Resolving Ambiguities in Confused Online Tamil Characters with Post Processing Algorithms

A G Ramakrishnan, Suresh Sundaram
Medical Intelligence and Language Laboratory
Indian Institute of Science, Bangalore, India
ramkiag@ee.iisc.ernet.in, suresh@ee.iisc.ernet.in

Abstract. This paper addresses the problem of resolving ambiguities in frequently confused online Tamil character pairs by employing script specific algorithms as a post classification step. Robust structural cues and temporal information of the preprocessed character are extensively utilized in the design of these algorithms. The methods are quite robust in automatically extracting the discriminative substrokes of confused characters for further analysis. Experimental validation on the IWFHR Database indicates error rates of less than 3 % for the confused characters. Thus, these post processing steps have a good potential to improve the performance of online Tamil handwritten character recognition.

Keywords: Online handwritten character recognition, Confusion Pairs, Substroke extraction and analysis, Fourier Descriptors.

1 Introduction

Tamil is a popular classical language spoken by a significant population in South East Asian countries. There are 156 distinct symbols in Tamil [1]. As far as earlier work on recognition of online Tamil characters is concerned, Deepu *et al.* [2] generate class specific subspaces using principal component analysis, while Niranjan *et al*. [1] have employed dynamic time warping for matching unequal length feature sequences. Hidden Markov models for recognition have also been reported in [3] [4]. In a recent work, we have studied the performance of the 2DPCA Algorithm [5], which was originally proposed for face recognition.

Each of the above schemes is found to give nearly similar generalization performances on a given test data. Most of the misclassifications of the given data, in general, are attributed to the fact that Tamil has many symbols that look visually similar. Any classifier that works on features at a global level fails to capture finer nuances that make these symbols distinct. One way to circumvent this drawback would be to incorporate a post processing step that employs local features to reduce the degree of confusion between frequently confused characters, and thereby improves the overall performance of the recognition. Specifically, this paper proposes algorithms for disambiguating frequently confused symbols. The approaches are developed, taking into account, the popular writing / lexemic styles of modern Tamil script. They can be applied irrespective of the nature of the classifier used for the recognition.

In a system that deals with recognition at the word level one could use language models. However, when recognizing isolated characters, one is devoid of any such additional information that can be used to correct errors. Thus, we resort to the use of robust structural features for post recognition disambiguation.

2 Confusion Pair Analysis

A careful analysis of the confusion matrices, obtained with different classifier frameworks, suggests that the frequently confused Tamil characters can be manually grouped into two categories A and B as shown in Table 1. Accordingly, we propose appropriate post-processing techniques to each group of the confusion pairs. The confusions in Group A appear between pairs of Tamil consonant-vowel combinations sharing the same base consonant but different vowel modifiers. The confused strokes contributing to errors are ி and ீ. These correspond to the modifiers of the vowels இ and ஈ respectively.

Apart from Group A pairs, there exist other pairs that differ predominantly in the end such as (ஐ ஜ) and (க கு). The structure to be analyzed for these pairs is strikingly different from those belonging to Group A. Consequently, these pairs are placed in Group B. Also, there exist certain character pairs that differ only at the start and middle of the trace such as (ஏ ர), (ள ன) and (மு ழு). These are also incorporated in Group B. As stated before, the spatial and temporal information provided by the online data is extensively utilized for the design of the algorithms. Prior to designing the appropriate post processing technique, the raw Tamil character is subject to pre processing modules [2] such as smoothing, size normalization and resampling.

Table 1. List of commonly confused online Tamil character pairs.

Group A	(கி,கீ) (ஙி,ஙீ) (சி,சீ) (ஞி,ஞீ) (ணி,ணீ) (தி,தீ) (நி,நீ) (லி,லீ) (வி,வீ) (ளி,ளீ) (னி,னீ) (ஜி,ஜீ)
Group B	(ஏ,ர) (ஐ,ஜ) (க,சு) (ல,வ) (ள,ன) (மு,ழு) (டு,டூ) (நு,னு) (நூ,னூ)

3 Disambiguation of Group A Confusion Pairs

In this section, we outline the post processing algorithm for disambiguating the vowel modifiers ி and ீ for a given base consonant. Popular writing styles of modern Tamil script suggest that the vowel modifiers ி and ீ always form the last stroke in any multistroke consonant-vowel combination character. However, for CV combinations written as a single stroke (where the vowel modifiers are written as a continuation of the base consonant), a subset of sample points, carefully chosen before the final PEN UP signal, is taken to be the vowel modifier. It is worth re-emphasizing that the confused pairs in Group A correspond to CV combinations sharing the same base consonant (BC). Let ω_1 and ω_2 denote the class labels of BC+ ி and BC+ ீ combinations, respectively. We outline below the algorithm proposed for distinguishing ω_1 and ω_2. Note that as soon as any 'If' condition in the algorithm is satisfied, the corresponding class label is assigned to the CV combination and we terminate.

