Recognition of online captured, handwritten Tamil words on Android

A G Ramakrishnan and Bhargava Urala K

Medical Intelligence and Language Engineering (MILE) Laboratory,
Dept. of Electrical Engineering, Indian Institute of Science, Bangalore
Email: bhargava.urala@gmail.com and ramkiag@ee.iisc.ernet.in

Abstract

The online handwriting recognition system currently assumes that whatever is written on the user window is a single word, ignoring the separating spaces even if they are present. The attention-feedback segmentation strategy is applied to segment the stream of strokes into separate stroke-groups that represent distinct, recognizable Tamil symbols. These segmented symbols are normalized after smoothing and resampled along the arc length to 64 equidistant points. A SVM-RBF classifier recognizes the symbols using the x and y values of these sequence of equidistant points along the arc and also their complex DFT coefficients. Character recognition accuracies of 95.8% and 83.2% are obtained on the test sets of 26, 926 isolated characters and 45,405 words (containing 2, 53, 095 characters), both collected on Tablet PCs. The latter is lower since it is the product of segmentation and recognition accuracies. On another test set of 1, 897 words (6,627 characters) collected on Genius GNote 7, the recognition rate is 89.2%. On a page of data written on HiTech digitizer, the accuracy is 86.6%. Thus, the performance is good across different devices with differing spatial and temporal resolutions. This MILE online handwriting recognition engine has now been ported to work on Android devices with a stylus.

Index Terms: OCR, ICR, Tamil, Android, handwriting recognition, online, word recognition, SVM-RBF, attention-feedback, dominant overlap, segmentation, stroke group, symbols.

Introduction

The term ’online’ refers to the acquisition of data as one writes, which makes the writing trajectory, with the sequence of spatial coordinates of the trace, available at uniform time intervals. A digital pen or a stylus is used to write on any surface (which could be touch-sensitive) and a digitizer captures the online handwritten data. The instances of pen contact with the surface (up or down) are also recorded as appropriate flags.

In the literature, recognition of online handwritten Tamil word generally follows one of two distinct approaches: 1) Lexicon-driven approach [1], where each stroke is treated as a basic unit of recognition and word models are trained using hidden Markov models (HMM). This method works well for applications involving a limited vocabulary. 2) Segmentation-driven

approach [2], where strokes are grouped into stroke groups (symbols) which represent a full or part Tamil compound letter (*koottezhuthu* or *akshara*). Each stroke group is recognised and a word is represented as a string of symbols. This method is used for open vocabulary problems. Here, we describe the development of an open (unlimited) vocabulary, handwritten word recognition system for Tamil on the Android platform.

Brief outline of the recognition system

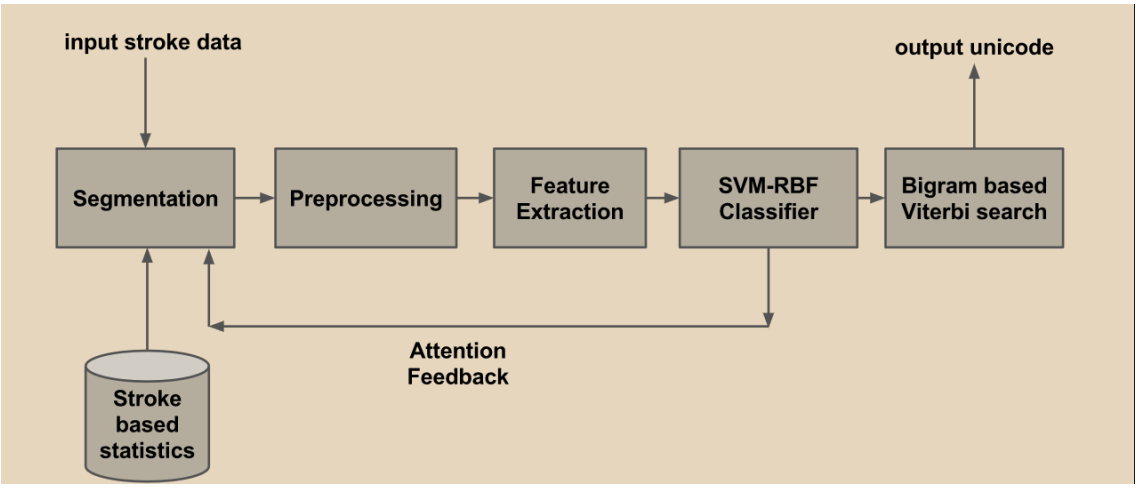

Figure 1. Block schematic of the handwriting recognition system.

