International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 6 (2014) pp. 528-534

Structural and Statistical Feature Based Multistage Recognition Approach for Handwritten Devanagari Script Recognition

Vijaya Rahul Pawar[1], Arun Natha Gaikwad [2]

[1] Research Scholar, Department of Electronics Engineering, College of Engineering,
Bharati Vidyapeeth Deemed University, Pune, Maharashtra, India
pawar_vr123@yahoo.com

[2] Department of Electronics Engineering, College of Engineering,
Bharati Vidyapeeth Deemed University,
Pune, Maharashtra, India
arungkwd47@gmail.com

***Abstract*: Devanagari is one of the basic Script widely used all over the India . Many Indian Languages Like Hindi, Marathi, Rajasthani etc are based on Devanagari Script. Devanagari Scripts Hindi language is third common language used all over the word. In this paper we propose statistical and structural method based feature extraction and an artificial neural network based classifier for the recognition of handwritten Devanagari characters . Optical isolated Hindi Characters are taken as an input image from the scanner. An input image is preprocessed and segmented in terms of various structural and statistical features like end points, middle bar, loop, end bar, aspect ratio etc.The feature vector is applied to Self organizing map which is one of the classifier of an artificial neural Network. Self organizing map is trained for such 500 different characters collected from 500 persons. The characters are classified into three different classes. The proposed classifier attains 93% accuracy.**

***Keywords*:** *Image Pre-processing, Segmentation, Feature Extraction, Self Organizing Map, Network Neighborhood.*

I. Introduction

Handwriting is one of the most effective way by which civilized people speak. A person's handwriting is unique just like fingerprints. A script is defined as a graphical form of system. Six different scripts popularly used all over the world are [1] Arabic, Roman, Cyrillic, Han, Hebrew, and Devanagari. Different script may follow the same writing method. In the present work we are focusing on Offline Devanagari word recognition. Devanagari script consists of 49 characters and 10 numerals, out of 49 characters 13 are vowels and 36 are consonants. Devanagari Script consists of 5 upper modifiers and 3 lower modifiers with side modifiers. Vowels can be written as independent letters, or by using a variety of diacritical marks which are written above, below, before or after the consonant they belong to. When vowels are written like this way they are known as modifiers and the characters so formed are called conjuncts. Sometimes two or more consonants can combine and take new shape . These new shape clusters are known as compound characters. All the characters have a horizontal line at the upper part, known as Shirorekha or headline. In continuous handwriting, from left to right direction, the Shirorekha of one character joins with the Shirorekha of the previous or next of the same word. In this fashion, multiple characters and modified shapes in a word appear as a single connected component joined through the common Shirorekha. Due to all these complexities, handwritten character recognition is challenging problem. Artificial Neural Network are becoming wave of the future in computing and becoming powerful class of model.

II. Past Review

Recognition of isolated handwritten numeral recognition of Indian scripts is proposed by U.Bhattacharya [2].Three MLP classifiers corresponding to three coarse-to-fine resolution levels in a cascaded manner. If rejection occurs even at the highest resolution, another MLP is used as the final attempt to recognize the input numerals by combining the outputs of three classifiers .However the classification accuracy is poor for the mixed language numerals.

A system of complete handwriting recognition, which has been implemented and tested on the database of cursive script is proposed by A. Senior [3]. Recurrent error propagation network is used with Hidden Markov Model for the recognition of offline script which attains 87% accuracy. For better result context dependent model has been suggested.

A system of statistical character structure modeling is proposed by In-Kim [5]. In structure modeling each stroke is defined by distribution of feature point. The neighbor relationship and probability approximation method is used.

The accuracy is higher for Chinese characters with great computational complexity.

J.Park described [6] an adaptive approach to offline handwritten word recognition. Critical optimization issue is excluded and that may be considered as a limitation. Relatively high computational power and large storage are required to build a successful handwriting recognition system. Since resources are always limited, the trade-off between the desired recognition rate and required system resources is an optimization problem, and is an important factor to be consider, when handwriting recognition system is designed .In proposed system an artificial neural network can learn complex mappings from inputs to outputs based solely on samples.

