Preprocessing Algorithms for the Recognition of Tamil Handwritten Characters

N Shanthi K Duraiswamy

Abstract

Handwriting has continued to persist as a means of communication and recording information in day-to-day life even with the introduction of new technologies. Handwriting is a skill that is personal to individuals. Recognition of characters is an important area in machine learning. Widespread acceptance of digital computers seemingly challenges the future of handwriting. However, in numerous situations, a pen together with paper or a small notepad is much more convenient than a keyboard. Handwriting data is converted to digital form either by scanning the writing on paper or by writing with a special pen on an electronic surface such as a digitizer combined with a liquid crystal display. The two approaches are distinguished as offline and online handwriting respectively. It is necessary to perform several document analysis operations prior to recognizing text in scanned documents. This paper presents detailed analysis of various preprocessing operations performed prior to recognition of Tamil handwritten characters and the results are shown.

Keywords : Indian Language, Handwriting Recognition.

0. Tamil Language

Tamil which is a south Indian language, is one of the oldest languages in the world. It has been influenced by Sanskrit to a certain degree[2]. But Tamil is unrelated to the descendents of Sanskrit such as Hindi, Bengali and Gujarati. Most Tamil letters have circular shapes partially due to the fact that they were originally carved with needles on palm leaves, a technology that favored round shapes. Tamil script is used to write the Tamil language in Tamil Nadu, SriLanka, Singapore and parts of Malaysia, as well as to write minority languages such as Badaga. Tamil alphabet consists of 12 vowels, 18 consonants and one special character (AK). Vowels and consonants are combined to form composite letters, making a total of 247 different characters and some Sanskrit characters. The complete Tamil alphabet and composite character formations are given in [5]. The advantage of having a separate symbol for each vowel in composite character formations, there is a possibility to reduce the number of symbols used by the alphabet.

1. Steps involved in Handwriting Recognition System

The major steps involved in recognition are shown in Fig.1. They are

1. Preprocessing and segmentation
2. Feature Extraction
3. Classification
4. Post processing

This paper presents about the first stage of handwriting recognition system known as preprocessing and the various steps to be performed before the recognition of Tamil handwritten characters.

