Hand printed Character Recognition using Neural Networks

Vamsi K. Madasu[1], Brian C. Lovell[2], M. Hanmandlu[3]

[1]*School of ITEE, University of Queensland, Australia*
[2]*NICTA and School of ITEE, University of Queensland, Australia*
[3]*Department of Electrical Engineering, I.I.T. Delhi, India*

Abstract

In this paper an attempt is made to recognize hand-printed characters by using features extracted using the proposed sector approach. In this approach, the normalized and thinned character image is divided into sectors with each sector covering a fixed angle. The features totaling 32 include vector distances, angles, occupancy and end-points. For recognition, both neural networks and fuzzy logic techniques are adopted. The proposed approach is implemented and tested on hand-printed isolated character database consisting of English characters, digits and some of the keyboard special characters.

1.0 INTRODUCTION

The problem of recognition of hand-printed characters is still an active area of research. With ever increasing requirement for office automation, it is imperative to provide practical and effective solutions. It has been observed that all sorts of structural, topological and statistical information about the characters does not lend a helping hand in the recognition process due to different writing styles and moods of persons at the time of writing. In this paper, we focus our attention on recognition of hand-printed English characters, which include lower case, upper case letters, numerals and some important special characters occurring on the keyboard. We consider a limited variation in shapes of characters. A brief review of literature is given in the following paragraphs.

Kimura and Sridhar [1] have worked on the recognition of handwritten numerals using multiple algorithms. Their first algorithm is a statistical classification technique that utilizes the histogram of the direction vectors derived from the contours of the character. Their second algorithm utilizes the features derived from the profile of the character in a structural configuration to recognize the numerals. They have tried to devise the best polling strategy between these two algorithms. The work in [2] starts with a random starting point and follows the pixels clockwise along the outer boundary of the symbol. A periodic function of the angular direction is constructed in terms of the normalized length of boundary from the starting point. Finally, this function is expressed in a Fourier series. The descriptors so obtained have the power to distinguish characters, which are mirror images of each other. Sridhar and Badraldin [3] have used the outer boundary of the character and expressed its X and Y coordinates as periodic functions. Finally, they have used the Fourier series to extract features from these functions. A radius function ρ (l) to represent the closed boundary of a character is described in

[4], where ρ (l) measures the length of the line connecting the boundary points to the centroid of the boundary curve. Then ρ (l) is expanded in a Fourier series whose coefficients act as the descriptors. An elliptic Fourier series to extract the features of the characters are given in [5].

The work of Taxt et al. [6] avoids traditional thinning and vectorization steps in the recognition process. The outer boundary is approximated by a spline curve. Using the parametric spline approximation, the curvature and circular graphs are calculated. The values measured at uniformly spaced positions on the spline approximation are used as descriptors in a statistical scheme. An autoregressive model is applied in [7] for closed boundaries of a character. They have used Mahalanobis distance as the measure while matching the features of unknown characters with those of the known. A new approach for the shape recognition of leaves using Complex Autoregressive (CAR) model is presented in [8]. In [9] it has been successfully shown that separately spaced character boxes provide superior machine readability and that writer idiosyncratic responses can be eliminated to achieve higher recognition rates.

The present work integrates the previous works of [10], [11]. In [11] a hybrid approach combining ring and sector approaches is presented. The features consist of distances and angles of characters enclosed in a fixed number of rings or sectors. The rings or sectors are formed by computing the centroid of all '1' pixels of a character, drawing a circle so as to enclose a character and then subdividing the circle either into rings or sectors. In the present study, we also use sectors but sectors are formed with respect to a fixed point, which is the center of the character image. This is in contrast to the earlier sector approach in which the centroid varies. The features here are taken to be the normalized vector distances and angles of all '1' pixels lying in different sectors and these features are found to be more effective as demonstrated in [10].

