FREEMAN CODE BASED ONLINE HANDWRITTEN CHARACTER RECOGNITION FOR MALAYALAM USING BACKPROPAGATION NEURAL NETWORKS

Amritha Sampath[1], Tripti C[2] and Govindaru V[3]

[1]Department of Computer Science and Engineering, Rajagiri School of Engineering and Technology, Kochi, India
amrithasampath@yahoo.com
[2]Department of Computer Science and Engineering, Rajagiri School of Engineering and Technology, Kochi, India
triptic@rajagiritech.ac.in
[3]Computational Linguistics, Centre for Development of Imaging Technology, Kerala, India
neithalloor@gmail.com

ABSTRACT

Handwritten character recognition is conversion of handwritten text to machine readable and editable form. Online character recognition deals with live conversion of characters. Malayalam is a language spoken by millions of people in the state of Kerala and the union territories of Lakshadweep and Pondicherry in India. It is written mostly in clockwise direction and consists of loops and curves. The method aims at training a simple neural network with three layers using backpropagation algorithm. Freeman codes are used to represent each character as feature vector. These feature vectors act as inputs to the network during the training and testing phases of the neural network. The output is the character expressed in the Unicode format.

KEYWORDS

Freeman code;Backpropagation Neural Networks; Unicode

1. INTRODUCTION

Optical character recognition (OCR) can be based on conversion of typewritten or printed characters as in textbooks or it can deal with conversion of handwritten text into machine editable form. Both have their own applications. Conversion of handwritten characters is important for making several important documents related to our history, such as manuscripts, into machine editable form so that it can be easily accessed and preserved. A search is difficult when information is available in a form which is not recognizable by the machine. If it is converted into a machine recognizable form, the search becomes fast and easier. The method of conversion of already existing information is called Offline character recognition. Such systems are called OCR systems.

Online recognition is important as an alternate method of data input. Languages like Malayalam have large character set, hence difficult to have a keyboard which can be used easily. So for online data acquisition a *digital pen* or *stylus* can be used. But due to limitations of the device or speed of writing, or tremble, it is possible for a single character to be broken into different parts, hence creating confusion in recognition.

Handwritten character recognition is generally more difficult than conversion of printer and typed characters since, in latter there is a standard set of fonts to which it can be mapped.

DOI : 10.5121/acij.2012.3407

However, handwritings vary from person to person and also for a person it may vary from time to time according to his/her mood, urgency, etc. Hence handwritten character recognition is a difficult task. Existing character recognition software in Malayalam focuses on conversion of printed or typed texts.

Handwritten character recognition has been developed for several languages. But its difficulty in Malayalam can be attributed to several reasons like complexity and similarity in the way characters are written and also due to large character set. Malayalam can be written either in the old lipi or in new lipi as shown in Table 1. Hence number of characters to be recognized will almost be doubled.

Table 1. Characters in old and new lipis.

Old scripts	New scripts
കൃ	കൃ
ഗ്ല	ഗ്ല
ത്ല	ത്ല
ച്ല	ച്ല
ന്ത	ന്ത

Online and offline character recognition requires basically four steps.

1. Pre-processing
2. Feature extraction
3. Recognition
4. Post processing

The method used in these basic steps varies according to the application.

2. RELATED WORK

Though there has been a lot of study on the handwritten character recognition in many languages, an efficient system in Malayalam has not yet been developed. Most of the research has been based on the offline character recognition and on typed text. Malayalam consists of characters with loops and curves, with most of the characters being written in the clockwise direction.

An OCR system for Malayalam has been developed which uses the number of horizontal and vertical lines for the identification of the characters[1]. It includes pre-processing, character extraction and skeltonization phases before the actual recognition takes place. The recognition module include functions which calculate the number and position of horizontal and vertical lines which forms the feature that distinguishes each character from another. Offline recognition of Malayalam characters using chain code histogram and normalised chain code histogram has also been developed[2]. Chain code is used to represent the boundary of the character and is stored as location and direction of line segments of specified length. Centroid of the image is

also taken to improve the result. Online system which uses a combination of context bitmap and normalised (x,y) co-ordinates has also been developed[3]. It uses Kohonen network for recognition. A recognition system developed in Tamil[4], which is another prominent language used in south India, uses a post-processing stage to distinguish between two confusing characters.

3. METHOD OF IMPLEMENTATION

The method proposes different processing techniques for each of the four steps mentioned.

3.1. Pre-processing

Pre-processing includes noise removal.

A noise is a mark made on the writing surface which is not to be taken as a part of the input. Noise will be different from the actual input in its characteristics. A stroke can be defined as a set of points taken from a pen-down position to pen-up position. It is a trajectory followed by the pin tip from the point when it makes the first contact with the writing surface to the point when it leaves the surface. The time taken to make a noise stroke will be either too high or too low when compared with the average time to make a stoke of the actual character. Also, if noise is much away from the rectangular area and number of pixels is less than a threshold, it can be removed as noise.

