A Multistage Handwritten Marathi Compound Character Recognition Scheme using Neural Networks and Wavelet Features

Sushama Shelke
Dept. of E & TC,
Pune Institute of Computer Tech.,
Pune-411 043, Maharashtra, India
Email: sush_aditi11@yahoo.com

Shaila Apte
Dept. of E & TC,
Rajarshi Shahu College of Engg.,
Pune-411 033, Maharashtra, India
Email: sdapte@rediffmail.com

Abstract

Compound characters which are one of the features of Marathi script, derived from Devanagari, occur frequently in the script. Recognition of these characters poses challenges to the researchers due to their complex structure. This paper presents a novel approach for recognition of unconstrained handwritten Marathi compound characters. The recognition is carried out using multistage feature extraction and classification scheme. The initial stages of feature extraction are based upon the structural features and the classification of the characters is done according to the structural parameters into 24 classes. The final stage of feature extraction employs wavelet transform. Single level wavelet decomposition is used to generate the approximation coefficients which are used as features. These coefficients are further modified and then used as another set of features. Both the wavelet approximation features as well as the modified wavelet features are applied to neural network for recognition. A separate neural network block is built for each of the 24 classes. The average recognition rate is found to be 96.14% and 94.22% respectively for training and testing samples with wavelet approximation features and 98.68% and 96.23% respectively for training and testing samples with modified wavelet features.

Keywords: *Compound characters, structural classification, wavelet decomposition, neural network*

1. Introduction

Handwritten character recognition, irrespective of the script, finds potential application areas for automation in various fields like postal automation [1, 2], bank automation [3, 4], form filling etc. Handwritten character recognition for Indian scripts [5] is quite a challenging task for the researchers. This is due to the various characteristics of these scripts like their large character set, complex shape, presence of modifiers, presence of compound characters and similarity between characters. Marathi script derived from Devanagari, is an official language of Maharashtra. It is the 4th most spoken language in India and 15th most spoken language in the world. Marathi script consists of 16 vowels and 36 consonants making 52 alphabets. Marathi is written from left to right. It has no upper and lower case characters. Every character has a horizontal line at the top called as the header line. The header line joins the characters in a word. Vowels are combined with consonants with the help of specific characteristic marks. These marks occur in line, at the top, or at the bottom of a character in a word. Marathi also has a complex system of compound characters in which two or more consonants are combined forming a new special symbol. Compound characters in Marathi script occur more frequently in the script as compared to other

languages derived from Devanagari. The occurrence of compound characters in Marathi is found to be about 11 to 12% whereas in other scripts of Devanagari and Bangla script, it is just 5 to 7% [6]. The compound characters exhibit following features: the consonants in the compound character are not joined in an arbitrary manner but the combination of some specific characters is done in order to give a meaningful combination.

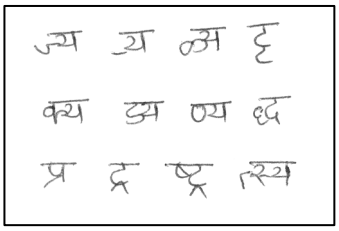

Figure 1. A small subset of Compound characters in Marathi script

The compound character can have two or more characters joined together in various ways as shown in Figure 1. One way of forming compound character is by removing the vertical line of a character and then joining it to the other on its left hand side. This type of joining is more common. Another way of connection of characters in the compound character is by just joining the characters side by side or one above the other. More than two consonants also join in various ways to form a compound character. In another way, one of the characters completely changes its form and then gets connected to the other to form a compound character. As seen from Figure 1, the compound characters not only exhibit a variation in the shape of the character but also in the aspect ratio as per the joining strategy. One might get tempted to use the features like aspect ratio or number of end points, but the various joining strategies limits the use of these features to achieve acceptable recognition accuracy. All these challenges cannot be met by just a single feature extractor or a single classifier. Moreover, a feature extractor and classifier combination may recognize a character which may not be recognized by the other feature extractor and classifier combination. Hence a multistage system is needed that can recognize the characters over a wide range of varying conditions.

