Performance Optimization and Comparative Analysis of Neural Networks for Handwritten Devanagari Character Recognition

Sushama Shelke
Dept. of E & TC Engg.,
College of Engineering,
Pune, India
sds.extc@coep.ac.in

Shaila Apte
Dept. of E & TC Engg.,
Rajarshi Shahu College of Engineering,
Pune, India
sdapte@rediffmail.com

Abstract— **Devanagari script has character set with rich structural features that makes the recognition of unconstrained handwritten Devanagari characters difficult However, these features can be used to divide the characters into different categories. This paper presents few techniques for optimizing the recognition accuracy at pre-classification stage, feature extraction stage and recognition stage. Initially, the pre-classification of the characters is done into different classes using various structural features. Then features are extracted using optimized feature extraction techniques. Finally, the recognition is done using neural network. In this paper, different neural networks are implemented and their performances are analyzed.**

Keywords- character recognition; classification; zonal average features; neural network recognition

I. INTRODUCTION

Optical character recognition converts character images into text or speech. The text output can be stored or edited. Such systems are used in various Postal Automation for address recognition, in Banking for cheque recognition, Offices for Data entry and form filling etc. While the speech output can be useful in reading systems for physically handicapped or can be used as a teaching and learning tool. Devanagari has a large number of characters set. It has 16 vowels and 36 consonants [1]. Their shape is complex with varied structural features. Every character has a horizontal line below which the characters are written. This horizontal line is called as header line. The header line also connects the words in a sentence. The vowels take different shapes and are connected to the consonants in different styles, which are termed as modifiers. The modifiers occur above the header line, in line with the character or are joined at the bottom of the character. The character set also includes conjunct or compound characters that are formed by joining two or three characters. Based upon the source of the characters, the character recognition systems can be classified into two types, namely, printed character recognition systems and handwritten character recognition systems. While, based upon the nature of character acquisition process, the systems are classified into two types, namely offline recognition systems and online recognition systems. Plenty of research is carried out in offline character recognition for various Indian scripts including printed and handwritten characters, where the character images are stored in the memory prior to applying to the recognition engine. At first, work on printed Devanagari script for recognizing characters was carried out in 1970s [2-4]. This was followed by the development of systems for handwritten Devanagari digit recognition [5]. The success in handwritten numeral recognition encouraged the researchers to develop systems for off-line handwritten Devanagari character recognition systems [6].

Further advances led to the use of soft computing techniques like Artificial Neural Network and Fuzzy logic [7-10] for Devanagari character recognition in order to improve the recognition accuracy, especially in handwritten character recognition systems. Researchers also attempted to enhance the recognition rate of handwritten Devanagari characters by implementing multiple features, multiple classifiers and multi-stage classifiers [11-12].

Among different soft computing techniques, neural networks are found to be more popular for recognition of Indian numerals and characters, either printed or handwritten [13-18]. This is because of the characteristics of neural networks like parallelism, modularity and dynamic adaptation. A neural network has a parallel structure. It recognizes by learning through the samples. The learning helps the network to generalize and recognize the pattern in a particular application. The neural network also has the characteristics like nonlinear nature, mapping the non-linear input to the linear output, adapting to a particular application, tolerance to noise etc. It has also been observed that Feed-forward back-propagation network is commonly implemented for character recognition application. This paper explores other back-propagation neural networks along with few other optimization techniques to analyze the recognition accuracy for handwritten Devanagari characters.

The next part of the paper is organized as follows: Section II discusses the features of Devanagari script. Section III explains the proposed system. Section IV presents the optimized structural classification. In Section V, optimized feature extraction technique is presented. Section VI explains the neural network architectures used for recognition. Section VII and Section VIII presents the results and the conclusion respectively.

