Segmentation of Text Lines and Characters in Ancient Tamil Script Documents using Computational Intelligence Techniques

N. Sridevi
Research Scholar

P. Subashini, Phd.
Associate Professor

Avinashilingam Institute of Home Science and Higher Education for Women
Bharathi Park Road, Coimbatore, India.

ABSTRACT
Document image segmentation is one of the critical phases in handwritten character recognition system. Correct segmentation of individual characters decides the accuracy of the recognition system. It is used to decompose the sequence of characters into individual characters to segmenting text lines and then words. Ancient Tamil scripts documents consist of vowels, consonants and various modifiers. Hence proper segmentation algorithm is required. In existing methods, segmentation of overlapping lines and characters are difficult. In order to overcome this problem, two methods are proposed one for line segmentation and another for character segmentation, first method uses projection profile and PSO for line segmentation. In second method combination of connected components along with nearest neighborhood methods are used to segment the characters. Experimental results show that these methods give better results when compared to other methods.

Keywords: Character segmentation, Projection profile, connected components, nearest neighborhood, PSO.

1. INTRODUCTION
Script segmentation is an important task for any Character Recognition System. Segmentation is the process of splitting the document image into text lines, words and then into characters which is extremely useful for many applications like classification, translations. However, this task is extremely challenging for handwritten documents, since the characters structure and content vary considerably. The accuracy of the OCR system depends on the segmentation. If the characters are segmented properly the recognition system gives best results.

Segmentation divides an image into regions or objects. Basically segmentation, tries to extract basic constituent of the script, which are certainly characters. This is needed because the classifier recognizes these characters only [1]. Segmentation phase is also crucial in contributing to this error due to touching characters, which the classifier cannot properly tackle. Even in good quality documents, some adjacent characters touch each other due to inappropriate scanning resolution [2].

The paper is organized as follows: Section 2 describes the characteristics of Tamil scripts and text lines. In Section 3, existing segmentation methods are explained. Section 4 deals with the proposed character segmentation method. Section 5, presents the experimental results and, finally, Section 6 gives the conclusion.

2. CHARACTERISTICS OF TAMIL SCRIPTS AND TEXT LINES

2.1 Tamil scripts
Tamil which is one of the ancient languages in India is the native language of south India. Tamil letters have circular shapes as they were originally carved with needles on palm leaves [3]. The Tamil script has 12 vowels (*uyire̲luttu*), 18 consonants (*meyye̲luttu*) and one character, the *āytam*, which is classified in Tamil grammar as being neither a consonant nor a vowel, though often considered as part of the vowel set. The script, however, is syllabic and not alphabetic. The complete script, therefore, consists of the thirty-one letters in their independent form, and an additional 216 combinant letters representing a total 247 combinations (*uyirmeyye̲luttu*) of a consonant and a vowel, a mute consonant, or a vowel alone. These combinant letters are formed by adding a vowel marker to the consonant. Some vowels require the basic shape of the consonant to be altered in a way that is specific to that vowel. Others are written by adding a vowel-specific suffix to the consonant, yet others a prefix, and finally some vowels require adding both a prefix and a suffix to the consonant [4].

2.2 Text lines
The text line structure becomes the dominant physical structure of a historical document image. Some definitions about text line components are [5]

***Baseline*:** fictitious line which follows and joins the lower part of the character bodies in a text line

Median line: fictitious line which follows and joins the upper part of the character bodies in a text line.

Upper line: fictitious line which joins the top of ascenders.

Lower line: fictitious line which joins the bottom of descenders.

Overlapping components: overlapping components are descenders and ascenders located in the region of an adjacent line.

Touching components: touching components are ascenders and descenders belonging to consecutive lines which are thus connected. These components are large but hard to discriminate before text lines are known.

3. SEGMENTATION ALGORITHMS

Segmentation is an important step in recognition system as it extracts meaningful regions for further analysis. The process of segmentation mainly contains the following [6]:

- Identify the text lines in the page.
- Identify the words in individual line.
- Finally identify individual character in each word.

