A Fuzzy based Classification Scheme for Unconstrained Handwritten Devanagari Character Recognition

Sushama Shelke
Electronics and Telecommunication
NBN Sinhgad School of Engineering,
Pune, India
sushama.shelke@sinhgad.edu

Shaila Apte
Electronics and Telecommunication
Rajarshi Shahu College of Engineering
Pune, India
sdapte@rediffmail.com

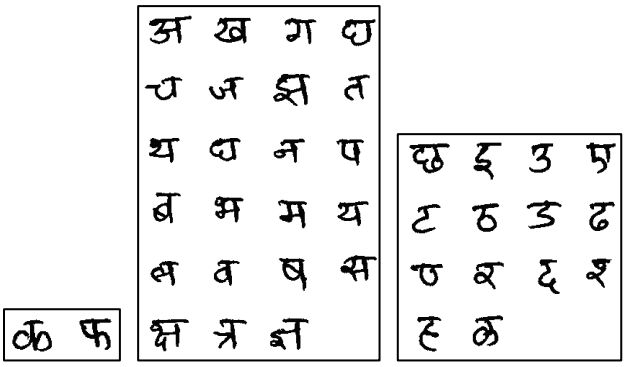
***Abstract*— The large data set and similar structural features of the characters in Devanagari script demand a highly efficient classification and recognition system. This paper presents a novel approach for the recognition of unconstrained handwritten Devanagari characters. The system is based on multi-stage classification scheme. The classification stages categorize the characters into smaller groups. The classification is done using two stages, first stage is based on fuzzy inference system and second stage is based on structural parameters. The fuzzy system improves the classification over crisp classification. The classified characters are passed to the feature extraction stage. The final stage implements feed forward neural network for character recognition. The recognition accuracy achieved by the proposed method is 96.95%.**

Keywords*— *handwritten Devanagari characters recognition; structural features; fuzzy logic; neural network

I. INTRODUCTION

Handwritten character recognition remains to be a challenging area for researchers in this area since few decades due to various reasons. Firstly, handwriting recognition finds application in many fields like Postal Automation [1], Banking, Form filling etc. The automation in handwriting recognition greatly reduces the efforts of character recognition and data entry. Secondly, the handwritten characters exhibit large variations in writing style. Thirdly, the recognition accuracy is affected by various parameters like the acquisition device, pen width, pen ink color, physical and mental condition of the writer and so on. Handwritten Devanagari character recognition is even more complex. The reasons are the large number of characters, character structure, shape and presence of modifiers. A Devanagari character set consists of 16 vowels and 36 consonants [2]. These characters are written below a horizontal line called as header line, unlike English, where characters are written above a horizontal line. The header line also joins the characters in a word. The script has many character pairs which are almost similar to each other. The script is further made complex by the modifiers which may lie above, below or in line with the character. Another complexity in Devanagari script is the conjunct characters, where two or more characters are joined together to form a special character. Researchers have attempted recognition by pre-classifying the characters into different classes on the basis of structural parameters [3]. About 60% of Devanagari characters exhibit a vertical line in them. This vertical line is at the center in two of the characters while in the rest, it is towards the end. The remaining 40% of the characters do not have a vertical line. Fig. 1 shows the characters grouped on the basis of this structural parameter, the vertical line. In Fig. 1 a), the vertical line is present at the center of the character. In Fig.1 b), the vertical line is present towards the end of the character, while in Fig. 1 c), there is no vertical line in the characters.

Due to the variations in the writing style and pre-preprocessing operations, the samples of the same character are misclassified, when the structural parameters are used for classification. This further affects the recognition accuracy. In order to reduce this misclassification rate, there is a need to design a classifier that classifies the characters correctly independent of the variations in the writing style and pre-processing output. In this paper, we propose a system for unconstrained handwritten Devanagari character recognition, where the pre-classification of the characters is done using fuzzy logic. Further, the recognition is done by applying the extracted features to feedforward neural network.



(a) (b) (c)

Fig. 1. Characters with different structural features.