For the preprocessed CV combination resampled to N points, let $S = \{(x_i, y_i)\}_{i=b}^{N}$ denote the pen coordinates of the extracted vowel modifier. Here (x_b, y_b) denotes the starting sample point of the vowel modifier. A point (x_i, y_i) in S is said to be an 'interest point' if the following two conditions are satisfied.

(i) $y_i < y_{i-1}$ and $y_i < y_{i+1}$.

(ii) $x_{i+1} < x_i$. **(1)**

```
Using the aforementioned condition, compute the number of interest points $I$.

Find the sample point $(x_s, y_s)$ satisfying the relation $y_s = \max_{i>=b} y_i$ .

If $(x_s\ y_s)$ corresponds to the last sample point of the modifier,
    Accept class $\omega_2$ if $I>0$ and $\omega_1$ if $I=0$.
End

If $I > 0$
    Assign character to class $\omega_2$ (Fig. 1(a)).
End
```

Locate the sample point (x_m, y_m) satisfying the relation $x_m = \max_{i>s} x_i$.

```
 Define the quantity      $r = x_m - x_N$                                (2)
 If  $r \geq \varepsilon$  and  $y_N > y_b$
   Assign the character to class  $\omega_2$  (Fig. 1(b));
 else
   Assign it to class  $\omega_1$  (Fig.1(c)).
End
```

Here ε is a threshold, empirically set to 0.07.

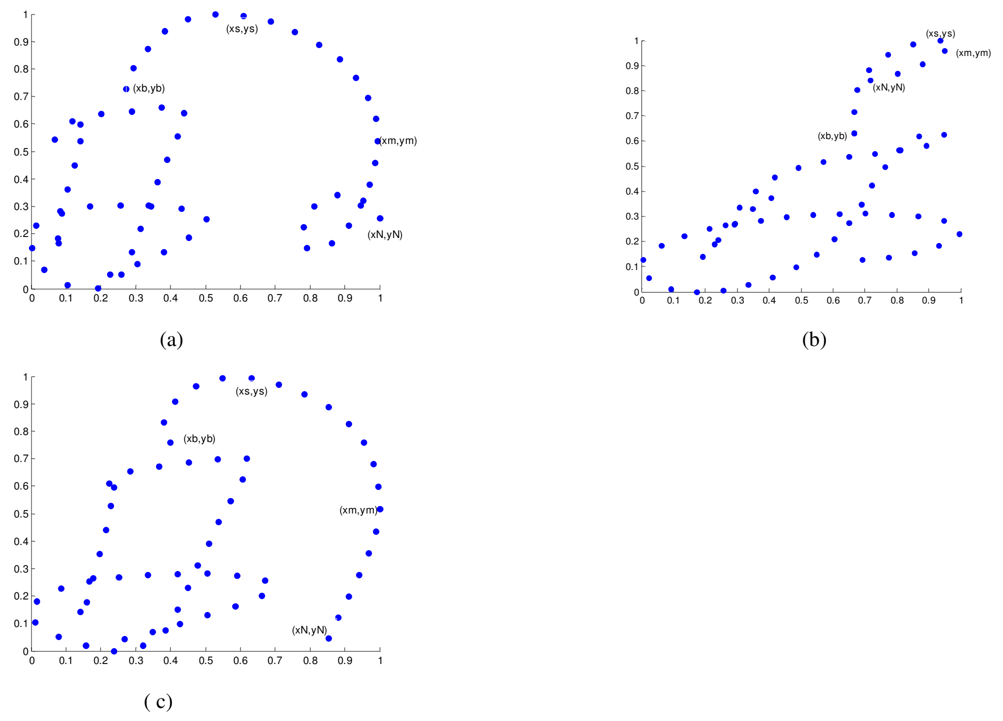


Fig. 1. Disambiguation of (Group A) confused pairs by structural features.