3.1 Connected Components Method

The connected component method first labels the pixels in the image. The pixels that are connected are labeled with the same blob [7]. This connectivity can be 4 or 8. After labeling, the labeled components are extracted from the image. The Connected Components method solves the overlapping character segmentation problem, but it separates the simple characters into their constituent glyphs which may increase the recognition complexity [8]. The characters are segmented into two glyphs each. These glyphs are to be reassembled to preserve the character shape if the recognition phase uses the shapes of the basic characters [9].

Connected components labeling scans an image and groups its pixels into components based on pixel connectivity, *i.e.* all pixels in a connected component share similar pixel intensity values and are in some way connected with each other. Once all groups have been identified, each pixel is labeled with a gray level or a color according to the component it was assigned to. Connected component labeling works by scanning an image, pixel-by-pixel (from top to bottom and left to right) in order to identify connected pixel regions, that is regions of pixels which have the same intensity values. The connected components labeling operator scans the image by moving along a row until it comes to a point *p* (where *p* denotes the pixel to be labeled at any stage in the scanning process) for which *V={1}*. When this is true, it examines the four neighbors of *p* which have already been encountered in the scan (*i.e.* the neighbors to the left of *p*, above it, and the two upper diagonal terms). Based on this information, the labeling of *p* occurs as follows:

- If all four neighbors are 0, assign a new label to *p*, else
- if only one neighbor has *V={1}*, assign its label to *p*, else
- if more than one of the neighbors have *V={1}*, assign one of the labels to *p* and make a note of the equivalences.

After completing the scan, the pairs with equal labels are sorted into similar classes and an exclusive label is assigned to each class. Finally, a second scan is made through the image, during which each label is replaced by the label assigned to its similar classes.

- Advantage:
  The Connected Components method solves the overlapping character segmentation problem

- Disadvantage:
  It separates simple characters into their constituent glyphs.

3.2 Projection Profile Method

A natural choice for line segmentation of gray scale images is the projection profile method [10]. Gaps between the text lines can be found by finding the maximum projections values, the projection value is calculated by summing the pixel values along the horizontal directions of the document image. There are two main advantages for the projection profile approach in the context of historical document. First, it does not require binarization of the image, which makes it directly applicable to gray scale images. Second, it is very robust to noise and other degradations [11].

To segment the text lines, from the document image, the horizontal projection profile is calculated. The horizontal projection profile is the histogram of the number of intensity values of the pixels along every row of the image. The space between text lines is used to segment the text lines. The projection profile will have histogram of zero height between the text lines. Line segmentation is done at these points [12].In order to segment the word from the text line the vertical projection profile of an input text line is calculated. Vertical projection profile is the sum of ON pixels along every column of the image which is used to separate the word from the text line. Character is segmented from the word by taking the vertical projection profile of a word.

- Advantage:
  This method is suitable for segmenting image documents that are well spaced without overlapping and touching

- Disadvantage:
  When the characters are overlapped or touched this method can't segment

4. PROPOSED SEGMENTATION METHOD

In order to overcome the shortcomings of the existing method the proposed method is described. The space between the lines is used to separate the lines. Normally the distances between two lines are larger than the distances between words, thus lines can be segmented by comparing this distance against a suitable threshold. To determine an optimal threshold, Particle Swarm Optimization technique is used. It is known from literature, Particle Swarm Optimization (PSO) algorithm is used to solve many of difficult problems in the field of pattern recognition [13]. Hence, PSO is used to compute an optimal value.

