Development of OHWR System for Tamil

A. G. Ramakrishnan, Bhargava Urala K, Suresh Sundaram, Harshitha PV
Department of Electrical Engineering, Indian Institute of Science, Bangalore

Abstract: A comprehensive recognition system has been developed for open vocabulary, online handwritten text in Tamil language. A page of text can be segmented at the line, word and then the symbol level. The symbols are recognized using a SVM classifier with RBF kernel trained to recognize 155 distinct Tamil symbols, which can make up all the 313 different characters in Tamil. By analyzing the cross-validation performance of the classifier, the sets of confused symbols have been identified. If the recognition label of a symbol corresponds to that of a confused symbol, then the feature vector of the corresponding stroke group is fed to an expert classifier trained only on the set of confused symbols. Then, the recognized symbols of each word are corrected using a symbol level bigram model derived from a huge text corpus. Finally, the sequence of symbol labels corresponding to each word is converted to the Tamil Unicode sequence using a set of rules. The recognition engine at the level of a handwritten word has been developed in C as a .dll and integrated with the census data collection application developed by CDAC Pune. On the annotated dataset of 45,405 words collected from over a hundred Tamil writers, the engine has a recognition performance of 83.2% at the symbol level and 54.2% at the word level, without the use of expert classifiers. A separate SVM has been trained to recognize the Indo-Arabic numerals 0 to 9, with a cross validation accuracy of 98%.

Key words and phrases: Tamil, xml standard, handwriting recognition, page recognition, dominant overlap segmentation, attention-feedback segmentation, delayed strokes, stroke, stroke group, symbol, Fourier descriptor, support vector machine, SVM, bigram models, expert classifiers, census data collection, tablet PC, Android.

1. Choosing the recognition primitives

Most Indian languages, excluding Bangla, are written with individual letters, non-cursively. So, one can attempt to segment and recognize the individual characters and through that the word, sentence and so on. The original Tamil script contains 12 pure vowels (5 short vowels, 5 long or stressed vowels and 2 diphthongs), 18 pure consonants (6 stops, 6 nasals and 6 semivowels) and a special character /ah/. Each pure consonant can combine with each vowel to generate a total of 18 × 12 = 216 consonant-vowel (CV) combinations. These add up to a total of 247 Tamil characters. In this work, however, we have included five additional pure consonants (used to represent the consonants borrowed from Sanskrit) [1] and another special symbol /sri/. These consonants contribute an additional 5 × 12 = 60 CV combinations. The

complete 313 character set consists of 276 CV combinations, 12 vowels, 23 pure consonants and two special characters [2]. All of these characters are supported by Unicode.

Based on an analysis of the complete character set, we come up with a strategy to choose the minimum number of entities/ symbols for recognizing the 313 characters, taking into account the fact that many of the symbols may be written with a single or multiple strokes. With the above analysis, it is found that the set of 155 distinct classes (henceforth referred to in this work as 'symbols') is sufficient to form (and hence recognize) all the 313 characters considered. The constituents of the 155 distinct symbols are:

- 11 pure vowels (excluding /au/)
- 23 pure consonants
- 23 base consonants
- 23 CV combinations of /i/
- 23 CV combinations of /I/
- 23 CV combinations of /u/
- 23 CV combinations of /U/
- 6 Additional symbols ((VM of /A/) , (VM of /e/) , (VM of /E/), (VM of /ai/), /ah/ and /sri/.)

It is to be noted that the HP Labs Tamil online handwritten character dataset, which was created for a competition in IWFHR 2006, consists of samples across 156 symbols or stroke groups [3]. However, the 156^{th} class in the dataset is a combination of already present classes 10 and 28 and is therefore neglected. We use the rest of the 155 classes as recognition primitives in our classifier.

In a recent paper on similar recognition study [4] by HP Labs, only 83 distinct symbols of written Tamil script have been considered. In the above paper, the entire work reported deals with only 85 distinct Tamil words and 70 distinct Hindi words. The authors use a finite lexicon to recognize the test words. In their work, the matra for /i/ and /I/ are taken as separate symbols. While generally these are written as separate strokes, we have come across many cases in our database, where the writer writes the CV combination involving /i/ or /I/ matra as a single stroke. The approach dealing with them as separate symbols will fail in such cases, but in a limited lexicon problem, it can be corrected easily by lexicon-driven recognition or by post-processing with the lexicon, as the above authors have performed.

Tamil is a morphologically highly rich language and any verb root can get modified by one or more suffixes corresponding to gender, number, tense, person, etc. and form over 6000 distinct word. So, Tamil cannot be contained within any finite vocabulary. Hence, our aim is come up with a strategy for open vocabulary recognition, and we want to see how far we can go with that. It is for this purpose, that we chose a larger set of symbols, which facilitates us to handle the different ways in which a CV is written by people - single, two or multistrokes.

2. Data collection and annotation

Our isolated word database consists of 45,405 words, with a total of 2,53,095 symbols. These words were collected on a Tablet PC with WACOM digitizer and were written by 181 different regular Tamil writers from several schools and colleges in Coimbatore and Chennai. This database is composed of 2000 unique words, which were selected so as to include all of the consonants, vowels and consonant vowel combinations possible in Tamil script. This entire database has been annotated at the stroke group level [2]. An XML standard has been proposed by the online handwriting recognition consortium members for this annotation [5], which is applicable for all Indian languages. It starts at the level of a page and hierarchically involves paragraphs, lines, words, stroke groups, strokes and substrokes, some of which can be optional.

