A Novel Approach of Handwritten Devanagari Character Recognition through Feed Forward Back Propagation Neural Network

Shubhra Saxena
Jaipur National University,
Jaipur, Rajasthan, India

P. C. Gupta
Department of Computer Science and Informatics
University of Kota
Kota, Rajasthan, India

ABSTRACT

Handwritten character recognition plays an important role in the modern world. It can solve more complex problems and makes human's job easier. The present paper portrays a novel approach in recognizing handwritten devanagari character through feed forward back propagation neural network. All the experiments are conducted by using the Artificial Neural Network tool of Matlab.

Keywords

Devanagari Character Recognition, Character Segmentation, Feature Extraction, Neural Network Model, Matlab .

1. INTRODUCTION

Over the last decade Handwritten Character Recognition systems have proven to be a boon for the society owing to their stupendous advent of digital computers. The documents of transaction on paper like cheques, envelopes, forms and other manuscripts have been used in various forms of application in banks, libraries and other publishing houses. It has been estimated that approximately millions per year is spent on searching information from paper documents due to the cost of human labour. Automatic recognition of handwritten information by the use of computers can be used to reduce significant cost. The solution lies in the intersection of the fields of pattern recognition, image processing field. Optical character recognition is a process of mechanical or electronic translation of scanned images of handwritten or printed text into machine –readable text. Essentially, handwritten character recognition can be segregated into two domains: Online and Offline. Online character recognition involves the identification of character in the process of writing and dealing with pen up and down movement. Mobile communication systems such as Personal Digital Assistant (PDA), electronic pad and smart phone have online handwriting capability integrated in them. Offline character recognition recognizes already written character pattern in a scanned digital image. It uses static representation of a digitized document such as check, form, mail, or technical document.

Recognition of handwritten character gets complicated due to numerous variations involved in the shape of characters, different writing style, overlapping and the interconnection of the neighbouring characters, It also depends on the individual since we do not write the same character in that same way, Since developing an OCR system with high recognition accuracy for Devanagari script is a daunting task. The data conceived should be preprocessed accurately to achieve high recognition .Following steps aid in acquiring high accuracy.

1. Image Preprocessing
2. Binarization
3. Character Segmentation
4. Feature Extraction and selection
5. Recognition

The above Steps have been explained in paper [1-4]. The organization of the paper is as follows. A characteristic of Devanagari Script is described in Section 2. Literature Review is given in Section 3. The proposed Methodology is given in Section 4. Experimental results and discussions are provided in Section 5 followed by conclusion and references in Section 6 and 7 respectively.

2. CHARACTERISTICS OF DEVANAGRI SCRIPT

India is a multi-lingual and multi-script country. Devanagari scripts are originated from Brahmi script through various transformations. It was originally developed to write Sanskrit but was later used to write many other languages like Bhojpuri, Bhili, Magahi, Maithili, Marwari, Newari, Pahari, Santali, Tharu, Marathi, Mundari, Nepali and Hindi. Moreover, Hindi is the national language of India and the third most popular language in the world [2]. In Devanagari, words are written as they are pronounced. Devanagari has 13 vowels and 34 consonants. Text, characters, and numerals are written from left to right in Devanagari. Apart from vowels and consonants, words are written using consonant and vowel that together form compound character. The shape of compound character depends on the modifiers that it is placed to the top, bottom, left or right of the consonants. Top modifiers are placed above the horizontal line drawn on the top of the word. This line is the distinctive feature of Devanagari and known as header line or "Shirorekha". The lower modifier are placed below the character .which may or may not touch the character.

3. LITERATURE REVIEW

Many stalwarts in the field of research have worked towards the off-line handwritten Devanagari Character Recognition. The first research work on handwritten devanagari character was published in 1977 [3]. Gradually

numerous works continued to solve the problems associated with character recognition. U.Pal et al. [4] proposed A comparative study of Davanagari handwritten character recognition using 12 different classifiers and four set of features. Feature sets used in the 12 different classifier are computed based on curvature and gradient information obtained from binary as well as gray-scale images.

G.G .Rajput et al. [5] proposed a novel method towards multi-script identification at block level . This recognition is based on the feature that was extracted while using Discrete Cosine Transform and Wavelets of Daubechies family.

Bikash et al. [6] proposed a continuous density HMM to recognize a word image. The histogram of chain-code directions in the image script, scanned from left to right by a sliding window, is used as the feature vector. A handwritten word image is assumed to be a string of several image frame primitives. One HMM is constructed for each word. To classify an unknown word image, its class conditional probability for each HMM is computed. The class that gives highest probability is finally selected.