4.1 PSO Algorithm

Let X and V denote the particle's position and its corresponding velocity in search space respectively. At iteration K, each particle i has its position defined by $X_i^k = (x_{i1}, x_{i2}, ...x_{in})$ and a velocity is defined by $V_i^k = (v_{i1}, v_{i2}, ...v_{in})$ in search space n. Velocity and position of each particle in next iterations can be calculated using following equation (1) and (2)

$$v_{ij}^{k+1} = wv_{ij}^{k} + c_1 r_1 (pbest_{ij}^{k} - x_{ij}^{k}) + c_2 r_2 (gbest_{ij}^{k} - x_{ij}^{k}) \quad (1)$$

$$x_{ij}^{k+1} = x_g^k + v_g^k \quad (2)$$

where k is the current iteration number,w is inertia weight, v_{ij} is then updated velocity on the j^{th} dimension of the i^{th} particle, c_1 *and* c_2 are acceleration constants, c_1 and c_2 are positive constant parameters, usually $c_1 = c_2 = 2$. r_1 and r_2 are the real numbers drawn from two uniform random sequences of U (0, 1).

The algorithm starts by generating randomly initial population of the PSO. In PSO, every particle is initialized with locations and velocities using the equations (1) and (2).These locations consist of the initial solutions for the optimal threshold.

The procedure of the proposed PSO algorithm is described as follows:

Step 1: Initialize N particles with random positions $x_1, x_2, \ldots . x_n$ according to Eq. (1) and velocities V_i where i = 1, 2...N.
Step 2: Evaluate each particle according to equation (3)

$$f(t) = \omega_0(t) \times \omega_1(t) \times (\mu_0(t) - \mu_1(t))^2 \quad (3)$$

where, t is a gray level between 0 and 255 which can be obtained through the particle's position.
Step 3: Update individual and global best positions. If $f(pbest_i) < f(x_i)$, then $pbesti = x_i$, and search for the maximum value f_{max} among $f(pbest_i)$, If max $f(gbest) < f_{max}$, *then* $gbest = x_{max}$, x_{max} is the particle associated with f_{max}.
Step 4: Update velocity: update the i^{th} particle velocity using the Eq. (2) restricted by maximum and minimum threshold v_{max} and v_{min} .
Step 5: Update Position: update the *i*th particle position using Eq. (1) and (2).
Step 6: Repeat step 2 to 5 until a given maximum number of iterations is achieved or the optimal solution so far has not been improved for a given number of iteration

The best threshold value is obtained using PSO in order to segment text line from the image. To segment the word from the text line vertical projection profile is calculated. In the profile, the zero valley peaks may represent the character or word space. To differentiate whether it is character or word spacing, find the maximum character space cluster and use it for separating the words.

4.2 Character segmentation

Character segmentation from word is little difficult because vowel modifiers placed on top of base characters and consonant modifiers attached to left or right or bottom of the base character. When the characters are touched or overlapped projection profile method doesn't give good results therefore the following algorithm is used [14]

1. Remove consonant modifiers from the word. For this,
   a. Determine the middle row using bounding box
   b. Compute horizontal profile and identify the bottom base line using this profile. The bottom base line is the highest peak row in the profile down from the middle row.
   c. The Connected Components down the bottom base line are consonant modifiers. Remove them from the word and add to the consonant modifiers group.
2. Remove vowel modifiers by finding the top base line computed as in the above step and modifiers to the vowel modifier group.
3. Using the vertical profile separate the base characters using the white space between them. Then add vowel and consonant modifiers using nearest neighborhood method with horizontal relationship heuristics.

Advantage:

This method is suitable for segmenting image documents contained overlapping characters documents

Disadvantages:

- Tamil scripts are composed of two parts, namely the basic character and a modifier symbol corresponding to each of the basic character [15]. If the space between the basic character and the modifier symbol is more, the proposed method couldn't segment it properly.
- It couldn't segment the touching lines and characters

5. RESULTS AND DISCUSSIONS

The data samples are collected from thee script which belongs to Tamil brahmi script (250 BC – 250 AD) and Tiruvalangadu plates of Rajendra chola I (11^{th} century) [16]. Figure 1, 2 and 3 shows the output of the existing method and the proposed method. From the experimental results it is clear that the proposed method segment the characters properly even if they are overlapping or touching each other. The proposed method shows better results when compared to other methods. The limitation of this method is that it results in segmentation error of touching lines and characters.