III. Recognition of Handwritten Devanagari OCR System.

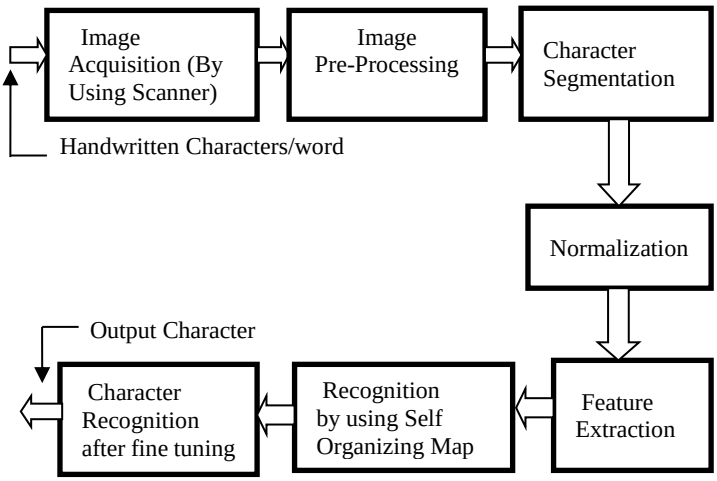

Figure 1. Recognition of handwritten Devanagari characters

Six major steps involved in recognition of handwritten characters are image acquisition (scanning), Image pre-processing, RGB to grey conversion, Segmentation, Feature extraction, Self organizing map based character classification, Post-processing.

Handwritten Devanagari words are captured from scanner and are converted into digital images. The scanner used here is 300 dpi flat bed scanners. The Devanagari words are taken from different 1000 persons which are size invariant .Preprocessing relates to image preparation for the later analysis and use. The main aim of preprocessing is to make the image ready for the feature extraction. Color image can be converted into grey scale image by using the formula [10]

GRAY=0.299*R+0.587*G+0.114*B

There are many methods to remove the noise from the image but Median filter with 3*3 matrix is used for noise removal [10] .It is the statistic filter. Scanning process introduces irregularities such as ‘’speckle noise”, “salt and pepper noise” in the digital image. Median Filtering has been employed, to remove such noise and its effects [19].

The original value of pixel is included in the computation of the median. Median filters are quite popular because for a certain type of random noise, they provide excellent noise reduction capabilities, with considerably less blurring than linear smoothing filter. Median filter are particularly effective in presence of both bipolar and unipolar impulse noise [10].

Gray scale image will be converted in terms of two levels 0 and 1. 0 as a white level and 1 as a black level the function used for this conversion is [9]

Input image (i, j) < threshold, output image (i, j)=1
Input image (i, j) ≥ threshold, output image (i, j)=0

For edge detection canny edge detector is used. Thinning is done by using morphological thinning function in Matlab.

Devanagari characters segmentation process has been illustrated as follows. The pre-processed word image is taken as an input image .The most dominating line is called shirorekha [1][2][3][10]. Shirorekha connects all the alphabets together. Remove the shirorekha from the word. Extract the sub images that are separated from the adjacent letters. The sub image may contain more than one component called as modifier. Separate the modifiers [18]. Segmented image is sent for feature extraction [8].

Input file Bhavan.jpg

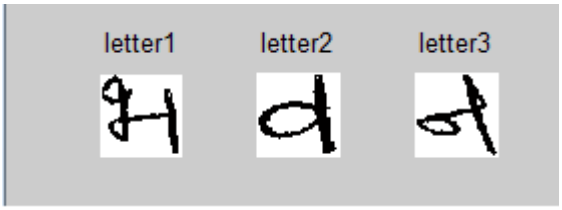

Figure 2. Segmentation of Devanagari Word Bhavan

A. Feature Extraction

Three classes of core characters are decided based on the coverage of core stripe and bar present End Bar, Middle Bar, No Bar Characters. Twenty one features are extracted and applied to the neural network. End points, Junctions, loops are structural features and aspect ratio is statistical feature [20].

