On-Line Handwritten Character Recognition using Kohonen Networks

Sreeraj.M
Department of Computer Science
Cochin University of Science and Technology
Cochin, India
msreeraj@cusat.ac.in

Sumam Mary Idicula
Department of Computer Science
Cochin University of Science and Technology
Cochin, India
sumam@cusat.ac.in

***Abstract*—This paper presents an efficient Online Handwritten character Recognition System for Malayalam Characters (OHR-M) using Kohonen network. It would help in recognizing Malayalam text entered using pen-like devices. It will be more natural and efficient way for users to enter text using a pen than keyboard and mouse. To identify the difference between similar characters in Malayalam a novel feature extraction method has been adopted-a combination of context bitmap and normalized (x, y) coordinates. The system reported an accuracy of 88.75% which is writer independent with a recognition time of 15-32 milliseconds.**

Keywords- Malayalam handwritten characters; Artificial Neural Network; Feature extraction; Kohonen network (SOM)

I. INTRODUCTION

Man has decided to record history, culture, facts and phenomenon for communication through language. Each language has a script that held the characters of alphabet. Manuscripts formed the major knowledge base where man recorded his ideas and creativity. With the advent of printing technology, sharing of information became easier and thrust for printing technology expanded. Later on information like the typewriter and computer were in demand by the working class to communicate faster. Today the relevance of handwritten notes has increased as data collection has expanded out into the field [1]. It is more convenient to make small handwritten notes for both the technical and non technical people. Since the data collected have to be processed by digital computers these handwritten notes have to be reinterpreted and digitized. Many devices like PDA can record the handwriting and digitize it. Handwritten character recognition has been a popular field of research for many decades. We can find many literatures on the handwriting recognition of Western, Chinese and Japanese scripts, but there are very few related to the recognition of Indic script such as Malayalam. This paper presents a method for recognizing Malayalam handwriting characters using Artificial Neural Network.

Handwriting data is converted to digital form either by scanning the writing on paper or by writing with a special pen or an electronic surface such as a digitizer combined with a liquid crystal display. The two approaches are distinguished as off-line and on-line handwriting respectively. In the on-line case, the two dimensional coordinates of successive points of the writing as a function of time is stored, the order of strokes made by the writer. In the off-line case, the completed writing is available as an image. Even though some effort for the offline recognition of Malayalam characters are reported [2] [3], very little effort was reported for the on-line recognition.

The System described in this paper is developed using Java. The system has four major modules namely data acquisition and preprocessing, Pixel Matrix, Trainer and Character Recognizer. The system is first trained with the character set before using as recognizer.

The organization of the paper is as follows. Section 2 describes Malayalam script characteristics. Section 3 gives the overview of the system architecture, feature extraction and learning algorithm. Section 4 gives the implementation and performance analysis and section 5 represents the concluding remarks.

II. MALAYALAM SCRIPT CHARACTERISTICS

Malayalam is a Dravidian language spoken by about 35 million people. It is spoken mainly in the state of Kerala and in the Lakshadweep Islands.

Notable features of Malayalam script are follows.

A syllabic alphabet in which all consonants have an inherent vowel. Diacritics, which can appear above, below, before or after the consonant they belong to, are used to change the inherent vowel. When they appear at the beginning of a syllable, vowels are written as independent letters. When certain consonants occur together, special conjunct symbols are used which combine the essential parts of each letter. There are about128 characters in the Malayalam alphabet which includes Vowels (15), consonants (36), chillu (5), anuswaram, visargam, chandrakkala-(total-3), consonant signs (3), vowel signs (9), conjunct consonants (57). Out of all these characters mentioned, only 64 of them are considered to be the basic ones as shown in Figure 1.

The properties of Malayalam characters are the following.