978-1-4799-5522-0/15/$31.00 ©2015 IEEE

OCR work on printed Devanagari script started in early 1970s. Oivind Due Trier et al [4] discussed various feature extraction techniques for character recognition. An extensive research on printed Devanagari text was carried out by Veena Bansal and R. M. K. Sinha [5-7]. First system for hand-written numeral recognition of Devanagari characters was proposed by R. Bajaj et al [8]. P. M. Patil and T.R. Sontakke [9] also presented an algorithm for handwritten Devanagari numeral recognition which was rotation, scale and translation invariant using fuzzy neural network. Fuzzy logic is also used for unconstrained handwritten character recognition [10-13]. Researchers have also applied multiple classifiers [14] and multi-stage classifiers [15] to improve the recognition rate of handwritten Devanagari characters.

The rest of the paper is organized as follows: Section II presents the proposed system. Section III describes the fuzzy based classification. In section IV, the feature extraction and in Section V the recognition technique is given. Section VI discusses the results obtained and Section VII finally discusses the conclusion.

II. THE PROPOSED SYSTEM

The proposed system flow is shown in Fig. 2. It recognizes 39 Devanagari characters discussed in previous section.

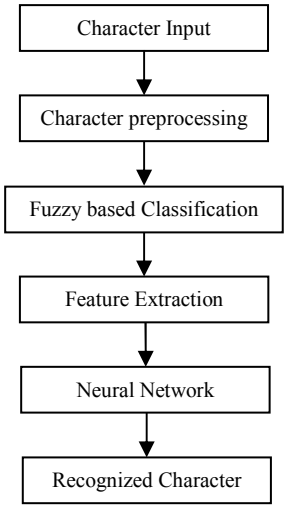


Fig. 2. Proposed system design.

At first, the handwritten Devanagari characters are passes through a pre-processing stage. They are further pre-classified into various classes using fuzzy classifier. The features are then extracted from the characters and applied to the neural network built for each class. Finally the output of the neural network points to the recognized character.

The handwritten characters are scanned at 300 dpi in bmp file format. The header line is removed by finding the average width of the header line. Next, preprocessing is carried out before fuzzy classification. The pre-processing stage is critical due to various reasons. Handwritten characters do not exhibit same width over the entire shape of the character. Generally, the width of the character tends to reduce at the curvature. The binarization process may produce a gap at such places. An averaging filter prior to binarization bridges such gaps. The unwanted strokes are further removed by performing morphological opening operation on the binarized image. After preprocessing, the characters are passed to fuzzy classifier to find the presence of vertical line and its position in the character and classify the characters accordingly. The classification of the characters also employs other structural features like enclosed region and end points.The fuzzy based classifier is explained in the next section.

III. FUZZY BASED CLASSIFICATION

This section presents the classification of the pre-processed characters using fuzzy rules and other structural parameters. The classification of the characters using crisp rules results into their misclassification due to variation in the writing style, pre-preprocessing output and slant in the characters. These variations are taken care of by the fuzzy inference system to improve the classification accuracy. A typical fuzzy interference system is shown in Fig. 3.

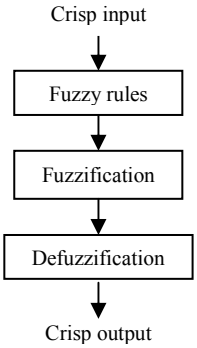


Fig. 3. Typical fuzzy inference system.

It includes the following blocks: a rule base block containing the fuzzy rules based upon the database, a fuzzification block which transforms the crisp inputs based upon the rules. Finally, the defuzzification block transforms the fuzzy results of the inference into a crisp output [16]. The designing of fuzzy inference system is given below.

A. Creating fuzzy rules

The fuzzy rules are formulated for determining the vertical line and its position in the character. If the vertical line is present towards the end, the character contains end bar. If the character contains the vertical line towards the middle, the character contains mid bar. If there is no vertical line in the character, the character is a no bar character. The pre-processed image is rotated clockwise and anti-clockwise through 30 degrees with an increment of one degree. At each rotation, the image is cropped and histogram of the image is obtained. The histogram generates the column with maximum pixel count. Let the cropped image be *f(m,n)*, the maximum pixel count be *max_value* at n_{max} column. Thus using these values, we obtain the parameters *vt* for finding the presence of vertical line and *per* for position of the vertical line, where,

978-1-4799-5522-0/15/$31.00 ©2015 IEEE

$$vt = \frac{max\_value}{m} \quad (1)$$

$$per = \frac{n_{max}}{n} \times 100. \quad (2)$$

The input membership function for the presence of vertical line is given in Fig. 4 and the membership function for the position of the vertical line is given in Fig. 5. The membership function for the output of the fuzzy inference system is shown in Fig. 6.

