Hand-Written Malayalam Character Recognition
An Approach Based On Pen Movement

Jayababu G and Sumam Mary Idicula

Jayababu G, Sumam Mary Idicula,
Department of Computer Science, Cochin University Of Science & Technology
Cochin, Kerala, INDIA

ABSTRACT

In this paper we introduce a novel approach for character recognition based on the pen movement i.e., recognition based on sequence of pen strokes. A Back-propagation Neural Network is used for identifying individual strokes. The recognizer has a two-pass architecture i.e., the inputs are propagated twice through the network. The first pass does the initial classification and the second for exact recognition. The two-pass structure of the recognizer helped in achieving accuracy of about 95 percent in recognizing Malayalam letters. The training set contains samples of all independent strokes that are commonly used while writing Malayalam. Input values to the network are the directions of pen movement. A "minimum error" technique is used for finding the firing neuron in the output layer. Based on the output of First-Pass the network is dynamically loaded with a fresh set of weights for exact stroke recognition. Analyzing the stroke sequences identifies individual characters. This work also demonstrates how a statistical pre-analysis of training set reduces training time.

Key Words

Character Recognition, Neural Networks, Statistical Analysis.

1.0 INTRODUCTION AND MOTIVATION

The objective of this work is to build an efficient recognizer for Hand-written Malayalam letters. Malayalam is one of the prominent regional languages of Indian sub continent. Malayalam language has more than 100 commonly used characters that contain vowels, consonants, prefix & suffix symbols as well as joined letters. But a computer keyboard couldn't support all these characters, which restricted the native users from using joined letters. For inputting a joined letter he has to use a combination of more than one symbol, which is really awkward and not of his common practice. This motivated us to develop an inputting system that helps the users to enter Malayalam text in a natural manner. As pen like devices such as stylus came as the convenient input devices, recognition of characters based on pen movement became the major task.

As it is very difficult to codify the rules for recognizing a particular character, we selected neural network for learning the rules. Back-propagation network, which uses supervised algorithm, is used for learning different independent pen strokes. The network is trained using a certain number of samples of the pen-strokes. The learned network is used for recognition.

The recognizer has a two-pass structure. The inputs to the network are the directions of pen movement. Eight values are given to the eight possible directions. Once a pen-stroke is over, a certain number of equal-distant pen directions are taken as inputs. First the inputs are given to the neural network loaded with the weights for initial classification. This is the first pass of the recognizer, the output of which gives the possible set of strokes. During the second pass the network dynamically loads new set of weights based on the output of the first pass. The inputs are again applied to the network to recognize the exact stroke. The streams of strokes are given to the analyzer, which identifies the exact letter based on the sequence.

In this report we include the details of pen-stroke recognition, which is the core of the work and an abstract design for continues text recognizer.

The work is aimed at achieving the following objectives: -

- To build a recognizer, that recognizes the hand written Malayalam letters based on pen movement.
- The recognizer must have the ability to identify letters irrespective of their size.
- The slight variations of writing styles should not affect recognition process.
- There should be some techniques for reducing the training time.

2.0 GENERAL CHARACTERISTICS OF MALAYALAM HAND WRITING

This section gives an overview of Malayalam letters as well as the way of writing them by a layman. Normally Malayalam letters are written by individual stroke of pen. So by analyzing the sequence of strokes it is easy to identify a letter. The basic pen movement to print one stroke is almost same for most of the persons. The use of joint letters is also common practice. Many independent strokes are used for writing Malayalam letters. Table 1 contains some examples of the handwritten Malayalam letters.