Sukalpa Chanda et al. [7] propounded a two stage approach for word-wise identification of Devanagri, English , and bangle script . The first stage allows identifying script with high speed. The advanced second stage processes only those samples that yield low recognition confidence in the first stage. Features used in first stage are a 64-dimensional chain-code -histogram feature, while 400-dimensional gradient features are used in the second stage. Final classification of a word to a particular script is done through majority voting of each recognized character component of the word. Correct classification of 98.51% on 11,123 test words is achieved while recognition confidence is as high as 95 % at both stages.

In [8] Mahesh Jangid proposed a methodology that relies on a three feature extraction techniques. The first technique was based on recursive subdivision of the character image so the resulting sub – images at each iteration have balanced number of foreground pixels, as far as this is possible. Second technique is based on the zone density of the pixel and third is based on the directional distribution of neighbouring background pixels to foreground pixels. The 314 sized feature vectors is formed from the three feature extraction techniques for a handwritten Devanagari character. The dataset of 12240 samples is used and obtained the 94.89% recognition accuracy.

Kumar and Singh [9] proposed a Zernike moment feature based approach for Devanagari handwritten character recognition. They used an artificial neural network for classification.

R. Kapoor, D. Bagai, T. Kamal, [10] proposed HMM based approach, using junction points of a character as the main feature. The character has been divided into three major zones. Three major features i.e. number of paths, direction of paths, and region of the node were extracted from the middle zone.

S.Arora et al.[11] proposed a handwritten devnagari character recognition based on SVM and ANNs classification methods, After pre-processing the character image, they extracted shadow features, chain code histogram features, view based features and longest run features. These features are then fed to Neural classifier and in support vector machine for classification.

Prachi Mukherji et al.[12] proposed a new shape based technique for recognition of isolated handwritten Devanagari characters. The thinned character is segmented into segments , using basic structural features like endpoint, cross point, junction points and adaptive thinning algorithm. The segments of characters are coded using Average Compressed Direction Code (ACDC) algorithm. Their location in the image frame is based on a fuzzy classification. Characters are pre-classified using a tree classifier. Subsequently unordered stroke classification based on mean stroke features is used for final classification and recognition of characters. The system tolerates slant of about 10 deg left and right and a skew of 5 deg up and down. The average accuracy of recognition of the proposed system is 86.4% .

Ashutosh Aggarwal et al.[13] proposed an approach, where he used the gradient representation as the basis for extraction of features **.** Initially the Gradient Vector is calculated at all image pixels and sample image is divided into 9x9 sub-blocks. Then in each sub-block Strength of Gradient is accumulated in each of 8 standard directions in which Gradient Direction is decomposed. Finally image is down sampled to 5x5 blocks from 9x9 blocks using a Gaussian Filter giving a feature vector of dimensionality 200 (5x5x8). Accuracy of 94% is obtained using Support vector Machines (SVM) as classifier.

Sandhya Arrora et al. [14] proposed a new method for recognition of offline Handwritten non- compound devanagari characters in two stages. Two MLP's are used separately to recognize the characters. For one of the MLP's the characters are represented with their shadow features and for the other chain code histogram feature is used. The decision of both MLP's is combined using weighted majority scheme. Top three results produced by combined MLP's in the first stage are used to calculate the relative difference values. In the second stage, based on these relative differences character set is divided into two. First set consists of the characters with distinct shapes and second set consists of confused characters, which appear very similar in shapes. Characters of distinct shapes of first set are classified using MLP. Confused characters in second set are classified using modified Harris corner detection technique of minimum edit distance method. Experiment on this method is carried out on a database of 7154 samples. The overall recognition is found to be 90.74%.

4. METHODOLOGY

The first stage is Image pre-processing stage in which data in a paper document are captured by optical scanner, then pixel based images are created by painting programs such as Microsoft's Paintbrush. These pixels may have values: OFF (0) or ON (1) for binary images, 0-255 for gray-scale images, and 3channels of 0-255 color values for color images. The images are read into the matlab environment using imread function, whose basic syntax is imread ('filename') . Here file name is a string containing the complete name of the image file and then the image can be displayed on Matlab workspace using function imshow ( ).

In next stage image is enhanced with some standard image processing function. The pre-processing stage yields a

clean document with minimum noise. The next stage is segmenting the image into four sub parts. It is important stage because separation of character directly affects the recognition rate. After segmentation the extracted feature directly feed in the training phage .