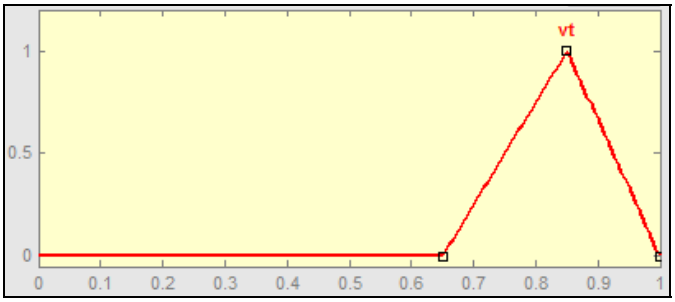


Fig. 4. Membership function for vertical line detection.

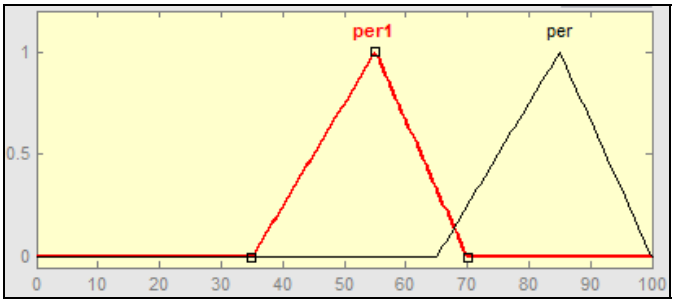


Fig. 5. Membership function for detecting position of vertical line.

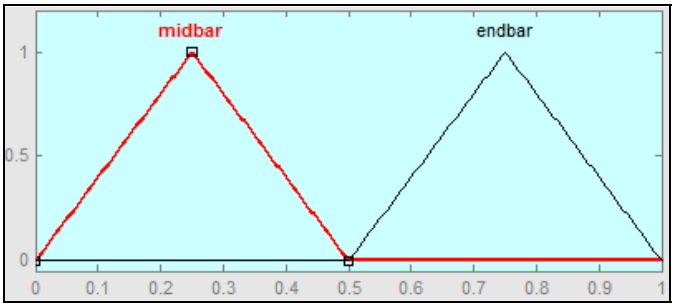


Fig. 6. Membership function for output of fuzzy inference system.

The fuzzy rules for detection of the end bar and the mid bar are:

- If the input is *vt* and input2 is *per1* then output is endbar
- If input is *vt* and input2 is *per* the output is midbar

B. *Fuzzy inference method*

The proposed system uses Mamdani's fuzzy inference method. Mamdani type inference expects the output membership functions to be fuzzy sets. A two input, two rule Mamdani fuzzy inference system is shown in Fig. 7 and Fig. 8.

C. *Defuzzification method*

The defuzzification is done by the centroid method. This is done to obtain a crisp output to classify the characters. A threshold *T* is set to 50 to obtain a crisp output. If the output of the fuzzy inference system output is greater than *T*, the character belongs to mid bar class else belongs to end bar class. Fig. 7 shows the example of classifying the character to mid bar class, while Fig. 8 is the example of classification of the character to end bar class.

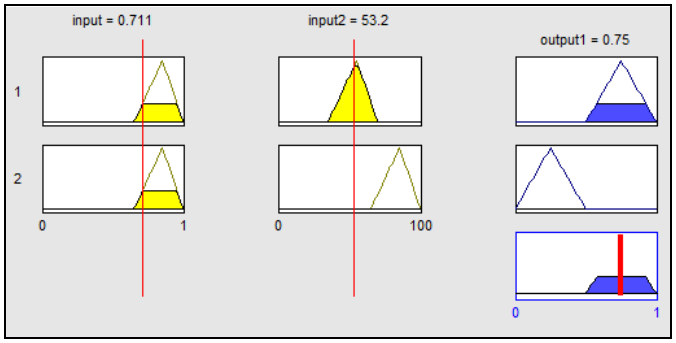


Fig. 7. A two input, two rule fuzzy inference system for mid bar character.