- Since Malayalam script is an alphasyllabary of the Brahmic family they are written from left to right.
- Almost all the characters are cursive by themselves. They consist of loops and curves. The loops are written frequently in the clockwise order

978-1-4244-5612-3/09/$26.00 ©2009 IEEE

- Several characters are different only by the presence of curves and loops.
- Unlike English, Malayalam scripts are not case sensitive and there is no cursive form of writing.
- Malayalam is a language which is enriched with vowels, consonants and has the maximum number of sounds that are not available in many other languages as shown in Figure 2.
- Two prominent ways of writing Malayalam scripts exists today. One followed by older generation and the other followed by younger generation. But the later has become standard form even though usage of the former is still common. Some samples are given in Figure 3.

Vowels

അ	ആ	ഇ	ഉ	ഋ	എ	ഏ	ഒ

Consonants

ക	ഖ	ഗ	ഘ	ങ	
ച	ഛ	ജ	ഝ	ഞ	
ട	ഠ	ഡ	ഢ	ണ	
ത	ഥ	ദ	ധ	ന	
പ	ഫ	ബ	ഭ	മ	
യ	ര	ല	വ	ശ	
ഷ	സ	ഹ	ള	ഴ	റ

Dependent Vowel Signs

ാ	ി	ീ	ു	ൂ	ൃ	െ	േ	ൗ

Anuswaram	**Visargam**	**Chandrakala**
ം	ഃ	്

Consonant Signs

ൻ	ൽ	ർ

Chillu

ൻ	ൽ	ർ	ൾ	ൺ

Figure 1. 64 basic characters of Malayalam

ണ, ള, ഴ, റ, ഓ, ഏ, ൾ, ൺ, ർ

Figure 2. Rare Sounds of scripts available only in Malayalam language.

Old scripts	New scripts
ക്ക	ക്കു
ഗ്ഗ	ഗ്ഗു
ത്ത	ത്തു
ച്ച	ച്ചു
ണ്ണ	ണ്ണു

Figure 3.old and new scripts of Malayalam

III. SYSTEM OVERVIEW

The system for on-line recognition of Malayalam characters (OLCR-M) is developed in JAVA using different Kohonen (SOM) algorithms. Figure 4 gives the schematic representation of the system.

A. Pen device & Data sets

The raw data is collected using the device Wacom Graphire 4 CTE-640. Standards are important for the creation of handwriting datasets to ensure that resources created can be used by others. UNIPEN is still the de-facto standard for encoding of handwriting data because of its simplicity and widespread usage [8] [9]. For the attainment of maximum accuracy, a database of 40 writers consisting of 64 basic characters have been collected and trained. Then the test has been done by different schemes based on writer dependant and writer independent strokes.

The system can recognize similar characters in Malayalam scripts using a combination of context bitmap and normalized feature. The system consists of preprocessing feature extraction, training and recognition modules. They are described below.

B. Preprocessing

Prior to any recognition, the captured data is generally preprocessed. It is for reducing spurious noise, normalizing the various aspects of the trace and segmenting the signal into meaningful units [4]. The main steps for noise reduction are smoothing, dot detection and dehooking.

1) Smoothing: Smoothing or filtering is the process of removing unwanted cusps and self intersection. The strokes are smoothened using a larger filter so that unwanted cusps and intersections are removed and we obtain a smooth curve with lesser number of points. Since Malayalam characters are curvaceous in nature, smoothening of curves is essential for recognition.

To remove jitter from the handwritten text, we replace every point (x(t), y(t)) in the trajectory by the mean value of its neighbors:

$$x'(t) = \frac{x(t-N) + \ldots + x(t-1) + \alpha x(t) + x(t+1) + \ldots + x(t+N)}{2N + \alpha}$$

and

$$y'(t) = \frac{y(t-N) + \ldots + y(t-1) + \alpha y(t) + y(t+1) + \ldots + y(t+N)}{2N + \alpha}$$