Input stroke sequence – From the Android device, the sequence of strokes corresponding to one word are input at a time. Consecutive strokes need to be grouped into symbols.

Handling delayed and overwritten strokes – Sometimes, the *pulli* (dot on top of any pure consonant) or some other valid part of a character is left out to begin with and is written later after the rest of the word has been completed. In the first pass through the input stroke sequence, such delayed strokes are reordered based on their position vis-a-vis the earlier strokes. If the reordered stroke is fully contained within a stroke group, it is assumed to be an overwritten, probably beautifying part-stroke or correction, and hence, is deleted. Otherwise, it is combined with the overlapping stroke group.

Dominant overlap criterion segmentation – Two or more strokes are grouped into a stroke group, if the horizontal overlap between them exceeds a statistically determined threshold [2]. If segmented properly, such stroke groups correspond to distinct Tamil symbols.

Normalization of stroke groups – Smoothing removes the effect of jitter; normalisation handles size variations and resampling along the trace eliminates the variations in the time sequence of data due to variations in the writing speed [3].

Extraction of global and local features - The global and local shapes of the symbol are captured as the features and concatenated. Global features are the discrete Fourier transform coefficients of the preprocessed coordinates, treated as a sequence of complex numbers [4]. Local features are the resampled pen trajectory points themselves [5].

Classification using SVM-RBF - Support vector machine with radial basis function kernel is trained on the feature vectors extracted from a database of isolated stroke groups [6].

Correction of symbol segmentation - Recognition scores and inter-stroke distances are used to detect any possible errors that may have occurred in dominant overlap segmentation. Two suspected adjacent stroke groups are merged tentatively and if its confidence score is higher than the average score of the individual stroke groups, they are merged permanently. This is the attention-feedback segmentation proposed in [7].

Postprocessing using language models – Bigram probabilities of all pairs of Tamil symbols (including spaces) have been computed from a large Tamil text database. After segmentation and recognition of each word, a lattice is constructed with each stroke group as the nodes and bigram probabilities as transition weights. The most likely symbol string to represent the word is found using maximum likelihood (Viterbi) decoding [8].

Output Unicode string generation – The string of symbols output by the classifier is converted to the corresponding Unicode string, using Tamil script grammar and the resultant text (actually, the top three choices given by Viterbi) is displayed.

The entire recognition engine is implemented in C++ and has been ported to an application in Tablet PC [9]. We have employed Java Native Interface (JNI) along with a simple front end created using Java, so that the same code and engine used for Tablet PC application development can be readily used for Android application development. The application runs fairly fast (~50 ms per character) and has robust recognition rates despite the training data for the classifier having been collected on a different device (Tablet PC) with a different sampling rate and spatial resolution. The application currently runs on tablets with Android version 4.4.2 or higher.

Future work

The recognition speed and accuracy will be improved on the Android platform by optimizing the code and revising the recognition strategy. The graphical user interface will also be enhanced to create a smooth user experience. The current word recognition module will be extended to handle the recognition of line and page level text to support practical and usable applications. The idea is to develop a regular word processing application using handwriting that supports recognition of numerals, punctuation marks and other symbols.

Acknowledgment

The authors are grateful to Technology Development for Indian Languages (TDIL), Department of Information Technology (DeitY), Government of India for funding the above research and development project as a national research consortium. We are also thankful to Mr. Swapnil Belhe and his team at CDAC Pune for helping with the Android porting. We also thank INFITT and Kani Thamizh Sangam for their continued encouragement.