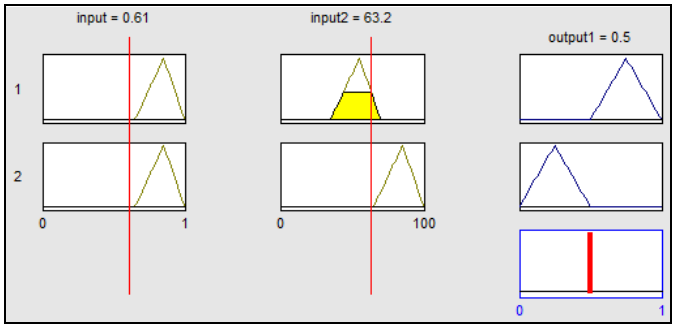


Fig. 8. A two input, two rule fuzzy inference system for end bar character.

D. *Detection of other structural features*

Further, 8-adjacency is used to find the presence of connected components or the enclosed regions within the character. Table I indicates the classification of the characters based upon these fuzzy inference system and presence of enclosed region in the character. The characters are classified into six classes based upon the presence or absence of these structural features. The cropped binary image *f(m,n)* is thinned to yield a single pixel wide character. This character is then passed to hit-or-miss transformation to find the endpoints of the character.

978-1-4799-5522-0/15/$31.00 ©2015 IEEE

TABLE I. FUZZY BASED CLASSIFICATION

Class	Mid bar	End bar	Enclosed region
No bar not enclosed	0	0	0
No bar enclosed	0	0	1
End bar not enclosed	0	1	0
End bar enclosed	0	1	1
Mid bar not enclosed	1	0	0
Mid bar enclosed	1	0	1

Eight structuring elements are used to detect the location of endpoints in all eight directions. The image is then partitioned into four quadrants as shown in Fig. 9. Here quadrants 2 and 3 only are of interest. The presence of end points in quadrants 2 and 3 classify the character into four classes further, namely, 00, 01, 10 and 11, where, class 00 indicates that there is no end point in quadrant 2 and 3, whereas, 01 indicates that there is an endpoint in quadrant 2 and no endpoint in quadrant 3, 10 indicates that the end point is in quadrant 3 and so on.

Fig. 9. Character partitioning for end point detection.

TABLE II. SECOND STAGE STRUCTURAL CLASSIFICATION

Class	Mid level features	
	End point in quadrant 3	*End point in quadrant 2*
00	Absent	Absent
01	Absent	Present
10	Present	Absent
11	Present	Present

Table II illustrates this classification. At this stage, after the classification of characters based on two stage classification using structural features, the characters are classified into 24 classes.

IV. FEATURE EXTRACTION TECHNIQUE

This section discusses the feature extraction method. Features are extracted by finding the density of pixels in non-overlapping zones into which the character is partitioned. The binary characters are spread over 70x50 pixels dimension. The cropped character image is used to calculate normalized pixel density features. The characters are partitioned into 35 non-overlapping zones of size 10x10 and the number of zeros in each zone is calculated. The normalized pixel density features *npd(x, y)* are calculated as,

TABLE III. RECOGNITION PERFORMANCE

Neural Network Parameter	Value
Number of inputs	35
Number of hidden layers	1
Number of neurons in hidden layer	square root of the product of input & output
Hidden layer activation function	Hyperbolic tangent sigmoid
Number of outputs	characters in the class
Output layer activation function	Linear
Goal	0.001
Error function	Mean square error
Maximum number of epochs	300
Training algorithm	Levenberg-Marquardt
Training parameter initial value	0.001
Training parameter decrease factor	0.1
Training parameter increase factor	10

$$npd(x,y) = \frac{100 - g(x,y)}{100} \tag{3}$$

where,

$$g(x,y) = \sum_{i=1}^{10}\sum_{j=1}^{10} g(x,y) \tag{4}$$

is the density of the pixels in the 10x10 window, x = 1, 2, …,7 and y = 1, 2,…,5. These 35 features are applied to neural network during training and testing of the characters.