Suppose the Character is अ then the matrix are

1	0	1
0	0	0
1	0	1

0	1	1
0	1	0
0	0	0

0	1	1
0	1	0
0	0	0

Figure 3.a)End Points b) Loop c) junction points

And the aspect ratio is [1 0 0] So the total code feature vector is

1 0 1 0 0 0 1 0 1 0 1 1 0 1 0 0 0 0 0 1 1 0 1 0 0 0 0 0100

If the character is ब then the matrix are

0	0	1
0	0	0
0	0	1

0	0	0
1	0	1
0	0	0

0	0	0
0	0	1
0	0	0

Figure 4. a)End Points b) Loop c) junction points

And the aspect ratio is [1 0 0] so the total code feature Vector is [7][8] [9][10]

0 0 1 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0

Like this we can calculate code vector or feature vector for all Devanagari characters. Features vector is applied as an input vector to an artificial neural network.

1) Self Organizing Feature Map.

Self Organizing map is topology preserving Map[13]. It has property of human brain which is not observed in many other artificial neural network classifiers. It is based on the concept of closet cluster wins (minimum Euclidian distance) .It clusters the data efficiently & having generalization ability. Character classification and recognition is based on Artificial Neural Network. The network can organize a topological map from any random point. The resulting map shows the natural relationship among the patterns that are given to the network. The network combines an input layer with the competitive layer and is trained by competitive learning. [4]. Similarities among the patterns are mapped into the closeness relationship on the competitive layer grid.

2).Basic Structure

The first layer of the network is an input layer. Fig 5 shows connections from input vector to a single unit in the competitive layer Typically the second layer is organized as a two dimensional grid.[4] All the connections go from the first layer to second layer. Two layers are fully interconnected [31] .

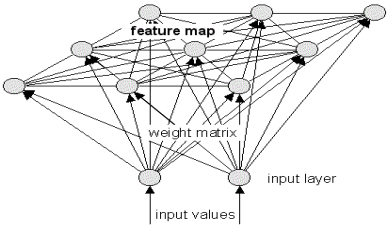

Figure 5 Self Organizing Maps [4]

When an input pattern is presented,[31] each unit in the first layer takes the values of the corresponding entry, from an input pattern. The second layer units then sums the inputs and compute to find a single winning unit. Every unit is associated with its weight value which is initially random. Each input pattern vector is uniformly distributed between 0 and 1.We can apply input pattern vector with n entries. As a result the input pattern is uniformly spread over d dimensional hypercube.

The input pattern applied to the network can be written as [4]

$$X= [x_1, x_2, x_3, x_4, ---x_n,] \quad (1)$$

The weights are given by

$$W= [w_1, w_2, w_3, w_4, -- w_n,] \quad (2)$$

Where W identifies the units in the competitive layer. The first step in the operation of the Kohonen network is to compute a matching value for unit i is [4]

$$\| X - W_i \| \quad (3)$$

which is the distance between vector E and Ui

$$\sum (x_J - w_{ij})2 \quad (4)$$

The unit with the lowest matching value (the best match) wins the competition. Here we denote The unit with the best match as unit c_i and c is chosen such that

$$\| X - W_i \| = \min \{ \| X - W_i \| \} \quad (5)$$

Where minimum is taken over all the units i in the competitive layer. If two units have same matching value then by convention, the unit with the lower index value i is chosen. After the winning unit is identified, the next step is to identify the neighborhood around it. The neighborhood in this case consists of the units that are within the square that is centered on the winning unit c. The size of the neighborhood changes as shown by squares of different sizes in figure 6. The neighborhood is denoted by set of units Nc. Weights are updated for all neurons that are in the neighborhood of the winning unit

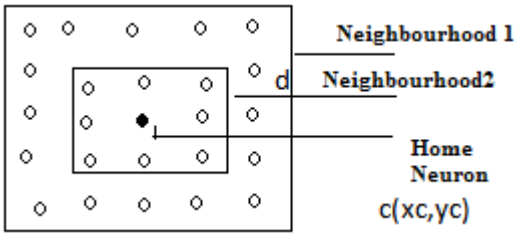

Figure 6 Network Neighborhood

The update equation is

$$Wij = \{ \sum (x_j - w_{ij}) \} \quad (6)$$

If the unit i is the neighborhood Nc otherwise And

$$w_{ijnew} = w_{ijold} + w_{ij} \quad (7)$$

The adjustments results in the winning unit and its neighbors having their weights modified becomes more like the input pattern[31]. The winner then becomes more likely to win the competition. Two parameters that should be specified, the value of α_i, learning rate parameter and neighborhood size Nc ,the learning rate, [4]
Is specified as

$$\alpha = \alpha_0 \{1- (t/T) \} \quad (8)$$

Where t = the current training iteration
T = Total no. of training iterations to be done .