(a) This sample is assigned to class கீ (by Step 2). Here $I = 1$.

(b) This sample is assigned to கீ (by Step 3). Here r $> \varepsilon$, $y_N > y_b$ and $I = 0$.

(c) This sample is assigned to கி (by Step 3). Here r $> \varepsilon$, $y_N < y_b$ and $I = 0$.

4 Disambiguation of Group B Confusion Pairs

The discriminative substrokes of confused pairs in Group B may fall in any region of the trace. Hence, the post processing algorithms for this group must be able enough to automatically extract and analyze parts of strokes that differ. Unlike in Group A, where a single algorithm is used for disambiguation, separate post processing algorithms are generated for each confusion pair in Group B. The need for using dedicated post processing scheme to disambiguate a specific confusion pair arises mainly due to the structural characteristics of the pair under consideration. For example, the post processing scheme for extracting the discriminative substroke of the (ள ன) pair will be different from that used for extracting the substroke in (ழ ழு) pair. Note that, however, in both these pairs, the finer nuance that makes the confused characters distinct is at the middle of the trace. So, for a given pair, the proposed algorithms for

Group B take into consideration the temporal information, writing styles and structural cues of the confused characters. The illustrations that follow will throw light on the effectiveness of the methods in both substroke extraction and analysis.

A) Consider the characters ஜ and ஐ. Instead of feeding the (x,y) coordinates of these characters as a whole to the post processing module, we focus on the shape of substrokes forming the tails of these characters and extract Fourier descriptors from them. In order to extract the shape of interest, we compute the length of the character and divide it to 4 equal segments. Sample points that lie in the last segment form the tail of the character and are resampled to 30 points before deriving the features. The number of Fourier coefficients chosen is set empirically to 10. A nearest neighbor classifier is used to obtain the final recognition label of a test character.

B) As a next illustration, consider the characters ன and ண. Since the subtle difference in these characters is observed in the middle of the trace, Fourier descriptors do not form a robust feature for discrimination. Our post processing algorithm first automatically extracts the substroke of interest as follows. Let $\{(x_i, y_i)\}_{i=1}^{N}$ be the sample points of preprocessed character (ன or ண). Locate the first minimum (x_c, y_c) that satisfies the condition.

$$y_c < y_{c-1} \text{ and } y_c < y_{c+1}. \qquad \textbf{(3)}$$

Starting from (x_c, y_c), move along the trace and locate the point (x_b, y_b) whose x coordinate just exceeds x_c. A substroke is extracted with (x_b, y_b) as the starting point. The extracted substroke can thus be described as

$$S = \{(x_i, y_i)\}_{i=b}^{N}. \qquad \textbf{(4)}$$

Having extracted the substroke S, we shift our focus to analyzing the same. Let (x_{int}, y_{int}) be the first encountered minimum in S for which

$$y_{int-1} < y_{int} < y_{int+1}. \qquad \textbf{(5)}$$

Fig. 2 illustrates the aforementioned explanation.

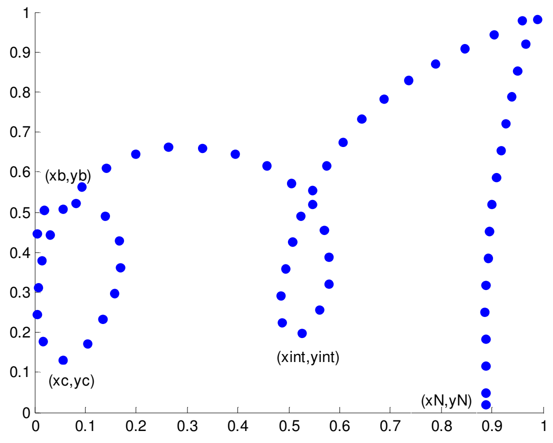


Fig. 2. Illustration of how (x_{int}, y_{int}) is located in the character ண .

The following conditions are applied to the point (x_{int}, y_{int}) for finalizing our decision.

1) If $x_{int-1} < x_{int+1}$, assign symbol to ள. (Fig. 3 (a))

2) If the angle θ between successive pen directions defined at (x_{int}, y_{int}) is greater than 150 deg, assign the symbol to ள. (Fig. 3 (b)). θ is the angle formed by the vectors $(x_{\text{int}} - x_{\text{int}-1} \quad y_{\text{int}} - y_{\text{int}-1})$ and $(x_{\text{int}+1} - x_{\text{int}} \quad y_{\text{int}+1} - y_{\text{int}})$.