2.0 SECTOR APPROACH FOR FEATURE EXTRACTION

The recognition rate solely depends on the efficiency of features extracted from the character. Features could be topological, structural and geometrical (angles, distances). Topological or structural features work well for machine printed characters, as the shapes of these characters do not have drastic variations. However, these features alone are not suitable for recognition of hand printed characters due to some variations in writing styles. These variations could result in barbs and deformation in character shapes. In this study, geometrical features are explored in view of difficulty with topological and structural features for achieving high recognition rates.

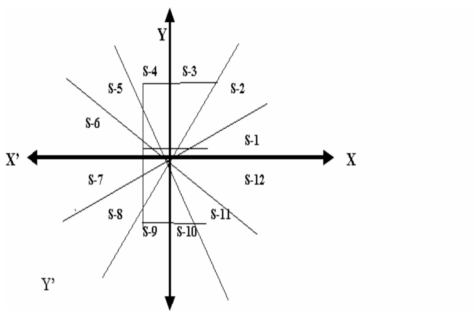

Fig. 1: Formation of sectors

In the sector approach, we consider the center of the character matrix as the fixed point. This change makes the features more robust as they do not depend on the centroid. In this method, the normalized and thinned image of size (42x32) is partitioned into a fixed number of sectors from the center of the image by selecting an angle. The number of sectors could be increased or decreased by changing the angle. However, from the experimentation, an angle of 30 degrees that leads to 12 sectors has been found to be an optimum choice. Here, we consider geometric features consisting of vector distances and angles. For extracting other features such as endpoints and occupancy, we have used 4-sectors with a sector angle of 90 degrees. The pictorial representation of character 'E' subdivided into 12-sectors is shown in Fig. 1. The first sector is from 0 to 30 degrees; the second sector is from 30 to 60 and so on as given in Table 1. Once the character is bifurcated into sectors, the portions lying in each sector is used for the extraction of features. It is noted that a character my not present in all the sectors.

Table 1: Ranges of angles in sectors

S-1: Sector 1 : 0 < Range1 <= 30	S-7: Sector 7 : 180 < Range7 <= 210
S-2: Sector 2 : 30 < Range 2 <= 60	S-8: Sector 8 : 210 < Range8 <= 240
S-3: Sector 3 : 60 < Range 3 <= 90	S-9: Sector 9 : 240 < Range9 <= 270
S-4: Sector 4 : 90 < Range 4 <= 120	S-10: Sector 10 : 270 < Range10 <= 300
S-5: Sector 5 : 120 < Range 5 <= 150	S-11: Sector 11 : 300 < Range11 <= 330
S-6: Sector 6 : 150 < Range 6 <= 180	S-12: Sector 12 : 330 < Range12 <= 360

2.1 Extraction of distance and angle features

The normalized distances and angles are shown to be robust in [10], hence, these are now calculated for all the sectors in the present approach. Let n_k be number of '1' pixels present in a sector k, with k=1,2,....,12. For each sector, the normalized vector distance, which is the sum of distances of all '1' pixels in a sector divided by the number of '1' pixels present in that sector is

$$D_k = \frac{1}{n_k}\sum_{i=1}^{n_k}\{(x_m - x_i)^2 + (y_n - y_i)^2\}^{1/2} \tag{1}$$

where, (x_i, y_i) are the co-ordinates of a pixel in a sector and (x_m, y_n) are the co-ordinates of the center of the character image.

This normalized vector distance D_k is taken as one set of features. Next, for each sector the corresponding angles of pixels are also calculated. The normalized angle, A_k which is taken as another set of features, is calculated as,

$$A_k = \frac{1}{n_k}\sum_{i=1}^{n_k}\mathbf{tan}^{-1}\left[\frac{y_n - y_i}{x_m - x_i}\right] \tag{2}$$

Both vector distance D_k and vector angles A_k constitute 24 features from 12 sectors. These features when plotted would give an approximate pattern of the original character as shown in Fig. 2.