In this paper, we propose a system for unconstrained handwritten Marathi compound character recognition without separation of the characters in the compound character. No work for handwritten compound character recognition is found so far to the best of our knowledge. OCR work for printed and handwritten characters in various Indian scripts [7-9] is carried out by researchers but major work is found for Bangla [10, 11] and Devanagari. Work on printed Devanagari script started in early 1970s. An extensive research on printed Devanagari text was carried out by Veena Bansal and R. M. K. Sinha [12, 13]. First system for hand-written numeral recognition of Devanagari characters was proposed by R. Bajaj et al. [14]. P. M. Patil and T.R. Sontakke [15] also presented an algorithm for handwritten Devanagari numeral recognition which was rotation, scale and translation invariant. U. Pal et al. [16] presented a system for off-line handwritten character recognition of Devanagari using directional information for extracting features. Multilayer perceptron was also used for classification by Sandhya Arora et al. [17] for handwritten Devanagari characters along with different features

separately. The final classification result was obtained by a decision algorithm based on weighted majority voting technique. In [18], U. Pal et al combined two classifiers to get higher accuracy of Devanagari character recognition with the same features. Combined use of SVM and Modified Quadratic Discriminant Function (MQDF) are applied for better performance of Devanagari character recognition. Recently, a comparative study of various features and classifiers used for handwritten Devanagari character recognition was done by U. Pal et al. [19]. They found that Mirror Image classifier was the best classifier. Recognition of handwritten Bangla compound character was attempted by U. Pal et al. [20] using gradient features. The rest of the paper is organized as follows: Section 2 discusses the wavelet transform, while section 3 describes the neural network. Section 4 presents the proposed system. Section 5 discusses the results obtained and section 6 finally discusses the conclusion.

2. Wavelet transform

The wavelet transform exhibits the features like separability, scalability, translatability, orthogonality and multiresolution capability. The discrete wavelet transform [21] of an image *f(x,y)* of size *M*x*N* is

$$W_{\varphi}(j_0,m,n)=\frac{1}{\sqrt{MN}}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)\varphi_{j_0,m,n}(x,y) \tag{1}$$

$$W_{\psi}^{i}(j,m,n)=\frac{1}{\sqrt{MN}}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)\psi_{j,m,n}^{i}(x,y) \tag{2}$$

where,

$$\varphi_{j,m,n}(x,y)=2^{j/2}\varphi(2^{j}x-m,2^{j}y-n) \tag{3}$$

and

$$\psi_{j,m,n}^{i}(x,y)=2^{j/2}\psi^{i}(2^{j}x-m,2^{j}y-n) \tag{4}$$

are the two dimensional scaling and wavelet functions respectively and the index *i* identifies the directional wavelets that takes the values H, V and D i.e. horizontal, vertical and diagonal details respectively. j_0 an arbitrary starting scale and the $W_{\varphi}(j_0,m,n)$ coefficients define an approximation of *f(x, y)* at scale j_0. The $W_{\psi}^{i}(j,m,n)$ coefficients add horizontal, vertical and diagonal details for scales $j \geq j_0$. Normally, $j_0 = 0$ $N = M = 2^J$ so that $j = 0, 1, 2\ldots, J\text{-}1$ and $m, n = 0, 1, 2,\ldots, 2^J\text{-}1$. The discrete wavelet transform can be implemented using digital filters and down samplers. The high pass or detail component characterizes the image's high-frequency information with vertical orientation; the low-pass, approximation component contains its low-frequency, vertical information. Both sub images are then filtered column wise and down sampled to yield four quarter size output images. These sub images are shown in Figure 2.

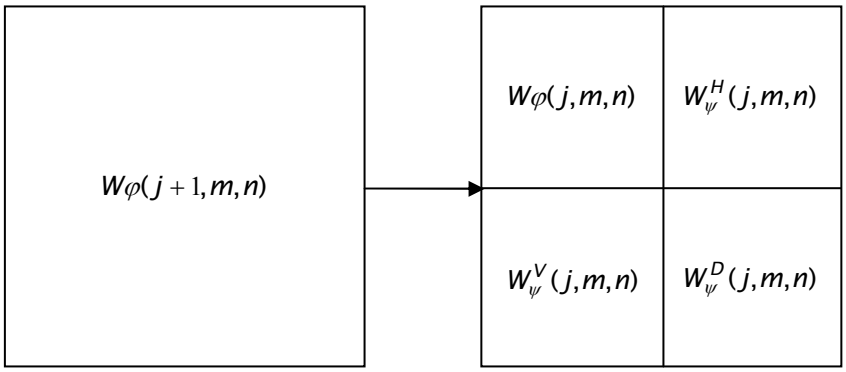

Figure 2. Single level wavelet decomposition

3. Neural networks

Artificial neural networks are one of the popular techniques used for classification due to their learning and generalization abilities. They have been traditionally used for character recognition application. Out of various architectures, multilayer perceptron (MLP) is widely used. The MLP is a fully connected network, where every neuron in a layer is connected to each and every neuron in the next layer by a weighted link through which the state of the neuron is transmitted. It consists of an input layer, a hidden layer and an output layer. Such a network is shown in Figure 3. The feature vector is applied as the input signal to the neurons in the hidden layer from the input layer. 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.