II. FEATURES OF DEVANAGARI SCRIPT

This section describes the prominent structural features found in Devanagari characters. These features form the basis of dividing the characters into different classes. Researchers have done pre-classification of the characters into different classes prior to recognition [18]. First structural feature is the vertical line in the character. A large number of characters in Devanagari script have a vertical line in them. This vertical line is an important structural feature. In two of the characters, the vertical line is at the center. Let these characters be called as 'Mid-line' characters. Such characters, that are used in the proposed system are shown Fig. 1a). The vertical line is towards right hand side in about 60% of the total characters, i.e. 'End-line' characters. Such characters are shown in Fig. 1 b). The number of characters in the end-line class is large. The remaining characters do not have this feature in them. Let us call such characters as 'No-line' characters. Fig. 1 c) shows no-line characters. Second structural feature is the enclosed region in the character. As seen from Fig. 1, many characters consist of enclosed region. The third structural feature which is used in this system is the position or number of end-points in the character. Devanagari script also has character pairs which are much similar in shape.

There is a wide variety of writing style in case of handwritten characters. Also, while pre-processing, the character attributes change. As a result, the samples of the same character may show variations in the structural features discussed above. For example, the vertical line in the character may be so short that it may get classified to No-line class. If the character is not written properly, it may not contain the enclosed region and so on. This results in the misclassification of the characters to another class. This has an effect on the recognition accuracy. The misclassification rate can be reduced by applying a fuzzy based pre-classifier is implemented in this system [19] which is optimized further in the proposed system. This optimization is done for the End-line character class, since characters falling in this class give lower recognition rate as compared to other classes. The optimization is explained in the next part of the paper.

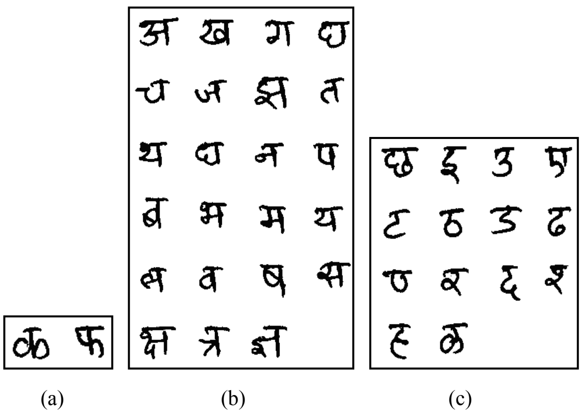

Fig. 1. Devanagari characters used in the proposed system.

III. PROPOSED SYSTEM DESIGN

The proposed system is designed to recognize 39 unconstrained Devanagari characters indicated in the previous section. The flow of the proposed system is shown in Fig. 2. The handwritten Devanagari character images are scanned in bmp file format at 300 dpi using a flatbed scanner. These images in grayscale are pre-processed to remove noise and convert into binary images. The binary images of the characters are segmented and cropped to be stored in the database. The database is created by classifying the characters based upon the structural features like presence of vertical line, enclosed region in the character and position of the endpoints.

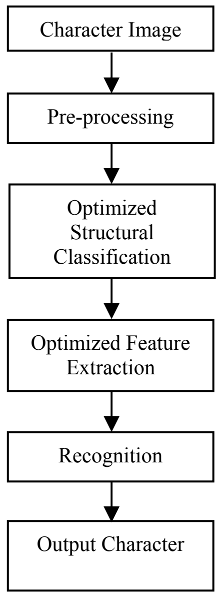

Fig. 2. Proposed system design.

The recognition of the characters is done using neural network. At first, the neural network is trained using optimized pixel density feature extraction technique. Finally, the neural network is tested by applying the similar features from the test characters. Here, three different neural networks are implemented and their performance is analyzed. The neural network output corresponds to the recognized character which is then displayed in text format. The next section explains the optimized structural classification.

