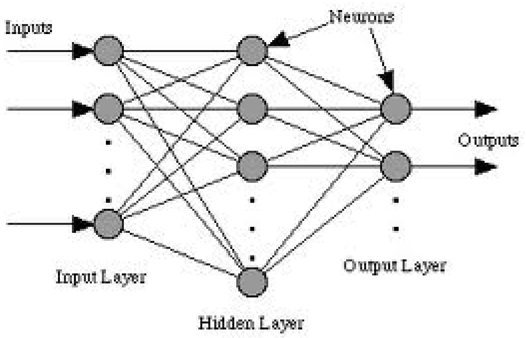

Fig.7: Multi-Layer Perceptron Network (MLPN)

Error back-propagation learning consists of two passes through the different layers of the network and they are: a forward pass and a backward pass. In forward pass, an activity pattern (input vector) is applied to the sensory nodes of the network, and its effect propagates through the network layer by layer. Finally, a set of outputs is produced as the actual response of the network. During the forward pass, the synaptic weights are fixed. During the backward pass, the synaptic weights are adjusted in accordance with an error-correction rule. The actual response of the network is subtracted from the desired (target) response to produce an error signal. This error signal is then propagated backward through the network against the direction of synaptic connections.

1) The MLP Learning Algorithm

Step I. Initialize the network, with all weights set to random numbers between -1 and +1.

Step II. Present the first training pattern, and obtain the output.

Step III. Compare the network output with the target output.

Step IV. Propagate the error backwards.

(a) Correct the weights of output layer using the following formula-

$$w_{ho} = w_{ho} + (\eta\delta_o o_h)$$

Where w_{ho} = weight connecting hidden unit h with output unit o, η = learning rate, o_h = output at hidden unit h.
δ_o is given as

$$\delta_o = o_o(1 - o_o)(1 - o_o)$$

Where o_o = output at node o of the output layer, and t-o = target output for node o.

(b) Correct the input weights using the following formula.

$$w_{ih} = w_{ih} + (\eta\delta_h o_i)$$

Where w_{ih} = weight connecting node i of the input layer with node h of the hidden layer, o_i = input at node i of the input layer, η = learning rate. δ_h is calculated as follows

$$\delta_h = o_h(1 - o_h)\sum(\delta W_{ho})$$

Step V. Take the average difference between the target and the output vector to calculate the error. The following function could be used-

$$E = \frac{\sqrt{\sum_{n=1}^{p}(t_0 - o_0)^2}}{p}$$

Where p is the number of units in the output layer.

Step VI. Repeat from Step II for each pattern in the training set to complete one epoch.

StepVII. Shuffle the training set randomly.

Step VIII. Repeat from Step II for a set number of epochs, or until the error ceases to change.

In MLPN with Back-propagation training algorithm, the procedure and calculations given as

$$f_j(x) = 1/(1 + e^{-net}) \text{ and } net = \Sigma w_{ij} o_i$$

Where o_i = output of unit i, w_{ij} = weight from unit i to unit j. The cost function can be minimized by updating the weights of the neural network using the generalized delta algorithm:

$$E = 1/2(\Sigma(D_{pk} - O_{pk}))^2$$

Where D_{pk} = desired value of the output unit k and O_{pk} = actual value of the output unit k and training pair p. By updating the weights convergence is achieved. For this the formula can be given as follows:

$$W_{ij}(n + 1) = W_{ij}(n) + \Delta W_{ij}(n) \quad (1)$$

$$\Delta W_{ij}(n) = \eta\delta X_j + \alpha(W_{ij}(N) - W_{ij}(n - 1)) \quad (2)$$

Where η is the learning rate, α is the momentum, $W_{ij}(n)$ is the weight from hidden node i or from an input to node j at n^{th} iteration, X_i is either the output of unit i or is an input, and δ_j is an error term for unit j [1]. If unit j is an output unit, then error term can be given as

$$\delta_j = O_j(1 - O_j)(D_j - O_j)$$

If unit j is an internal hidden unit, then error can be given as
$\delta j = O_j(1 - O_j)\, \Sigma\delta_k W_{kj}$

VI. RESULT AND CONCLUSION

Different data sets were used for training and testing. Table-I shows the experimental result. Four hundred samples were collected from 10 persons, 40 samples each. 200 samples were used for training (training data) and 200 samples were usedfor testing the data. In this paper, a handwritten English character recognition system has been developed. An experimental result shows that Fourier descriptors with back-propagation network yields higher accuracy.

TABLE I
RESULT OF HANDWRITTEN ENGLISH CHARACTER USING MLPN

No. of Hidd en nodes (neur ons)	*Learn ing Rate*	*Momentum Factor*	*No. of Epochs*	*Recognition % Training Set Test Set*	
12	0.2	0.8	40	80	88
24	0.2	0.8	80	80	93
36	0.2	0.8	160	80	95

REFERENCES

[1] Anita Pal, Dayashankar Singh, "Handwritten English Character Recognition Using Neural Network" IJCSC, vol.1 no.2, jul-dec-2010, pp141-144

[2] Verma B.K, "Handwritten Hindi Character Recognition Using Multilayer Perceptron and Radial Basis Function Neural Network", IEEE International Conference on Neural Network, 4, pp. 2111 2115, 1995.

[3] Gilsela Klette "Skeletons in Digital Image Processing", Computer Science Department of the University of Auckland CITR at Tamaki Campus, CITR-TR-112, july 2002.

[4] J. Sutha, N. Ramraj, "Neural Network Based Offline Tamil Handwritten Character Recognition System", IEEE International Conference on Computational Intelligence and Multimedia Application, 2007, 2, 13-15, Dec.2007, Page(s): 446-450, 2007.

[5] Dayashankar Singh, Maitreyee Dutta and Sarvpal H. Singh, "Neural Network Based Handwritten Hindi Character Recognition", ACM International Conference (Compute 09), Jan. 9-10, 2009, Bangalore.

[6] Renato Kresch, and David Malah, "Skeleton-Based Morphological Coding of Binary Images" IEEE Transactions on image processing, vol. 7, no. 10, october 1998 1387.

[7] F. Kimura, T.Wakabayashi, S.Tsuruoka, and Y.Miyake,"Improvement of Handwritten Japanese Character Recognition using Weighted Direction Code Histogram," Pattern Recognition, 30, No.8, pp. 1329 1337, 1997.

[8] N.Kato, M.Suzuki, and S.Omachi, "A Handwritten Character Recognition System Using Directional Element Feature and Asymmetric Mahalanobis Distance", IEEE Trans. on PatternAnalysis and Machine Intelligence, 21, No.3, pp. 258-262, 1999.

[9] Sumedha B. Hallale, Geeta D. Salunke, "Offline Handwritten DigitRecognition Using Neural Network", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 2, Issue 9, September 2013.

[10] S.V. Rajashekararadhya, Dr P. Vanaja Ranjan, "Efficient Zone Based Feature Extration Algorithm For Handwritten Numeral Recognition Of Four Popular South Indian Scripts ", Journal of Theoretical and Applied Information Technology, 2005-2008.