V. RECOGNITION TECHNIQUE

This section discusses the final recognition stage. The recognition is done using multilayer feedforward backpropagation neural network. It consists of an input layer, a hidden layer and an output layer [17]. The feature vector is applied as the input signal to the neurons in the hidden layer from the input layer. The neurons are connected to each other by links which are associated with weights. A bias is similar to weight. It acts exactly as a weight on a connection from a unit whose activation is always one. Each neuron in the hidden layer includes a nonlinear activation function. Once the network weights and biases are initialized, the network is ready for training. The hidden neurons enable the network to learn complex tasks by extracting progressively more meaningful features from the input vectors. A learning rule is a procedure for modifying weights and biases of a network. The purpose of the learning rule is to train the network to perform a pattern recognition task. During training the weights and biases of the network are iteratively adjusted to minimize error. The training process requires a set of examples of proper network behavior, network inputs and the target outputs. As

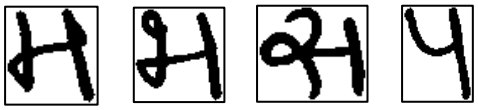
978-1-4799-5522-0/15/$31.00 ©2015 IEEE

each input is applied to the network, the network output is compared to the target. The error is calculated as the difference between the target output and the network output. The goal is to minimize the average of the sum of these errors. The network parameters such as activation function, number of neurons in the hidden layer and the network training function can be varied and selected for the optimum performance of the network. The neural network parameters selected in the proposed system are given in Table III.

VI. Experiments and Results

The handwritten Devanagari character dataset is collected from more than 50 writers with more than 500 samples per character. This results to more than 40,000 samples in the database. 60% of these samples are used for training, 20% is used for validation and 20% is used for testing. There is no limitation on the size of the database. The parameters for mid bar and end bar detection are selected after testing about one third of the characters in the database.

At first, the characters are passed through fuzzy inference system for detection of the vertical line and its position in the character. The fuzzy based classification is combined with the enclosed region feature to classify the characters into 6 classes. This is further followed by the classification based on the end points in the character. The characters are classified into 4 classes each based upon the position of the end points in the character. Finally, after both the levels, 24 classes are obtained.



(a) (b) (c) (d)

Fig. 10. Characters misclassified by crisp classification to (a) mid bar enclosed-11, (b) mid bar enclosed-11, (c) mid bar enclosed-11, (d) no bar not enclosed-01.

The handwritten characters may take different shapes as per the writing style of the writer. This may result in classification of the same character to different classes in case of crisp classification. Fig. 10 shows character samples misclassified using crisp classification Here, the first sample in Fig. 10 (a) is classified to the class 'mid bar enclosed-11', the second in Fig. 10 (b) is classified to 'mid bar enclosed-11', the third in Fig. 10 (c) is classified to 'mid bar enclosed-11' while the forth in Fig. 10 (d) is classified to 'no bar not enclosed-01'. After using fuzzy inference system, whereas classified correctly using fuzzy based classification. characters in Fig. 10 (a), Fig. 10 (b) and Fig. 10 (c) are classified to end bar enclosed -11, and Fig. 10 (d) is classified to end bar not enclosed 01. Thus misclassification of the characters results into increasing the number of character classes in each class out of 24. This also adds to increased complexity of the neural network. Since as the number of character classes increase, the output neurons increase, thereby increasing the hidden neurons. This also adds to the computational cost.

The pixel density features are extracted from the characters to be tested. The matching is done by applying the 35 pixel density features to the neural network built for one of the 24 classes to which the test character belongs to. The neural network output with maximum value points to the recognized character class. The overall recognition rate for the proposed system is 96.95%. Results show that some characters which exhibit large variations get classified into many classes in case of crisp classification. This misclassification is reduced in case of fuzzy based classification. The characters with lesser shape variations give more than 98% recognition rate. Table IV presents the recognition rates for the recognition using crisp classification and fuzzy based classification, keeping the same feature extraction and recognition methods.

TABLE IV. Recognition Performance

Classifier	Recognition Rate
Crisp classification	95.00 %
Fuzzy based classification	96.95 %

VII. Conclusion

Here in this paper, we propose a novel approach for Devanagari character recognition based on fuzzy classification. The approach involves multi stage classification, feature extraction and neural network recognition scheme for unconstrained handwritten Devanagari characters. The preprocessed character passes through a two stage structural classification. The first stage is fuzzy based while the second stage is based upon other structural features like enclosed region and end points. The fuzzy classification is based upon the presence of the vertical bar and its position in the character. The classification stages classify the characters to one of the 24 classes. Features based upon pixel density are then extracted and are applied to feed forward back propagation neural network. The fuzzy based classification improves the recognition over the crisp classification. It also reduces the burden on the feature extraction and recognition stages to improve the recognition accuracy.