Numeral data 0 - 9 was collected by the project staff of MILE lab from 75 different writers and approximately 400 samples per class have been obtained. Each user has written 5-7 samples. A classifier was trained using these data, with an SVM cross validation accuracy of over 98%.

Our page and paragraph database consists of 40 paragraphs of data. 25 paragraphs were collected on TabletPC and 15 paragraphs were collected on A4 sheets using HiTech digitizer. We also assisted Sushil M of CDAC Pune in the collection of isolated word data, paragraph and sample census data from SRM University, Chennai towards independent testing by CDAC.

We have also used the HP Labs Tamil online handwritten character dataset, which has been made publicly available for research. It consists of 50385 training samples and 26926 test samples across 156 symbols or stroke groups, out of which 155 are used by us.

3. Isolated Word Recognition Engine

Figure 1 shows the overall block diagram of our recognition engine. The major steps that constitute the Tamil recognition engine for isolated, online handwritten words are:

- Segmentation of the input word (a sequence of strokes) into symbols or stroke groups - based on the horizontal overlap between the bounding boxes of strokes and certain pen displacement cues.
- Pre-processing of these symbols for noise-free, scale and velocity invariant feature extraction.
- Extraction of global (Fourier descriptor) and local features from the pre-processed symbols.
- Recognition of the symbols by an SVM-RBG classifier.
- Correction of segmentation of the word, based on recognition labels and their scores.
- Post-processing of the recognized sequence of class labels that represent the word using symbol-level bigram models.

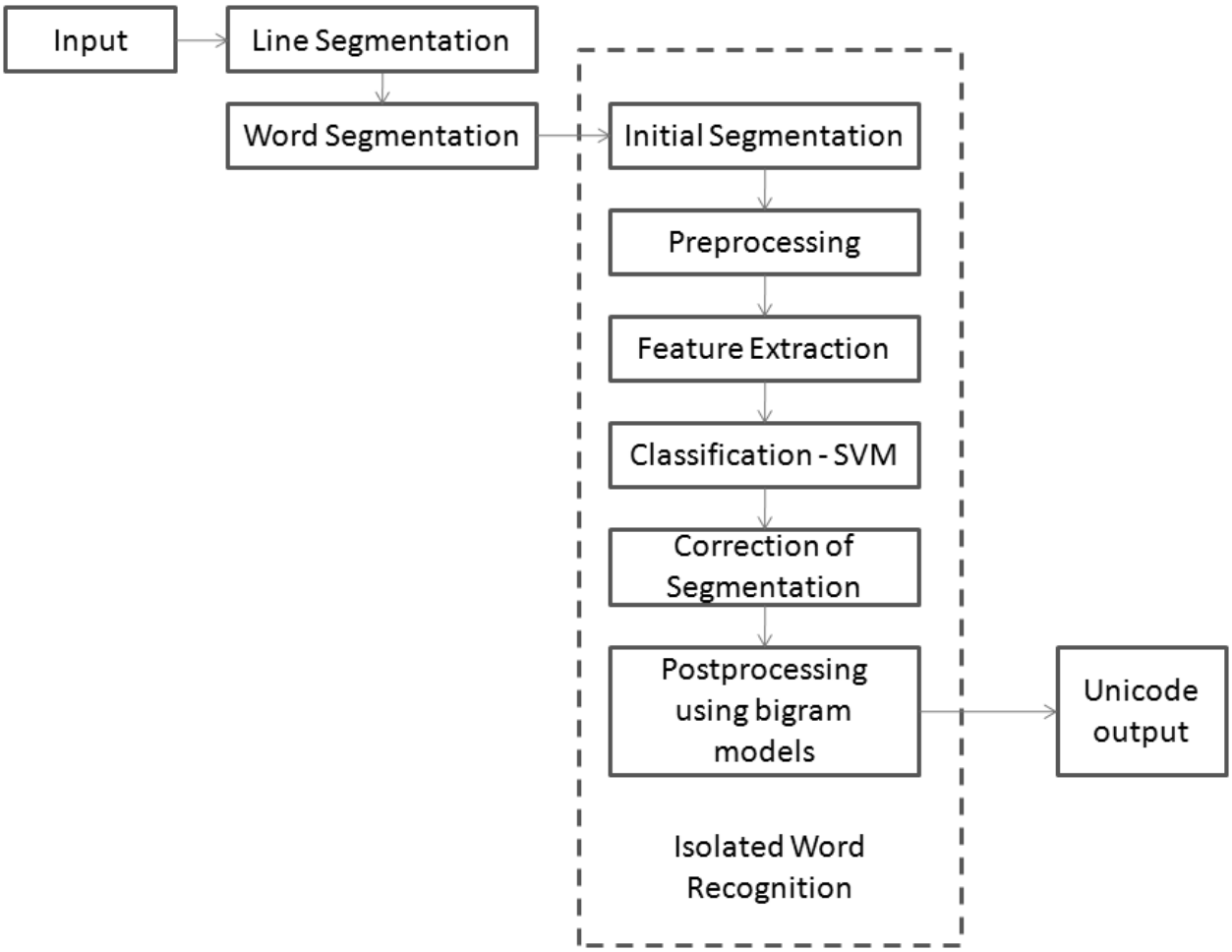


Fig. 1. Block diagram of the recognition engine for Tamil online handwriting.