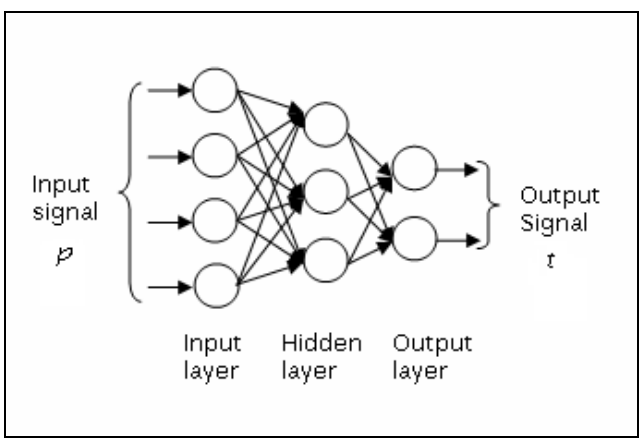

Figure 3. Two layer feed forward neural network

This operation is described by [22]

$$a^{m+1} = f^{m+1}(W^{m+1}a^{m} + b^{m+1}), \tag{5}$$

for $m = 0, 1,\ldots, M$-1, where M is the number of layers in the network. The neurons in the first layer receive external inputs:

$$a^0 = p, \tag{6}$$

which provides the starting point for the network. The outputs of the neurons in the last layer are considered the network outputs:

$$a^M = t \tag{7}$$

Once the network weights and biases are initialized, the network is ready for training or learning. 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 *p* and target outputs *t*. As 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 for a particular application.

4. The proposed scheme

The proposed system is designed to recognize 70 characters including 40 compound characters and 30 split characters. The compound characters used in the proposed system are shown in Figure 4. The split characters are the characters in the compound character which may be split into two separate characters due to writing style or due to pre-processing operation. These characters are shown in Figure 5. The flow of the proposed scheme is depicted in Figure 6. It consists of training phase and testing phase. At first, the characters are pre-classified based upon their structural features. A two stage structural classification technique is implemented in both the phases.