References

[1] K. Roy, S. Vajda, U. Pal, and B. B. Chaudhuri, "A system towards Indian postal automation," in *Proc. Int. Workshop Frontiers Handwrit. Recognit.*, 2004, pp. 580–585.

[2] U. Pal and B. B. Chaudhari, "Indian script character recogntion: a survey", *Pattern Recognit.*, vol. 37, pp. 1887-1899, 2004.

[3] S. Shelke, S. Apte, "A novel multi-feature multi-classifier scheme for unconstrained handwritten Devanagari character recogntion", in *Proc. 12th Int. Conf. Frontiers Handwrit. Recognit.*, 2010, pp. 215-219.

[4] Oivind Due Trier, Anil K. Jain and Torfinn Taxt, "Feature extraction methods for character recognition – a survey", *Pattern Recognit.*, vol. 29, pp. 41-62, 1996.

[5] V. Bansal, R.M.K. Sinha, "Partitioning and searching dictionary for correction of optically read Devanagari character strings", *Int. J. Document Anal. Recognit.*, vol. 4, pp. 269–280, 2002.

978-1-4799-5522-0/15/$31.00 ©2015 IEEE

[6] V. Bansal and R. M. K. Sinha, "On how to describe shapes of Devanagari characters and use them for recognition", in *Proc. 7th Conf. Document Anal. Recognit.*, 1999, pp. 410-413,

[7] V. Bansal, "Integrating knowledge sources in Devanagari text recognition", *Ph.D. Thesis*, IIT Kharagpur, 1999.

[8] R. Bajaj, L. Dey and S. Chaudhary, "Devanagari numeral recognition by combining decision of multiple connectionist classifiers", *Sadhana*, vol. 27, pp. 59-72, 2002.

[9] P. M. Patil and T. R. Sontakke, "Rotation, scale and translation invariant handwritten Devanagari numeral character recognition using general fuzzy neural network", *Pattern Recognit.*, vol. 40, pp. 2110-2117, 2007.

[10] M. Hanmandlu, A. V. Nath, A. C. Mishra, and V. K. Madasu, "Fuzzy model based recognition of handwritten hindi numerals using bacterial foraging," in *Proc. Int. Conf. Comput. Inf. Sci.*, 2007, pp. 309–314.

[11] M. Hanmandlu, J. Grover, V. K. Madasu, and S. Vasikarla, "Input fuzzy modeling for the recognition of handwritten Hindi numerals," in *Proc.Int. Conf. Inf. Technol.*, 2007, pp.208–213.

[12] M. Hanmandlu and O. V. R. Murthy, "Fuzzy model based recognition of handwritten numerals," *Pattern Recognit.*, vol. 40, pp. 1840–1854, 2007.

[13] M. Hanmandlu, K. R. M. Mohan, S. Chakraborty, S. Goyal, and D. R. Choudhury, "Unconstrained handwritten character recognition based on fuzzy logic," *Pattern Recognit.*, vol. 36, no. 3, pp. 603–623, 2003.

[14] U. Pal, T. Wakabayashi, and F. Kimura, "Comparative study of Devanagari handwritten character recognition using different features and classifiers," in *Proc. 10th Conf.Document Anal. Recognit.*, 2009, pp. 1111–1115.

[15] S. Arora, D. Bhatcharjee, M. Nasipuri, and L. Malik, "A two stage classification approach for handwritten Devanagari characters," in *Proc. Int. Conf. Comput. Intell. Multimedia Appl.*, 2007, pp. 399–403.

[16] S. Sivanandam, S. Sumati, S. Deepa, Introduction to Fuzzy Logic using MATLAB, Springer, Germany, 2010.

[17] M. Hagan, H. Demuth and M, Beale, *Neural Network Design*, Thomson Learning, India, 2003.

978-1-4799-5522-0/15/$31.00 ©2015 IEEE