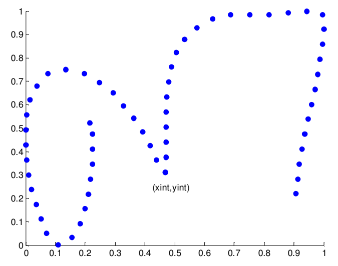
3) If points in the neighborhood of (x_{int}, y_{int}) are sufficiently close to each other (less than a threshold), the character is assigned to ள. (Fig. 3 (c))
The 'closeness' condition is defined as follows: Let $W = \{(x_i, y_i)\}_{i=\text{int}-3}^{\text{int}+3}$ be a window size of 7 centered at (x_{int}, y_{int}). Using this window, compute three distances D_1, D_2 and D_3 as defined below.

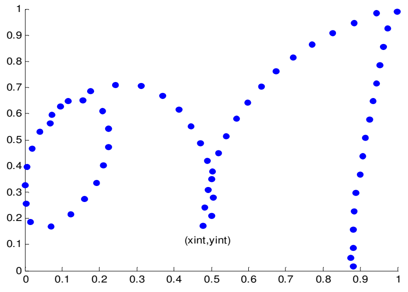
$$D_j = \text{dist}\,((x_{int-j} \quad y_{int-j}), (x_{int+j} \quad y_{int+j})) \qquad j = 1, 2, 3 \tag{6}$$

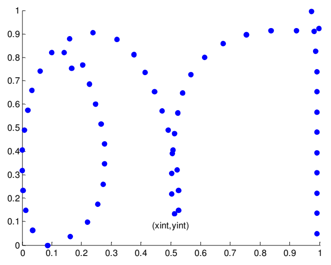
Our final decision for the label of the character can be formulated as follows:

$$\text{Assign character to ள if } \sqrt{D_1^2 + D_2^2 + D_3^2} < 0.1. \tag{7}$$

4) If neither condition holds good, character is assigned to ன. (Fig. 3 (d))


(a)


(b)


(c)

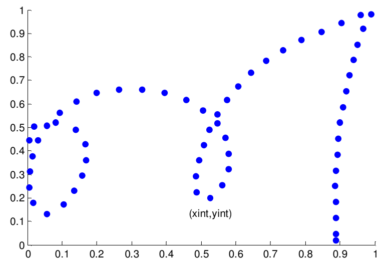

(d)

Fig. 3. (a) (b) (c) indicate samples of ள that satisfy conditions (1) (2) and (3) respectively. Fig. 3 (d) depicts the sample ன that violates the above conditions.

C) As a final illustration, we describe the strategy proposed to reduce the confusion between the symbols (மு மூ). The rationale for segmenting the substroke of interest S is as follows. Locate the first sample point (x_b, y_b) for which the following 2 conditions are simultaneously satisfied.

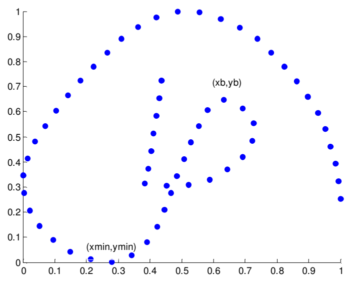
(i) $y_{b+1} < y_b$ and $y_{b-1} < y_b$.

(ii) $x_{b+1} < x_b < x_{b-1}$. **(8)**

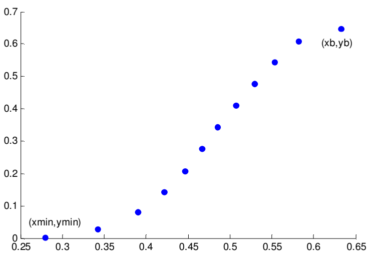
$(x_b,\ y_b)$ serves as the starting point for our substroke. The minimum y coordinate of the symbol, $y_{\min}$, is the end point. Stated in another way,

$$S = \{(x_i, y_i)\}_{i=b}^{\min} \ . \qquad \textbf{(9)}$$

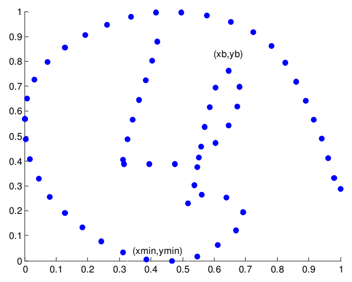
Snapshots of ழ (Fig. 4 (a)) and ழ (Fig. 4 (c)) with the extracted substrokes are shown in Figs. 4 (b) and (d) respectively.



