A HMM Based Online Tamil Word Recognizer

Rituraj Kunwar Shashi Kiran

Suresh Sundaram A G Ramakrishnan*

Medical Intelligence and Language Engineering Laboratory

Indian Institute of Science, Bangalore -560012, India.

1 Introduction

In Online handwriting recognition, a machine recognizes, as a user writes on a pressure sensitive screen with a stylus. The stylus captures information about the position of the pen tip as a sequence of points in time. The sequence of point between a PEN DOWN and PEN UP signal defines a stroke. This spatio-temporal information of the character being traced is the only input available to the online recognition system. Also given a character, one can capture the different writing styles using the information from the stylus. Tamil is a popular South Indian language for a significant population in countries such as Singapore, Malaysia and Sri Lanka besides India. There are 313 characters in Tamil alphabet. Of these, there are 12 pure vowels and 23 pure consonants (including the 5 consonants derived from Sanskrit (Grantham consonants)), the remaining being consonant vowel combinations, wherein the vowels modify the consonants and 2 special characters ∴ and ஸ்ரீ. It has been found that to represent all the possible 313 characters, a set of 155 symbols is sufficient. This paper deals with the problem of recognizing online handwritten Tamil words with a Hidden Markov Model framework. Given a Tamil word, we first run a segmentation algorithm to identify the individual symbols. A Tamil symbol may be written with different number of strokes. The extracted symbols are subjected to the following preprocessing modules: smoothing to remove noise, resampling to a fixed number of points for speed normalization and size normalization. A set of seven features are derived at each sample point of the preprocessed Tamil symbol. These features are then fed to the Hidden Markov Model classifier for recognition of the Tamil symbol. . Based on Unicode generation rules derived from the language, stroke groups are generated from the Tamil symbols. A Tamil word corresponds to a set of stroke groups. Fig 1 gives a snapshot of a handwritten Tamil word **கவசம்**, collected with a TABLET PC, together with the recognized output using the Hidden Markov Models as the classifier.

2. Segmentation

Word level data is to be segmented to character level as the modeling of the data is done at the character level. Then the recognition is performed at the character level and the results are concatenated to form the words. Segmentation of the Tamil words into characters is simple when compared to the English cursive handwriting. In Tamil script, two strokes of the same character either overlap or touch at some point. The vowel or consonant modifiers are written as separate symbols and their models are built separately.

* Corresponding Author email id : ramkiag@ee.iisc.ernet.in

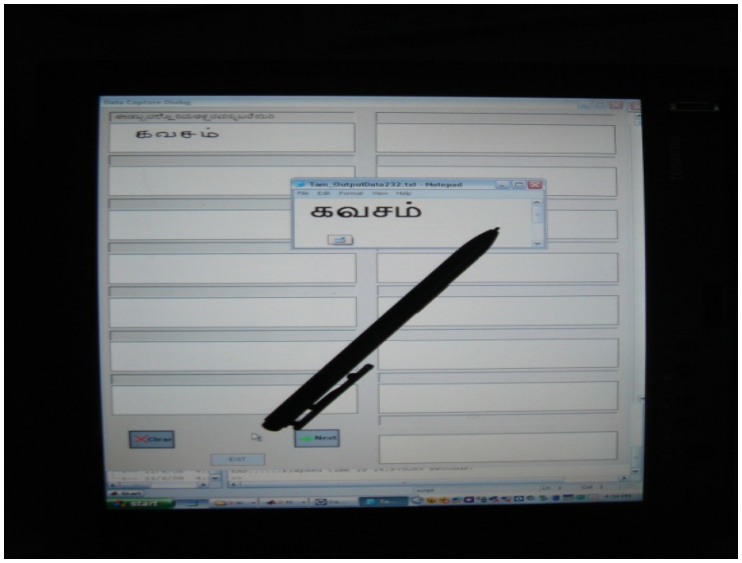


Fig 1. Data Collection Device with recognized output.

For each current stroke, the next stroke is taken and checked whether the x co-ordinate of its starting point is less than the maximum x coordinate of the previous (i.e. current) stroke. Saying in other words, we see if the next stroke overlaps with the current stroke. If there is an overlap (within a threshold empirically set) then the successive stroke is concatenated with the current stroke to form the same symbol (Fig. 2). Other wise the future stroke is considered as the current stroke / new symbol and the same procedure is repeated.

However, there are cases wherein the last part of the trace of the successive stroke overlaps with the current stroke, as shown in Fig 3. In such scenarios, one needs to treat these strokes as 2 separate entities. Hence we formulate our condition as follows: If the x co-ordinate of the last point of the current stroke is less than the x max of the previous stroke, we do not concatenate these strokes.

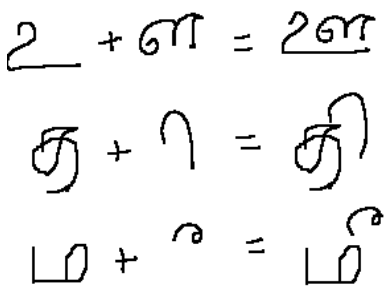


Fig 2. Simple overlap of the first part of the successive stroke with the current stroke

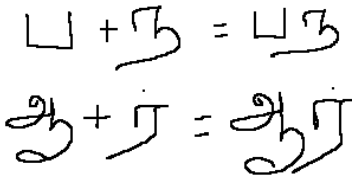


Fig. 3 Simple overlap of the last part of the successive stroke with the current stroke

3 Preprocessing

The raw character, captured from the device, comprising of N points sampled uniformly in time, $\{x_i^{raw}, y_i^{raw}\}_{i=1}^{N}$ is first smoothed using a Gaussian mask, that is applied to the x and y coordinates independently.