3.1 Dominant overlap segmentation

The extent of horizontal overlap between a stroke group and the successive stroke is calculated using equation (1) and a merge or split decision is taken, respectively, if the value is above or below an empirically determined threshold (0.2 in our case) [6, 7]. A stroke group consists of one or more strokes and corresponds to one of the 155 symbols that partially or wholly describe a distinct Tamil akshara.

$$O_k^c = max\left[\frac{x_M^{S_k} - x_m^{s_c}}{x_M^{S_k} - x_m^{S_k}}, \frac{x_M^{S_k} - x_m^{s_c}}{x_M^{s_c} - x_m^{s_c}}\right] \quad (1)$$

where, s_c and S_k indicate the current stroke and stroke group, respectively; x_M and x_m refer to the bounding box maximum and minimum in the horizontal direction.

Certain styles of writing result in spurious merge decisions being taken. Therefore, we calculate two displacement values d_x and d_y between s_c and S_k using the equations (2) and (3). If $d_x >= 0$ or if $d_y <= 0$, then the decision is not to combine the stroke group with the next stroke.

$$d_x = x_1^{s_c} - x_M^{S_k} \quad (2)$$

$$d_y = y_1^{s_c} - y_{last}^{S_k} \quad (3)$$

3.2 Pre-processing

The main purpose of pre-processing is to nullify the effect of noise, account for variations in writing size and style and remove duplicate points in the stroke. It is carried out in three steps - smoothing, normalisation and resampling [8]. A sample Tamil letter before and after pre-processing (smoothing, normalization and resampling along the trace) is shown in Fig. 2.

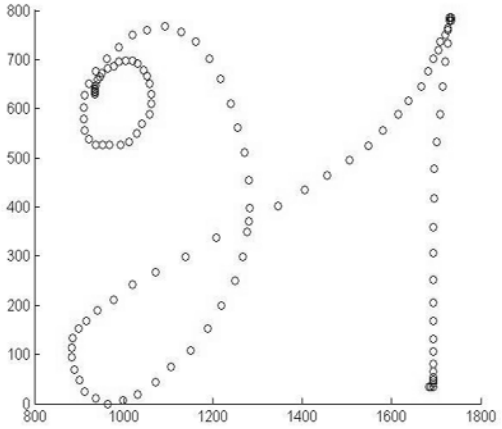


Fig. 2 (a) Raw handwritten character (vowel /a/)

- Smoothing - Every stroke of the stroke group is independently smoothed using a Gaussian filter.
- Normalisation - Range normalisation of the stroke group is carried out by linear mapping of its x and y coordinates to the range 0 to 1.
- Resampling - We uniformly sample the stroke group along its arc length. The number of resampled points is fixed at 64. In case a stroke group consists of more than one stroke, then the number of points allotted to each stroke is proportional to the arc length of the stroke and the sum of the number of points in all the strokes is ensured to be 64.

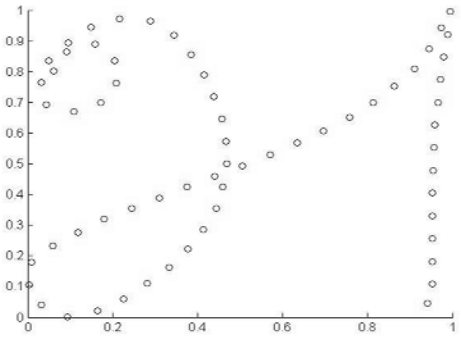


Fig. 2(b) Preprocessed character.

3.3 Experiments on feature extraction and classification

We have experimented with three different types of features:

- Local features only: Using the HP training dataset, we trained a classifier on the preprocessed (x,y) features alone and performed recognition on the test dataset.
- Global features only: We independently used discrete Fourier transform (DFT) and discrete cosine transform (DCT) of the preprocessed (x,y) coordinates of each symbol to train the classifier. We have also investigated the effect of truncating the Fourier transform feature vector.
- Combination of local and global features: We trained the classifier on feature vectors obtained by concatenating the local features and truncated Fourier features. We also experimented with using first derivative features along with the previously mentioned features.

3.3.1 Local features – point sequence

The number of points in any preprocessed symbol is 64, which results in a feature vector length of 128. An SVM classifier with RBF kernel is trained on the preprocessed data from the IWFHR database and a grid search with 5-fold cross validation is performed to find the training parameters that lead to the best performance with the given training data.

3.3.2 Global Features – Fourier descriptor

The discrete Fourier transform is used to extract global features from the handwritten data. We treat the preprocessed (x,y) coordinates as a 64-point complex-valued vector and then take its Fourier transform, i.e

$$F(k) = \sum_{i=0}^{N-1} z_i e^{-j2\pi ik/N}$$

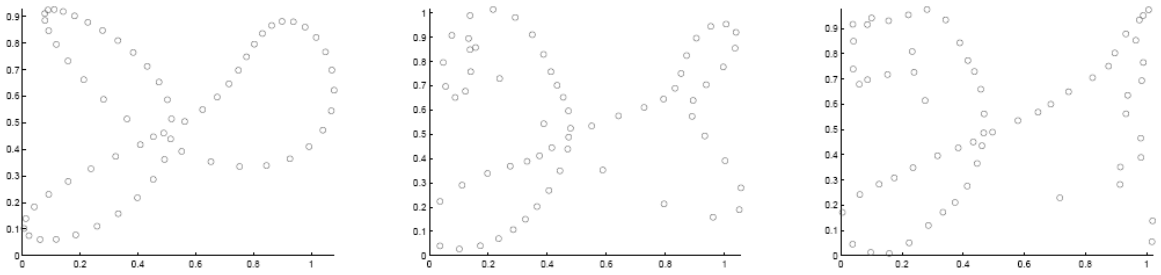
where, N = 64 and $z_i = x_i + jy_i$. Figure 3 shows the magnitude of the DFT of the preprocessed character shown in Fig. 2(b).