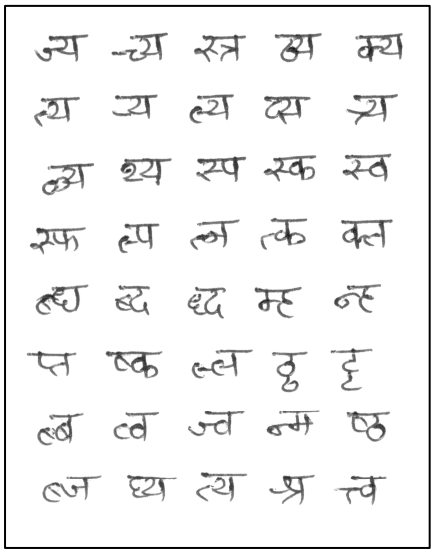

Figure 4. Compound characters used in the proposed system

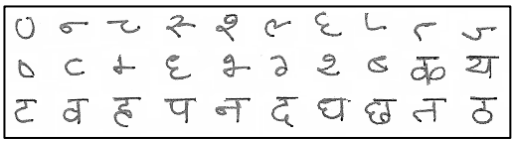

Figure 5. Split characters

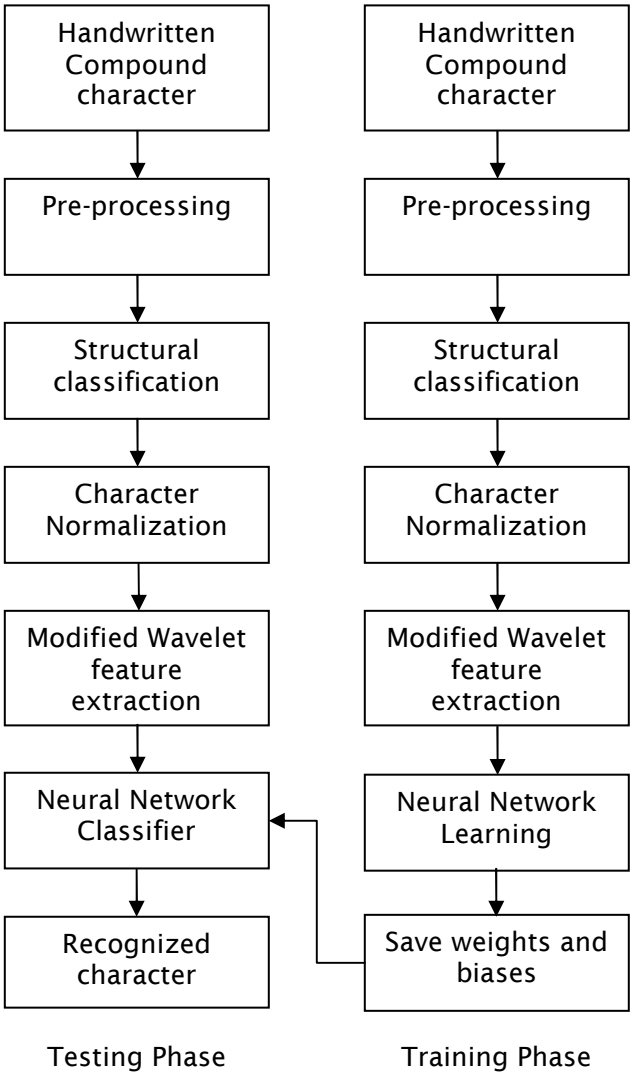

Figure 6. Proposed system design

In the training phase, modified wavelet featured are extracted from the normalized characters and used for training the neural network so as to fix the weights and biases for each character, after applying sufficient samples per character. In the testing phase, similar features are extracted from the character. This is done again after pre-processing and structural classification of the character. The features are applied as inputs to the neural network. The

output of the neural network yields the final recognition result. The next section discusses the proposed system in detail.

4.1. Data collection

The database for handwritten compound characters is created by scanning the characters at 300 dpi using a flatbed scanner. The images are stored in bmp file format. Here we assume that the consonants in the compound characters are either touching or overlapping. However, sometimes they may not be touching or overlapping due to the writing style or may result into a gap or separate after binarization and pre-processing operations. These split components of a compound character result into separate entities after segmentation. In order to take these separate entities into account, we also consider the split components of the compound character for recognition. More than 500 samples per character are scanned resulting into about 35000 characters in the database. No standard database for compound characters is available.

4.2. Pre-processing

A point operator converts the gray scale character images to binary. This operator separates pixels that have values within specified range i.e. the object from the rest or the background. This is done by choosing a threshold that separates the object and the background. Here, the threshold is chosen by using uniform thresholding after normalization. In uniform thresholding, pixels above a threshold are set to white and those below the threshold are set to black. Uniform thresholding requires the knowledge of the gray levels otherwise the target features might not get selected or may get misclassified after the thresholding process. The handwritten characters were tested and checked for the global features for various threshold values before finalization of a threshold. On testing about one third of the characters in the database, the normalized threshold value of 0.85 was found to be an optimum value that gave correct feature selection of global features in most of the cases. Pre-processing plays an important role in handwritten character recognition as in any other pattern recognition task. Handwritten characters show various undesirable effects like unwanted strokes, gaps or breaks which occur due to binarization. Many a times when a character is handwritten, it exhibits lesser width at the curvature than at other parts of the character. This point is more likely to break during binarization. Hence, a 3x3 averaging operator is implemented before binarization, which blurs the image resulting into bridging small gaps and retaining the actual shape of the character. The unwanted strokes occur more often between the pen lifting and placing points and their occurrence depend upon the writing style and the ink viscosity. These strokes may result into unwanted feature detection after binarization. In order to avoid this, the binarized image should be cleaned. This is done by using morphological opening operator. Morphological opening removes thin protrusions, breaks thin connections and smoothes the object contour. The morphological opening of image I by structuring element B is simply erosion of I by B followed by dilation of the result by B as indicated in equation 8. Here the structuring element B used removes all objects smaller than 40 pixels using 8 – connectivity.

$$I \circ B = (I \Theta B) \oplus B \tag{8}$$

where, $I \circ B$ indicates the opening of the image I with the structuring element B.