3.2. Feature Extraction

Feature extraction is the next step after pre-processing. We need to identify unique features that can be used to uniquely identify every character in the character set of the language.

Feature extracted can be either low level or high level. Low level features include width, height, curliness, aspect ratio etc of the character. These alone cannot be used to distinguish one character from another in the character set of the language. So, there are a number of other high level features which include number and position of loops, straight lines, head lines, curves etc. One feature that can be used for identification is *direction information* which is collected online. It is based on Freeman codes as shown in Figure 1.

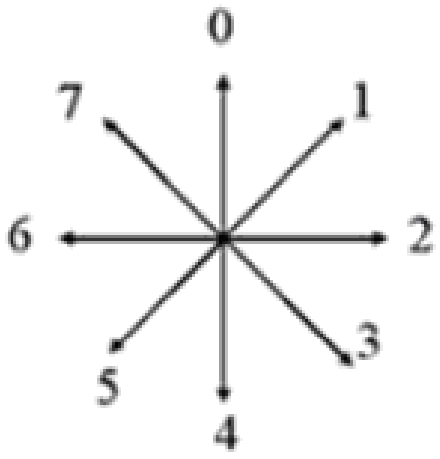

Figure 1. Freeman codes[5]

Starting from the point when first contact is made with the writing surface, direction in which the pen tip moves is recorded. 1 for NE, 2 for E, 3 for SE etc will be stored as a single dimensional array. Direction is recorded only when there is change in direction to avoid

dependence of length of line segments in the character. Also in order to mark crossings of line segments, a character 9 can be used. This array is used as a feature vector for classification.

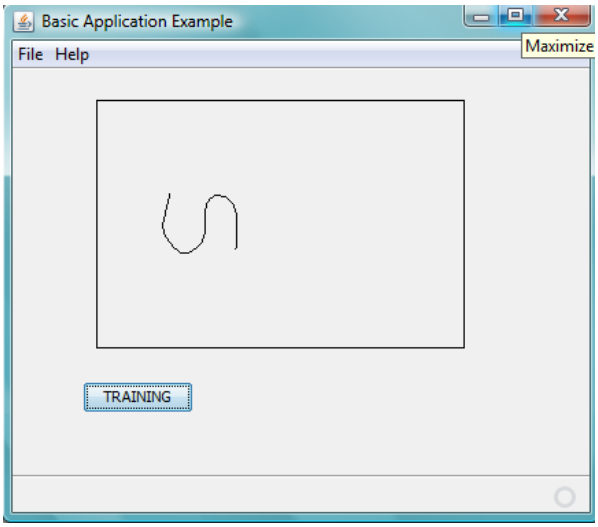

Figure 2. Sample input given during training

Figure 2 shows a sample input 'ga' given as input for training. The input will be coded into feature list which is stored as a linked list. For the given sample input, the feature vector is '[3, 4, 3, 2, 3, 2, 1, 2, 1, 0, 1, 0, 1, 0, 1, 2, 3, 4, 3, 4, 5]'. An issue, that arise when creating the feature vector based on direction of pen movement is that, instead of storing a '1' in the feature vector for the NE direction, it may store it as '2' followed by '0'. This issue arises due to irregularities in writing caused due to the inexperience of the user in using the device, shivering during writing etc. This can be avoided by extracting the direction formed between points 2 pixels apart rather than adjacent pixels. This greatly helps to reduce the size of feature vector and makes it more accurate.

3.3. Classification

Several techniques such as k-Nearest Neighbor (k-NN) [6], Bayes Classifier, Neural Networks (NN), Hidden Markov Models (HMM), Support Vector Machines (SVM), etc exist for the purpose of classification. One of the commonly used techniques is neural networks. Neural networks consist of a number of nodes and links arranged as different layers as shown in Figure 3. Different links which connect different layers are associated with weights. Input to a node is the sum of product of activation and weight associated with the link. The weights must be selected such that inputs map to their corresponding outputs.

Usually a neural network consists of a training phase, a validating phase and a testing phase. During the training phase, features extracted will be used for training the network to map input to output. The *training set* consists of feature vectors for each of the characters that must be recognized by the network. A *training cycle* consists of a forward pass and a reverse pass. Backpropagation algorithm is a supervised training algorithm which is used to train the network by adjusting the weights according to Delta rule.

Initially, the weights in the neural network are assigned small random weights. The input is applied to the network. In each layer, the input is multiplied with the corresponding weights and

an activation function such as sigmoid function is applied at each node. This acts like a squashing function.