(a) Preprocessed character (vowel /a/)

(b) Magnitude of its DFT

Figure 3. Fourier descriptors obtained from the preprocessed character in Fig. 2(b).

We experimented with truncating the feature vector F to 8, 16 and 32 complex points and reconstructing the symbol by taking the inverse Fourier transform of the truncated feature vector. The reconstructions can be seen in Figs. 4 a), b) and c) respectively. Truncating to 32 complex points and subsequent inverse transform produces an acceptable reconstruction and hence is used for the experimentation for recognition performance.



(a) From 8 point complex-valued Fourier coefficients

(b) From 16 point complex-valued Fourier coefficients

(c) From 32 point complex-valued Fourier coefficients

Fig. 4. Fidelity of reconstruction of an online Tamil handwritten character from its Fourier descriptor truncated to different extents.

The discrete cosine transform (DCT) was also considered as a possible global feature. DCT is obtained using,

$$C(k) = \sum_{i=0}^{N-1} w_k z_i \cos \frac{\pi k(2i+1)}{2N}$$

where, N = 64, $z_i = x_i + jy_i$, $w_k = 1/\sqrt{N}$, for k = 0 and $w_k = 2/\sqrt{N}$, for 1 <= k <= N - 1.

3.3.3 Combined Global and Local Features

Global features are good at capturing the overall shape information of the character. However, they generally do not work well with similar classes that have minor point-wise variations. In contrast, local features are extracted at each point and therefore can facilitate inter-class separation. To get both the advantages, we experimented with the concatenation of global and local features.

First derivative features were also computed and concatenated to the above vector and used to train an SVM with RBF kernel. The first derivative features were computed using the equations

$$d_x(i) = \frac{(x_i - x_{i-1}) + \frac{(x_{i+1} - x_{i-1})}{2}}{2}, 2 \leq i \leq 63$$

and

$$d_y(i) = \frac{(y_i - y_{i-1}) + \frac{(y_{i+1} - y_{i-1})}{2}}{2}, 2 \leq i \leq 63$$

As we can see from the above equations, the value of first and last points of d_x and d_y cannot be computed. Therefore, we make $d_x(1) = d_x(2)$, $d_y(1) = d_y(2)$, $d_x(64) = d_x(63)$ and $d_y(64) = d_y(63)$. Table I shows the cross validation and test accuracies [9] for different features on the complete test set of IWFHR 2006 online Tamil handwritten character database.

3.4 Other recognition experiments

The current Tamil engine actually has the above features, namely the combined global and local features fed to a SVM-RBF classifier. However, earlier we had carried out a number of other experiments with different classifiers. One of the earliest experiments obtained the principal component subspace for each symbol and used the distance of the test sample to each subspace for classification [8]. In another experiment, we explored the use of other features such as quantized slopes and dominant point coordinates for recognition with elastic matching algorithms (dynamic time warping classifier) for writer dependent recognition of isolated

characters [10]. Further experiments compared the subspace and DTW based classifiers under writer dependent, independent and adaptive conditions [11]. Dynamic space warping of strokes within the characters and a perceptual distance measure were explored for recognition [12]. The traces of the characters were fit with polynomials and the coefficients of the polynomial were used as features, along with second derivative features on two different classifiers: statistical dynamic time warping (SDTW) and GMM-HMM classifier [13]. We also tested hierarchical classification with PCA based nearest neighbour classification for the first stage and three DTW based classifiers for the second stage and in addition to the features mentioned above, quartile features were proposed and experimented with [14]. A bigram model at the level of the multisymbol characters was used to reduce the search space for recognition of each symbol in a word and this was combined with expert classifiers to disambiguate confused classes [15].

Table I. Results of recognition experiments with different combinations of features. No. of training and test samples for the 155 distinct Tamil symbols are 50385 and 26926, respectively.

L: Length of feature vector, CVA: Cross validation accuracy, TA: Test accuracy

Feature	L	CVA (%)	TA (%)
(x,y)	128	91.8	92.46
DFT	128	91.7	95.78
DCT	128	91.6	93.84
Truncated DFT	64	91.5	95.69
(x,y)+ truncated DFT	192	91.3	95.85
(x,y)+ truncated DFT+first derivative	320	91.5	95.86

3.5 Attention-feedback based correction of segmentation

To check for possible errors in segmentation, each suspected stroke group in the word is merged with the nearest stroke group. This merged stroke group is pre-processed and recognised by the classifier. If the average SVM confidence of the individual stroke groups is less than the confidence of the merged stroke group, the two stroke groups are merged [6,7] and we continue along the word.

