Identification of Tamizh script on Tablet PC

Bhargava Urala and A G Ramakrishnan
Department of Electrical Engineering, Indian Institute of Science, Bangalore, India.

Abstract

Research on Tamil script has mostly dealt with the identification of isolated letters. Different classifiers such as, hidden Markov models (HMM), neural networks and support vector machines (SVM) have been used along with a selection of features such as Fourier, wavelet, angular and directional features. The challenge of Tamil word recognition that has been addressed in the literature so far can be divided broadly into two approaches: (i) Recognition of individual strokes and then concatenating them using appropriate models to detect a word or a compound character. (ii) Grouping of strokes to form symbols or stroke groups, which may or may not constitute a compound character and recognition of these symbols to form words. The second method needs a framework for efficient segmentation of strokes into stroke groups and post processing techniques based on statistical language models. Use of word lists have also been investigated to enhance Tamil word recognition rates.

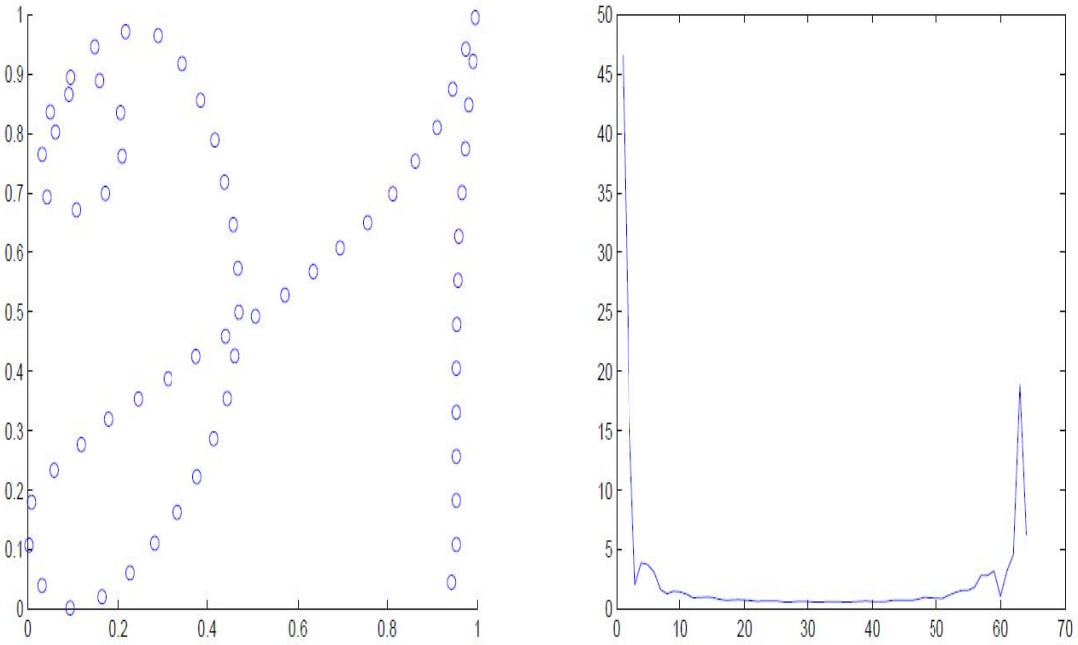
We portray experiments with preprocessed (x,y) coordinates, Fourier and derivative features for recognizing individual Tamil characters and the integration of segmentation, feature extraction, classification and post-processing steps into a dynamic link library for use on a Tablet PC.

Preprocessing

Each stroke group is smoothed with a Gaussian kernel, normalised to a 1 X 1 box and resampled to contain a fixed number (64) of points. This preprocessing removes gitter and accounts for variations in size and speed. An example of a preprocessed stroke group is shown in Figure 1(a).

Deriving Features

We have studied the use of local and global features extracted from the preprocessed data for training the classifier. As the names suggest, local features capture minor variations occurring in the data in small neighbourhoods and global features capture the overall shape of the character. We use x,y coordinates of normalized symbols and their first derivative as the local features and truncated Fourier coefficients as the global features.