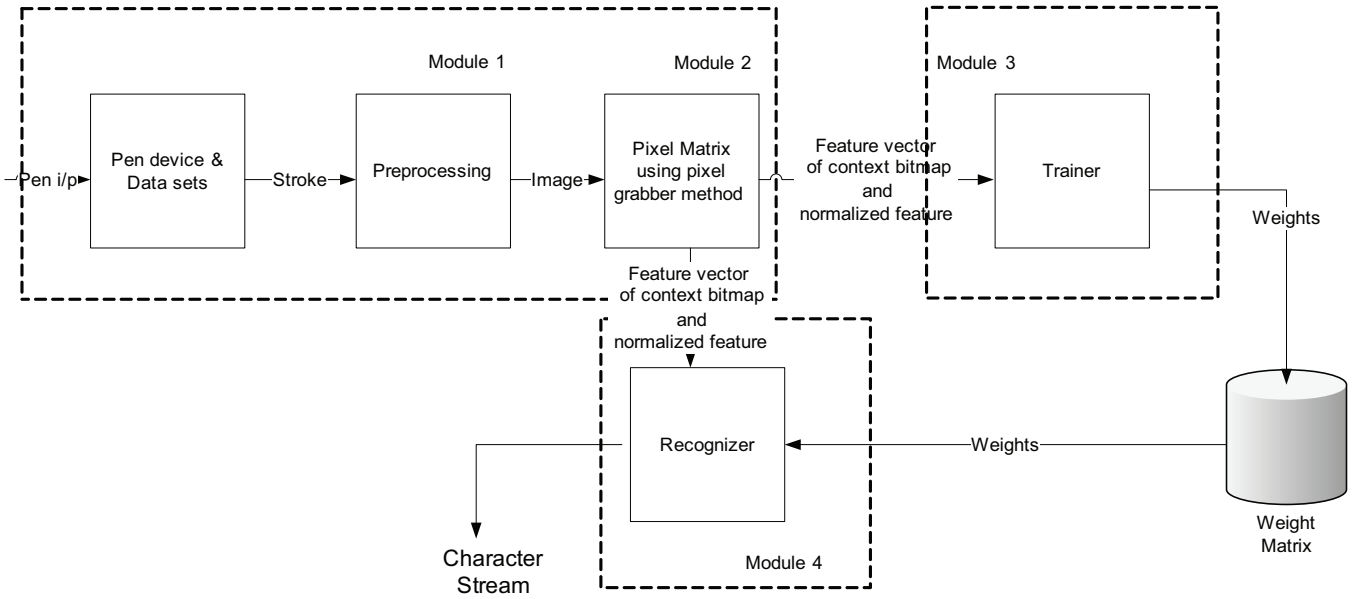

Figure 4. System architecture

The parameter α is based on the angle subtended by the preceding and succeeding curve segment of $(x(t), y(t))$ and is empirically optimized. This helps to avoid the smoothing of sharp edges, which provide important information when there is a sudden change in direction [12].

2) Dot detection: Any stroke, if it has to be considered as a dot should be below the normalized dot size threshold. The threshold value is 0.01 and it is expressed in real length terms (inches) and converted internally to points using the knowledge of the device's spatial resolution. If the width and height is both less than this threshold then it treated as a dot.

3) Dehooking: Dehooking is done to eliminate stray strokes that appear due to inaccuracies in pen down position or rapid erratic motion in placing the stylus on the tablet. Strokes are detected by comparing the number of points with a threshold value. The mark is retained if the value is greater than the threshold value or removes it otherwise. A threshold value of length of 0.13 was chosen for dehooking process.

4) Normalization: Scaling is the next stage in preprocessing. Handwritten characters have different sizes necessitating normalization. Scaling involves the process of converting all characters to the predefined constant width and height. This ensures that each handwritten characters have a canonical representation, so that the size makes no difference in recognition.

5) Equidistant Resampling. This resamples each stroke at equal intervals in space along its trajectory, and removes speed variations while writing across the writers. The resampling is performed such that a constant number of points are obtained from any trace [10].

Finally, the output of the preprocessor module is the image for the creation of context bitmap which is further used in feature extraction. The resulting character image is sharp and of standard size. This makes training and recognition phase more efficient and accurate.

