Handwritten English Character Recognition Using New Quadrant Splitting Feature Extraction Technique

Dayashankar Singh[1], Maitreyee Dutta[2], Sarvpal H. Singh[3], P. K. Singh[4]

[1]M.M.M.Engineering College, Gorakhpur,
[2]NITTTR, Chandigarh,
[3]M.M.M.Engineering College, Gorakhpur,
[4]M.M.M.Engineering College, Gorakhpur,

{dss_mec, d_maitreyee, singh_sarvpal}@yahoo.co.in, topksingh@gmail.com

Abstract- In this paper, a new feature extraction technique based on divide and conquer approach has been developed. Each character divided in four quadrants is processed to search each quadrant in some predefined patterns. The values assigned to resultant patterns are combined and fed to the neural network as input for training the network. The technique named Quadrant Splitting Feature Extraction (QSFE) is implemented using Back-propagation Neural Network with one hidden layer and one output layer. An analysis has been carried out on Experimental results to show that the newly developed feature extraction technique QSFE requires less training time and provides high recognition accuracy of about 96%.

Keywords: Neural Network, Feature Extraction Technique, Quadrant Splitting, Recognition Accuracy, Backpropagation neural network

1. Introduction

Neural Networks has been extensively used in the area of pattern recognition. Character recognition is one such aspect of pattern recognition that is being addressed by researches heavily. Many researches done in several languages such as Chinese, Japanese, Arabic, Farsi, etc. have been reported but still efficiency improvements in work related to hand-written English words and their conversions into Hindi using NN is an open problem. In many Indian offices such as Passport, Banks, Railway and sales tax etc., both Hindi and English languages are used [19]. This leaves a vast scope for English to Hindi translation.

Handwritten character recognition is a challenging problem in pattern recognition field. The difficulty is mainly because of large variations of individual's writing styles. Therefore, robust feature extraction to improve the performance of handwritten English character and word recognition has become very important to improve the performance of handwritten character recognition. In continuation to these efforts, a technique is being presented here that recognizes handwritten English characters and words. The work has also been enhanced to convert the words written in English into Hindi with high recognition accuracy and reduced training time. This technique is being referred to **Quadrant Splitting Feature Extraction (QSFE)** and is based on Back-propagation Neural Network with one hidden layer and one output layer.

Character recognition in several languages such as Chinese, Japanese etc has been an interesting issue and a good amount of work has been reported in this field. In India, about 80 percent people speak or know Hindi and a major part of this population also understand English. Therefore, considering the age of globalization and Information Technology, it is greatly felt that auto recognition of English words and their conversion into Hindi words are the needs of hour so that million of Hindi aware people around the world could get benefit with this revolutionized age of Information Technology. The methodology suggested in this paper has been proved to be the one recognizing English Character and words with high accuracy in less training and classification time.

The application of neural network to recognize characters is not new and the size of the input to such a net has remained a main; as the size of the input increases, the training time of network also increases. The QSFE minimizes the number of inputs to the neural network by dividing a character into four parts and then identifying the features of the character in each of the four parts. A complete character to the neural network is not needed to be given for recognition purpose; rather the values of the features of each quadrant are taken as inputs to the network. This requirement makes the feature extraction a key issue.

The organization of paper is as follow: Section-II describes the Extraction of features. Section-III provides detailed description of newly developed feature extraction technique named as Quadrant Splitting Feature Extraction (QSFE) technique. Section-IV offers experimental results and their analysis while Section-V covers conclusion and future scope.

2. Extractions of Features

Feature can be defined as a measurement taken on the input pattern to be classified. Features play an important role in handwritten character recognition in order to influence the recognition performance. When the input data to an algorithm is is too large to be processed and it is suspected to be notoriously redundant (much data, but not much information) then the input data will be transformed into a reduced representation set of features (also named features vector). Transforming the input data into the set of features is called features extraction.

The main objective of feature extraction is to divide the pattern by means of minimum number of features, which are effective in discriminating the different pattern classes.

Typically, it is being looked for the features that will provide a definite characteristic of that input type. The classifier is then supplied with a list of measured features, so that it maps these input features onto a classification state. On giving the input features, the classifier must decide which type of class or category they match most closely. Classification is rarely performed using a single measurement or feature taken from the input pattern. Usually, several measurements are required to be able to adequately distinguish between inputs that belong to different classes.