The logical function is used to convert gray image to logical. It return an array that is further used to perform logical indexing. Logical can have the values 0 and 1. After that Im2bw function of matlab is used to convert image to binary image from indexed image. To do this it firstly convert it to gray image and then coverts this to binary image. The output Binary image BW has values of 1 (white) for all pixels with luminance greater then level and 0 for all other pixels. Now hist function is used to finds the histogram values of all the image subparts.

After finding the histogram equivalent of each part of character image , these values are imported as input of neural network tool and some random value are assigned as target value to train , test , and validate each sample part individually using feed-forward back propogation network . After training of each part we have taken all the four trained output as a single input and trained it for a unique value. This whole process described above is repeated for the full character image without doing segmentation.

4.1 Data Collection

We performed our experiments over a database generated by Collecting 20 handwritten samples of Devanagari characters from five different writers. In this work we have only used Devanagari consonants. The image files are stored in jpg format. Here we have shown only one sample image of Kha character. Sample table is given in Annexture 1

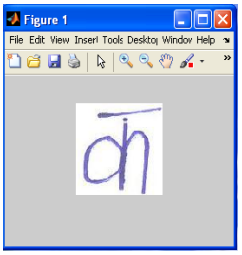

Fig1: Sample Image

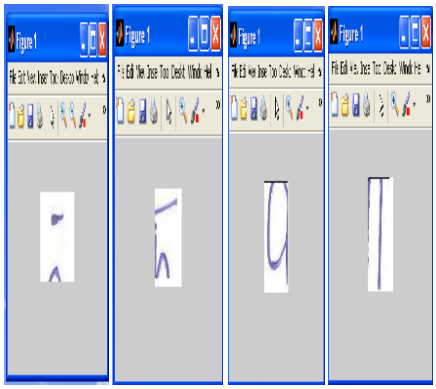

Fig 2: Segment Image

5. EXPERIMENTS AND RESULTS

The methodology relies on three techniques. The first technique is based on randomly subdivision of character sample and uses it as input of network. The second technique based on taking full image values as input of the network. The third technique based on taking all the sample image values as input and subdivision values as target to recognize the character pattern.

5.1 Proposed Approach 1:

Suppose that im(a, b) is a handwritten character Sample. Following steps are used to create network:

Step 1: Read Scanned Image. Suppose im(a, b) is the image . a and b is the height and width of the image.

Step 2: Segment the image horizontally and vertically into 4 Sub –parts.

Step 3: Calculate the hist feature of four Sub-parts .

Step 4: Set the four Sub-parts value as Input of neural tool and select random values for target .

Step 5: Train the Network till we get good performance. Plot the performance curve.

Step 6: Calculate the Mean Square Error and Plot the Regression values.

Step 7: Test the network again with same input and target value and calculate the Error and Regression values.

Network Model 1:

In our approach we use a two-layer feed-forward network with sigmoid hidden neurons and linear output neurons .It is also used to fit multi- dimensional mapping problems. The network will be trained with Levenberg-Marquardt back propagation algorithm. The neural network tool evaluate its performance using mean square error and regression analysis. We set the 20 hidden neuron to fit in the hidden layer of neural tool.

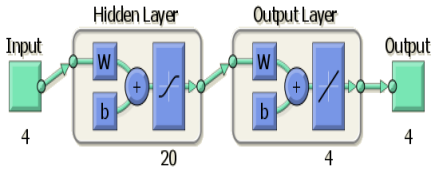

Fig 3: Sigmoid Model

The different Samples that we have applied on the network perform three major activities that are:

Training:

Neural Network has the discriminative property, what it has learned has been distributed over the whole network. The samples that are presented during training is adjusted according to the weight.

Validation:

The samples that are presented in the network are used to measure the network generalization.

Testing:

These samples are used to provide independent measure of network performance during and after training.

Training Session: 1

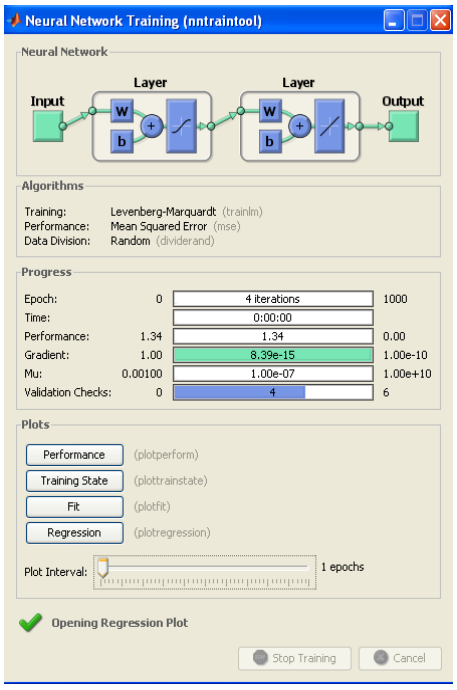

Fig 4: Training

We have used 1000 Epoch and 4 Iteration to train the network. The sample that is presented to the network during training is adjusted according to its error, here 4 validation checks is performed.