IV. OPTIMIZED STRUCTURAL CLASSIFICATION

The number of characters in Devanagari script is large the handwritten characters show large amount of shape variations. Thus efforts to enhance the recognition rate of the handwritten Devanagari characters are taken by pre-classifying them into several classes based upon the structural features [19]. A two-stage classification is implemented for the same using various structural features. At first-stage, the characters are classified based upon the vertical line and enclosed region into 6 classes. If the character is a mid-line character as shown in Fig. 1a), it is assigned to the class 'Mid-line-enclosed' and 'Mid-line-not-enclosed' based upon the presence and absence of the enclosed region in it respectively. Similarly, the other classes are ‘End-line-enclosed’, ‘End-line-not-enclosed’, ‘No-line-enclosed’ and ‘No-line-not-enclosed’. This classification is indicated in Table I.

TABLE I. STRUCTURAL CLASSIFICATION USING VERTICAL LINE AND ENCLOSED REGION

Class	Mid line	End line	Enclosed region
No-line-not-enlcosed	Not found	Not found	Not found
No-line-enclosed	Not found	Not found	Found
End-line-not-enclosed	Not found	Found	Not found
End-line-enclosed	Not found	Found	Found
Mid-line-not-enclosed	Found	Not found	Not found
Mid-line-enclosed	Found	Not found	Found

In the next stage these 6 classes are further classified based upon the location of end-points. Each character is divided into four quadrants (Fig. 3) and lower two quadrants are used for classification of the character into further 4 classes. The end-points are detected using hit-and-miss algorithm [20] with eight directional structuring elements. The lower two quadrants, i.e. quadrant 2 and quadrant are considered. If there are end points these quadrants, then the characters are divided into four classes 1, 2, 3 and 4. Here class 1 means, there is no end point in quadrant 2 and 3; class 2 has an endpoint in quadrant 2 and no endpoint in quadrant 3; class 3 has the end point in quadrant 3 and so on. After two-stage structural classification, finally 24 classes are obtained. With this method, the character in Fig. 3 is classified to ‘No-line-not-enclosed-4’ class, since there are endpoints in both lower quadrants 2 and 3 respectively.

Altogether 32 classes are thus obtained, 16 for ‘End-line-enclosed’ class and 16 for ‘End-line-not-enclosed’ class. Fig. 4 gives an example of a character classified to ‘End-line-not-enclosed-16 class, which means it has a vertical line towards on the right hand side, with no enclosed region and has end points in all the four quadrants. After two-stage structural classification, total number of classes obtained is 48. A neural network is built for each class out of 48 and the characters are trained and tested.

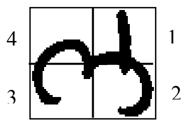

Fig. 3. Partitioning for end-point detection.

However, it was observed that the recognition rate of ‘End-line-characters’ is much less as compared to ‘Mid-line’ and ‘No-line’ characters. The recognition rate of ‘Mid-line’ and ‘No-line’ classes is upto 99% whereas for End-line-enclosed and End-line-not-enclosed, it falls below 97%. This needs some optimization at this level. Hence instead of considering only lower two quadrants, the second-stage classification for the characters in ‘End-line’ class is done using all four quadrants. The classes then obtained are indicated in Table II.

TABLE II. STRUCTURAL CLASSIFICATION USING END POINTS IN ALL QUADRANTS

Class	*End point in quadrant 4*	*End point in quadrant 3*	*End point in quadrant 2*	*End point in quadrant 1*
1	Not found	Not found	Not found	Not found
2	Not found	Not found	Not found	Found
3	Not found	Not found	Found	Not found
. . .	. . .	. . .	. . .	. . .
16	Found	Found	Found	Found

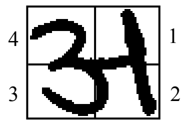

Fig. 4. Character assigned to class ‘End-line-not-enclosed-16.