In this paper, various feature extraction techniques have been studied e.g. Conventional Feature Extraction, Boundary Tracing and Gradient Feature Extraction etc. A new feature extraction technique has been developed and it has been named as Quadrant Splitting Feature Extraction. The newly developed feature extraction technique (QSFE) has been implemented for English character recognition that provides high recognition accuracy and reduced training time.

3. Quadrant Splitting Feature Extraction (QSFE)

In this new approach, a 32x 32, black & white image in binary format is taken as input .The pixels which are covering the shape of the character are taken as the values 1 and rest of the pixels have the values 0. Now, the image is split into four quadrants each with size 16 x 16. Each character has its part in each of the four quadrants. Shape numbers are assigned to each part of the character which is lying in different quadrants.

Each quadrant has been separately scanned and matched the pixel pattern to different shapes. If all the four quadrant shapes are matching to a particular character, then the features of that character are extracted. In this way, 1024 input values are reduced to 4 values corresponding to the character part lying in each of the four quadrants.

For word recognition, the word is scanned column wise. The appearance of first empty column i.e. column having no character pixel, indicates the end of very first character. The number of empty columns between two consecutive characters will be the empty space between them. This procedure continues till the last column (up to column number 96).Characters scanned in this way are stored in separate arrays and they are made of the size 32 X 32.Thus by recognizing individual characters, the word made by their combination is recognized. Once the English word has been recognized, the process of conversion from English to Hindi starts. For word recognition, pattern matching technique has been performed. In this way, English words have been converted into Hindi words. Experiment has been carried out on these word conversions and it shows good result.

3.1. Features for English Characters

The features for characters of English Alphabet are given in Figure 1.

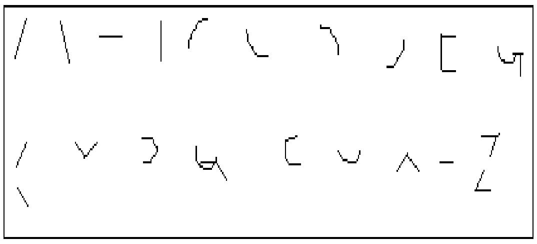
Figure1. Features of English Characters

3.2 Shape Representation

The various shape numbers which have been assigned for their corresponding features in various quadrants are given below.

3.2.1 First quadrant: Features and their corresponding values of shape numbers of first quadrant are given in figure 2.

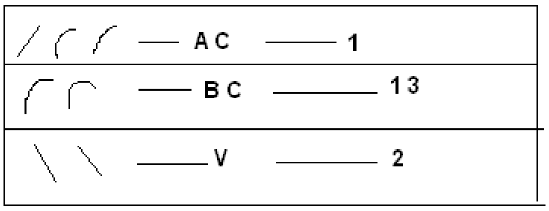

Figure 2. First Quadrant Features and their shape numbers

3.2.2 Second Quadrant*:* Features and their corresponding values of shape numbers of second quadrant are given in figure 3.

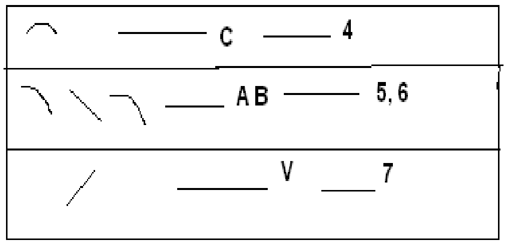

Figure 3. Second Quadrant Features and their shape numbers

3.2.3Third Quadrant: Features and their corresponding values of shape numbers of third quadrant are given in figure 4.

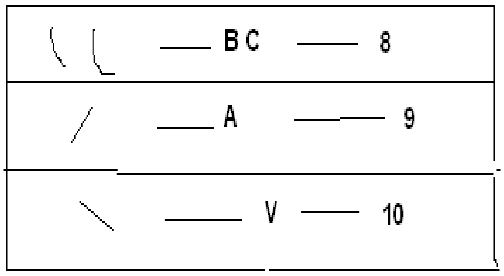

Figure 4. Third Quadrant Features and their shape numbers

3.2.4 Fourth Quadrant: Features and their corresponding values of shape numbers of fourth quadrant are given in figure 5.