In the proposed work, compound characters are written on plain paper. The lines and characters are segmented and used for database creation as well as testing. The lines and characters are written in such a way that they do not overlap. The lines and characters are segmented using horizontal and vertical projection profiles. Peaks of the projection profiles separate the lines and the characters in the document. The number of lines and number of characters in each line are separated.

4.3. Structural classification

The large number of compound character set with a wide range of variations in the writing style demands a pre-classification of the characters before the final recognition. The pre-classification is done using a two stage classification based upon the structural features. The first stage employs classification using global features like presence of vertical line in the character, its position in the character and the presence of holes. These features can be termed as global features. The detection of global features is followed by the detection of the local features. The detection of these features is explained in the next section.

4.3.1. Detection of vertical bar and its position: At first, a character is segmented and preprocessed to remove the unwanted gaps and strokes. Further, the character is binarized using adaptive thresholding. The binarized character is then cropped. Vertical projection profile of the cropped image $I(m,n)$ is further calculated in order to find the column with maximum number of pixels n_{max}. An average height of the vertical bar is considered 85 percent of the total height of the image. This value is set as a threshold T_V to find the presence of a vertical bar in a character. Thus if,

$$\begin{aligned} n_{max} &\geq T_V \quad \text{vertical bar = present} \\ &< T_V \quad \text{vertical bar = absent} \end{aligned} \tag{9}$$

If the vertical bar is detected, further its location is found to further classify the character as per its location within the character. The vertical line in the character can be in middle or towards the end. Again an average threshold T_{mid} is set to be 30 percent for the position of the vertical bar in the character. If,

$$\begin{aligned} T &\geq T_{mid} \quad \text{mid bar = present} \\ &< T_{mid} \quad \text{end bar = present} \end{aligned} \tag{10}$$

where, $$T = ((n-n_{max}) / n) \times 100. \tag{11}$$

4.3.2. Enclosed region detection: Further, the detection of enclosed region is carried out. Here, 8-adjacency is used to find the presence of connected components or enclosed regions. Two foreground pixels p and q are said to be connected if there exists an 8-connected path between them, consisting entirely of foreground pixels. The decisions taken so far are based upon the global features, which are the presence of the vertical line and the loops in the character. The decisions result into six classes. These six classes are namely EBE, EBNE, MBE, MBNE, NBE, NBNE. When the vertical line is towards the end, the classes obtained are EBE (end-bar enclosed) and EBNE (end-bar

not enclosed), with presence and absence of closed loop respectively. When the vertical bar is at the center, two classes are again derived based on the presence of loop namely, MBE (mid-bar enclosed) and MBNE (mid-bar not enclosed). Finally, the two classes namely NBE (no-bar enclosed) and NBNE (no-bar not enclosed) do not have a vertical line. Table 1 indicates the first stage structural classification based upon the global features as discussed in this section.

Table 1. Structural classification based on detection of vertical line, its position and enclosed region detection

Class	Mid bar	End bar	Enclosed region
No-bar enclosed (NBE)	Absent	Absent	Present
No-bar not enclosed (NBNE)	Absent	Absent	Absent
Mid-bar enclosed (MBE)	Present	Present	Present
Mid-bar not enclosed (MBNE)	Present	Absent	Absent
End-bar enclosed (EBE)	Absent	Present	Present
End-bar not enclosed (EBNE)	Absent	Present	Absent

4.3.3. End point position detection: The second stage structural classification is based on the local features. The local feature used here is the presence of end points. However we don't go for finding number of endpoints in the character but we find the position of the endpoints. The detection of end points is done in two steps as follows:

- Partitioning character into 4 quadrants
- Detection of endpoint in 3rd and 4th quadrants

The character obtained after global feature-based first stage structural classification is further passed to hit-or-miss operator to find the endpoints in the character. 8-directional structuring elements are used for finding endpoints in all the directions. The structuring elements used for end point detection are shown in Figure 7.

```
-1  1 -1    -1 -1  1    -1 -1 -1    -1 -1 -1
-1  1 -1    -1  1 -1    -1  1  1    -1  1 -1
-1 -1 -1    -1 -1 -1    -1 -1 -1    -1 -1  1


-1 -1 -1    -1 -1 -1    -1 -1 -1     1 -1 -1
-1  1 -1    -1  1 -1     1  1 -1    -1  1 -1
-1  1 -1     1 -1 -1    -1 -1 -1    -1 -1 -1
```