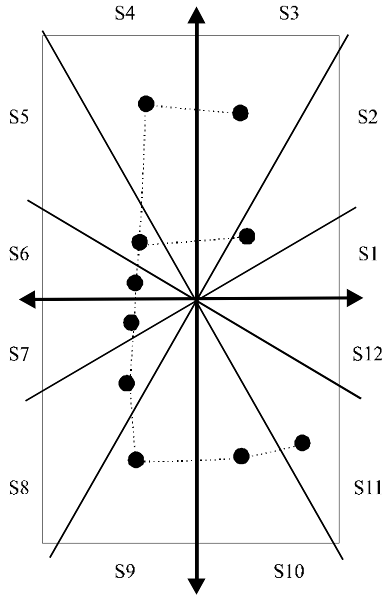

Fig. 2: Reconstructed shape of 'E' using features

2.2. Extraction of occupancy and end points features

Shape profile though depends on D_k and A_k , we augment these features proposed originally in [10] with other features such as occupancy and end points of a character for use in a recognition system. The occupancy quantifies the proportion of '1' pixels in a sector with respect to all '1' pixels in a character. We have used only four sectors for determining occupancy and accordingly, we have four values for occupancy and four values for end points. The endpoints are demarcated by tracing the character. For this, we start from any pixel and keep moving in the direction of '1' pixels until we find none and then search the neighboring five pixels in all directions (except the direction of tracing). If no '1' pixel is present, then that point is declared as an end point and sector in which it is lying is also noted as shown in Fig.3.

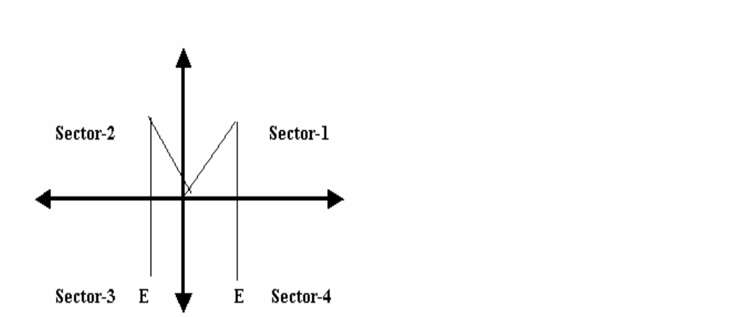

Fig. 3: End-points of character 'M'

If an end point is not present in any sector, it is taken as 0 for that sector. The location of end points in a sector do not depend very much on writing styles, as they lie within a sector only. We now have, a total of 32 features consisting of 12 vector distances, 12 vector angles, 4 occupancies and 4 end points. Using these features, a recognition system based on both Neural Network and fuzzy logic approaches has been developed.

3.0 TESTING AND IMPLEMENTATION

A database of hand printed characters is created for training and testing. The database is user independent, unbiased and random, written by thirty different writers. Writers are only briefed to maintain the legibility of characters. All the characters have been scanned in a flat bed A4 size scanner with 200 dpi resolution. After converting the scanned characters into binary patterns, the individual character images are extracted, separated category wise and stored in separate files for feature extraction. Table 3 presents category wise training, validation and test samples.

Table 3: Category wise total number of samples

Category	*Total number of samples in Training set*	*Total number of samples in validation set*	*Total number of samples in test set*
Numerals	3898	1292	1059
Capital letters	10359	3084	2927
Small letters	9921	3209	3038
Special characters	3305	1080	952

4.0 RECOGNITION SYSTEM

The recognition system consists of both neural network and fuzzy logic approaches, these approaches are widely used for document processing and therefore they merit a separate discussion as in the following.