Thus $\alpha 0$ is decreased until it reaches 0. The decrease is linear with number of training iterations completed. The size of neighborhood is second parameter to be specified. Typically the initial neighborhood width is relatively large and the width is decreased over many training iterations. Consider the neighborhood as shown in figure 6 which is centered on winning unit c, at position (x_c, y_c). Let d be the distance from c to the edge of the neighborhood. The neighborhood is then all (x,y) Such that

C-D < X < C+D And C-D < Y < C+D (9)

This defines a square neighborhood about c. Since the width of the neighborhood decreases over the training iterations, the value of d decreases. Initially d is set at a chosen value denoted by d_0. The value of d is then decreased according to the equation [4]

$d = d_0 \{1- (t/T) \}$ (10)

Where t = the current training iteration
T = Total no. Of training iterations to be done. This process assures a gradual linear decrease in d, starting with d_0 and going down to 1. The same amount of time is spending at each value.

IV. Handwritten Character Recognition System

The scheme is implemented and discussed as below.

A) Input Data
The scope of the system is restricted to Devanagari Script the characters are scanned in the view of the limitations imposed by the training time. The data base was obtained by scanning handwritten numerals and characters , words of different persons .

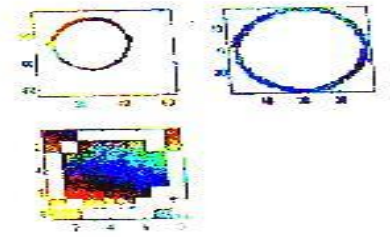

Figure 7. Scanned image, minimum rectangle, bitmap image

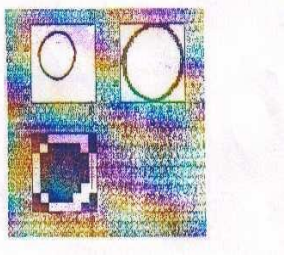

Figure 8 . Scanned image, minimum rectangle, bitmap image

The samples are as shown in fig.7 and fig.8 In the process of obtaining bitmap image 64 windows are obtained for each character as shown in fig. 7. for each local window average of the pixel value is computed and threshold value is set to 1 otherwise 0. The final vector is of 64 elements as shown in the fig.9

0 0 1 1 1 1 1 0
0 1 0 0 0 0 1 0
1 0 0 0 0 0 0 1
1 0 0 0 0 0 0 1
1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 1
0 1 0 0 0 1 1 0
0 0 1 1 1 1 0 0

Figure 9. Feature vector

For Devanagari characters also we have implemented the system in the view of the limitations imposed by the training time. The data base is obtained by scanning handwritten characters from different 500 persons. In the process of obtaining bitmap image 64*64 size windows is obtained for each character as shown in fig. 8. for each local window average of the pixel value is computed and threshold value is set to 1 otherwise 0

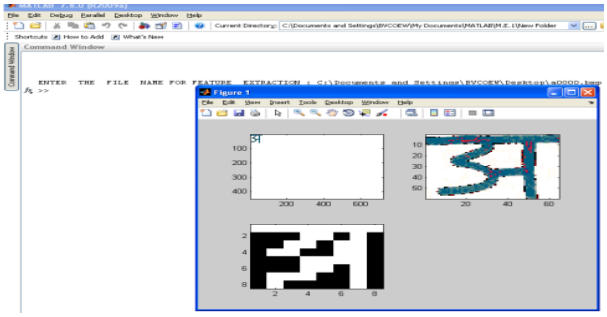

Figure 10. a) Scanned image b) minimum rectangle c) bitmap image .

The final vector is of 64 elements as shown below.