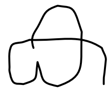
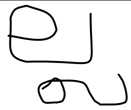
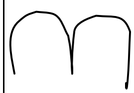
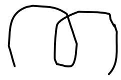
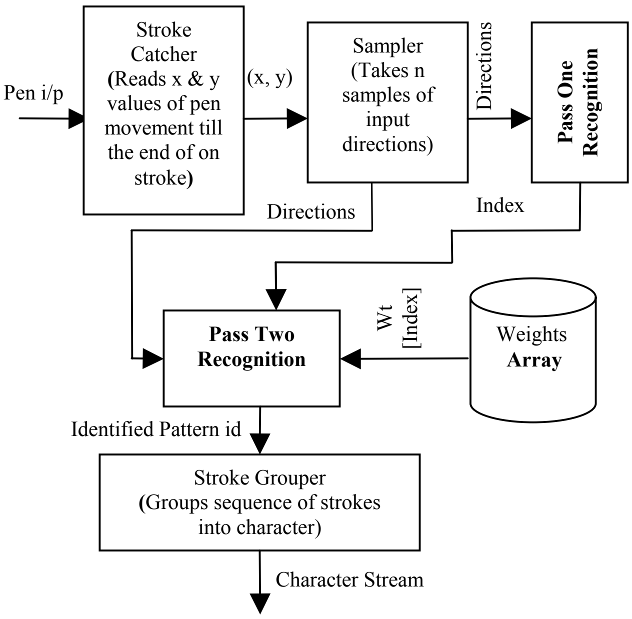
Example of a letter written by a single stroke	Example of a joint letter written by more than one independent strokes	Examples for letters that are written by almost similar pen strokes.	
			
Letter “KA”	Letter “LLA”	Letter “NA”	Letter “THA”

Table 1 :Malayalam letters

This table gives some idea about the difficulty of character recognition based on pen strokes. Some letters are written by single stroke, while others by a sequence of strokes. The problem of dealing with extremely similar-stroked characters is also a matter of consideration.

3. 0 CONTINUES TEXT RECOGNIZER



4.0 PEN-STROKE RECOGNIZER

The pen-stroke recognizer contains a Back-propagation [5] neural network to codify the features of each stroke and this knowledge is used for recognition as well. The following sections describe the details of the neural structure, training and recognition.

4.1 The ANN Architecture

Through years artificial neural networks (ANN) are used as an effective tool for pattern recognition in many areas such as image processing, speech recognition etc [3]. In this particular problem also, it is very difficult to give precise rules for identifying each pen-stroke. A neural network can codify these rules in a better manner if trained properly.

Out of the many ANN models, Back-propagation (BP) network is the most commonly used and the most flexible one. Unlike some networks like Perceptron, ART1 etc where inputs are restricted to binary values, BP can use real values as well [2]. In our problem also we had to deal with the real values.

The BP network contains one input layer, one output layer and any number of hidden layers. There is no restriction over the number of neurons in each layer. Trial and error is the only way to find the optimal number of neurons and layers for learning a set of patterns [4]. Neurons in one layer are connected to the neurons in the next layer through a weighted link. The activation for a particular neuron is the sum of products of weights and inputs.

Consider a network contains **n** inputs and **m** neurons in the next layer. Input can be represented by a row matrix I [n] and weights can be represented by two-dimensional matrix Wt [n][m], where each column contains connection weights of each input neuron to a neuron in the next layers. So the activation of i^{th} neuron can be calculated by the following equation: -

$$NET_i = \sum_{j=1}^{n} I_j * Wt_{ji} \qquad (1)$$

The activation of the i^{th} neuron is calculated using sigmoid function [2]: -

$$OUT_i = 1 / (1 + e^{-NET_i}) \qquad (2)$$

Where $_i$ ranges from 1 to m.

In our problem we used a neural network model that contains 30 input, 60 hidden and 10 output neurons. A “minimum error” technique is used to find out the firing neuron and is given in Section 4.6.

4.2 Input Value Selection

The input values are selected based on the direction of the pen movement. Two consecutive points are selected and the x & y values are compared to get the direction of pen movement. In our experiments we used the following set of values for the eight directions.

Table 2: Pen movement values

←	↖	↑	↗	→	↘	↓	↙
0.01	0.15	0.29	0.43	0.57	0.71	0.85	0.99

Consecutive values are separated by a difference of 0.14, which is the maximum possible within the range 0.01 to 0.99.

4.3 Preparing Training Set

The training set is prepared by taking 20 samples of each of the independent pen-stroke and total set contains 2000 training-patterns. As the neural network contains 30 input nodes, each training pattern contains 30 equal-distant direction-values. There are only eight directions and the values for each direction is given in Table 1.