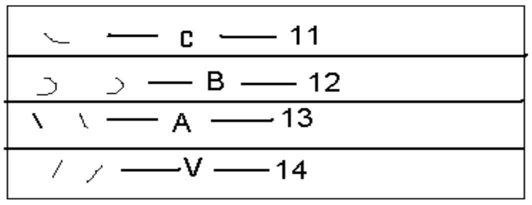

Figure 5. Fourth Quadrant Features and their shape numbers

3.4 Procedure

The procedure of implementing **QSFE** technique can be illustrated as follows:

- Capture the image of handwritten English character in 32 x32 pixels.
- Perform binarization on captured image into 32 x 32 pixels.
- Apply thinning process on binarized image.
- Split the image into four quadrants of 16x16 each.
- Scan each quadrant to match the pixel pattern of different shapes.
- All the four quadrant shapes are matching to a particular character and then the features of that character are extracted.
- In this way, 1024 input values have been reduced into 4 corresponding values to the character part lying in each of the four quadrants.
- Four corresponding values of a character are given as input to the Back propagation neural network for training as well as for simulation.

3.3 Flow Chart

The flowchart of automatic English character recognition system is given in figure 6.

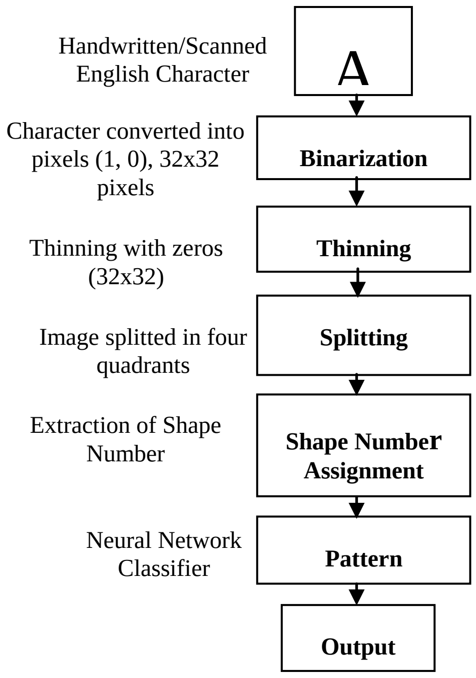
Handwritten/Scanned English Character
A
Character converted into pixels (1, 0), 32x32 pixels
Binarization
Thinning with zeros (32x32)
Thinning
Image splitted in four quadrants
Splitting
Extraction of Shape Number
Shape Number Assignment
Neural Network Classifier
Pattern
Output

Figure 6. Flowchart

Binarization: A process that converts whole image into pixel values 0 and 1.

Thinning: Through this operation skeleton of the image is obtained.

Splitting: Skeletonized image is splitted into four quadrants.

Shape Number Assignment: Different shape numbers have been assigned to the parts of the character which are lying in different quadrants. Pattern Recognition Pixel pattern is matched to find a particular shape number in each quadrant

4. Neural Network Classifiers

The neural network classification techniques such as multilayer perceptron (MLP) trained by Error backpropagation (EBP) algorithm has been used in this work. The feed-forward backpropagation network does not have feedback connections, but errors are backpropagated during training. Adjusting the two set of weights between the pair of layers and recalculating the output is an iterative process that is carried on until the error falls below a tolerance level. Learning rate parameters scale the adjustments to weights [13, 14]. The layered diagram of neural network is shown in Figure7.