Figure 7. Structural elements for end point detection

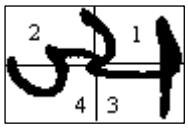

Figure 8. Partitioning for end point detection

Table 2. Structural classification based on end point position detection

Class	End point in quadrant 4	End point in quadrant 3
1	Absent	Absent
2	Absent	Present
3	Present	Absent
4	Present	Present

The character is partitioned into four quadrants as shown in Figure 8. Only end points in lower two quadrants are used for further classification. Each class in the previous step is sub divided into four classes as per the presence of the end points in the third and fourth quadrants as indicated in Table 2. If there are no endpoints in both the lower quadrants, i.e. quadrant 3 and quadrant 4, the character is assigned to class 1. If the end point is detected in quadrant 3 only, the character is classified to class 2. If the endpoint is found in quadrant 4 only, the character is classified to class 3 and if endpoints are in both lower quadrants, the character is assigned to class 4. Thus, the character in Figure 8, after decision tree classification is classified to EBNE-2, which indicates that there is a vertical line towards the end, there is no enclosed region in the character and the endpoint/endpoints is/are found in the third quadrant. The structural classification ends here resulting into 24 classes. The training phase also employs similar classification procedure. The character used for testing is first pre-classified as discussed and further features are extracted and matching is done between the test character and the characters in the class out of 24 classes, to which it is assigned after structural classification.

4.4. Feature extraction

After two stage structural classification, the character is resized to a fixed size of 16x16. Wavelet transform is used for feature extraction. Features are extracted using single level wavelet decomposition as discusses earlier. The approximation coefficients obtained for every character after single level decomposition is considered. The single level decomposition leads to 8x8 approximation features. The modified wavelet features are also generated in order to improve the recognition results. The modified wavelet features are obtained by convolving the approximation features with themselves. Convolution between a 2D signal *f(x, y)* and a template *g(x, y)* is defined as

$$f * g = \sum_{(x,y)\in T} f(x',y')g(i-x',j-y') \tag{12}$$

where, $x' = x + i$ and $y' = y + j$.

Convolution operation generates an output of size 15x15. This is down sampled to a feature vector of 75 features. Both the features, wavelet approximation as well as modified approximation wavelet features are applied as the input to the neural network. The convolution operation in the modified wavelet feature generation procedure increases the dynamic range of the signal over a large extent. This leads to generation of a unique feature for each character thereby increasing the inter class and intra class distance between the characters.

4.5. Recognition

A separate MLP for each of the 24 classes is built. When a character is being tested, its structural class out of 24 classes is obtained first and the MLP of that class is utilized for the final recognition. The neural network implemented is a two-layer network with hyperbolic tangent sigmoid transfer function for the hidden layer and linear activation function for the output layer. The number of inputs equal to the features derived in the previous section. The number of outputs varies as per the number of characters in the particular class obtained after structural classification. The number of neurons in the hidden layer is taken to be equal to the square root of the product of the inputs and the outputs of the network. The weights and biases are updated using Levenberg-Marquardt algorithm.

5. Results and Discussion

The handwritten Marathi compound character dataset is collected from different individuals. About 500 samples per character were collected resulting in about more than 35000 character samples. Out of these, two third of the characters were used for training and rest were used for testing. No standard database is available for handwritten Devanagari compound characters so far. At first, the presence of vertical line, its position and enclosed region are detected and the character is classified into one of the six classes as shown in the Table 1. Further, the six classes obtained are classified into four classes each based on local features as indicated in Table 2. Thus there are 24 classes at the output of the second stage of structural classification. 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 structural classes. The characters are then normalized to a fixed size of 16x16 after structural classification for feature extraction. A single level wavelet decomposition of the resized character image generates the approximation and the detail coefficients. The decomposition is done with respect to Daubechies wavelet. Here, $j_0 = 0$ and $M = N = 16$ which is equal to 2^j, hence j=0, 1, 2,…, 7 and m, n=0, 1, 2,…, 15. 64 wavelet approximation features as well as 75 modified wavelet features are used to train the neural network. The inputs to the neural networks in all the 24 classes equal to the number of features derived, but the number of hidden neurons and the outputs depend upon the characters in that class. The performance goal is kept to 0.001 and 300 epochs are sufficient to train the network using Levenberg-Marquardt algorithm. The Levenberg-Marquardt algorithm is the fastest back propagation algorithm but it requires more memory as compared to other algorithms. This put a limit to the size of the input

vector. Resizing the character to dimensions greater than that of 16x16, may improve the result but it leads to memory shortage during neural network training. The time required for training the neural networks in each of the 24 classes depends upon the number of characters and the number of samples per character in that class. The testing time is the same irrespective of the structural class.