C. Feature Extraction.

This is the module where the features of handwritten characters are analyzed for training and recognition. A preprocessor module dehooks and smoothens the raw data before normalization process.

Feature vector consists of the combination of normalized (x, y) coordinates and context bitmap. The process involves calculation of the character boundary. Once the character boundary is detected the stroke within it is converted into a grey scale bitmap B= {b (i, j)} where b (i, j) indicates the number of normalized points falling into a pixel (i, j).

Assume (x, y) falling into a bitmap pixel (i, j) where a local 24x24 section of bitmap b is centered around (i, j). Further a sequence of low resolution bitmaps L centered around (x, y) by averaging a 3x3 grey scale bitmap is derived [7]. Using the Otsu Algorithm the grey scale bitmap is converted into binary image [11].

All images are also normalized; this prevents the neural network from being confused by size and position.

1) Pixel Grabber Method

This method is to generate the pixel matrix. The image of the character within the boundary is created. The pixel grabber method uses the algorithm [5] in Figure 5.

Finally the combination of the feature of normalized (x, y) coordinates and the binary representation of the image are

presented to the neural network input for training and recognition purposes.

i. *Find the text boundary of the whole image by scanning from top to bottom for upper border Y1, left to right for left border X1, right to left for right borderX2 and bottom to top for lower borderY2 from the top, left, right and bottom side of the drawing area.*
ii. *Scan the binary image from Y1 towards Y2,*
iii. *If there is a black pixel, then scan from X1 to X2 for that particular row to detect any white pixel.*
iv. *If no white pixel found, there is a row gap.*
v. *Repeat steps ii - iv for the whole image to find total number of row gaps.*
vi. *If a black pixel is found then the character pixel considered as '1'.*
vii. *If a white pixel is found then the character pixel considered as '0'.*
viii. *Repeat steps vi – vii for getting the binary representation of the image.*

Figure 5. Algorithm for Pixel grabber

D. Learning algorithm-Kohonen Network

Kohonen neural network is an artificial neural. It can be trained easily and provides fair deal of accuracy in character recognition [6]. Self-organizing maps (SOM) are different than other artificial neural networks in the sense that they use a neighborhood function to present the topological properties of the input space. Like most artificial neural networks, SOMs operate in two modes: training and mapping. Training builds the map using input examples. It is a competitive process and is also called vector quantization. Mapping automatically classifies a new input vector.

A self-organizing map consists of components called nodes or neurons. Associated with each node is a weight vector of the same dimension as the input data vectors and a position in the map space. The usual arrangement of nodes is a regular spacing in a hexagonal or rectangular grid. The self-organizing map describes a mapping from a higher dimensional input space to a lower dimensional map space. The procedure for placing a vector from data space onto the map is to find the node with the closest weight vector to the vector taken from data space and to assign the map coordinates of this node to our vector.

Figure 6 depicts the architecture of Kohonen Network used. The system involves the usage of learning algorithm of Kohonen network consisting of 550 input neurons and 64 output neurons.