0 0 1 1 1 1 1 0
0 1 0 0 0 0 1 0
1 0 0 0 0 0 0 1
1 0 0 0 0 0 0 1
1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 1
0 1 0 0 0 1 1 0
0 0 1 1 1 1 0 0

Figure 11. Feature vector .

V. Training and Testing

The feature vector computed for different data sets are stored in a file. In the processing of training, these features are applied to a self-organizing map subsequently, and the network is trained. In the training process features for each character and numeral are applied to self-organizing map .The network is trained for 1000 to 10,000 iterations. After training the resultant prototypes are used for fine-tuning. These prototype are used in the recognition process to test the performance of the network. The network is trained and tested for different parameters such as output nodes, neighborhood size and number of iterations.

The relations used for α_I and N_C is follows [4]

$\alpha_i = \alpha_0 \{1- (g/h) \}$ (11)
$N_C = N_{C0}\{1- (g/h) \}$ (12)

α_0 & N_{C0} are initial values. g = current training iterations.
h = total no. of training iterations to be done.

A) SOM Output after training with different parameters after fine tuning.
α_0 = 0.999,Nco= 90,Output Nodes =200,Performance of the Network for α_0 = 0.9,Number of cycle (iterations) =10000
Total no. of patterns =100

```
steps=3   cycles=500 nc=1   alpha=1.000000e-001 win_x=22
steps=4   cycles=500 nc=1   alpha=1.000000e-001 win_x=33
steps=5   cycles=500 nc=1   alpha=1.000000e-001 win_x=39
steps=6   cycles=500 nc=1   alpha=1.000000e-001 win_x=24
steps=7   cycles=500 nc=1   alpha=1.000000e-001 win_x=62
steps=8   cycles=500 nc=1   alpha=1.000000e-001 win_x=14
steps=9   cycles=500 nc=1   alpha=1.000000e-001 win_x=89
steps=10  cycles=500 nc=1   alpha=1.000000e-001 win_x=89
steps=11  cycles=500 nc=1   alpha=1.000000e-001 win_x=72
steps=12  cycles=500 nc=1   alpha=1.000000e-001 win_x=75
steps=13  cycles=500 nc=1   alpha=1.000000e-001 win_x=48
steps=14  cycles=500 nc=1   alpha=1.000000e-001 win_x=17
steps=15  cycles=500 nc=1   alpha=1.000000e-001 win_x=86
steps=16  cycles=500 nc=1   alpha=1.000000e-001 win_x=8
steps=17  cycles=500 nc=1   alpha=1.000000e-001 win_x=59
steps=18  cycles=500 nc=1   alpha=1.000000e-001 win_x=75
steps=19  cycles=500 nc=1   alpha=1.000000e-001 win_x=56
steps=20  cycles=500 nc=1   alpha=1.000000e-001 win_x=67
steps=21  cycles=500 nc=1   alpha=1.000000e-001 win_x=62
steps=22  cycles=500 nc=1   alpha=1.000000e-001 win_x=4
steps=23  cycles=500 nc=1   alpha=1.000000e-001 win_x=17
steps=24  cycles=500 nc=1   alpha=1.000000e-001 win_x=48
steps=25  cycles=500 nc=1   alpha=1.000000e-001 win_x=44
steps=26  cycles=500 nc=1   alpha=1.000000e-001 win_x=84
steps=27  cycles=500 nc=1   alpha=1.000000e-001 win_x=1
steps=28  cycles=500 nc=1   alpha=1.000000e-001 win_x=80
steps=29  cycles=500 nc=1   alpha=1.000000e-001 win_x=28
steps=30  cycles=500 nc=1   alpha=1.000000e-001 win_x=1
traning complete
>>
```

Figure 12. Self organizing Map Training .