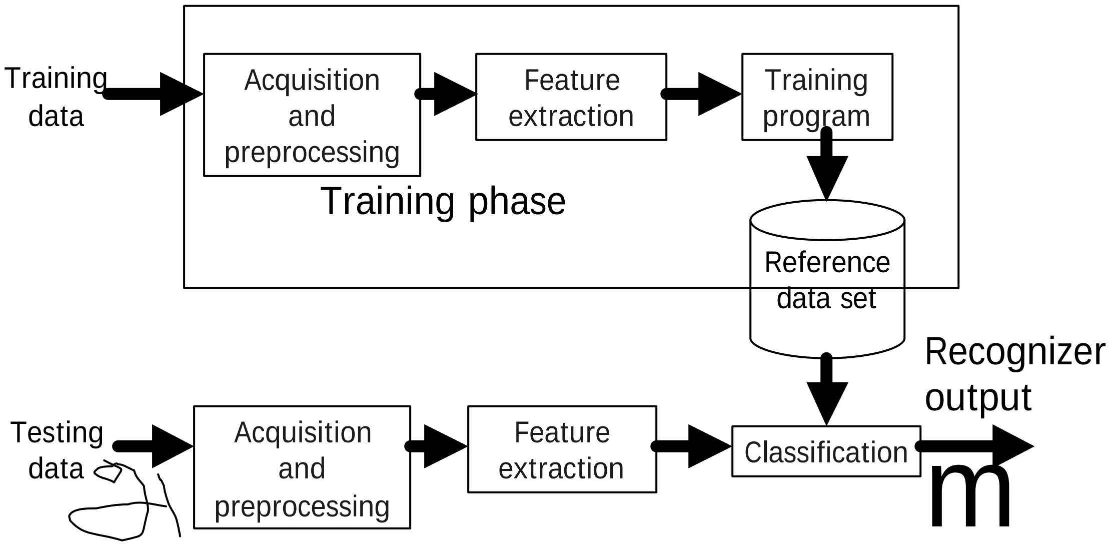

Fig.1. Steps in handwriting recognition system

2. Preprocessing

The raw input of the digitizer typically contains noise due to erratic hand movements and inaccuracies in digitization of the actual input. Original documents are often dirty due to smearing and smudging of text and aging [1]. In some cases, the documents are of very poor quality due to seeping of ink from the other side of the page and general degradation of the paper and ink. Preprocessing is concerned mainly with the reduction of these kinds of noise and variability in the input. The number and type of preprocessing algorithms employs on the scanned image depend on many factors such as paper quality, resolution of the scanned image, the amount of skew in the image and the layout of the text. Some of the common operations performed prior to recognition are: thresholding, the task of converting a gray-scale image into a binary black-white image; skeletonization, reducing the patterns to thin line representation; line segmentation, the separation of individual lines of text; and character segmentation, the isolation of individual characters [3].

3. Thresholding

The task of thresholding is to extract the foreground from the background. A number of thresholding techniques have been previously proposed using global and local techniques. Global methods apply one threshold to the entire image while local thresholding methods apply different threshold values to different regions of the image[Leedham]. The histogram of gray scale values of a document image typically consists of two peaks: a high peak corresponding to the white background and a smaller peak corresponding to the foreground. So the task of determining the threshold gray-scale value is one of determining as optimal value in the valley between the two peaks. Here Otsu's method of histogram-based global thresholding algorithm is used and is described below [6].

4. OTSU's Method for Image Thesholding

An image is a 2D grayscale intensity function, and contains N pixels with gray levels from 1 to L. The number of pixels with gray level i is denoted f_i, giving a probability of gray level i in an image of

$$p_i = f_i / N \tag{1}$$

In the case of bi-level thresholding of an image, the pixels are divided into two classes, C_1 with gray levels [1, …, t] and C_2 with gray levels [t+1, …, L]. Then, the gray level probability distributions for the two classes are

C_1: $p_1/ù_1(t)$, …. $p_t/ù_1(t)$ and

C_2: $p_{t+1}/ù_2(t)$, $p_{t+2}/ù_2(t)$,…, $p_L/ù_2(t)$,

where $$ù_1(t) = \overset{t}{\underset{i?1}{?}} p_i \tag{2}$$

and

$$ù_2(t) = \overset{L}{\underset{i?t?1}{?}} p_i \tag{3}$$

Also, the means for classes C_1 and C_2 are

$$ì_1 = \overset{t}{\underset{i?1}{?}} \mathrm{i}\, \mathrm{p}_i \,/?_1(t) \tag{4}$$

and

$$ì_2 = \overset{L}{\underset{i?t?1}{?}} \mathrm{i}\, \mathrm{p}_i \,/?_2(t) \tag{5}$$

Let $ì_T$ be the mean intensity for the whole image. It is easy to show that

$$ù_1ì_1 + ù_2ì_2 = ì_T \tag{6}$$

$$ù_1 + ù_2 = 1 \tag{7}$$

Using discriminant analysis, Otsu defined the between-class variance of the thresholded image as

$$ó_B^2 = ù_1 (ì_1 - ì_T)^2 + ù_2 (ì_2 - ì_T)^2 \tag{8}$$

For bi-level thresholding, Otsu verified that the optimal threshold t* is chosen so that the between-class variance $ó_B^2$ is maximized; that is,

$$t^* = \mathrm{Arg} \quad \mathrm{Max}\,\{ó_B^2(t)\} \tag{9}$$

1d” t< L

5. Skeletonization

Skeletonization is the process of peeling off a pattern as many pixels as possible without affecting the general shape of the pattern. In other words, after pixels have been peeled off, the pattern should still be recognized. The skeleton hence obtained must be as thin as possible, connected and centered. When these are satisfied the algorithm must stop. A number of thinning algorithms have been proposed and are being used. Here Hilditch’s algorithm is used for skeletonization [4].

Consider the following 8-neighborhood of a pixel p_1

P9	P2	P3
P8	P1	P4
P7	P6	P5

Consider a decision is to be taken whether to peel off P1 or keep it as part of the resulting skeleton. For this purpose the 8 neighbors of P1 in a clock-wise order and two functions are defined.

B(P1) = number of non-zero neighbors of P1

A(P1) = number of 0,1 patterns in the sequence P2,P3,P4,P5,P6,P7,P8,P9,P2

The algorithm consists of performing multiple passes on the pattern and on each pass, the algorithm checks all the pixels and decide to change a pixel from black to white if it satisfies the following four conditions.