V. OPTIMIZED FEATURE EXTRACTION

The concentration of the pixels in the character is commonly used for extracting features [19]. Here, prior to extracting the shape features, the cropped character is divided into non-overlapping zones after resizing the image. In every zone, the average of the pixels is calculated. The zonal average features for each zone are calculated as,

$$Z(m,n) = \frac{\sum_{i=1}^{10}\sum_{j=1}^{10} f(i,j)}{100} \qquad (1)$$

Considering, image is resized to 90 x 60, and then 9 x 5 = 54 zonal average features are obtained. Along with these features, average of horizontal projection profile and average of vertical projection profiles for the resized character is obtained. These average projection profiles are given by,

$$H(x) = \frac{\sum_{x} g(x,y)}{x} \qquad (2)$$

$$V(y) = \frac{\sum_{y} g(x,y)}{y} \qquad (3)$$

respectively. The $H(x)$ and $V(y)$ features are appended with the zonal average features Z (m, n). In case of 90x60 character image, the $H(x)$ and $V(y)$ features are of size 9 and 6 respectively. Thus the optimized feature vector is of size 69.

These features are further applied to neural network for recognition.

VI. RECOGNITION

The features are finally applied to neural network. This is done in two phases, training and testing. In the training phase, the network is trained to recognize the given test character. Various neural networks [21] implemented in this system are given further.

A. Feed-forward back-propagation network (FFBPN)

A FFBPN is a layered network with input layer, hidden layer and output layer. The features are applied to hidden neurons through input layer. The neurons in hidden layer and output layer are connected to each other using weighted connections. Generally, the neurons in the hidden layer have a nonlinear activation function and neurons in output layer have a linear activation function. The neurons in the hidden layer are trained through the training samples, to obtain a pre-defined output. This is done with the help of a training algorithm that updates the weights of the network to obtain the pre-defined output.

B. Cascade-forward back-propagation network (CFBPN)

A CFBPN is a variation of the FFBPN network. It has additional connections from the input layer to every layer. It also has connections from each layer to all following layers.

C. Elman back-propagation network (EBPN)

EBPN [22] generally has two layers with recurrent connection. The first layer input receives a feedback from the first layer output to the first-layer input. This recurrent nature of EBPN produces different outputs even when identical inputs are applied with the same weights and biases. EBPN can also be trained by giving training patterns. At every iteration, the input features are given to the network. The error between the output and the target is calculated and is back-propagated to find gradients of errors for every weight and bias. The weights are updates using the gradient.

VII. RESULTS AND DISCUSSION

The proposed handwritten Devanagari character system is developed by collecting over 40,000 samples in the database. The database is divided into 60% samples for training, 20% samples for validation and 20% samples for testing.

At first, the characters are passed through two stage structural classification that categorizes the characters into one of 48 classes. Here the 'End-line' character is classified to one of the 32 classes, 'Mid-line' character is classified to one of 8 classes and 'No-line' character is classified to one of 8 classes, depending upon the structural features; vertical line, enclosed region and end-points. By increasing the number of classes for 'End-line' characters, we are reducing the number of characters in each class thus ensuring increased recognition accuracy and fast response time. This also reduces the computational complexity of each network, since the number of neurons in the hidden layer depends on the number of outputs in the respective class. Every structural class has its own neural network. Thus 48 neural networks are built altogether.

The feature extraction technique is optimized by adding the average projection profiles, thus including the details of the character shape and position of those details. Here 69 features are derived. Thus the number of inputs to each neural network is 69 and the number of outputs is equal to the number of characters to be classified in each structural class. The neural networks implemented here were discussed in the previous section. The network parameters chosen for training are common to all the three types.

Table IV gives the recognition rates for all the three neural networks namely, FFBPN, CFBPN and EBPN. It also gives a comparison of the recognition rate obtained before optimization and after optimization for all the three networks adopted in this system.

TABLE III. COMPARISON OF NEURAL NETWORK PERFOMANCE

Neural Network	Before optimization	After optimization
	Recognition Rate (%)	*Recognition Rate (%)*
FFBPN	96.95	97.20
CFBPN	97.32	97.46
EBPN	97.60	98.10

The results indicate that optimization at structural classification stage and feature extraction stage increases the recognition rate. Also EPBN gives highest recognition rate in both the scenarios. Each character is recognized in 0.25 sec approximately on an Intel Core i3 with 2 GB RAM. The recognized character is displayed in text file using Kiran font.