4.1. Recognition based on neural networks

The feed forward Back Propagation Neural Network (BPNN) with one input layer, two hidden layers and one output layer is selected for training. We have constructed four BPNN's, one for each of the four categories: numerals, upper case letters, lower case letters and special keyboard characters to simplify the BPNN architecture. The feature set for each character in a category is stored in a separate file. The 32 features are ordered as pairs ($D_{k,}$ $A_{k,}$), k=1, 2... 12 followed by 4 features of occupancy for four sectors and finally presence of end points four sectors. Thus, input features are 32. The output features are the 4-bit patterns for each character. Separate BPNN's are built up for each category of characters (lower case and upper case etc). The different architectures are summarized in Table 4.

The network for numerals (0-9) settles at 0.041 mean square error (MSE), which is achieved at 3000 cycles (epochs), where as MSE for both training and validation sets is arrived at 0.032. The results of applying test data of numerals are summarized in Tables 5 and 6. Tables 5 through 8 not only show the character-wise RR but also the percentage of the character recognized as some other character. For example in Table 7, the RR for the character 'B' is 99; however, it is recognized as character 'R' in one percent. The category wise RR of characters ranges from 99.23 to 99.73 percent with the complete character set having RR of 99.46 percent. This study proves the efficiency of the proposed features extracted using sector-based method. We will now discuss a simple approach using fuzzy logic.

Table 4: BPNN architectures for different categories

Type	*I*	*H*	*O*	*I*	*H1*	*H2*	*O*	*Training set*	*Validation set*	*Cycles*
Numerals	1	2	1	32	24	12	5	0.041	0.032	3000
Capital	1	2	1	32	24	12	5	0.062	0.043	8000
Small	1	2	1	32	24	12	5	0.070	0.051	8500
Special	1	2	1	32	24	12	5	0.023	0.001	1500

Where I: Input Layer, H: Hidden Layer, O: Output Layer, H1: 1st Hidden layer, H2: 2nd Hidden Layer

For testing, all test data sets of all categories are combined into a single data set to try on all BPNN's. The network for upper case letters takes 8000 cycles for a MSE of 0.043 and that for lower case letters, takes 8500 cycles for a MSE of 0.051. The MSE for the special characters is obtained as 0.001 in 1500 cycles. The Recognition Rate (RR) of a particular character is computed as the percentage of correctly recognized characters to the total number of characters used for testing. The results of applying test data of special characters, upper case letters and lower case letters are summarized in Tables 6, 7 and 8 respectively. The category wise RR's for BPNN are tabulated in Table 9.

Table 6: RR using BPNN for special characters

	?	**{**	**}**	**[**	**]**	#	**!**	<	>	+	=
?	**99**	0	0	0	0	0	2	0	0	0	0
{	0	**100**	0	0	0	0	0	0	0	0	0
}	0	0	**100**	0	0	0	0	0	0	0	0
[	0	0	0	**100**	0	0	0	0	0	0	0
]	0	0	0	0	**100**	0	0	0	0	0	0
#	0	0	0	0	0	**100**	0	0	0	0	0
!	1	0	0	0	0	0	**98**	0	0	0	0
<	0	0	0	0	0	0	0	**100**	0	0	0
>	0	0	0	0	0	0	0	0	**100**	0	0
+	0	0	0	0	0	0	0	0	0	**100**	0
=	0	0	0	0	0	0	0	0	0	0	**100**

Table 6: RR using BPNN for numerals

	0	1	2	3	4	5	6	7	8	9
0	**100**	0	0	0	0	0	0	0	0	0
1	0	**99.1**	0	0	0	0	0	0	0	0
2	0	0	**98**	0	0	0	0	0	0	0
3	0	0	0	**98**	0	0	0	0	0	0
4	0	0	0	0	**100**	0	0	0	0	0
5	0	0	0	0	0	**100**	1	0	0	0
6	0	0	0	0	0	0	**99**	0	0	0
7	0	0.8	1.5	0	0	0	0	**100**	0	0
8	0	0	0.5	2	0	0	0	0	**100**	0
9	0	0.1	0	0	0	0	0	0	0	**100**