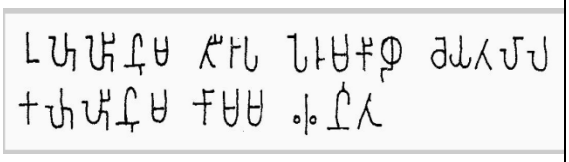
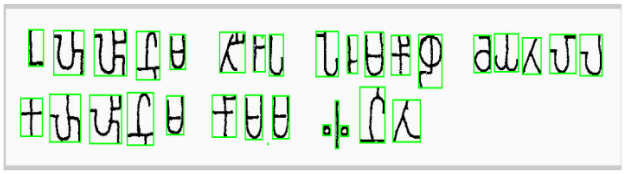
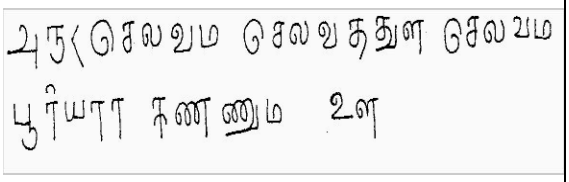
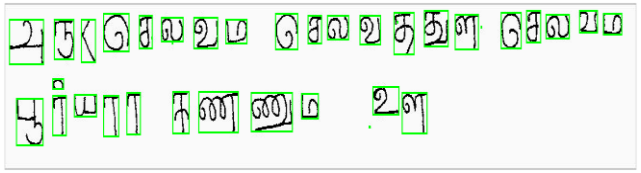
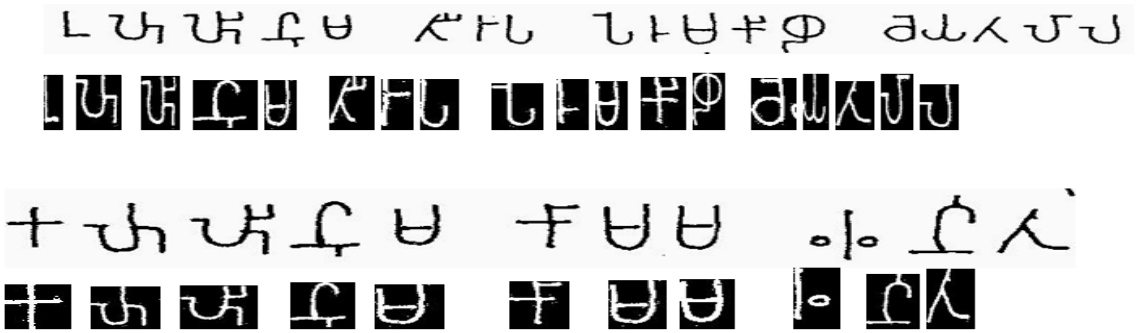
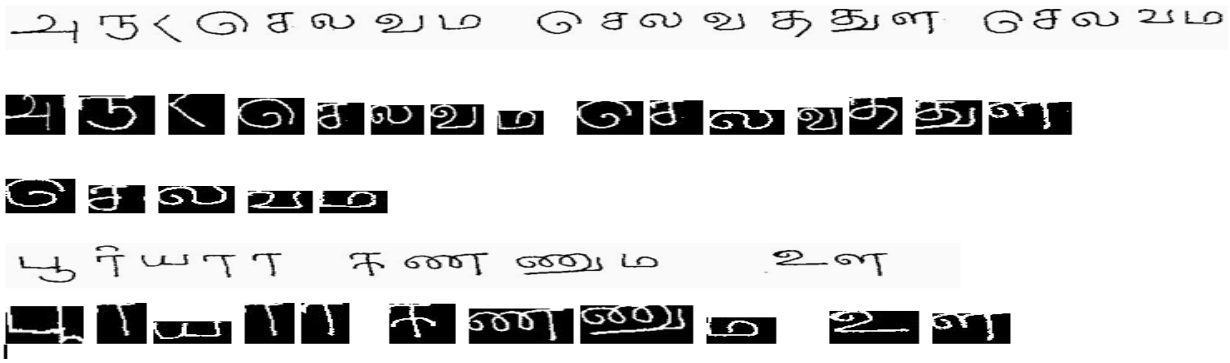
Input image	Output Image
	