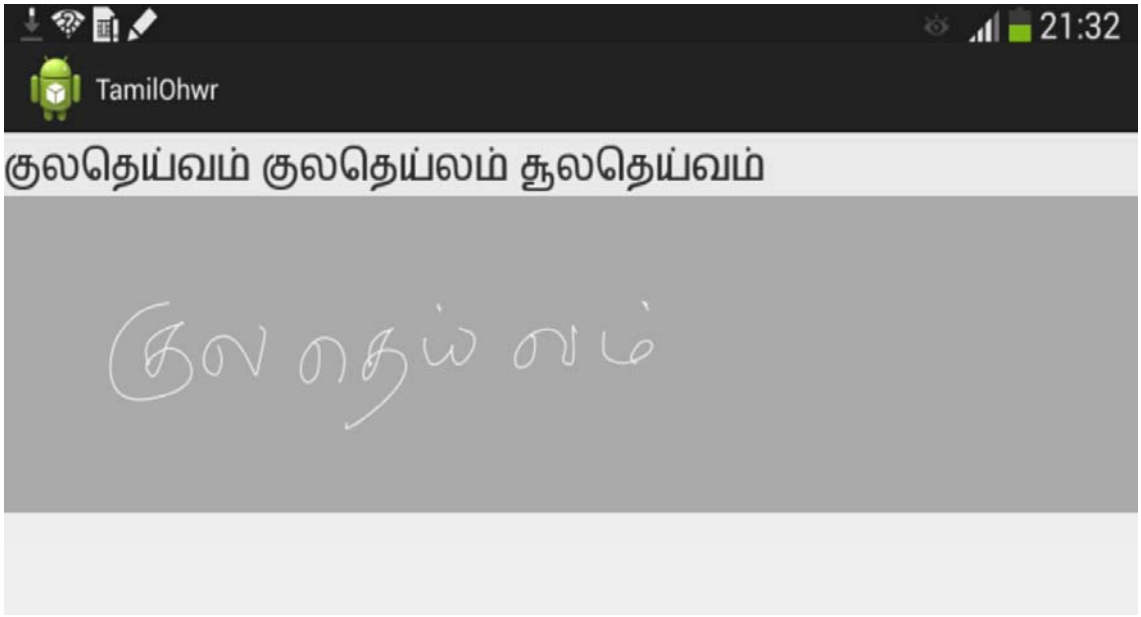

Fig. 2. A sample handwritten input on Samsung Tab 10.1
and the top three choices of the recognized word given by the Viterbi.

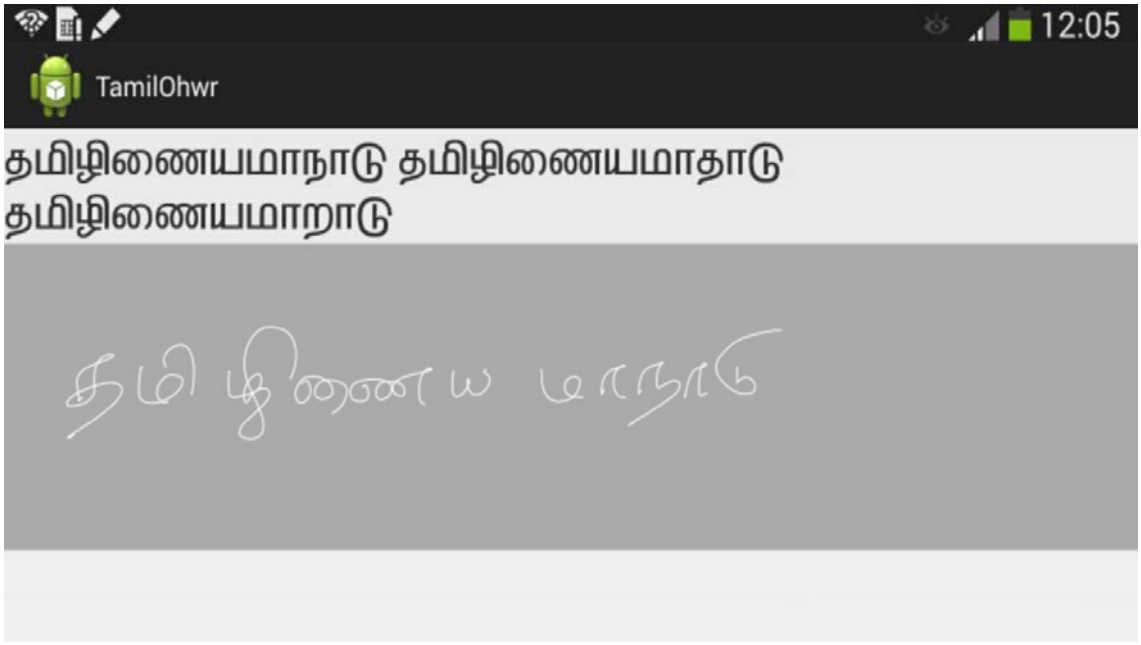

Fig. 3. Another sample handwritten input and the three best choices
for the recognized word obtained from the Viterbi decoder.