Performance: 1

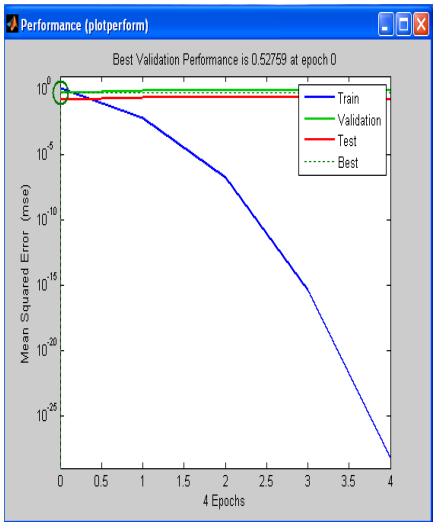

Fig 5: Performance

We get Best validation performance at epoch 0 in 4 iterations.

Error Session: 1

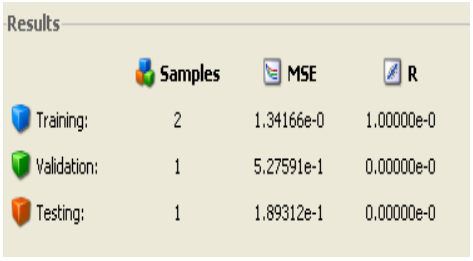

Fig 6: Result

Two Type of Value are considered:

MSE: Mean Squared Error is the average squared difference between outputs and targets. Lower values are better. Zero means no error.

R: Regression R Values measure the correlation between outputs and targets. An R value of 1 means a close relationship, 0 a random relationship.

Test Session: 1

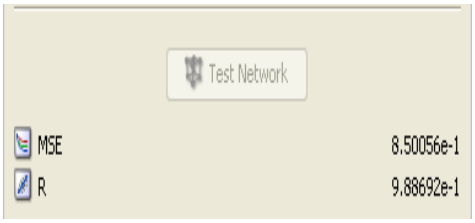

Fig 7: Testing

Second Session is also performed to check that network is performed well. Testing have no effect on training and so provide an independent measure of network performance during and after training.

5.2 Proposed Approach 2:

Step 1: Read Scanned Image. Suppose im(x, y) is the image. x and y is the height and width of the image.

Step 2: Segment the image horizontally and vertically into 4 Sub –parts.

Step 3: Calculate the hist feature of four Sub-parts.

Step 4: Calculate the hist feature of full Image value.

Step 4: Set the full image values as input and concatenated four Sub-parts value as target of neural tool.

Step 5: Train the Network till we get good performance. Plot the performance curve.

Step 6: Calculate the Mean Square Error and Plot the Regression values.

Step 7: Test the network again with same input and target values and calculate the Error and Regression values.

Network Model 2:

In Training Session 2 we take the full image sample values as input and 4 segment part concatenated values as target. We have taken 1000 epoch, 20 hidden neurons, and perform 4 validation checks till maximum Mu is reached. The network will be trained with Levenberg-Marquardt back propagation algorithm.

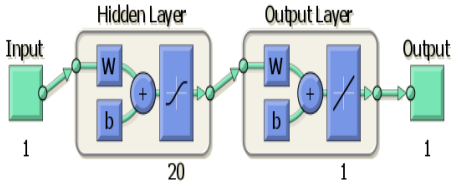

Fig 8: Sigmoid Model

Training Session 2:

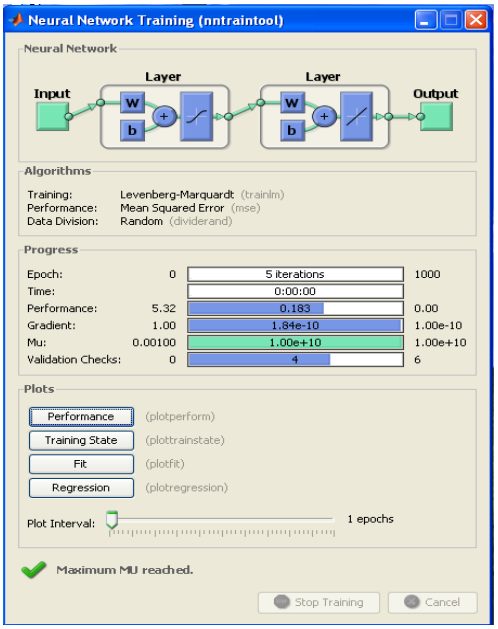

Fig 9: Training

Performance 2:

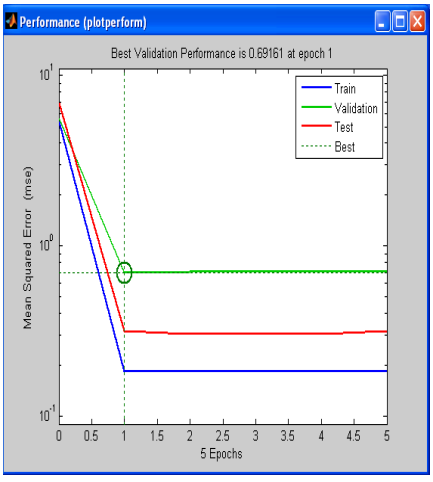

Fig 10: Performance

We have achieved best validation performance at epoch 1 in 5 iterations.

Training States:

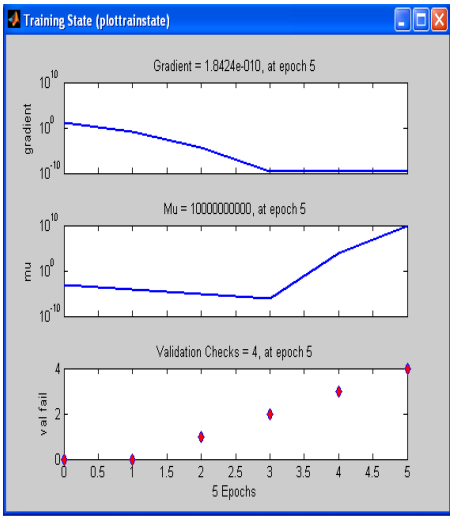

Fig 11 : States

During the training session completed it will take 1.8424e-010 gradient values at 5 epoch

Training Plot:

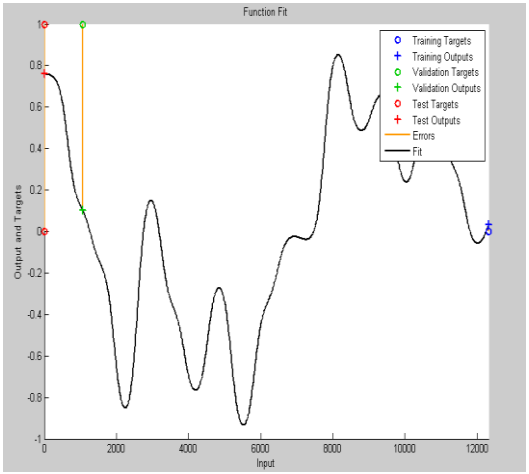

Fig 12: Function Approximation

Training Plot is used to define over fitting of data. It is describe Training, Validation, and Target input and output values.

Error Session 2:

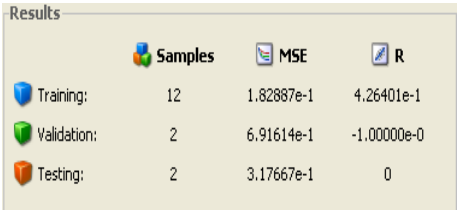

Fig 13: Result

It is used to define Mean Square Error and Regression during session.

Test Session 2:

Fig 14: Testing

The Testing session is performed after Training Session. We provide same set of input and target value to check system performance. Test vectors are used as a further check that the network is generalizing well, but do not have any effect on training.

5.3 Proposed Approach: 3

Step 1: Read all Scanned Image. Suppose im(x, y) is the image. x and y is the height and width of the image.

Step 2: Set all full image sample values of variation of same character as input and four sub-parts value as target of neural tool.

Step 3: Train the Network till we get good performance. Plot the performance curve.

Step 4: Calculate the Mean Square Error and Plot the Regression values.

Step 5: Test the network again with same input and target value and calculate the Error and Regression values.

Network Model 3:

We have taken 1000 epoch, 20 hidden neurons, during training. till maximum Mu is reached. The network will be trained with Levenberg-Marquardt back propagation algorithm.