2 <= B(P1) <= 6

A(P1) = 1

P2.P4.P8=0 or A(P2)!=1

P2.P4.P6=0 or A(P4)!=1

Stop when nothing changes. Hilditch's algorithm is a parallel-sequential algorithm. It is parallel because at one pass all pixels are checked at the same time and decisions are made whether to remove each of the checked pixels. It is sequential because this step just mentioned is repeated several times until no more changes are done.

6. Line Segmentation

Segmentation of handwritten text into lines, words, and characters has many sophisticated approaches. This is in contrast to the task of segmenting lines of text into words and characters, which is straight forward for machine-printed documents. It can be accomplished by examining the horizontal histogram profile.

7. Character Segmentation

Line separation is usually followed by a procedure that separates the text line into characters. Vertical histogram profile is used to separate the characters.

8. Experimentation and Results

The input image and the results of various preprocessing algorithm is shown below.

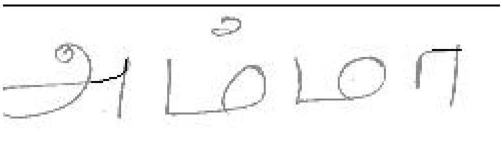

Fig.2 Original image

Fig.2 shows the original image which is used for the process of preprocessing. Data sample was collected and it was scanned using a flat-bed scanner at a resolution of 100 dpi and stored as 8-bit gray scale images.

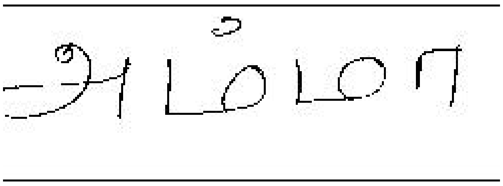

Fig.3 Thresholded image

Fig.3 shows the binary image after applying Otsu's global thresholding method to the image shown in Fig.1.

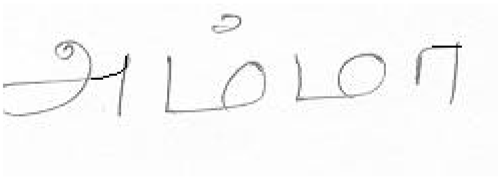

Fig.4 Skeleton of the image

Fig.4 is the skeleton of the image obtained after applying Hilditch's skeletonization algorithm.

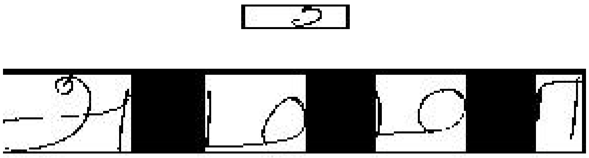

Fig.5 Segmented image

Fig.5 is the image obtained after applying the segmentation algorithm to the skeleton of the image.

9. Conclusion

This paper presents number of preprocessing algorithms that has to be performed before the process of feature extraction for Tamil handwritten character recognition system and the results are shown. The result shows that the algorithms are working reasonably well with sufficient accuracy. This work can be further extended by including few other preprocessing activities like smoothing of images, Slant correction, edge linking and size normalization. The preprocessed image can be given as input to the feature extraction phase.

10. References

1. Leedham et.al., “Comparison of some thresholding algorithms for text/background segmentation in difficult document images”, ICDAR 2003.
2. S.Hewavitharana, H.C.Fernando, “A two stage classification approach to Tamil Handwriting recognition”, Tamil Internet 2002, California, USA, pp.118-124
3. Srihari, “Online and Offline handwriting recognition: A comprehensive survey”, IEEE PAMI, Vol.22, No.1, Jan.2000.
4. C.J.Hilditch, “Comparison of thinning algorithms on a parallel processor”, Image Vision Computing, pp.115-132,1983.
5. P.Chinnuswamy and S.G.Krishnamoorthy, “Recognition of hand printed Tamil characters”, Pattern recognition Vol.12, pp141-152,1980.
6. N.Otsu, “A threshold selection method from grey level histogram”, IEEE Transaction. Syst. Man Cyber., vol.9 no.1, 1979, pp. 62-66.

About Authors

N Shanthi is a Assistant Professor in K. S. Rangasamy College of Technology
E-mail : shanthimoorthi@yahoo.com

Dr. K Duraiswamy is a Principal in K. S. Rangasamy College of Technology
E-mail : ksrctt@yahoo.com