4.4 Setting Target Values

The target values for the ten output neurons are assigned within the range 0.05 to 0.95, separated by a difference of 0.1 between successive ones. So the selected target values are: -

Table 3: Target values for output neurons

Output Neuron	1	2	3	4	5	6	7	8	9	10
Target Value	0.05	0.15	0.25	0.35	0.45	0.55	0.65	0.75	0.85	0.95

To associate a target value to a particular stroke, a statistical analysis conducted (as explained below) on the entire training set that contains 20 samples of 100 possible patterns.

Let T_i represents the input values for directions and AVG_i is the average number of i^{th} direction present for a particular stroke. Now we can calculate the factor F_j for the j^{th} stroke using the following equation: -

$$F_j = \sum_{i=1}^{8} T_i * AVG_i \qquad (3)$$

The stroke having minimum F value is selected as the **initial stroke**. A factor **distance from initial stroke** D_i of the i^{th} stroke is calculated using following equation: -

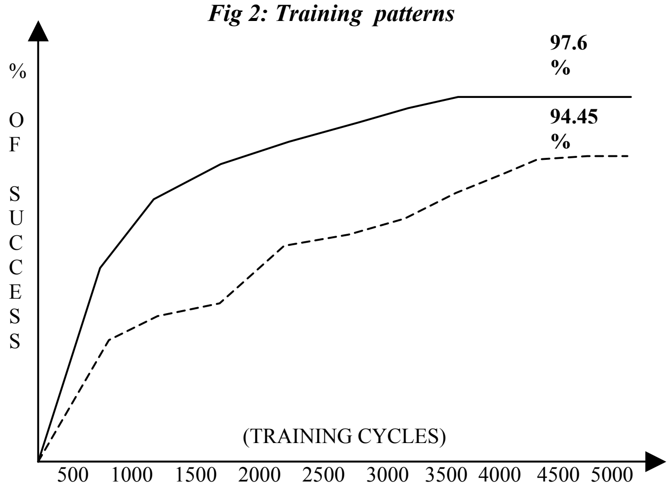
$$D_i = \sum_{j=1}^{30} |X_j - Y_j| * W_j \qquad (4)$$

Where X_j is the mean value of j^{th} input (that is going to apply to j^{th} input neuron during training) of **initial stroke** and Y_j is the corresponding input value of i^{th} stroke (the stroke for which the factor is calculating). Wj is a weight factor of the jth neuron, unequal weights are given for each neuron in the input layer. The weight factor is used for magnifying the difference between input values (for particular neuron) of two strokes.

The strokes are sorted based on the ascending order of **D value** and classified into ten groups that contain ten strokes in each one. Target values for the Pass1 classifier (group level classification) are given in such a way that the group contains low **D valued** strokes will get the lowest value (0.05) and the rest based on their order. Target values for the Pass 2 classifier is given to a group member (stroke) according to the order of it inside a group. Thus the values (0.05, 0.15, ……, 0.95) are assigned to strokes based on the ascending order of **D value** within the group. From the discussion it is clear that the stroke marked as **initial stroke** will be having a **D value** of zero and will be the first member in the first group.

Figure 2 shows the difference felt while we attempted to train neural network by classifying strokes based on two different factors. The training when the strokes are classified using F value (Equation 3) is represented using dotted line and when D value (Equation 4) is used for classification is represented by solid line. For this experiment we used a constant training rate of 0.05.

Fig 2: Training patterns



So the main benefits of setting target values based on this statistical pre-analysis of training set are: -

- Strokes having similar characteristics are grouped together.
- Best-fit target values are assigned to output neurons.
- The total training time reduced.
- The neural network easily converged to the solution without showing oscillation between values.
- The Back-propagation neural network never showed the problem of local-minima [4].

4.5 Training Using Back-propagation Algorithm

The network learns the hidden rules in patterns during training by adjusting its weights. There are 11 set of weights used by the recognizer, one for group level recognition in pass-1 and the remaining ten (one for each group) for exact stroke recognition in pass-2. As the network architecture remains the same for both pass-1 & pass-2, the only change for training another Pass/Group is that of loading a separate set of weights corresponding to that group. The training for the Pass-1 recognition is conducted by using alternate stroke-patterns from each group. A separate stroke within each group is selected for next iteration. The training of stroke-level recognition (Pass-2, ten groups) is done separately for each group using a separate set of weights. Only the stroke-patterns within a particular group are used for training. The basic training technique is common for all of these 11 groups and is narrated below.