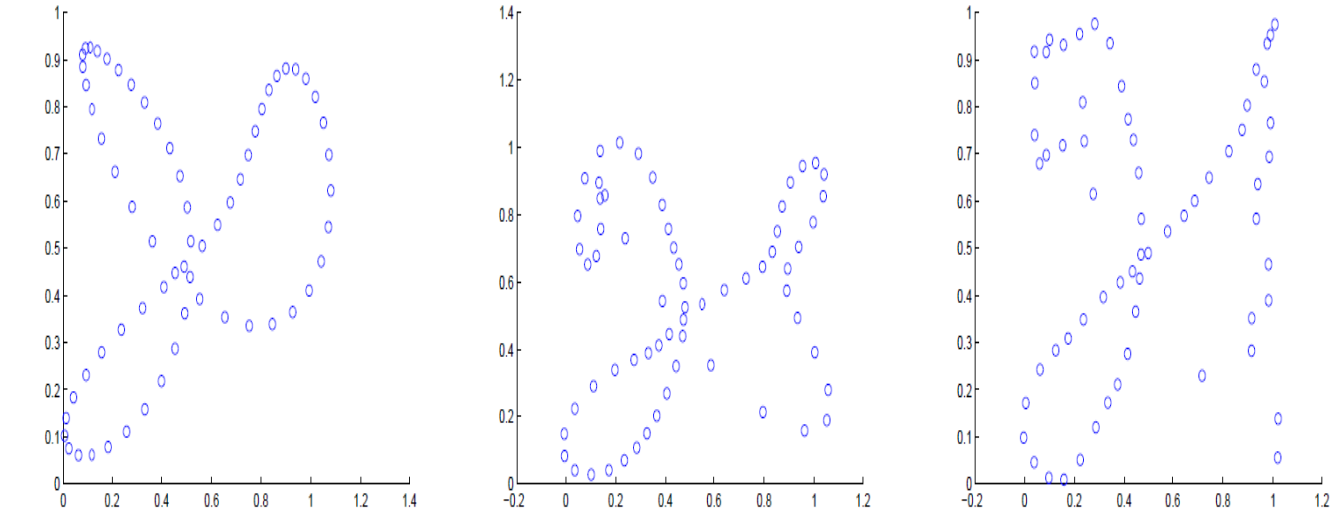
(a) Preprocessed character (vowel /a/) (b) Magnitude of its DFT

Figure 1: A sample preprocessed character and the magnitude of its DFT

It is well known that in the case of discrete Fourier transform (DFT), most of the energy in the signal is often contained in very few coefficients. Therefore, we also experimented with truncating the Fourier transform to less number of coefficients. The features were extracted from the samples of the training set of IWFHR isolated Tamil symbol database [1] and validated on the test set of the same database. We list below the features (local, global and combined local and global features) that we experimented with.

1. Preprocessed (x,y) coordinates: Each normalized symbol consists of 64 (x,y) points. Therefore, the length of this feature vector is 64 x 2 = 128.
2. Fourier descriptor: In this case, the preprocessed coordinate vector is treated as a vector of 64 complex points z = x + jy [2] and its DFT is taken, which results in a 64-point complex valued vector. Therefore, the length of feature vector is 64 x 2 = 128.
3. Truncated DFT: In this case, the Fourier coefficients obtained from DFT are truncated to 32 complex points. We experimented with lengths of 8, 16 and 32. The reconstructions from 8, 16 and 32 complex points can be seen in Figs. 2(a), 2(b) and 2(c), respectively. It can be seen that 32 complex points give an acceptable reconstruction. Hence, the chosen feature vector length is 32 x 2 = 64.

4. Combination of preprocessed (x,y) coordinates and truncated DFT features - Feature vector length = 128 + 64 = 192.
5. Concatenation of the sequence of preprocessed (x,y) coordinates, truncated DFT coefficients and the sequence of first derivative features, Accordingly, the feature vector length is 128 + 64 + 128 = 320.



(a) From 8 point complex-valued Fourier coefficients

(b) From 16 point complex-valued Fourier coefficients

(c) From 32 point complex-valued Fourier coefficients

Figure 2: Reconstruction of the character from the truncated Fourier transform coefficients

Segmentation framework

Segmentation of handwritten strokes into Tamil symbols consists of two steps. The initial segmentation is based on the horizontal overlap between bounding boxes of consecutive strokes. The second step is called attention feedback [3] and involves class labels and SVM confidence values obtained from the classifier stage as well as certain pen and stroke displacement statistics measured from a large Tamil handwritten character database to improve the results of initial segmentation.

SVM Classifier

We use a support vector machine (SVM) with radial basis function (RBF) as kernel trained on the features extracted from the training database, which consists of a large number of samples

of 155 unique Tamil symbols. The SVM gives a class label and an associated confidence level (between 0 and 1), both of which are used for improved word recognition.

Post processing

Statistics of co-occurrence of Tamil symbols estimated from the Emille corpus of Tamil text along with SVM confidence levels are used to generate N-best choices of symbol strings for a given handwritten word [4], using a Viterbi lattice. The symbol string is then converted to a Unicode string using a finite state transducer.

Application and usage on Windows Tablet PC

The Windows Tablet PC has a touch screen and a pen interface, which can be used to write and record handwritten data. An application developed by GIST, CDAC Pune serves as the front end, whereas dynamic link libraries (.dll) developed by us in C++ with functions and blocks to achieve the tasks described in the previous sections, act as the recognition engine. Along with the recognition of Tamil words, provision for recognition of Indo-Arabic numerals has been made with the same engine as well. A snapshot of the application in use is shown in Fig. 3. The engine has a high recognition rate exceeding ninety percent and a speedy performance with the average recognition time for a single symbol or stroke group being about 45 msec. Since the application is meant for practical use, a heuristics-based algorithm has also been developed and implemented with the engine, which detects and corrects the delayed strokes and also detects and removes the overwritten strokes. The online handwritten word recognition engine has been integrated with multiple form-filling applications.