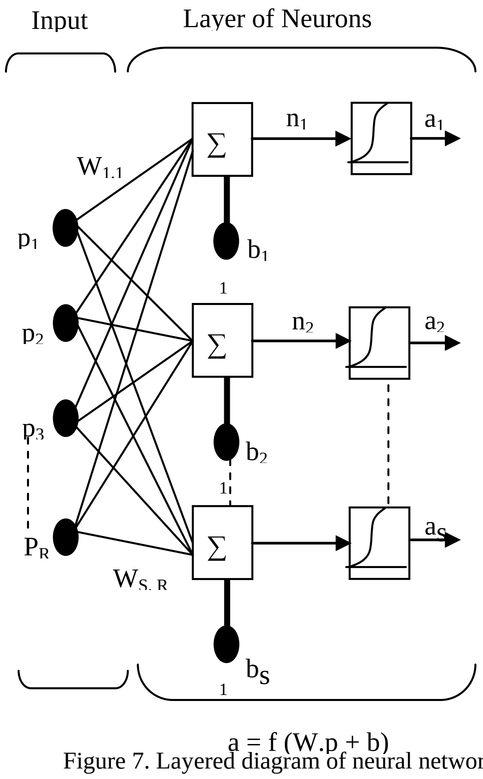

Figure 7. Layered diagram of neural network

In the layered diagram of neural network, $p_1, p_2 \ldots . p_R$ are inputs to the neural networks, $W_{11} \ldots W_{S,R}$ are the weights, $b_1, b_2 \ldots . b_S$ are biases given to the network initially 1 and a_1, $a_2 \ldots a_S$ are outputs of the neural network representing the number of inputs in the layer and S represents the number of neurons in the layer. At the hidden layer, tan sigmoid and at output layer, pure linear functions have been taken.

4.1 Backpropagation Training Algorithms

The simplest implementation of backpropagation learning updates the network weights and biases in the direction in which the performance function decreases most rapidly - the negative of the gradient. One iteration of this algorithm can be written as

$$W_{k+1} = W_k - \alpha_k . g_k$$

Where W_k is a vector of current weights, g_k is the current gradient, and α_k is the learning rate.

If gradient is greater than the threshold value then it performs next iteration. The batch steepest descent training function is traingd. The weights and biases are updated in the direction of the negative gradient of the performance function. There is only one training function associated with a given network.There are various training parameters associated with traingd: epochs, show, goal, time, min_grad, and lr. The learning rate lr is multiplied with the negative of the gradient to determine the changes to the weights and biases. The larger the learning rate, the bigger the step. If the learning rate is made too large, the algorithm becomes unstable. If the learning rate is set too small, the algorithm takes a long time to converge.The training status is displayed for every iteration of the algorithm. The other parameters determine when the training stops. The training stops if the number of iterations exceeds epochs, if the performance function drops below goal and if the magnitude of the gradient is less than mingrad, etc. Learning rate has been taken as 0.2 which is the optimum value.

4.2 Creating a Network (newff)

The first step in training a feedforward network is to create the network object. The function newff creates a feedforward network. It requires four inputs and returns the network object. The first input is an R by 2 Rx2 matrix of minimum and maximum values for each of the R elements of the input vector. The second input is an array containing the sizes of each layer. The third input is a cell array containing the names of the transfer functions to be used in each layer. The final input contains the name of the training function to be used.The following command creates a two-layer network. There is one input vector with two elements. The values for the first element of the input vector range between -1 and 2, the values of the second element of the input vector range between 0 and 5. There are three neurons in the first layer and one neuron in the second (output) layer. The transfer function in the first layer is tan-sigmoid, and the output layer transfer function is linear. The training function is traingd.

Net=newff([-12;0 5],[3,1],{'tansig','purelin'},'traingd');

This command creates the network object and also initializes the weights and biases of the network; therefore the network is ready for training. There are times when you may want to reinitialize the weights, or to perform a custom initialization. The next section explains the details of the initialization process.

4.3 Initializing Weights (init)

Before training a feedforward network, the weights and biases must be initialized. The newff command will automatically initialize the weights, but their reinitialization may be done with command init. This function takes a network object as input and returns a network object with all weights and biases initialized. Here is how a network is initialized (or reinitialized):

```
net = init(net);
```

4.4 Simulation (Sim)

The function sim simulates a network. sim takes the network input p, and the network object net, and returns the network outputs a. sim is called to calculate the outputs for a concurrent set of three input vectors. This is the batch mode form of simulation, in which all of the input vectors are place in one matrix. This is much more efficient than presenting the vectors one at a time.

```
p = [1 3 2; 2 4 1];
a=sim(net,p)
a = -0.1011   -0.2308   0.4955
```

4.5 Training

Once the network weights and biases have been initialized, the network is ready for training. The network can be trained for function approximation (nonlinear regression), pattern association, or pattern classification. The training process requires a set of examples of proper network behavior - network inputs p and target outputs t. During training the weights and biases of the network are iteratively adjusted to minimize the network performance function net.performFcn. The default performance function for feedforward networks is mean square error mse - the average squared error between the network outputs a and the target outputs t.