	

Fig 1: Resultant image obtained using connected components labeling method

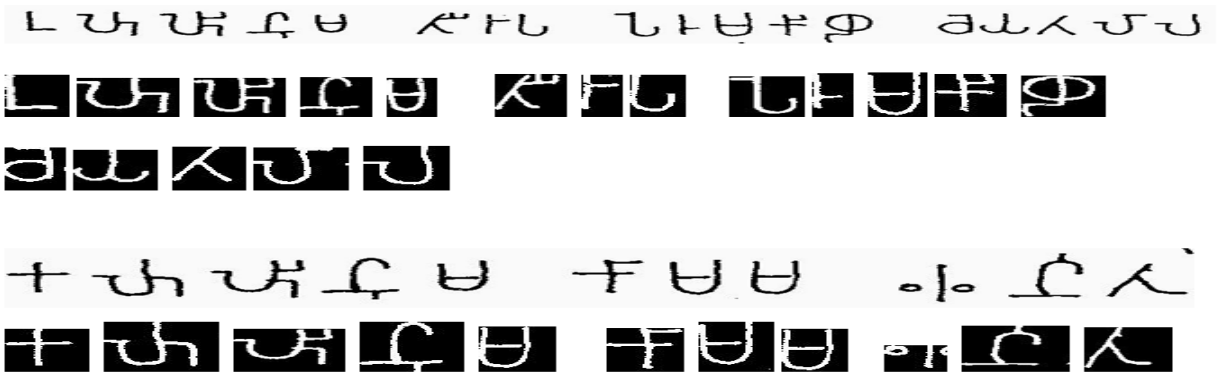


(a)

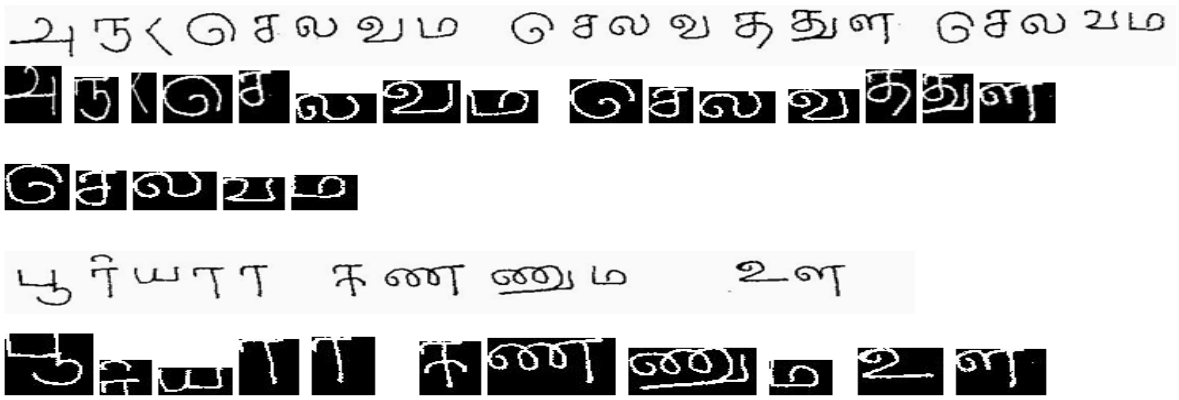


(b)