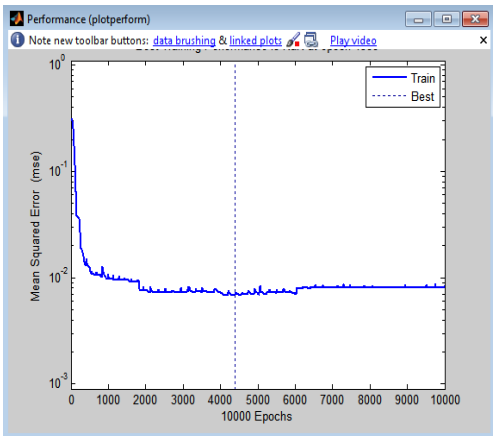

Figure13 . Mean Square Error

Mean squared error (MSE) of classifier is one of many ways to quantify the difference between values implied by classifier and the true values of the input being estimated. Figure 6 shows the squared value of error and the performance of the classifier. Mean square error is progressing towards zero as training time progresses.

Fig.14 shows the relationship between Target output and actual output which nearly linear and regression R=0.88. Output of self organizing map is applied to Learning vector quantization ,Fine tuningof winner Index is done as shown in fig.15. Fig 16 shows the final weight vector for character Ma. Fig. 17 shows the recognition of numeral 0 (zero) after fine tuning of the weight vector

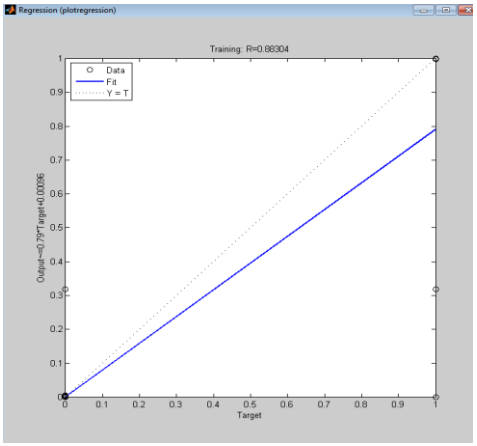

Figure 14. Regression of training with R=0.88

```
winnerindex=27
winnerindex=35
winnerindex=21
winnerindex=32
winnerindex=38
winnerindex=24
winnerindex=21
winnerindex=13
winnerindex=16
winnerindex=16
winnerindex=1
winnerindex=27
winnerindex=47
winnerindex=16
winnerindex=47
winnerindex=9
winnerindex=47
winnerindex=27
winnerindex=24
winnerindex=42
```

Figure 15 . Winner Index after Fine Tuning.

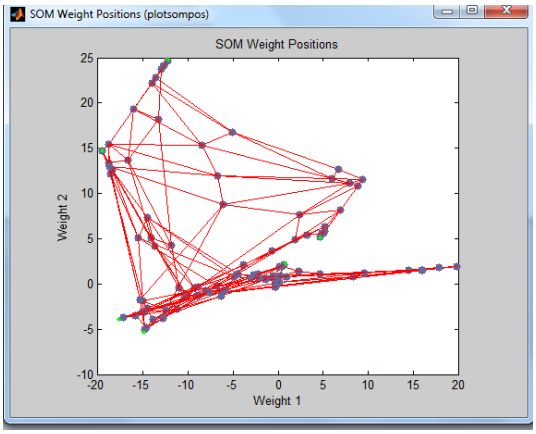

Figure 16. Example of Organization Final Weight Vector for character म (ma).

```
ENTER  THE  FILE  NAME FOR FEATURE  EXTRACTION : a05.bmp
charis =
     0
fx >>
```

Figure 17 Character Recognition after fine tuning.

Fig. 18 depicts the gradient, validity check and the learning rate for self organizing map network.

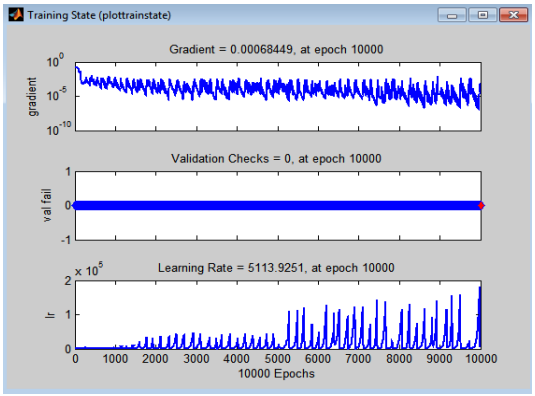

Figure 18. Gradient, Validity Checks, Learning rate for 1000 cycles.