The proposed system is implemented using MATLAB R2008b on Intel Core 2 Duo CPU, running on 2GHz with 2 GB RAM. The time required to test a character is approximately 0.045 seconds. Table 3 indicates the recognition results for both the feature extraction techniques. The results show that the modified wavelet features improve the results considerably for both the training as well as testing samples.

Table 3. Recognition results

Feature Extraction Technique	Recognition rate	
	Training samples	Testing samples
Wavelet features	96.14%	94.22%
Modified wavelet features	98.68%	96.23%

6. Conclusion

A new method for unconstrained handwritten Marathi compound character recognition using various features and neural network classifier is proposed in this paper. Prior to feature extraction, the characters are pre-classified into 24 classes using structural features. The wavelet approximation features and modified wavelet features are extracted from normalized character and then applied to train the neural network built for each structural class. Testing follows the similar feature extraction procedure and the neural network for the particular structural class to which the character is classified, is used for final recognition result. In order to reduce the training time, a faster training algorithm, named Levenberg-Marquardt is used. The modified wavelet feature generation method increases the range of the signal to a considerable amount, generating a unique feature for every character, thereby increasing the recognition accuracy.

References

[1] K. Roy, S. Vaidya, U. Pal, and B. B. Chaudhuri, A. Belaid, "A System for Indian Postal Automation", Proc. 8th Int. Conf. Document Analysis and Recognition, Seoul, Korea, Aug. 31-Sep. 1, 2005, pp.1060-1064.
[2] U. Pal, R. K. Roy, K. Roy, and F. Kimura, "Indian Multi Script Full Pin-code String Recognition for Postal Automation", Proc. 10th Int. Conf. Document Analysis and Recognition, Barcelona, Spain, Jul. 26-29, 2009, pp. 456-460.
[3] B. B. Chaudhari, Digital Document Processing - Major Directions and Recent Advances, Springer, London, 2007.
[4] Simone Marinai, and Hiromichi Fujisawa, Machine Learning in Document Analysis and Recognition, Springer-Verlag, Berlin Heidelberg, 2008.
[5] U. Pal, and B. B. Chaudhari, "Indian Script Character Recognition: a Survey", Pattern Recognition, vol. 37, 2004, pp. 1887-1899.
[6] Utpal Garain, and B. B. Chaudhuri, "Segmentation of Touching Characters in Printed Devanagari and Bangla Scripts using Fuzzy Multifactorial Analysis", IEEE Trans. Systems, Man and Cybernetics-Part C, Applications and Reviews, vol. 32, no. 4, 2002, pp. 449-459.
[7] K. Roy, T. Pal, U. Pal, and F. Kimura, "Oriya Handwritten Numeral Recognition System", Proc. 8th Int. Conf. Document Analysis and Recognition, vol. 2, Seoul, Korea, Aug. 31-Sep. 1, 2005, pp. 770 – 774.
[8] Arun Pujari, C. Dhanunjaya Naidu, M. Sreenivasa Rao, and B. C. Jinaga, "An Intelligent Character Recognizer for Telugu scripts using Multi resolution Analysis and Associative Memory", Image and Vision Computing, vol. 22, 2004, pp. 1221-1227.
[9] U. Bhattacharya, S. K. Ghosh, and S. K. Parui, "A Two Stage Recognition Scheme for Handwritten Tamil Characters", Proc. 9th Int. Conf. Document Analysis and Recognition", Parana, Sept. 23-26, 2007, pp. 511 – 515.
[10] B. B. Chaudhuri, and U. Pal, "A Complete Printed Bangla OCR System", Pattern Recognition, vol. 31, no. 5, 1998, pp. 531-549.
[11] A. F. R. Rahman, R. Rahman, and M. Fairhurst, "Recognition of Handwritten Bengali Characters: A Novel Approach", Pattern Recognition, vol. 35, 2002, pp. 997-1006.
[12] Veena Bansal, and R. M. K. Sinha, "On How to Describe Shapes of Devanagari Characters and Use them for Recognition", Proc. 5th Int. Conf. Document Analysis and Recognition, Bangalore, India, Sept. 20-22, 1999, pp. 410-413.