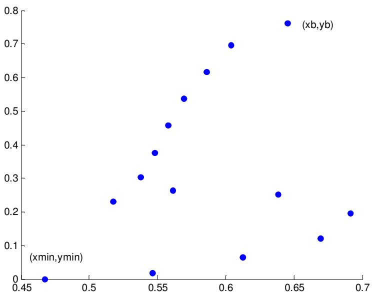
(a)



(b)



(c)



(d)

Fig. 4. (a) (c). Samples of ழ and ழ with the desired substrokes in Figs. 4. (b) and (d) respectively.

We now proceed with analyzing the substroke S as follows. Calculate the number of sample points, f, in S for which $y_{i+1} > y_i$. If $f > 0$, the character is assigned to ழ. Otherwise ழ is selected. Using this condition, we see that the character in Fig. 4 (a) is assigned to ழ since $f = 0$ for this sample. Fig. 4(c) provides a case for which $f = 1$. Hence, by our logic, it is assigned to ழ.

5 Experimental Results

The aforementioned techniques are tested on the combined training and testing corpus of the IWFHR Competition dataset [6]. Table 2 depicts the recognition accuracies obtained by incorporating the post processing scheme for disambiguating confused characters. It can be noted that the ambiguity amongst the symbols has been resolved significantly, as indicated by the substantially high classification rates. The algorithms can also be extended to disambiguate confusion quadruples like (ளி ளீ ணி ணீ). Here, we first invoke the Group A post processing algorithm to distinguish the vowel modifiers ி and ீ, and then discriminate the base consonants ள and ண (Group B pair disambiguation). Employing the scheme to the quadruple gives a recognition rate of 96.2 %.

However, it is important mentioning here that the accuracies quoted in Table 2 are applicable only to the confused characters in the pair under consideration. Needless to say, the methods can be applied irrespective of the classifier used for the recognition, though the nature of the confusion matrix may slightly vary.

Table 2. Recognition accuracies after invoking appropriate post processing algorithms to confusion pairs.

GROUP A		**GROUP B**	
Confusion Pairs	Accuracy	Confusion Pairs	Accuracy
(ஙி ஙீ)	99.2%	(ஐ ஜ)	98.9%
(ணி ணீ)	98.5%	(ள ன)	98.2%
(ளி ளீ)	98.1%	(னி ளி)	98.3%
(னி னீ)	97.6%	(ளீ னீ)	97.2%
(கி கீ)	99.1%	(மு ழு)	97.1%

In this work, we have designed post processing algorithms to reduce the ambiguity between frequently confused Tamil characters. Structural cues are utilized in the design of these methods, which can be applied independent of the classifier used for the recognition. Future areas of research would involve improving these methods to handle characters, whose strokes have been broken due to unintentional lifting of the stylus.

6 References

1. Niranjan Joshi, G Sita, A G Ramakrishnan and Sriganesh Madhavanath.: Comparison of Elastic Matching Algorithms for Online Tamil Handwritten Character Recognition. Proc. Intl Workshop Frontiers Handwriting Recog. pp 444-449 (2004).
2. Deepu V, Sriganesh Madhavanath, A G Ramakrishnan.: Principal Component Analysis for Online Handwritten Character Recognition.: Proc. Intl Conf .Pattern Recog. 2: pp 327-330 (2004).
3. Alejandro H Toselli, Moises Pastor, Enrique Vidal.: On-Line Handwriting Recognition System for Tamil Characters, Proc. Iberian conf. Pattern Recog. Image Anal., Lecture Notes Comp. Science Vol. 4477(I), pp 370-377 (2007).
4. Sai Prasad.: Online Handwriting Recognition for Tamil. Masters Thesis Report 2008. Indian Institute of Science, Bangalore, India.
5. Suresh Sundaram, A G Ramakrishnan.: Improvement of Online Tamil Character Recognition Engine using Post Processing Methods. Accepted for publication in Proc. Tenth Intl Conf on Doc Anal and Recog. (ICDAR) 2009.
6. HP Labs Isolated Handwritten Tamil Character Dataset. http://www.hpl.hp.com/india/research/penhw-interfaces-1linguistics.html#datasets

Acknowledgements. The authors would specially like to thank TDIL, DIT for funding this research work.