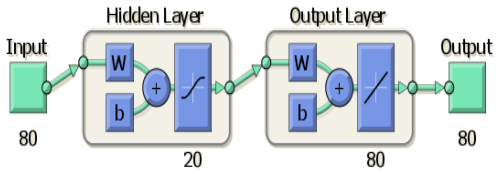

Fig 15 Sigmoid Model

Training Session 3:

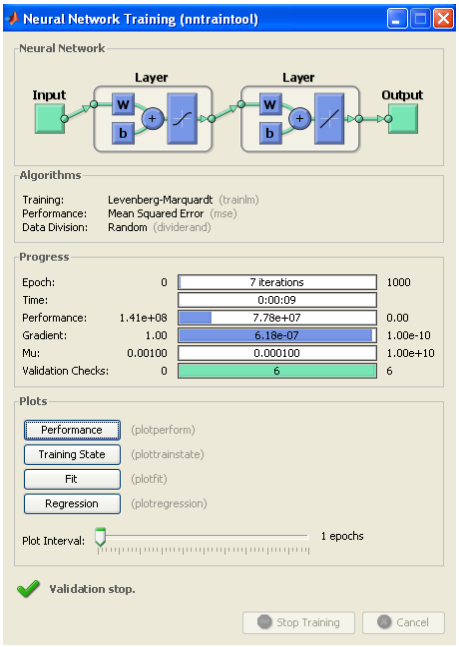

Fig 16 : Training

We divided sample data randomly and train the network till validation stop. Validation vectors are used to stop training early if the network performance on the validation vectors fails to improve or remains the same for max_fail epochs in a row.

Training States:

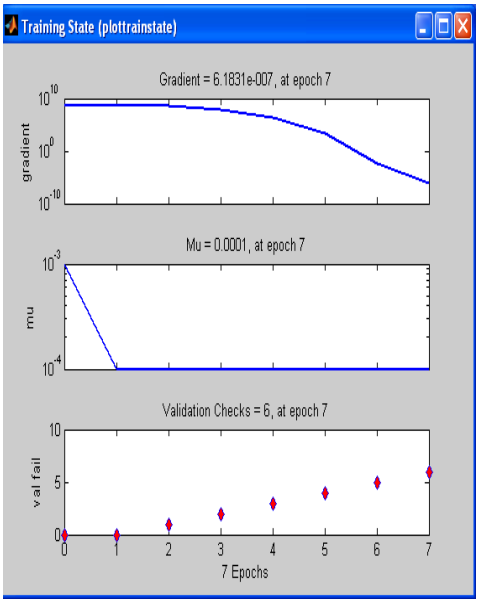

Fig 17: States

During the training session the training states is used to Show the gradient, Mu and validation check separately.

Performance 3:

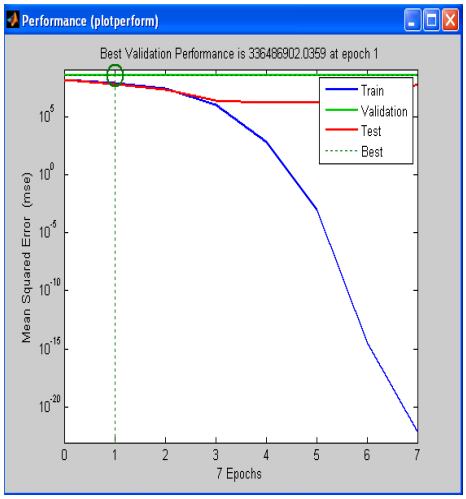

Fig.18: Performance

We have achieved best validation performance at epoch 1 in 7 iterations.

Error Session 3:

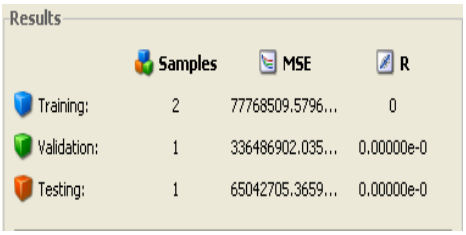

Fig 19: Result

It describes the consolidated result of all three type of samples i.e. training, validation and testing.

Test Session 1:

Fig 20 : Testing

We have made several training session on the sample set to train the network. The Testing session is performed after training session. We provide same set of input and target value to check system performance.

6. Conclusion

In this Research work a new approach has been implemented to recognize Devanagari handwritten characters.