[13] Veena Bansal, "Integrating Knowledge Sources in Devanagari Text Recognition", Ph.D. Thesis, IIT Kharagpur, India, 1999.
[14] R. Bajaj, L. Dey, and S. Chaudhary, "Devanagari Numeral Recognition by Combining Decision of Multiple Connectionist Classifiers", Sadhana, 27, 2002, pp. 59-72.
[15] P. M. Patil, and T. R. Sontakke, "Rotation, Scale and Translation Invariant Handwritten Devanagari Numeral Character Recognition using General Fuzzy Neural Network", Pattern Recognition, vol. 40, 2007, pp. 2110-2117.
[16] U. Pal, N. Sharma, T. Wakabayashi, and F. Kimura, "Off-line Handwritten Character Recognition of Devanagari Script", Proc. 9th Int. Conf. Document Analysis and Recognition, vol. 1, Parana, Sept. 23-26, 2007, pp. 496-500.
[17] Sandhya Arora, Debotosh Bhattacharjee, Mita Nasipuri, Deepak Basu, and Mahantapa Kunda, "Combining Multiple Feature Extraction Techniques for Handwritten Devanagari Character Recognition", IEEE Region Colloquium and the Third ICIIS, Kharagpur, India, 2008, pp. 342-347.
[18] U. Pal, S. Chanda, T. Wakabayashi, and F. Kimura, "Accuracy Improvement of Devanagari Character Recognition Combining SVM and MQDF", 11th Int. Conf. Frontiers in Handwriting Recognition, Québec, Canada, Aug. 19-21, 2008, pp. 367-372.
[19] U. Pal, T. Wakabayashi, and F. Kimura, "Comparative Study of Devanagari Handwritten Character Recognition using Different Feature and Classifiers", Proc. 10th Int. Conf. Document Analysis and Recognition, Barcelona, Spain, Jul. 26-29, 2009, pp. 1111-1115.
[20] U. Pal, T. Wakabayashi, and F. Kimura, "Handwritten Bangla Compound Character Recognition using Gradient Feature", Proc. 10th Int. Conf. Information Technology, Orissa, India, Dec. 17-20,2007, pp. 208-213.
[21] R. Gonzalez, and R. Woods, Digital Image Processing, Second edition, Pearson Education, India, 2006.
[22] M. Hagan, H. Demuth, and M, Beale, Neural Network Design, Thomson Learning, India, 2003.

Authors

Sushama D. Shelke received her BE and ME degrees from Walchand College of Engineering, Sangli, India in 1999 ad 2001 respectively. Currently she is pursuing her PhD in the Electronics Department, Walchand College of Engineering under Shivaji University, Kolhapur. Presently she is working as an Assistant Professor in the Post Graduate Department of E & TC in Pune Institute of Computer Technology, Pune, India. She has published more than 25 papers in various National and International conferences. She was the Principal Investigator, of a research project funded by University of Pune entitled 'Document Image Analysis and Understanding' during the year 2008-2010. Her research interests include Image Processing, Pattern recognition, Signal Processing, and embedded system design.

Dr. Shaila D. Apte completed her ME and PhD (Electronics Engineering) from Walchand College of Engineering, Sangli. She has a professional experience of 30 years. She worked in Walchand College of Engineering for 27 years and currently working as a Professor in Post Graduate Department in Rajarshi Shahu College of Engineering, Pune, India. She has a patent to her credit related to generation of mother wavelet from speech signal. She has published 16 research papers in reputed International journals and over 30 papers in International conferences. She is PhD guide in Shivaji University, Kolahpur. She is in the reviewer panel for International Journal of Speech Technology, Springer Publication and International Journal on Signal Processing, Elsevier Publication. She is the author of the book entitled "Digital Signal Processing" published by Wiley India. Her research areas include Speaker verification, Speech recognition, Speech synthesis, Watermarking of speech and music signals, handwritten character recognition, etc.