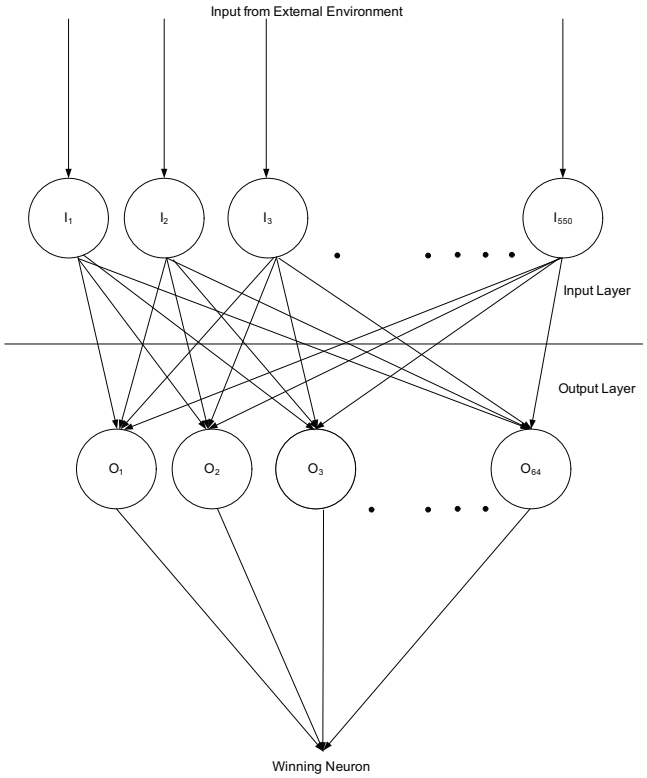

Figure 6. Network Diagram

Learning is the process of selecting a neuron weight matrix that will correctly recognize input patterns. A Kohonen neural network learns by constantly evaluating and optimizing a weight matrix. Table 1 describes the learning algorithm used in the system.

Table 1: Algorithm for Self Organizing Malayalam Character Learning.

Step 0.	Initialize weights w_{ij}.
	Set topological neighborhood parameters.
	Set learning rate parameters.
Step 1.	While stopping condition is false, do steps 2-8.
Step 2.	For each input vector **x**, do steps 3-5.
Step 3.	For each j, compute: $D(j)=\sum_i (w_{ij}-x_i)^2$.
Step 4.	Find index J such that $D(J)$ is a minimum.
Step 5.	For all units j within a specified neighborhood of J, and for all i: $w_{ij}(\text{new}) = w_{ij}(\text{old}) + \alpha[x_i - w_{ij}(\text{old})]$.
Step 6.	Update learning rate.
Step 7.	Reduce radius of topological neighborhood at specified times.
Step 8.	Test stopping condition.

E. Recognizer

The raw stroke data of a user was passed into the preprocessing module. An image that was created through the preprocessing module has to be sharp and standard in size for the generation of the feature vector. The feature extraction module resulted in a combination of context bitmap and normalized feature vector. This feature vector was further passed as an input of the network. The result of the input neuron would be the one nearest to the set of 64 characters provided in the sample.

IV. IMPLEMENTATION

The system was implemented in JAVA. The recognition module was created as a thread. It reduced the average recognition time per character which was found to be 15-32 milliseconds. So the system is faster in recognizing each character. Figure 7 shows the screen shot of one character recognition.

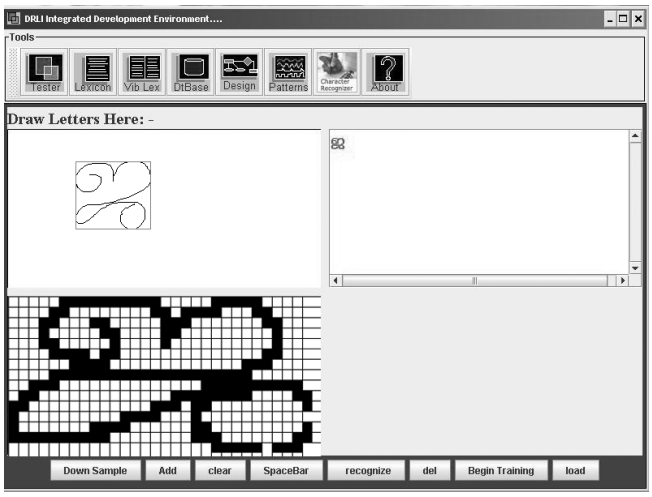

Figure 7. Screen shot of Character 'Ja'

A. Performance Analysis

A total of 2560 samples collected from 40 people are used for training the system. The system was tested according to two schemes. In writer dependent scheme 20 people whose writing samples were used in training phase were put into test. In writer independent scheme writings of 20 new users whose writing samples were not used in training phase were put to test.