The formula of sigmoid activation is: $f(x) = 1/(1 + e^{-input})$.

The output obtained from the last layer is compared with the expected output. This gives the error. This is propagated backwards and the weights associated with the links in each layer is modified as weight(old) + learning rate * output error * output(neurons i) * output(neurons i+1) * (1 - output(neurons i+1))

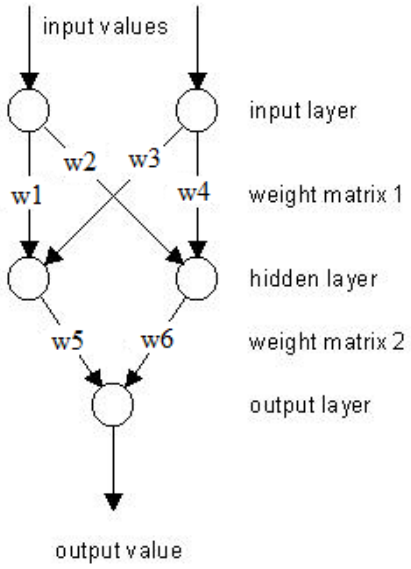

Figure 3. A simple neural network

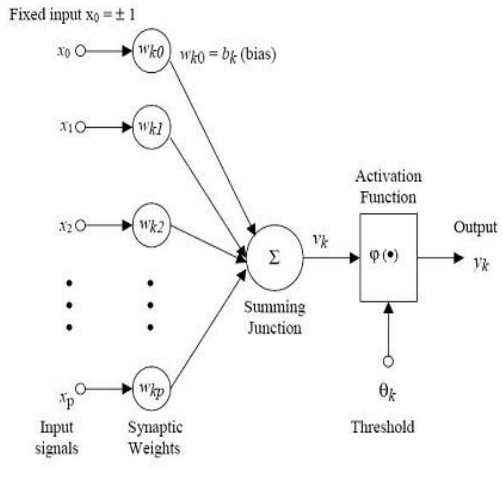

Figure 4. A node in a neural network[7]

After one training cycle is complete, the next input feature vector is applied and the process is repeated for all the feature vectors in the training set. This completes one training *epoch.* The

network requires several training epochs before the network learns to recognize the characters in the training set.

After training, the input feature vectors are applied to test the network. In the validating phase, feature vectors that were not applied during traing is given as input and the results are verified.

3.4. Post-processing

Post-processing [4] involves steps to be taken after classification using neural network is completed. It may include steps like representing the output in Unicode format and also disambiguation of confusing pairs such as 'Pa' and 'Va'(shown in Figure 5).

Figure 5. Pair of characters in confusion set.

This pair will have almost same direction feature vectors. So some additional disambiguating technique should be used for such confusing pairs. Eg. In case of 'Pa' and 'Va', the number of pixels above and below the horizontal axis can be compared. Such disambiguation technique is to be devised as post-processing mechanism for every confusing pair identified during the training of the classifier.

Hence the entire methodology is shown in Figure 6.