3.6 Post-processing: Symbol level bigram models

3.6.1 Generation of bigram statistics

The data used to generate bigram statistics for the language model is obtained from the EMILLE corpus and the copyright free book data from the Project Madurai site. Tamil text corpus which is a collection of sentences where each word is a sequence of Tamil characters. The unicode sequence of every word in the Tamil text corpus is mapped to a class label sequence corresponding to the Tamil symbols used as recognition primitives. To generate bigram models, we first obtain the following statistics by counting the respective occurrences in the corpus:

- N_w - total number of words in the text corpus
- $N_s(w_i)$ - total number of occurrences of symbol w_i
- $N_{ss}(w_i, w_j)$ - total number of occurrences of symbol pair (w_i, w_j)
- $N_b(w_i)$ - total number of occurrences of symbol w_i at the starting position
- $N_e(w_i)$ - total number of occurrences of symbol w_i at the last position

A specific word W can be expressed as a series of p symbols $W = \{w_i\}$, $i \in [1, p]$. In the bigram model, we assume that the probability of occurrence of a symbol depends only on the previous symbol. Thus, probability of the word, using a first-order Markov dependency can be written as

$$P(W) = P_b(w_1)P(w_2|w_1)...P(w_i|w_{i-1})...P(w_p|w_{p-1})P_e(w_p)$$

where,

$$P(w_i|w_{i-1}) = \frac{N_{ss}(w_{i-1}, w_i)}{N_s(w_{i-1})}$$

Probabilities of class w_i being at the beginning/ end of a word are computed using

$$P_b(w_i) = \frac{N_b(w_i)}{N_w}$$

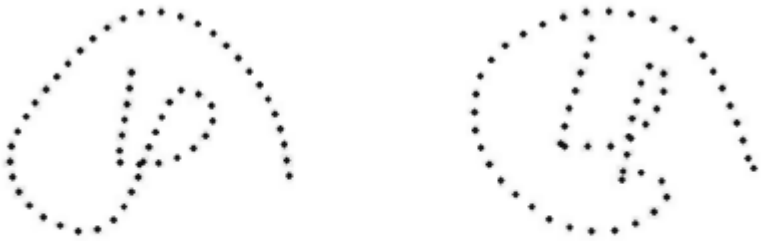
$$P_e(w_i) = \frac{N_e(w_i)}{N_w}$$

3.6.2 Using bigram models for post-processing

As mentioned previously, a word is treated as a first order Markov process, where each symbol depends upon the symbol that precedes it. The top N likely class labels to represent each stroke group and their respective recognition scores are taken as the states and a lattice is constructed with bigram probabilities as the transition weights. The standard Viterbi algorithm is then used to derive the N most likely class label strings that represent the word [16]. In our case, a value of N = 3 is found to be satisfactory.

3.7 Post processing: Disambiguation

In Tamil script, there are certain sets of symbols, which are similar in nature and are therefore confused by the SVM classifier. We have identified six such commonly occurring confusion pairs (see Table II) and built expert classifiers trained on (x, y) coordinate features in the discriminating regions between the two confused symbols [17]. An illustration of one of the confused pairs is shown in Fig. 5.



(a) Symbol for /mu/ (b) Symbol for /zhu/

Figure 5. An example pair of confused Tamil characters.

After the segmentation, we consider the most likely class label and corresponding likelihood as obtained from the SVM for each stroke group. If the recognized label belongs to the list of confused pairs, we use the appropriate expert classifier to reevaluate the input symbol to disambiguate between the two confused classes.

Table II. Some of the commonly confused pairs of Tamil symbols and the accuracies of the primary classifier for those pairs.

Symbol pair	Total no of symbols	No of confusions	Classifier Accuracy
/mu/ and /zhu/	349	26	92.6%
/Na/ and VM of /ai/	351	32	90.9%
/Ni/ and /Li/	364	32	91.2%
/La and /Na/	353	23	93.5%
/ki/ and /ci/	355	23	93.5%
/la/ and /va/	359	14	96.1%

4. Line and Word Segmentation

Both line and word level segmentation are carried out using script independent properties of the strokes such as the coordinates of the centroids of every stroke (x_c, y_c), bounding box minima (x_m, y_m), maxima (x_M, y_M) and heights and widths of every stroke.

4.1 Line Segmentation

To detect each line, we compare x_c values of consecutive strokes and check if they are in increasing order. To eliminate vowel modifiers from being detected as the start of new lines (due to decrease in x_c), we check for horizontal bounding box overlap with the previous stroke and also measure the difference in the y_c value between the two strokes. If the y_c value decreases beyond a certain threshold (1.25 * average stroke height in the page) and if there is no horizontal overlap with the previous stroke, we confirm and perform a line break. A further check is to see that the subsequent stroke also has a negative x-displacement relative to the previous stroke.

4.2 Word Segmentation

For word segmentation, we obtain the mean x-displacement between successive strokes, after merging all the successive strokes that have a horizontal overlap between them. A new stroke with horizontal overlap with the previous one indicates that the current stroke is possibly a vowel matra that is superposed on the previous stroke. Thus, its displacement is excluded from the computation of mean x-displacement. We define and compute the following quantities:

- Displacement of the i^{th} stroke, $b_x^i = x_m^i - x_M^{i-1}$.
  If b_x^i is negative, combine i^{th} and $(i-1)^{th}$ strokes and recompute the x_m and x_M of the combined stroke.
- Width of the i^{th} stroke, $w^i = x_M^i - x_m^i$
- Width threshold T_w is obtained as,

$$T_w = (1.25/N) \sum_{i=1}^{N} w^i$$

Now, any stroke (say, k^{th}) in the line is marked as the first stroke of a new word, if $b^k_x \geq T_w$. This works well, except when the writer puts a punctuation like comma or period almost in the middle of two words. They introduce two types of possible errors:

- Comma or period being falsely marked as the first stroke of a new word.
- Some word beginnings are missed because the computed value of stroke separation b_x is affected by the presence of the preceding comma or period.