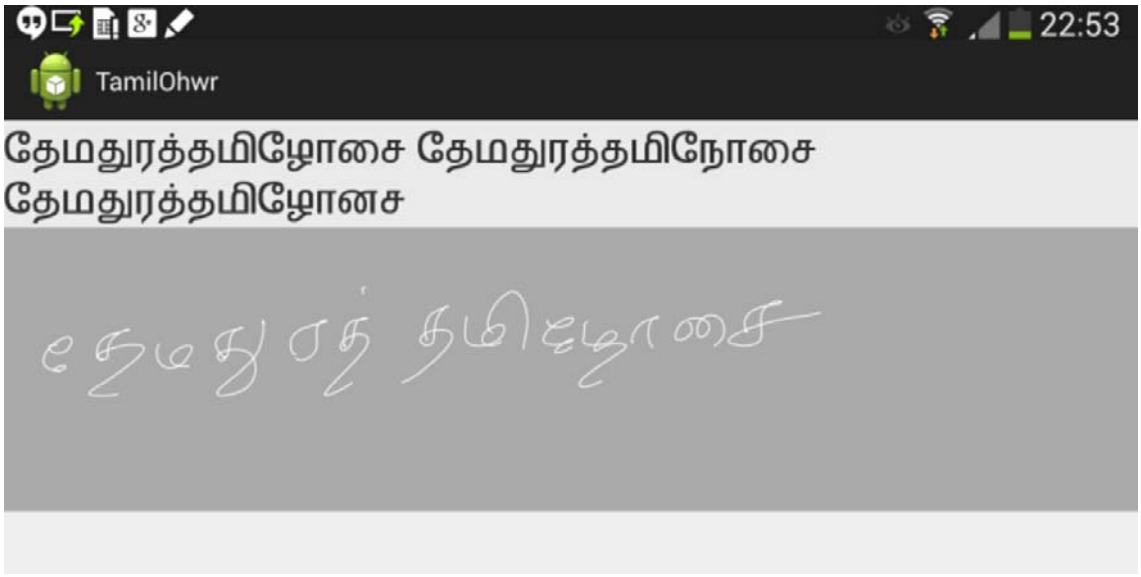

Fig. 4. A third sample handwritten Tamil input and the top three possibilities
for the recognized word obtained using bigram models and the SVM recognition scores.

References

[1] A. Bharath and S. Madhvanath. Hidden Markov Models for online handwritten Tamil word recognition. In Proc. IWFHR, pages 506–510, 2007.
[2] S. Sundaram and A. G. Ramakrishnan. Attention-feedback based robust segmentation of online handwritten isolated Tamil words. ACM Transactions on Asian Language Information Processing, 12(1), March 2013.
[3] N. Joshi, G. Sita, A. G. Ramakrishnan, and S. Madhvanath. Tamil handwriting recognition using subspace and DTW based classifiers. In Neural Information Processing, pages 806–813. Springer, 2004.
[4] A. G. Ramakrishnan and B. Urala. Global and local features for recognition of online handwritten numerals and Tamil characters. In Proc. International Workshop on Multilingual OCR (MOCR), 2013.
[5] V. Deepu, S. Madhvanath, and A. G. Ramakrishnan. Principal component analysis for online handwritten character recognition. In Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, volume 2, pages 327–330. IEEE, 2004.
[6] HP Labs India. Isolated Tamil Handwritten Character Dataset. http://lipitk.sourceforge.net/hpl-datasets.htm.
[7] S. Sundaram and A. G. Ramakrishnan. Attention feedback based robust segmentation of online handwritten words. Indian Patent Office Reference No., (03974/CHE/2010).
[8] S. Sundaram, B. Urala, and A. G. Ramakrishnan. Language models for online handwritten Tamil word recognition. In Proc. Workshop on Document Analysis and Recognition, 2012.
[9] B. Urala and A. G. Ramakrishnan. Identification of Tamizh script on Tablet PC. In Proc. of the Tamil Internet Conference (TI 2013), Kuala Lumpur, Malaysia, August 2013.