4.5.1Batch Gradient Descent (traingd)

The batch steepest descent training function is traingd. The weights and biases are updated in the direction of the negative gradient of the performance function. If training a network using batch steepest descent is desired, the network trainFcn should be sat to traingd, and the function train is then called. There is only one training function associated with a given network.
Out of seven training parameters associated with traingd: epochs, show, goal, time, min_grad, and lr, the learning rate lr is multiplied with the negative of the gradient to determine the changes to the weights and biases. The larger the learning rate, the bigger the step. If the learning rate is made too large, the algorithm becomes unstable. If the learning rate is set too small, the algorithm takes a long time to converge. The training status is displayed for every show iteration of the algorithm.The other parameters determine when the training stops. The training stops if the number of iterations exceeds epochs, if the performance function drops below goal, if the magnitude of the gradient is less than mingrad, or if the training time is longer than time seconds. max_fail, which is associated with the early stopping technique is discussed in the section on improving generalization.
The following code creates a training set of inputs p and targets t. For batch training, all of the input vectors are placed in one matrix.

```
p = [-1 -1 2 2;0 5 0 5];
t = [-1 -1 1 1];
```

Next the feedforward network is created. Here e the function minmax is used to determine the range of the inputs to be used in creating the network.

```
net=newff(minmax(p),[3,1],{'tansig','purelin'},'traingd');
```

At this point, some of the default training parameters might be modified as follows:

```
net.trainParam.show = 50;
net.trainParam.lr = 0.05;
net.trainParam.epochs = 300;
net.trainParam.goal = 1e-5;
```

to use the default training parameters, the above commands are not necessary.
Now, the network is ready to be trained.

```
[net,tr]=train(net,p,t);

TRAINGD, Epoch 0/300, MSE 1.59423/1e-05, Gradient

2.76799/1e-10
TRAINGD, Epoch 50/300, MSE 0.00236382/1e-05,

Gradient
0.0495292/1e-10
TRAINGD, Epoch 100/300, MSE 0.000435947/1e-05,

Gradient
0.0161202/1e-10
TRAINGD, Epoch 150/300, MSE 8.68462e-05/1e-05,

Gradient
0.00769588/1e-10
TRAINGD, Epoch 200/300, MSE 1.45042e-05/1e-05,

Gradient
0.00325667/1e-10
TRAINGD, Epoch 211/300, MSE 9.64816e-06/1e-05,

Gradient
0.00266775/1e-10
TRAINGD, Performance goal met.
```

5. Experimental Results

Quadrant Feature Extraction

The matlab v7 has been taken for the implementation of newly developed **QSFE** technique. Experiment has been carried out on 120 samples of training set and 150 samples of test set. Result of this experiment is given below in Table 1.

Table: 6.1 Result of Handwritten English character recognition using QSFE Technique

Input to MLPN	No of Iterations	Training Time (sec)	Classification Time (ms)	Performance on Training set (%)	Performance on Test Set (%)
4 x 1 direction input	50	6.038	7.454	100	96

This technique is giving very good accuracy around 96% and it requires less time to train the network because of the discrete integer calculation being performed and due to more information content available for the network to learn and recognize handwritten English characters easily and quickly. There are small variations in direction values due to which network recognizes the character with high accuracy. Experiment shows that this technique requires less number of iterations for training the network.It also reduces training time. Experiment also shows that only 50 iterations are sufficient for training the network fully.

6. Conclusion & Future Scope

The work present in this paper is an effort towards recognition of hand-written English characters and words. This can be extended to any type of character provided it is given sufficient training time and sufficient inputs. The accuracy of this completed work depends on the level of training that has been given. This work demonstrates the application of MLP networks to the hand-written English character problem. The skeletonized and normalized binary pixels of English characters as well as features of these characters were used as the inputs of the MLP. As future extension to this research work, the recognition accuracy of the network may further be improved by using more training samples and by using more efficient feature extraction techniques. Finally the work cited in this paper is a tiny step towards the completion of a large goal which can bring new possibilities in the field of text recognition.