After the complete research the following points are derived:

a) Four Devanagari characters have been selected for this work.

b) Segmentation of each Character into four parts have been done.

c) Eighty Samples from these characters have been collected.

d) Each part is converted into equivalent Binary codes by using Binarization method of Matlab.

e) Feed Forward Backpropogation Neural network is used for training, testing and validation.

f) Training of each sample is applied up to 5 times to minimize the error.

g) 20 Unknown Samples have been applied in the network to obtain the result.

h) Network correctly recognizes 17 samples. Therefore, accuracy level of the network is around 85%.

From this work we are able to recognize a handwritten devanagari characters. The only limitation in this work is the number of characters which have been selected for the work. The future work is to select all the Devanagari character for recognition through this approach.

7. REFERENCES

[1] Shubhra Saxena and P.C.Gupta , 2012 , " Study paper of OCR for character recognition ", IJBER ,Intl. Journal of Business &Engineering research. Vol .5.

[2] U.Pal et.al. ,2004, " Indian script character recognition : a survey", Pattern Recognition ,Vol .37, pp.1887-1899.

[3] Ashutosh Aggarwal et. al., 2012, "Handwritten Devanagari Character Recognition Using Gradient Features", Intl. Journal of Advanced Research in Computer Science and Software Engineering, Vol. 2, Issue 5.

[4] U.Pal, T.Wakabayashi, and F.Kimura, 2009, "Comparative study of Devanagari handwritten character recognition using different features and classifiers", 10th Intl. Conf. on Document Analysis and Recognition, pp.1111-1115.

[5] G.G Rajput, Anita H.B, 2010,"Handwritten Script Recognition using DCT and Wavelet Features at Block Level", IJCA, Recent Trends in image processing and pattern recognition.

[6] Bikash Shaw, Swapan Kumar Parui, Malayappan Sridhar, 2008, "Offline handwritten Devanagari word recognition : A holistic approach based on directional chain code feature and HMM" IEEE .

[7] Sukalpa Chanda , Sikanta Pal, Katrin Franke, Umapada Pal ,2009 " Two- stage Approcah for Word-wise Script Identification" , IEEE .

[8] Mahesh Jangid, 2011, "Devanagari Isolated Character Recognition by using Statistical features", IJCSE, Intl. Journal on Computer Science and Engineering Vol.3 No.6.

[9] Satish Kumar and Chandan Singh, 2005, "A Study of Zernike Moments and its use in Devnagari Handwritten Character Recognition", Intl. Conf. on Cognition and Recognition, pp. 514- 520.

[10] R. Kapoor, D. Bagai and T.S. Kamal, "Representation and Extraction of Nodal Features of Devanagari Letters", 3rd Intl. Conf. on Computer Vision, Graphics and Image processing.

[11] Sandhya Arrora et.al., 2010, "Performance Comparison of SVM and ANN for Handwritten Devanagari Character Recognition", IJCSI, Intl. Journal of Computer Science Issues, Vol . 7, Issue 3.

[12] Prachi Mukherji, Priti P. Rege, 2009, "Shape Feature and Fuzzy Logic Based Offline Devanagari Handwritten Optical Character Recognition",Intl. Journal of Pattern Recognition Research pp.52-68 .

[13] Ashutosh Aggarwal et. al., 2012, "Handwritten Devanagari Character Recognition Using Gradient Features", Intl. Journal of Advanced Research in Computer Science and Software Engineering, Vol. 2, Issue 5.

[14] S. Arora, D. Bhattacharjee, M. Nasipuri, D. K. Basu & M. Kundu , "Recognition of Non-Compound Handwritten Devnagari Characters using a Combination of MLP and Minimum Edit Distance,", IJCSS , Intl. Journal of Computer Science and Security Vol. 4 , Issue 1.

Annexure 1:

Since the Devanagari Character representation requires more space, Therefore, I have used some Abbreviation to represent these characters, which have been shown in the following table.