Table 1. Classification Accuracy for Different Nodes.

No.of output Nodes	Neighborhood size	cycles	Classification Accuracy (SOM)
125	60	500	65%
150	75	500	67%
175	80	750	70%
200	99	750	78%
225	120	900	85%
250	120	900	88%
275	130	1000	90%
300	145	1000	93%

Table 2.

Character Accuracy By Using LVQ Classifier

Numerals	No. of Samples for Training	No. of Samples for Testing	Accuracy With LVQ Classifier.
0	1500	1000	99
१	1500	1000	99
२	1500	1000	99
३	1500	1000	99
४	1500	1000	99
५	1500	1000	99
६	1500	1000	99
७	1500	1000	99
८	1500	1000	99
९	1500	1000	99

Table 3. Performance summary of different classifiers.

MLFFNN		SOM+LVQ		Pattern Matching	
Character Database	Numeral Database	Character Database	Numeral Database	Character Database	Numeral Database
93%	98%	93%	95%	90%	95%

VI. Conclusion and Discussions.

The self-organizing map is used for pattern classification. The performance achieved by self- organizing map is better than back propagation algorithm. Generally back propagation algorithm is used for such tasks. The disadvantages associated with back propagation algorithm such as local minima and deciding number of hidden units is not observed in the implemented system. Self organizing map is trained for different parameters the classification accuracy obtained is 85% to 93% for 300 nodes, except some special *characters*. The training time is proportional to the number of patterns used for training, number of output nodes and iterations. After fine tuning the accuracy is increased by 2% . The data base from 2000 persons is taken for training and for testing 500 samples is taken per character from different persons. In SOM initially one hidden layer has been taken, as the training progresses, observed winner index of different classes are either same or closer to the previous winner index. The characters are divided into three different classes based on whether the character is no bar, Middle bar, end bar. The output is fine tuned for different classes.

VII. Future Scope

The above system can be implemented for various scripts like Arabic, Roman, Cyrillic, Han, and Hebrew etc. The data base could be extended up to 5000 samples. We may include Upper Modifiers, Lower modifiers, and Compound character in the recognition of Devanagari script.

Acknowledgement

The above work has been carried in Electronics department of Bharati Vidyapeeths Deemed University college of Engineering Pune-43.Prof. Dr. A.R.Bhalerao, Dean has constantly encouraged and helped us. The database has been prepared by the Undergraduate students of Electronics and Telecommunication Engineering Department.We are thankful to them.

References.

[1] AM.Namboodiri, A.K.Jain, “Online script recognition,” *IEEE Trans. Patten Analysis and Intelligence*,Vol.26,No.1, pp.124-130,January 2004.

[2] U.Bhattacharya,B.B.Choudhari, “Handwritten Numeral database of Indian Scripts and Multistage Recognition of Mixed Numbers” IEEE Trans. Patten Analysis and Machine Intelligence, Vol.31No.3 pp. 444-457, March.2009.

[3] A.W. Senior, J.Robinson,“An Offline Cursive

Handwriting Recognition System,"*IEEE Trans. Patten Analysis and Intelligence*, Vol.20 No.3 pp. 309-321, March 1998

[4] T.Kohonen, "The Self Organizing Map," *IEEE Trans. Patten Analysis and Intelligence*, Vol.78, No.9 pp. 1464-1480, Sept.1990.

[5] In-Jung Kim and Jin Hyung Kim,"Statistical character structure modeling and its application to handwritten Chinese recognition," *Pattern Analysis and Machine Intelligence*, Vol.25, No.11, pp.1422-1436, 2003.

[6] J Park, "An Adaptive Approach to Offline Handwritten Word Recognition,"*IEEE Trans. Patten Analysis and Intelligence*, Vol.24 No.7 pp. 919-931, July.2002.

[7] Macro Bressan and Jordi Vitria , " On the selection and Classification of Independent Features," *IEEE Trans. Patten Analysis and Intelligence*, Vol.25 No.10pp. 1312-1322, October 2003.