In order to handle these errors, all the strokes with $w_i \leq 0.4 * T_w$ are considered to be punctuation marks. If a punctuation stroke is marked as word beginning then the mark is simply removed. If it is not marked as beginning, then we compute stroke separation b_x by neglecting the punctuation.

5. Test Results

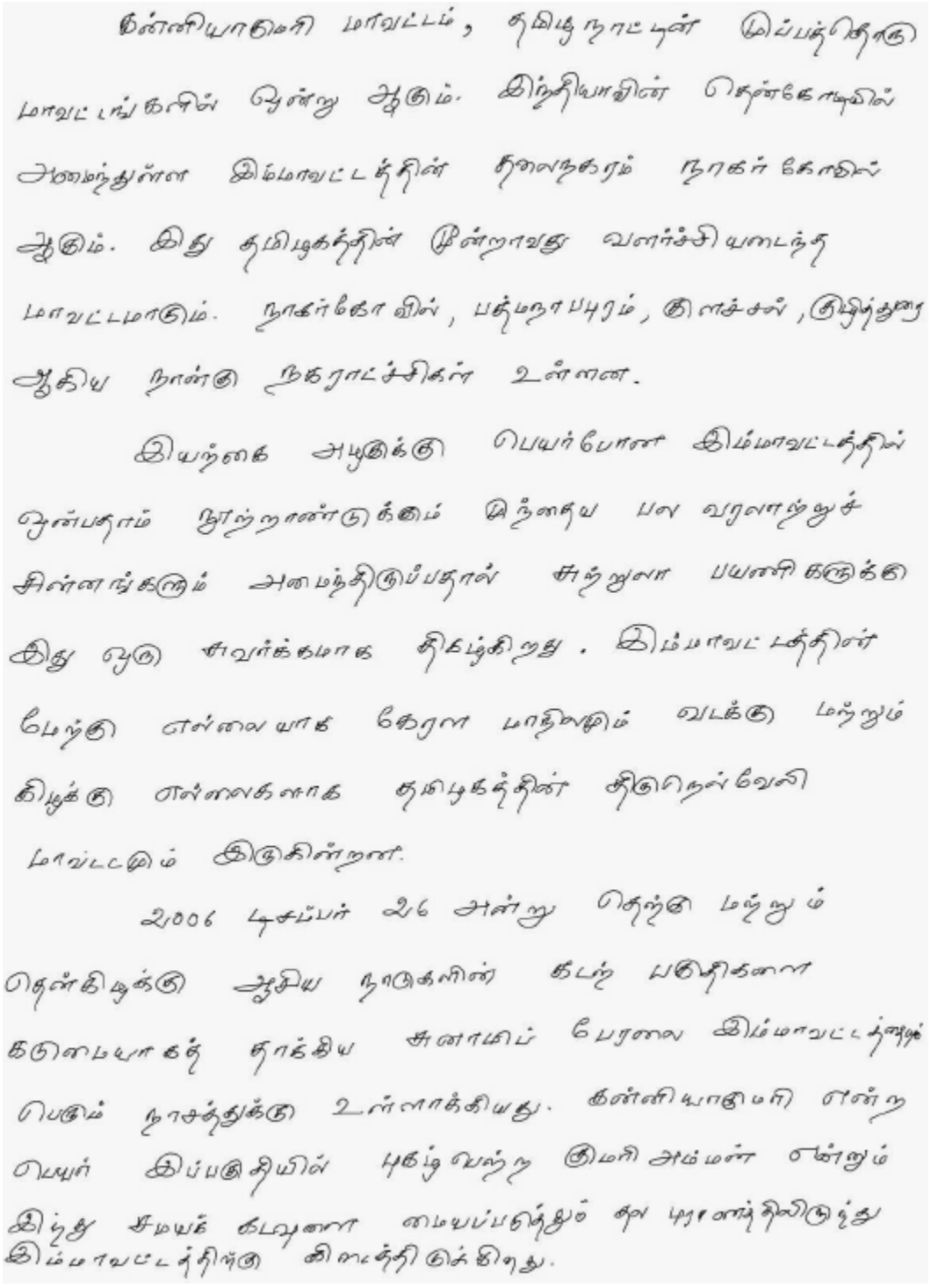
The results of running the isolated word recognition engine with and without the bigram language models at the level of symbols are listed in Table III.

Table III. Effectiveness of bigram language models on recognition of our word corpus.

SA: Symbol level accuracy, WA: Word level accuracy
Total No. of test words: 45, 405
Total No. of symbols in the test data: 2,53,095

Recognizer	SA (%)	WA (%)
SVM	78.52	40.05
SVM + bigram	83.22	54.2
SVM + bigram (Top 3 choices)	85.32	59.61

Figure 6 shows a sample handwritten page in Tamil language. The corresponding output of the recognition engine for the page data is shown in Fig.7 .


கன்னியாகுமரி மாவட்டம், தமிழ்நாட்டின் முப்பத்தொன்று மாவட்டங்களில் ஒன்று ஆகும். இந்தியாவின் தென்கோடியில் அமைந்துள்ள இம்மாவட்டத்தின் தலைநகரம் நாகர்கோவில் ஆகும். இது தமிழகத்தின் மூன்றாவது வளர்ச்சியடைந்த மாவட்டமாகும். நாகர்கோவில், பத்மநாபபுரம், களக்காடு, குழித்துறை ஆகிய நான்கு நகராட்சிகள் உள்ளன.