Acknowledgments

The sample application we used to illustrate the recognition performance of our Tamil engine has been developed by CDAC, Pune as part of a consortium project funded by Technology Development for Indian Languages (TDIL), Department of Information Technology, Government of India. We thank CDAC, Pune and TDIL for the same. We also thank Dr. Suresh Sundaram, Assistant Professor, IIT Guwahati for his suggestions on this work.

References

1. Madhvanath, S and Lucas, S M. IWFHR 2006 online Tamil handwritten character recognition competition. Proc. Intl Conf. Frontiers in Handwriting Recognition.

2. J. Rajkumar, K. Mariraja, K. Kanakapriya, S. Nishanthini, and V. S. Chakravarthy. Two schemas for online character recognition of Telugu script based on Support Vector Machines. In Proc. ICFHR, pp. 565–570, 2012.
3. Suresh Sundaram and A. G. Ramakrishnan, “Attention feedback based robust segmentation of online handwritten words,” Indian Patent Office Reference. No: 03974/CHE/2010.
4. Suresh Sundaram, Bhargava Urala and A. G. Ramakrishnan, “Language models for online handwritten Tamil word recognition,” Proc. Workshop on Document Analysis and Recognition (DAR 2012), 16 December 2012, IIT Bombay, Mumbai, India

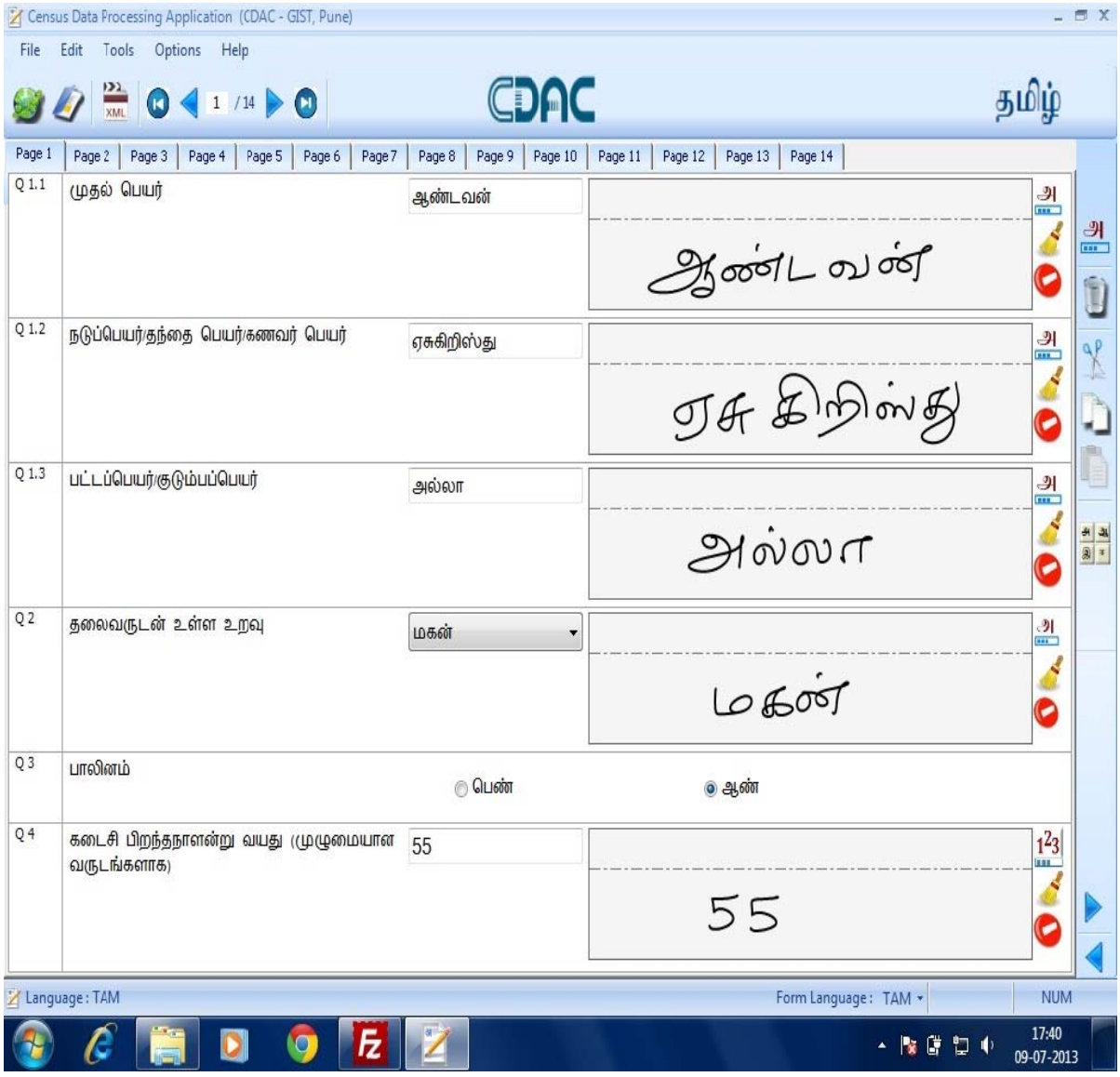


Fig. 3. A snapshot showing the recognized outputs of our Tamil handwriting recognition engine.