VIII. CONCLUSION

This paper proposes a novel approach for improving the recognition rate of Devanagari characters by incorporating some optimization at each level of this pattern recognition application. Here, increasing the number of structural classes for complex characters helps in improving the performance of the system with respect to accuracy and time both. Also, adding shape details of the characters during feature extraction enhances the recognition rate. Further, neural networks other than Feed-forward back-propagation, like Cascade-forward back-propagation and Elman back-propagation are applied and analyzed. The results indicate that the optimization improves the recognition rate and the Elman back-propagation network gives the highest recognition rate amongst all three networks.

Acknowledgment

The authors are thankful to the unknown reviewers and they acknowledge useful feedback received from the reviewers towards improving this paper. The authors are also thankful to the conference organizers for providing a platform to publish this research work.

References

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

[2] V. Bansal, and R.M.K. Sinha, “Partitioning and searching dictionary for correction of optically read Devanagari character strings,” 5th International Conference on Document Analysis and Recognition, 1999, pp. 653-656.

[3] V. Bansal, and R. M. K. Sinha, “On how to describe shapes of Devanagari characters and use them for recognition”, 5th International Conference on Document Analysis and Recognition, 1999, pp. 410-413.

[4] V. Bansal, “Integrating knowledge sources in Devanagari text recognition”, Ph.D. Thesis, IIT Kharagpur, 1999.

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

[6] U. Pal, N. Sharma, T. Wakabayashi, and F. Kimura, “Off-line handwritten character recognition of Devanagari script,” 9th International Conference on Document Analysis and Recognition, 2007, pp. 496-500.

[7] 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.

[8] 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.

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

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

[11] S. Arora, D. Bhattacharjee, M. Nasipuri, D. Basu and M. Kunda, “Combining multiple feature extraction techniques for handwritten Devanagari character recognition,” IEEE Region Colloquium and the Third ICIIS, 2008, pp. 1-6.

[12] 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.

[13] U. Bhattacharya and B. B. Chaudhuri, “Handwritten numeral databases of Indian scripts and multistage recognition of mixed numerals,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 3, pp. 444–457, 2009.

[14] S. Kompalli, S. Nayak, S. Setlur, and V. Govindaraju, “Challenges in OCR of Devanagari documents,” in Proc. 8th Conf. Document Anal. Recognit., 2005, pp. 1–5.

[15] C. V. Jawahar, P. Kumar, and S. S. R. Kiran, “Bilingual OCR for Hindi-Telugu documents and its applications,” in Proc. 7th Conf. Document Anal. Recognit., 2003, pp. 1–5.

[16] S. Arora, D. Bhattacharjee, M. Nasipuri, D. K. Basu, and M. Kundu, “Recognition of non-compound handwritten Devnagari characters using a combination of MLP and minimum edit distance,” Int. J. Comput. Sci. Security, vol. 4, no. 1, pp. 1–14, 2010.

[17] S. Arora, D. Bhattacharjee, M. Nasipuri, D. K. Basu, M. Kundu, and L. Malik, “Study of different features on handwritten Devnagari character,” in Proc. 2nd Emerging Trends Eng. Technol., 2009, pp. 929–933.

[18] S. Shelke, and 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.

[19] S. Shelke, and S. Apte, “A Fuzzy based Classification Scheme for Unconstrained Handwritten Devanagari Character Recognition,” International Conference on Communication, Information and Computing Technology, 2015, pp. 1-5.

[20] R. C. Gonzalez, and R. E. Woods and S.L Eddins, Digital Image Processing using MATLAB, 2nd ed., McGraw Hill: New Delhi, 2010.

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

[22] J. L. Elman, “Finding structure in time,” Cognitive Science, vol. 14, pp. 179-211, 1990.