இயற்கை அழகுக்கு பெயர்போன இம்மாவட்டத்தில் ஒன்பதாம் நூற்றாண்டுக்கும் முந்தைய பல வரலாற்றுச் சின்னங்களும் அமைந்திருப்பதால் சுற்றுலா பயணிகளுக்கு இது ஒரு சுவர்க்கமாக திகழ்கிறது. இம்மாவட்டத்தின் மேற்கு எல்லையாக கேரள மாநிலமும் வடக்கு மற்றும் கிழக்கு எல்லைகளாக தமிழகத்தின் திருநெல்வேலி மாவட்டமும் இருக்கின்றன.

2006 டிசம்பர் 26 அன்று தென் மற்றும் தென்கிழக்கு ஆசிய நாடுகளின் கடற் பகுதிகளை கடுமையாகத் தாக்கிய சுனாமிப் பேரலை இம்மாவட்டத்தையும் பெரும் நாசத்துக்கு உள்ளாக்கியது. கன்னியாகுமரி என்ற பெயர் இப்பகுதியில் புகழ் பெற்ற குமரி அம்மன் என்றும் இந்து சமயக் கடவுளை மையப்படுத்தும் கதை புராணத்திலிருந்து இம்மாவட்டத்திற்கு கிடைத்திருக்கிறது.

Figure 6. A sample handwritten Tamil page.

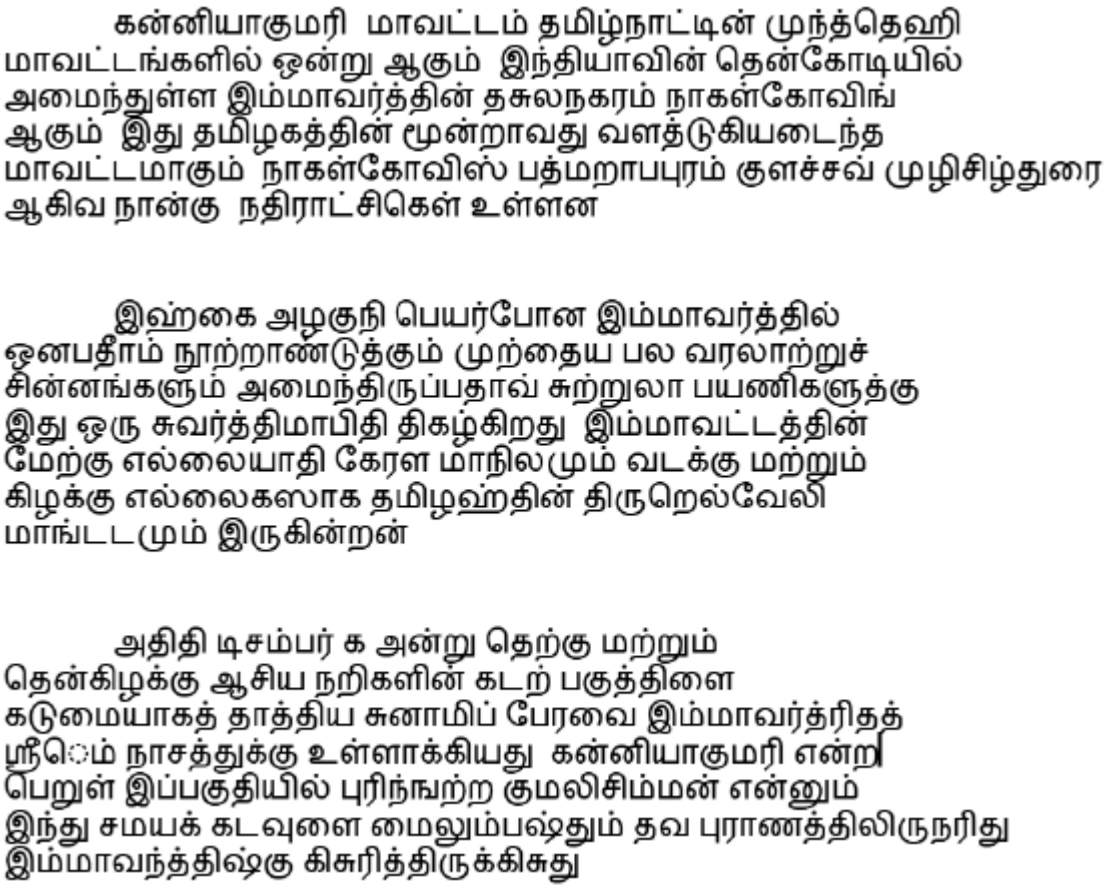


Figure 7. Result of recognition of the handwritten page of text in Fig. 6.

Results for line and word segmentation are 100% in line segmentation and 98.1% in word segmentation accuracies. When the two modules are combined and tested on a sample page data, a symbol level accuracy of 86.6% and a word recognition accuracy of 56.8% are obtained.

6. Status of Engine Integration with applications

Isolated word recognition engine as described above and a separate Indo-Arabic numeral recognition engine have been developed using C as a .dll, which has been successfully integrated with the Census data collection application built by CDAC-Pune [18]. This work got the Prof. M. Anandakrishnan best paper award in the 12-th International Tamil Internet Conference at University of Malaya, Kuala Lumpur, Malaysia in August 2013. The isolated word recognition engine has also been successfully ported to Android platform following a visit of the project staff to CDAC, Pune in January 2014.