REFERENCES

[1] Rajawelu, M.T. Husilvi, and M.V.Shirvakar, "A neural network approaches to character recognition." IEEE Trans. on Neural Networks, vol. 2, pp. 307-393, 1989.

[2] Parhami and M. Taragghi, "Automatic recognition of printed Farsi text,"IEEE Pattern Recognition, Vol. no. 8,
pp. 787-1308, 1990.

[3] C. Tappert, C.J. Suen and T. Wakahara,"The state of the art in outline handwriting recognition," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. PAMI-12, no.8, pp.707-808, 1990.

[4]Cheng-Lin Liu, “Normalization-Cooperated Gradient Feature Extraction for Handwritten Character Recognition” IEEE Transactions on Pattern Analysis and

Machine Intelligence, Volume 29, Issue 8, Aug. 2007 Page(s) 1465-1469.

[5] D.S. Yeung, "A neural network recognition system for handwritten Chinese character using structure approach," Proceeding of the World Congress on Computational Intelligence, vo1.7, pp. 4353-4358, Orlando, USA, June 1994.

[6] D.Y. Lee, "Handwritten digit recognition using K nearest-
neighbor, radial basis function and backpropagation neural networks,"IEEE Neural computation, vol. 3, Page(s)
440- 449.

[7] E. Cohen, 1.1. Hull and S.N. Shrikari, "Control structure for interpreting handwritten addresses," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 16, no. 10,
pp. 1049-1055, Oct. 1994.

[8] H. Almualim and S. Yamaguchi, "A method for recognition
of Arabic cursive handwriting," IEEE Trans. on Pattern and Machine Intelligence, vol. PAMI-9, no 5, pp.715-722,
Sept. 1987.

[9] Hailong Liu, Xiaoqing Ding, "Handwritten character recognition using gradient feature and quadratic classifier
with multiple discrimination schemes" Eighth IEEE International Conference on Document Analysis and Recognition, Vol. 1, Page(s): 19-23, August 29-1 Sept., 2005.

[10] I.S.I. Abuhaiba and S.A. Mahmoud, "Recognition of handwritten cursive Arabic Characters," IEEE Transaction on PA&MI vol.16, no 6, pp. 664-672, June 1994.

[11] J. Hertz, A. Krogh and R. Palmer, "Introduction to the theory of neural computation," Addison-Wesley Publishing Company, USA, 1991.

[12 K. Yamada and H. Kami, "Handwritten numeral recognition by multilayered neural network with improved learning algorithm," IJCNN Washington DC, vol. 2, pp. 259-266, 1989.

[13] Neural Computing Theory and Practices by Philip D. Wasserman.

[14] Neural Networks, Fuzzy Logic, and Genetic Algorithms by S. Rajasekaran and G.A. Vijaylakshmi Pai, PHI publication, India.

[15] P. Morasso, "Neural models of cursive script handwriting," IJCNN, WA, vol. 2, pp. 539-542, June 1989.

[16] S.J. Smith and M.O. Baurgoin, "Handwritten character classification using nearest neighbor in large database," IEEE Trans. on Pattern and Machine Intelligence, vol. 16, pp. 915-919, Oct. 1994.

[17] Starzyk,J.A.Ansari, " Feedforward neural network for handwritten character recognition", IEEE International Symposium on Circuits and Systems(ISCAS), Volume 6,
Page(s) 2884-2887, 1992.

[18] Sutha.J, Ramraj.N, "Neural Network Based Offline Tamil
Handwritten Character Recognition System", IEEE International Conference on Computational Intelligence and Multimedia Application, 2007 Volume 2, 13-15, Dec.2007, Page(s): 446-450, 2007.

[19] Verma B.K, "Handwritten Hindi Character Recognition Using Multilayer Perceptron and Radial Basis Function Neural Network", IEEE International Conference on Neural Network, vol.4, pp. 2111-2115, 1995.

[20] W.K. Verma, "New training methods for multilayer perceptrons," Ph.D Dissertation, Warsaw Univ. of Technology, Warsaw, March 1995.

[21 Weipeng Zhang; Yuan Yan Tang; Yun Xue. "Handwritten
Character Recognition Using Combined Gradient and Wavelet Feature" IEEE International Conference on Computational Intelligence and Security, Volume 1, Page(s) 662-667, Nov. 2006.