Fig 2: Resultant image obtained using Projection Profile method



(a)



(b)

Fig 3: Resultant image obtained using proposed method

6. CONCLUSION

In this paper, a new method is proposed using PSO in which an optimal threshold is calculated for segmenting the text lines from the document image. Connected components algorithm is combined with nearest neighborhood algorithm to segment the characters of ancient Tamil scripts. The proposed algorithm is compared with existing methods and tested using several document images. Even though the proposed method could segment all the documents in a robust way and gave good results, but it couldn't segment the touching lines and characters and also it could not segment the characters if the spaces between the basic character and modifiers are more. Segmentation of touching lines and characters needs some other approaches which could be consider as future work.

7. REFERENCES

[1] Raghuraj Singh. S. Yadav and Prabhat Verma" Optical Character Recognition (OCR) for Printed Devnagari Script Using Artificial Neural Network" , International Journal of Computer Science & Communication, Vol. 1, No. 1, January-June 2010, pp. 91-95.

[2] Vijay kumar and Pankaj K. Sengar, "Segmentation of Printed Text in Devanagari Script and Gurmukhi Script", International Journal of Computer Applications,Vol 3, No.8, June 2010,pp. 24-29

[3] N.Dhamayanthi, and P.Thangavel," Handwritten Tamil character recognition using neural network", Proceeding of Tamil Internet 2000, Singapore, July 22-24, 2000, pp.171-176.

[4] http://www.italki.com/notebook/entry/66643.htm.

[5] Laurence Likforman-Sulem, et.al," Text Line Segmentation of Historical Documents: a survey", Submitted to Special Issue on Analysis of Historical Document, International Journal on Document Analysis and Recognition, Springer, 2006.

[6] Vikas J Dongre and Vijay H Manka, "Devnagari Document Segmentation Using Histogram Approach", International Journal of Computer Science, Engineering and Information Technology (IJCSEIT), Vol.1, No.3, August 2011, pp. 46 -53.

[7] R.C. Gonzalez and R.E. Woods. (2004): *Digital Image Processing*, Pearson Education.

[8] Stephen Marchand Maillet ,"Binary Digital Image Processing- A Discrete Approach", 1999

[9] C V Lakshmi, C PAtardhan "A Multi-font OCR System for printed Telugu Text.", Proceeding of LEC'02, IEEE, 2002

[10] L. Likforman-Sulem, A. Zahour, B. Taconet," Text line segmentation of historical documents: a survey", International journal of Document Analysis and Recognition,Vol 9, 2007, pp. 123 – 138

[11] Itay Bar-Yosef et, al, "Line segmentation for degraded handwritten historical documents".

[12] R.Sanjeev Kunte and R D Sudhaker Samuel," A Simple and efficient optical character recognition system for basic symbols in printed kannada text", Sadhana, Vol 32, Part 5, October 2007, pp. 521 – 533.

[13] Oliveira .S.L., S. A. Britto, and R. Sabourin, " Optimizing Class-Related Thresholds with Particle Swarm Optimization", Proceeding of International Joint Conference on Neural Networks, IEEE, Montreal, Canada, July 31 – August 4, 2005,pp. 1511 – 1516.

[14] M Swamy Das et. al, "Segmentation of Overlapping Text Lines, Characters in Printed Telugu Text Document Images", International Journal of Engineering Science and Technology, Vol. 2, No.11, 2010,pp. 6606 – 6610.

[15] S.Santhosh Baboo, P.Subashini and M.Krishnaveni, "Combining Self-Organizing Maps and Radial Basis Function Networks for Tamil handwritten Character Recognition", International Journal of ICGST-GVIP, Vol. 9, No.4, August 2009, pp. 1- 7.

[16] Gift Siromoney, S Govindaraju, M.Chandrasekaran, "Thirukkural in Ancient Scripts", Department of Statistics, Madras Christian College, Tambaram, 1980.