Table 7: RR using BPNN for upper case letters

	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**
RR	100	99 (1 R)	100	98.2 (0.03 D, 1.5 O)	100	98.3 (1.2 B, 0.5 P)	97.3 (1.4 C, 0.3 E, 1.0 O)	100	100	99.1 (0.9 I)
	K	**L**	**M**	**N**	**O**	**P**	**Q**	**R**	**S**	**T**
RR	100	100	100	100`	99 (1 C)	98.8 (1.2 F)	100	100	100	100
	U	**V**	**W**	**X**	**Y**	**Z**				
RR	97.9 (2.1 V)	98.7 (1.3 U)	100	100	100	100				

Table 8: RR using BPNN for lower case letters

	a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**
RR	98 (1.2 d, 0.8 h)	99 (1 h)	100	98.2 (0.3 b, 1.5 o)	100	98.3 (1.2 b, 0.5 p)	97.3 (1.4 c, 0.3 e, 1 o)	100	100	99.1 (0.9 i)
	k	**l**	**m**	**n**	**o**	**p**	**q**	**r**	**s**	**t**
RR	100	100	100	100	99 (1 c)	98.8 (1.2 f)	100	95.7 (2.2 i, 2.1 s)	100	100
	u	**v**	**w**	**x**	**y**	**z**				
RR	97.9 (0.9 n, 2.1 v)	98.7 (1.3 u)	100	100	100	100				

Table 9: Category wise overall RR using BPNN

Category	Recognition Rate(RR)
Numerals	99.41
Capital case characters	99.47
Small case characters	99.23
Special keyboard characters	99.73
Overall Recognition Rates	**99.46**

4.2 Recognition based on fuzzy logic technique

The concept of fuzzy logic is proposed in [10] in order to take account of variability involved in a feature when several samples of a character are considered. We employ this concept for the fuzzification of features using Gaussian function. If there are ‘n’ possible features for each character with ‘m’ such samples, different values of the same feature for all samples form a fuzzy set. Here, we consider 16 features, i.e., 12 vector distances and 4 occupancies that will result in 16 fuzzy sets. Experimentation has shown that these 16 features are sufficient to give optimum recognition rates. To extract 16 fuzzy sets, the reference character data formed by combining both training and validation sets as given in Table 3 are used. For each reference character, 16 fuzzy sets are formed. The mean and variances computed for each of these 16 fuzzy sets are taken as Knowledge Base (KB). The procedure is repeated for all the characters in the reference data set. In the present study, there are 73 reference characters. Separate KB is created for every reference character. Given a very large number of samples, by choosing the Gaussian fuzzification function, the membership function of each feature value in the fuzzy set can be determined. However, there is no need to compute the membership function for the reference characters. What we require are the membership functions for the features of an unknown character. The features of unknown character need to be matched with all the features of reference characters available in the form of their membership functions. It is possible to compute the membership functions by associating the features of the input character with the fuzzy sets. The KB is created by the mean m_{ri} and variance σ^2_{ri} for each of 16 fuzzy sets of reference characters, using

$$m_{ri} = \frac{1}{N_i} \sum_{i=1}^{N_i} f_{ij} \tag{3}$$

$$\sigma^2_{ri} = \frac{1}{N_i} \sum_{j=1}^{N_i} (f_{ij} - m_{ri})^2 \tag{4}$$

where, N_i the number of samples in ith cluster and f_{ij} stands for jth feature value in ith cluster. For an unknown input character x we extract the 16 features using the sector approach. The membership value $_{xi}$ can be calculated by using the ith feature (x_i, i=1...16) of the unknown character and the known i^th fuzzy set mean and variance of a reference character in the following Gaussian fuzzification function;

$$\mu_{xi} = e - \left(\frac{(m_{ri} - x_i)^2}{2\sigma_{ri}^2} \right) \tag{5}$$

The fuzzy distance of input character x with respect to a particular reference character 'r' in KB with its mean m_{ri} is calculated using the equation:

$$d_{\mu_r}(x) = \frac{1}{N_f} \sum_{i=1}^{N_f} \mu_{xi}^2 (m_{ri} - x_i)^2 \tag{6}$$

where, N_f is the number of fuzzy sets. Similarly, the fuzzy distances with respect to all the reference characters in the KB are computed. Now, among the 73 fuzzy distances, the reference character 'r' corresponding to minimum $d_{\mu_r}(x)$ gives the identity of the unknown character.