[8] B.Wegmann, C.Zetzshe, "Feature – Specific Vector Quantization Of Images," *IEEE Trans.Image Processing*, Vol.5 No.2 pp. 274-288, Feb. 2000.

[9] J.A.Starzyk, Zhen Zhu, "Self – Organizing Learning Array," *IEEE Trans Neural Network*, Vol.16 No.2 pp. 355-363 March 2005.

[10] Image Compression with JBIG2," *IEEE Trans.Image Processing*, Vol.8 No.12 pp. 944-956,Feb. 2000.

[11] Luiz S. Oliveria,F. Bortolozzi,R.Sabourn "Automatic Recognition of Handwritten Numeral Strings:A Recognition and Verification Strategy ," *IEEE Trans. Patten An and Intelligence*, Vol.24 No.11 pp. 1438-1453,NOV.2002.

[12] Yu Qiao,M.Nishiara," A Framework Toward Restoration of Writing Order From Single – Stroked Handwriting Image ," *IEEE Trans. Patten Analysis and Intelligence*, Vol.28 No.11 pp. 1724-1437, NOV.2000.

[13] Kohonen Springer Verlag, *"Self Organizing Map,"*Berlin,NewYark, Oct2001.

[14] R.O.Duda, P.E.Hart and D.G.Strork," Pattern Classification," *Second ed. Wiley – Interscience*, Oct. 2000.

[15] Joshi, G. Nagy , "Online Handwriting Recognition Using Time Order of Lexical Signal co-occurrences ,"*Electronics Letters on Computer Vision and Image Analysis*,2005.

[16] Lionel ,l. Oudot, "Self Supervised Adoption for On Line Script Text Recognition,"*Electronics Letters on Computer Vision and Image Analysis*,2005.

[17] Weliwitage" Whole of Word Recognition Methods for Cursive Scripts," *Electronics Letters on Computer Vision and Image Analysis*,2005.

[18] K.Indira,S. Sethu Selvi, "An Offline Cursive Script Recognition system using Fourier Wavelet Features". in Proceeding of the 2007 International Conference on Computational Intelligence and multimedia , IEEE, Computer Society Publication

[19] Lu S,C Lim Tan, "Script and Language identification in noisy and degraded document images" IEEE Trans. Patten Analysis and Machine Intelligence, Vol.30No.1 pp.14-24, Jan.2008.

[20] V.R.Pawar;GaikwadA.N,"Multistage Recognition approach for handwritten Devanagari Script Recognition "Topic(s): Communication, Networking & Broadcasting ; Components, Circuits, Devices & Systems ; Computing & Processing(Hardware/Software) DigitalObjectIdentifier:10.1109/WICT.2012.6409156 Publication Year: 2012 , Page(s): 651 - 656 IEEE Conference Publications.

[21] V. Pawar and Dr. A. Gaikwad,"An Artificial Neural Network based Multistage Offline Handwritten Character Recognition".CiiT International Journal of Artificial Intelligent Systems and Machine Learning .Print: ISSN 0974–9667.

[22] V.R.Pawar,A.N.Gaiikwad, " Linear Vector Quantization of an Artificial Neural Network based Handwriting Recognition by using SOM" ,Proceedings of national Conference,NCET-2007,Pune,India, pp.152-155,April-2007.

[23] G. Louloudis, B. Gatos, I.Pratikakis, C. Halatsis. "Line and word segmentation of handwritten documents," 11th International Conference on Frontiers in Handwriting Recognition (ICFHR'08), Montreal, Canada, pp 247-252 ,August 2008.

Author Biographies

Vijaya Pawar has completed her M.E.(Electronics) from Shivaji University , Maharashtra ,India in 2002. Currently she is the research scholar at Bharati Vidyapeeth Deemed University College of Engineering, Pune, Maharashtra. Her area of research interest includes Digital Image Processing, Artificial Neural Network etc.

A.N.Gaikwad, received the Bachelors of Engineering (Electronics and Telecommunication Engineering) and Masters of Engineering (Electronics and Telecommunication Engineering) degrees from University of Pune in 1982 and 1995respectively .He has received Ph.D. degree from University of Pune His research interest includes Digital Signal Processing, Image Processing.