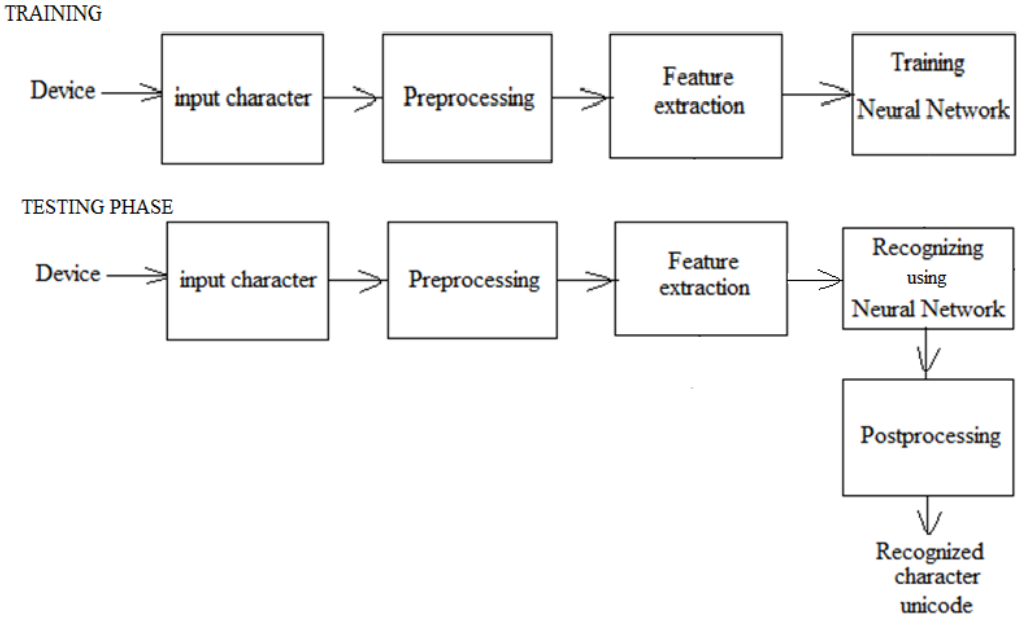

Figure 6. Steps involved in training and testing phases of a classifier network

4. OUTPUT

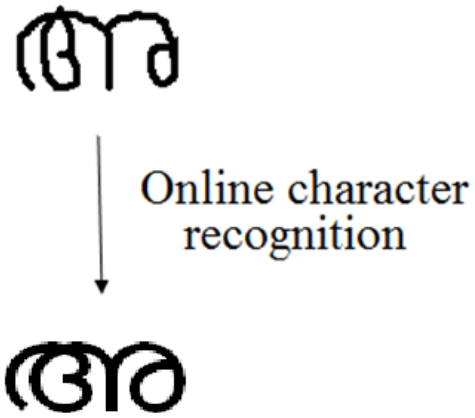

Figure 7. Handwritten character being converted to machine readable form

The handwritten Malayalam acquired by a digital pen or stylus will be converted into editable characters in the computer in one of the recognized fonts of the Malayalam language. Characters are recognised using Unicode 5.01 or above.

3. CONCLUSION AND FUTURE WORK

The method identified is used to recognize single character at a time. Here each character, whether it is a consonant character, a vowel character or a dependent vowel symbol (which usually comes along with consonant character) will be identified as separate character and will be assigned separate unicodes as shown in Figure 8.

Figure 8. Instead of 0D08, we store it as two separate characters with unicodes 0D07 and 0D57

The handwritten Malayalam acquired by a digital pen or stylus will be converted into editable characters in the computer in one of the recognized fonts of the Malayalam language. Characters are recognised using Unicode 5.01 or above.

When trying to extend the system to identify words, an additional step is required during post-processing which combines these two characters, which actually form a single entity, into a single character.

Disambiguation of characters can be done based on the position and meaning each character gives to the word, hence making the system more efficient. Eg. The letters ‘tta’ having Unicode 0D20 and the Malayalam sign *anuswara* having Unicode 0D02, have the same representation. They can be disambiguated based on position and neighboring character rules of the language.
Also, additional mechanisms such as automatic completion of the word, spell checker, etc can be incorporated into the system.

REFERENCES

[1] Abdul Rahiman M, M S Rajasree, Masha N, Rema M , Meenakshi R, Manoj Kumar G, "Recognition of Handwritten Malayalam Characters using Vertical & Horizontal Line Positional Analyzer Algorithm", IEEE, pp 268-274, 2011.

[2] Jomy John, Pramod K. V, Kannan Balakrishnan, "Offline Handwritten Malayalam Character Recognition Based on Chain Code Histogram", Proceedings ofICETECT, pp 736-741, 2011.

[3] Sreeraj.M, Sumam Mary Idicula,"On-Line Handwritten Character Recognition using Kohonen Networks", World Congress on Nature & Biologically Inspired Computing (NaBIC 2009),pp 1425-1430,2009

[4] Suresh Sundaram, A G Ramakrishnan, "An Improved Online Tamil Character Recognition Engine using Post-Processing Methods", 10th International Conference on Document Analysis and Recognition, pp 1216-1220, 2009

[5] Marwan Ali.H. Omer, Shi Long Ma, "Online Arabic Handwriting Character Recognition Using Matching Algorithm", IEEE, pp259-262, 2010

[6] Sreeraj.M, Sumam Mary Idicula, "k-NN based On-Line Handwritten Character recognition system", First International Conference on Integrated Intelligent Computing, pp 171-176, 2010

[7] http://www.learnartificialneuralnetworks.com/

Authors

Amritha Sampath is a postgraduate engineering student in Computer Science and Engineering at Rajagiri School of Engineering and Technology, Kochi, India. She completed her graduation in Computer Science and Engineering and secured high score in Graduate Aptitude Test in Engineering.

Tripti. C is is working as Assistant Professor in the Department of Computer Science and Engineering at Rajagiri School of Engineering & Technology, India. She is a postgraduate in Computer Science and Engineering from CDAC Noida, India and is now pursuing Ph.D from Cochin University of Science and Technology, in the area of vehicular and Adhoc networks. She did her graduation in electronics and communication engineering from Rajagiri School of Engineering and Technology.

Dr. Govindaru V did his Ph.d from ISEC, Bangalore, India. He did his post graduation from Jawaharlal Nehru University, India. Now he is working as head of Research and Development Division in C-DIT, Triruvananthapuram, India.