Fuzzy logic approach is implemented on the database given in Table 2. The knowledge Base (KB) for each character is constructed from vector distances and occupancies only. The RR ranges from 99.5 to 100 percent with no rejection as given in Table 10 for the test data.

Table 10: Category wise Overall RR using fuzzy logic technique

Category	Recognition Rate
Numerals	100
Capital case	99.96
Small case	99.51
Special characters	100
Overall Recognition Rates	99.98

5.0 CONCLUSIONS

Two approaches using both BPNN and fuzzy logic are presented for recognition of hand printed characters using the sector based feature extraction method. The features consist of normalized vector distances and angles. The sector approach of this chapter is specific to hand printed characters, because of consideration of limited variability in character shapes. The approach is global in nature as all sectors emanate from the center point of the character image. Hence, the local information within the sector that cannot be further partitioned for achieving increased recognition. We note here, the recognition strategies are able to take care of 15 to 20 percent of variation in the character shapes. Considering the computational time and RR as the desired parameters, fuzzy logic is the ideal choice. However, a large database is required for BPNN to learn and for fuzzy logic approach to have representative KB.

REFERENCES

[1] F.M Kimura, M. Shridhar (1991), "Handwritten Numeral Recognition Based on Multiple Algorithms", *Pattern Recognition*, Volume 24 (Issue 10), pp. 969-983.

[2] C.T Zahn and R.Z. Roskiers (1972), "Fourier Descriptors for Plane Closed Curves", *IEEE Transactions on Computers*, Volume 21(Number 3), pp. 269-281.

[3] M. Shridhar and A. Badraldin (1984), "High accuracy Character Recognition Algorithm using Fourier and Topological Descriptors", *Pattern Recognition*, Volume 17(Issue 5), pp. 515-523.

[4] F. Ghorbel, G. Cazuguel and J.L. Tonaye (1985), "Similarity-Invariant Analysis of handwritten Zip code using Fourier descriptors", *International Journal of Research & Engineering in Postal Application*s, Volume 1, pp. 1-5.

[5] F.P. Kuhl and C.R. Giardina (1982), "Elliptic Fourier features of a closed contour", *Computer Graphics and Image Processing*, Volume 18, pp. 236-258.

[6] T. Taxt, J.B. Olafsdotter and M. Daehlen (1990), "Recognition of Handwritten Symbols", *Pattern Recognition*, Volume 23(Issue 11), pp. 1155-1166.

[7] S.R. Dubois and F.H. Glanz (1986), "An Auto regressive model for two dimensional shape classification", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 8(Issue 1), pp. 55-66.

[8] L. Sekita, K. Torachi, R. Mori, K. Yamamoto and H. Yamada (1988), "Feature Extraction of Handwritten Japanese Characters by Spline Functions for Relaxation Matching", *Pattern Recognition*, Volume 21(Issue 1), pp. 9-17.

[9] M.D. Garris and D.L. Dimmick (1996), "Form Design for High Accuracy Optical Character Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 18(Issue 6), pp. 653-656.

[10] M. Hanmandlu, K.R.M. Mohan and V. Gupta (1997), "Fuzzy logic based character recognition", *Proceedings of IEEE International Conference on Image Processing*, pp. 26-29.

[11] M. Hanmandlu, K.R.M. Mohan and H. Kumar (1999), "Neural-based handwritten character recognition", *Proceedings of International Conference on Document Analysis and Recognition*, pp. 241-244.