Writer dependent testing was named Scheme 1 while writer independent testing was named as Scheme 2. After conducting the test schemes 1 and 2, it was notified that some characters had frequently erroneous nature. For a better quality of result more number of samples of frequently erroneous characters was also included in the training set. Now the training set consisted of 2591 samples. Figure 7 shows samples of erroneous characters.

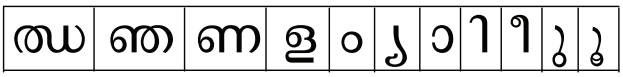

Figure 7. Sample of erroneous characters

Performance of the system is summarized in Table 2.

Table 2: Various schemes and its accuracies

Scheme	Total No. of training samples	Accuracy
1	2560	91.165%
2	2560	86.25%
1	2591	92.65%
2	2591	88.75%

V. CONCLUSION

The system for Online Character recognition for Malayalam developed here was able to read handwritten characters and match them to canonical representations it was trained for. Kohonen Network algorithm was used to train and recognize the character.

The system exhibited an accuracy 88.75% with a recognition time of 15-32 milliseconds in a machine having the configuration of AMD Athalon 2.0 with 512MB RAM. The recognition time was found to be reduced to 0-16 milliseconds in a high performance machine having the configuration of Intel(R) Xeon(R) 5160 @ 3.00GHz with 3.00GB of RAM.

REFERENCES

[1] Plamondon, R., & Srihari, S., "*On-line & Off-line Handwriting Recognition: A Comprehensive Survey*", IEEE PAMI, , 2000, vol. 22, no.1, pp. 63-84.

[2] G.Raju "Recognition of Unconstrained Handwritten Malayalam Characters using Zeero-crossing of Wavelet Coefficients," *Proc. of 14th international Conference on Advanced Computing and Communications,* 2006, pp 217- 221

[3] V.S Roshini, Shanifa Beevi, Revathy, "Machine Recognition of Malayalam Characters," *Proceedings of the International Conference on Cognition and Recognition,* 2005, pp 477-481.

[4] Brijesh Verma, Jenny Lu, Moumita Ghosh, Ranadhir Ghosh "A Feature Extraction technique for Online Handwriting recognition," IEEE International Joint Conference on Neural Networks, 2004, Hungary, IJCNN'04, pp. 1337-43.

[5] Muhammad Faisal Zafar, Dzulkifli Mohamad, and Razib M. Othman "On-line Handwritten Character Recognition: An Implementation of Counterpropagation Neural," *Proceedings of world academy of science,Enginering and Technology volume 10december 2005ISSN 1307-6884*.

[6] Neila Mezghani, Amar Mitiche, Mohamed Cheriet "On-line recognition of handwritten Arabic characters using A Kohonen neural network," *Proceedings of the Eighth International*

Workshop on Frontiers in Handwriting Recognition (IWFHR'02)
2002 IEEE

[7] S. Manke, M. Finke, and A. Waibel. Combining Bitmaps with Dynamic Writing Information for On-Line Handwriting Recognition. In *Proc. of the 12th International Conference on Pattern Recognition,* , 1994, pages 596-598.

[8] UNIPEN format, *http://unipen.nici.ru.nl/unipen.def*

[9] Guyon, I., Schomaker, L., Plamondon, R., Liberman, M., Janet, S., "UNIPEN Project of Online Data Exchange and Recognizer Benchmarks", International Conference on Pattern Recognition (ICPR 1994) ,Jerusalem, Israel(October 1994).

[10] Handwritten Gesture Recognition for Gesture Keyboard Balaji R., Deepu V., Madhvanath S., Prabhakaran J. In Tenth International Workshop on Frontiers in Handwriting Recognition (2006).

[11] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms", IEEE Transactions on Systems, Man, and Cybernetics, 1979.

[12] S.Jaeger, S. Manke, J. Reichert, A. Waibel, "Online handwriting recognition: the NPen++ recognizer,"*International Journal on Document Analysis and Recognition,* March, 2001, vol.3 (3,) pp. 169-180.