Back-propagation algorithm is used for training the network [2]. First all connection weights are initialized with some random values between –1 and +1. The inputs are applied one by one to the neurons in the input layer and the activation of the output layer neurons are calculated using equation 2. The neuron corresponds to the input pattern are set to exact target value, all others are set to a target value 0.2 more than that of original. For example if the input pattern is for Neuron-5 then the target value for it is set to exact value (i.e., 0.45) and all other neurons are set with a target value 0.2 more than that of its original (Neuron-1's target value is set as 0.25 where its actual value was 0.05). The difference between the actual and target value is calculated as the error of each neuron. This error is propagated back using the following method: -

Let OUT is the output layer neuron activation, OUTH is of hidden layer neuron and I represent the input vector. TARGET is the vector contains the desired target values. We can calculate the error of i^{th} output neuron using following equation.

$$\Delta_i = OUT_i * (1 - OUT_i) * (TARGET_i - OUT_i) \text{ --------------------(5.4)}$$

The error j^{th} hidden layer neuron is: -

$$\Delta H_j = OUTH_j * (1 - OUTH_j) * \sum_{i=1}^{10} (\Delta_i * Wt_{ji}) \qquad (5)$$

Where Wt_{ji} is the weight of the connection between j^{th}-hidden neuron to i^{th} output neuron.

Errors are propagated back by adjusting the weights. The new weight between j^{th} hidden neuron and i^{th} output neuron is calculated as: -

$$Wt_{ji} = Wt_{ji} + (\eta * \Delta_i * OUTH_j) \qquad (6)$$

The new weight between k^{th} input neuron and j^{th} hidden neuron is: -

$$Wt_{kj} = Wt_{kj} + (\eta * \Delta H_j * I_k) \qquad (7)$$

The factor η is the learning rate of the network. Normally learning rate is selected as a real value between 0 and 1. Low rate is used for slow learning and high rate is used for faster learning [2]. We used different rates at different stages of learning. During Pass 1 training, for initial iterations we used learning rate 0.5 and later stages learning rate of 0.02 is used. Table-4 gives the details of Pass 1 training.

Table 4. Pass1 training values

Learning Rate	**No of Iterations**	**% Of Successful recognitions**
0.5	100	85.3 %
0.2	100	91.8 %
0.1	100	93.4 %
0.05	500	95.9 %
0.02	700	97.6 %

The success % is based on the correct recognition of testing set patterns. For Pass 2 training (recognizing a stroke within a group) only an average of 250 cycles with a training rate of 0.05 was enough for getting success of more than 99%. So collectively the recognizer shows about 95% accuracy in recognizing testing patterns.

4.6 Recognition Using "Minimum Error"

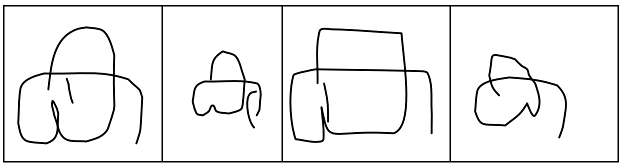
During recognition phase, initially the network is loaded with the weights for Pass-1 group level classification. The 30 equal-distant samples of directions are taken as input and applied to the network. The activation in the output layer neurons are calculated using Equation-2. Instead of threshold value [3], a "Minimum error" technique is used for finding the firing neuron i.e., out of the ten output neurons, the one showing minimum deviation from the actual target value (given in Table-3) is selected as the neuron pointing to the group containing the input stroke. The network is then loaded with the weights of the group that is pointed by the Pass-1 neuron for Pass-2 recognition. The same inputs used in Pass-1 are applied again to the network and the firing neuron is selected using the above method. Now the firing neuron points to the exact stroke. The identified stroke is given to the **stroke grouper**, which identifies letters by analyzing the sequence of strokes.

The design of the continues-text recognizer is given in Section-4 and a screen shot showing results of the recognition is given in table 6.

5. PERFORMANCE ANALYSIS

The network shows very good performance in recognizing letters in a font and size independent manner. For example the network is able to recognize the following variations of Malayalam letter "KA" given in table 5, out of which the first one is the exact letter.