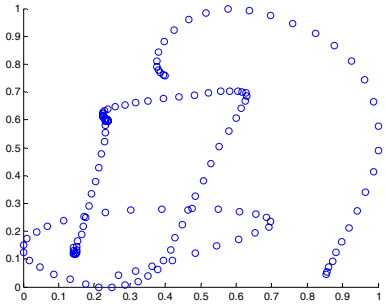
$$x_t^{smooth} = \sum_{i=-3\sigma}^{3\sigma} w_i x_{t+i}^{raw}$$

$$y_t^{smooth} = \sum_{i=-3\sigma}^{3\sigma} w_i y_{t+i}^{raw}$$

The weights w_i are given by $w_i = \frac{e^{\frac{-i^2}{2\sigma^2}}}{\sum_{i=-3\sigma}^{3\sigma} e^{\frac{-i^2}{2\sigma^2}}}$. We then normalize each character to a standard size using the transformation.

$$x_i = \frac{x_i^{smooth} - x_{\min}}{x_{\max} - x_{\min}} \qquad y_i = \frac{y_i^{smooth} - y_{\min}}{y_{\max} - y_{\min}}$$

Here x_{min} and x_{max} denote the minimum and maximum x coordinate of the raw character. y_{min} and y_{max} represent the minimum and maximum y coordinate. The characters are r-sampled to fixed number of points (in our work 60) uniformly in space. First, we find the length of each character and the length of all the strokes constituting it. We then assign different number of points to each stroke, depending on the ratio of the length of the stroke to that of the character. The first and last points of the character are retained and the points in between are found which are spaced uniformly. Fig 4(a) (b) depict a snapshot of the raw and preprocessed character **கி**.



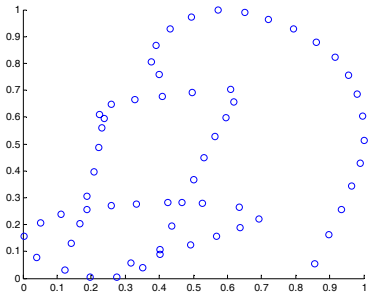


Fig 4 (a): Raw Character

4 (b): Pre processed Character

3. Feature Extraction

Each of the segmented characters are subjected to a feature extraction module. The following features are derived at each sample point of the characters.

Normalized x and y coordinates:

The x and y coordinates of the normalized sample are taken as two features for the recognition.

Normalized x and y first derivative:

The variations in the x and y coordinates are found independent of each other which give the shape features locally at each point. At the current point (x_j, y_j), a window is taken covering the past and the future points and the derivative is calculated using the formulae given below

$$x_j' = \frac{\sum_{i=1}^{2} i(x_{j+i} - x_{j-i})}{2\sum_{i=1}^{2} i^2} \qquad y_j' = \frac{\sum_{i=1}^{2} i(y_{j+i} - y_{j-i})}{2\sum_{i=1}^{2} i^2}$$

The normalized first derivatives in x and y direction at (x_j, y_j) is given by

$$x'_{norm} = \frac{x_j'}{\sqrt{x_j'^2 + y_j'^2}} \qquad y'_{norm} = \frac{y_j'}{\sqrt{x_j'^2 + y_j'^2}}$$

Normalized x and y second derivatives:

The second derivatives are computed similar to the first derivatives.

$$x''_j = \frac{\sum_{i=1}^{2} i(x'_{j+i} - x'_{j-i})}{2\sum_{i=1}^{2} i^2} \qquad y''_j = \frac{\sum_{i=1}^{2} i(y'_{j+i} - y'_{j-i})}{2\sum_{i=1}^{2} i^2}$$

The normalized second derivatives in x and y direction at (x_j, y_j) is given by

$$x''_{norm} = \frac{x''_j}{\sqrt{x''^2_j + y''^2_j}} \qquad y''_{norm} = \frac{y''_j}{\sqrt{x''^2_j + y''^2_j}}$$

Curvature

Curvature at a point on a curve is the inverse of the radius of the osculating circle at that point and can be found using the first and second derivatives as given below.

$$C = \frac{x''_{norm} y'_{norm} - y''_{norm} x'_{norm}}{(x'^2_{norm} + y'^2_{norm})^{3/2}}$$

4. Hidden Markov Models

The derived features are then fed to the Hidden Markov Model classifier for recognition. More details on the description of HMMs can be found in [1] [2]. Each Tamil symbol is modeled using a separate HMM .Training of the models is performed using the well known Baum Welch Estimation. The Bayesian approach is adopted for recognizing the label for the test symbol. The IWFHR Database has been used for our experiments. This database contains around 345 samples (written on a Tablet PC) for each of the 155 symbols. The resulting system gives an accuracy of 84% at the symbol level.

5. Acknowledgements

The authors are grateful to Technology Development for Indian Languages (TDIL), Department of Information Technology (DIT), MCIT, Government of India for funding this project and to AVM Matriculation Higher Secondary School, Chennai for contributing their students' time to collect training data for our project.

References

1. L. R. Rabiner and B. H. Juang, ``An introduction to hidden Markov models," *IEEE ASSP Mag.*, pp 4--16, Jun. 1986.

2. Duda, Hart, Stork, "Pattern Classification", Second Edition.