No.	Character (Ch)	Abbreviation
1.	KHA	A
2.	KHHA	B
3.	GHA	C
4.	GHHA	D

Sample Table: 1(Segment-Part-Value)

No	Ch	Part	Input Value				Target-Value
1	A1	1	75	0	0	4159	0011
2	A1	2	261	0	0	3973	1000
3	A1	3	41	0	0	4193	0101
4	A1	4	291	0	0	4016	1101
5	A2	1	90	0	0	4144	1001
6	A2	2	177	0	0	4289	1010
7	A2	3	131	0	0	3349	1100
8	A2	4	165	0	0	3141	0001
9	A3	1	125	0	0	4465	1110
10	A3	2	146	0	0	3044	0111
11	A3	3	382	0	0	3468	1011
12	A3	4	337	0	0	3623	0100
13	A4	1	300	0	0	3238	1000
14	A4	2	285	0	0	3075	1001
15	A4	3	303	0	0	3417	0010
16	A4	4	242	0	0	3478	1101
17	A5	1	334	0	0	4622	0011
18	A5	2	158	0	0	2510	1000
19	A5	3	542	0	0	6094	0111
20	A5	4	208	0	0	3440	0110
21	B1	1	152	0	0	5992	1010
22	B1	2	370	0	0	5807	1101
23	B1	3	260	0	0	4279	1100
24	B1	4	284	0	0	4591	1110
25	B2	1	186	0	0	4092	0001
26	B2	2	288	0	0	4682	0010
27	B2	3	213	0	0	3630	0110
28	B2	4	74	0	0	1276	1100
29	B3	1	216	0	0	4681	0010
30	B3	2	300	0	0	4260	0110
31	B3	3	264	0	0	4488	0100
32	B3	4	177	0	0	4101	1100
33	B4	1	281	0	0	6247	1010
34	B4	2	323	0	0	4297	0101
35	B4	3	305	0	0	7795	1011
36	B4	4	245	0	0	6055	0001
37	B5	1	312	0	0	6074	1010
38	B5	2	418	0	0	6446	0101
39	B5	3	272	0	0	6114	1000
40	B5	4	294	0	0	6426	0111
41	C1	1	188	0	0	3892	0001
42	C1	2	190	0	0	4490	0010
43	C1	3	76	0	0	3644	0110

44	C1	4	136	0	0	4424	1110
45	C2	1	230	0	0	5552	1101
46	C2	2	458	0	0	7872	1010
47	C2	3	235	0	0	5520	1100
48	C2	4	237	0	0	5628	0001
49	C3	1	395	0	0	5608	0001
50	C3	2	281	0	0	3595	0110
51	C3	3	383	0	0	7182	0100
52	C3	4	248	0	0	4825	1011
52	C4	1	299	0	0	4021	0100
54	C4	2	462	0	0	7450	1001
55	C4	3	250	0	0	5210	1000
56	C4	4	408	0	0	6000	0101
57	C5	1	265	0	0	5457	1101
58	C5	2	122	0	0	4498	1100
59	C5	3	424	0	0	5884	1010
60	C5	4	216	0	0	5524	1000
61	D1	1	119	0	0	4753	0111
62	D1	2	404	0	0	4468	1000
63	D1	3	174	0	0	5706	1010
64	D1	4	399	0	0	5901	0101
65	D2	1	168	0	0	4935	1000
66	D2	2	99	0	0	5004	1101
67	D2	3	258	0	0	4845	1000
68	D2	4	121	0	0	4982	1001
69	D3	1	170	0	0	4702	1011
70	D3	2	152	0	0	4720	1010
71	D3	3	184	0	0	5780	0001
72	D3	4	151	0	0	5813	0010
73	D4	1	218	0	0	5286	0110
74	D4	2	168	0	0	5124	0111
75	D4	3	75	0	0	4881	1010
76	D4	4	173	0	0	5378	1110
77	D5	1	311	0	0	5025	1000
78	D5	2	247	0	0	4451	1010
79	D5	3	138	0	0	4850	1101
80	D5	4	151	0	0	4547	0101

SAMPLE TABLE: 2 (Full –Image –Value)

Ch	**Input Value**				**Target Value**
A1	494	0	0	14782	0011100001011101
A2	351	0	0	12417	1001101011000001
A3	1064	0	0	12311	11100111101
A4	1040	0	0	13474	1000100100101101
A5	1054	0	0	16901	0011100001110110
B1	832	0	0	17264	1010110111001110
B2	74	0	0	12760	0001001001101100
B3	769	0	0	17015	0010011001001100
B4	722	0	0	23605	1010010110110001
B5	769	0	0	17015	1010010110000111
C1	391	0	0	16201	0001001001101110
C2	783	0	0	22002	1101101011000001
C3	216	0	0	5524	0001011001001011
C4	441	0	0	22225	0100100110000101
C5	930	0	0	20893	1101110010101000
D1	830	0	0	18754	0111100010100101
D2	541	0	0	15059	1000110110001001
D3	431	0	0	15937	1011101000010010
D4	612	0	0	22350	0110011110101110
D5	646	0	0	17321	1000101011010101