Table 5. Variants of letter "KA"



In other character recognition techniques, if the letter is larger or smaller then it must be scaled to some grid size before starting recognition [1]. This requires lot of processing time and errors will occur during scaling. But this new approach eliminates the need of scaling.

The major demerit of this approach is that, the training is conducted group-by-group and requires lot of patience. But there is a noted benefit for this, each group can be trained and tested separately. This will help in breaking the job of training and can be distributed among processors. As each group contains just ten independent target outputs, the network also shows faster convergence.

6. RESULTS

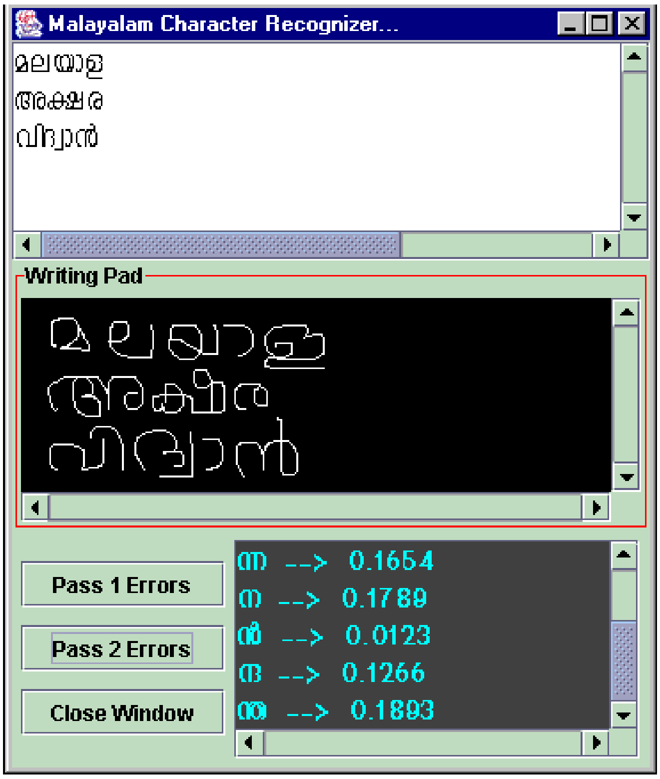


Table 6 : Sample screen shot

7. CONCLUSION

The research project achieved the target of creating an effective inputting technique for entering Malayalam text. Even though we experimented on Malayalam characters, this approach will be applicable to most of the Indian languages, which uses independent strokes of pen for writing words rather than the running letters that are found in English. This approach will be useful for hand held devices such as PDA, where Stylus is used as the input device. The study also gives some guidelines for creating a character recognizer that identifies letters in a size and font independent manner. The importance of analyzing the entire training set before training to reduce the convergence time is also demonstrated.

The future directions of this work are: -

- Further study is required to check the applicability of Back-propagation network that contains more hidden layers instead of Two-Pass structure of the recognizer. As the output layer contains neurons for all strokes, we can train the network as a single group.
- Extensive study is required to check the applicability of other statistical techniques such as Discriminative Statistics for grouping similar strokes together.
- A recurrent neural network [1] that takes decisions considering current inputs and previous outputs can be used for stroke grouping.
- An extensive study is required to check the effective application of NLP techniques in reducing recognition errors.

REFERENCES

[1]Dan W. Patterson. (1996) Artificial Neural Networks-Theory & Practice, ISBN 0-13-295353-6, pp 245-317, Prentice Hall, Singapore.

[2]Dr Valluru B. Rao & Hayagriva V. Rao. (1996) C++ Neural Networks & Fuzzy Logic, ISBN 81-7029-694-3, pp 109-176, BPB Publications, New Delhi.

[3]Patrick Henry Winston. (1992) Artificial Intelligence, ISBN 0-201-52552-6, pp 443-468, Addison Wesley, Reading.

[4]James A. Freeman & David M. Skapura. (1991) Neural Networks-Algorithms, Applications and Programming Techniques, ISBN 0-201-51376-5, pp 89-124, Addison Wesley, Reading

[5]Mohamad H. Hassoun (1998) Fundamentals of Artificial Neural Networks, ISBN-81-203-1356-9, pp 197-206, Prentice-Hall India, New Delhi.