7. Future Work

- Integration of line and word segmentation algorithms to CDAC applications such as sentence processing application.
- We would like to improve the segmentation of word into stroke groups by investigating the integration of segmentation, recognition and post-processing methodologies.
- Multiple script recognition of page data considering that most meaningful handwritten pages may contain numerals, special symbols and occasionally Latin script characters.
- Development of robust line and word segmentation algorithms immune to delayed strokes, overwritten and corrected strokes is also a challenge we would like to pursue due to its practical importance.

Acknowledgements

- We thank all the research students and the project staff at MILE lab, Indian Institute of Science for their different contributions over the past seven years.
- We thank Prof. Deiva Sundaram, Dr. Ponnavaikko, Dr. Ila Sundaram and Mrs. Lakshmi Balasubramanyam for facilitating us collect the Tamil handwriting data from hundreds of students.
- We also thank the management and students of AVM Meiyappan Matriculation Higher Secondary School, Chennai; Government Boys Higher Secondary School and Govt Girls Higher Secondary School, Sulur, Coimbatore; University of Madras; Presidency College, Chennai; and S R M University, Chennai for contribtuting to our data collection.
- We would like to specially thank Technology Development for Indian Languages (TDIL), Department of Information Technology, Government of India for funding this work.
- We thank all our research partners in the OHWR consortium project for the various discussions we have had over the time and knowledge inputs we have received.

References

1. Aparna, K. G. and Ramakrishnan, A. G. A complete Tamil optical character recognition system. Proc. Workshop on Document Analysis Systems (DAS), 2002, pp. 53–57.
2. B Nethravathi, CP Archana, K Shashikiran, AG Ramakrishnan, V Kumar, “Creation of a huge annotated database for Tamil and Kannada OHR,” Proc. Intern. Conf. Frontiers in Handwriting Recognition (ICFHR), 2010, pp. 415-420.
3. IWFHR 2006 - Online Tamil handwritten character recognition competition. http://algoval.essex.ac.uk:8080/iwfhr2006/index.jsp.
4. Bharath, A. and Madhvanath, S. HMM-based lexicon-driven and lexicon-free word

recognition for online handwritten Indic scripts. IEEE Trans. Pattern Analysis and Machine Intelligence 34, 4, 670–682, 2012.

5. Swapnil Belhe, Srinivasa Chakravarthy, A. G. Ramakrishnan, “XML standard for Indic online handwritten database,” ACM - Proceedings of the International Workshop on Multilingual OCR, 2009.
6. Suresh Sundaram and A. G. Ramakrishnan, “Attention feedback based robust segmentation of online handwritten words,” Indian Patent Office Reference. No: 03974/CHE/2010.
7. Suresh Sundaram and A. G. Ramakrishnan, “Attention-feedback based robust segmentation of online handwritten isolated Tamil words,” ACM Transactions on Asian Language Information Processing (TALIP), Vol. 12 (1), March 2013, Article No. 4.
8. V. Deepu, S. Madhvanath and A. G. Ramakrishnan, “Principal Component Analysis for online handwritten character recognition,” Proc. 17th Intern. Conf. Pattern Recognition (ICPR 2004), 23-26 Aug. 2004, Vol 2, pp. 327-330.
9. A. G. Ramakrishnan and Bhargava Urala, “Global and local features for recognition of online handwritten numerals and Tamil characters,” ACM - Proc. International Workshop on Multilingual OCR, (MOCR 2013), 24 Aug. 2013, Washington DC, USA.
10. Niranjan Joshi, G.Sita, A.G.Ramakrishnan and S.Madhvanath, “Comparison of elastic matching algorithms for online Tamil handwritten character recognition,” Proc. IWFHR-9, Tokyo, Japan, Oct. 26-29, 2004, pp. 444-449.
11. Niranjan Joshi, G. Sita, A. G. Ramakrishnan and Sriganesh Madhvanath, “Tamil handwriting recognition using subspace and DTW based classifiers,” Proc. 11th International Conference on Neural Information Processing (ICONIP 2004), Calcutta, Nov 22-25, 2004, pp. 806-813.
12. Amrik Sen, G. Ananthakrishnan, Suresh Sundaram, A. G. Ramakrishnan, “Dynamic Space Warping of Strokes for Recognition of Online Handwritten Characters,” IJPRAI 23(5): 925-943, 2009.
13. Shashi Kiran, Kolli Sai Prasada, Rituraj Kunwar, A. G. Ramakrishnan, “Comparison of HMM and SDTW for Tamil Handwritten Character Recognition,” IEEE International Conference On Signal Processing and Communications (SPCOM) 2010.
14. Venkatesh Narasimha Murthy and A. G. Ramakrishnan, “Choice of Classifiers in Hierarchical Recognition of Online Handwritten Kannada and Tamil Aksharas,” Journal of Universal Computer Science, Vol. 17, pp. 94-106, 2011.
15. Suresh Sundaram and A. G. Ramakrishnan, “Bigram language models and reevaluation strategy for improved recognition of online handwritten Tamil words,” ACM Transactions on Asian Language Information Processing (TALIP), revised version under review, 2014.
16. 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.

17. Suresh Sundaram and A. G. Ramakrishnan, "Performance enhancement of online handwritten Tamil symbol recognition with reevaluation techniques," Pattern Analysis and Applications, Dec. 2013.
18. Bhargava Urala and A G Ramakrishnan, "Identification of Tamizh script on Tablet PC," Proc. 12-th International Tamil Internet Conf., Kuala Lumpur, Malaysia, Aug. 15-18, 2013. [received Prof. M. Anandakrishnan best paper award]