Recognition of open vocabulary, online handwritten pages in Tamil script

Bhargava Urala K
Dept. of Electrical Engg.
Indian Institute of Science
Bangalore, India
Email: bhargava@mile.ee.iisc.ernet.in

A G Ramakrishnan
Dept. of Electrical Engg.
Indian Institute of Science
Bangalore, India
Email: ramkiag@ee.iisc.ernet.in

Sahil Mohamed
Dept. of Electrical Communication Engg.
Indian Institute of Science
Bangalore, India
Email: sahil.mohamed987@gmail.com

***Abstract*—In this work, we describe a system, which recognises open vocabulary, isolated, online handwritten Tamil words and extend it to recognize a paragraph of writing. We explain in detail each step involved in the process: segmentation, preprocessing, feature extraction, classification and bigram-based post-processing. On our database of 45,000 handwritten words obtained through tablet PC, we have obtained symbol level accuracy of 78.5% and 85.3% without and with the usage of post-processing using symbol level language models, respectively. Word level accuracies for the same are 40.1% and 59.6%. A line and word level segmentation strategy is proposed, which gives promising results of 100% line segmentation and 98.1% word segmentation accuracies on our initial trials of 40 handwritten paragraphs. The two modules have been combined to obtain a full-fledged page recognition system for online handwritten Tamil data. To the knowledge of the authors, this is the first ever attempt on recognition of open vocabulary, online handwritten paragraphs in any Indian language.**

I. INTRODUCTION

Handwriting recognition is generally broadly divided into two categories - online and offline. The term 'online' refers to the availability of temporal information along with handwritten data. Typically, a pen or stylus is used to write on a touch-sensitive surface and a digitizer captures a sequence of (x, y) coordinates uniformly in time. The beginning and ending of the individual strokes are indicated by 'pen-down' and 'pen-up' signals by the digitizer. In contrast, offline handwritten data refers to images obtained by scanning handwritten material. The emergence and ubiquity of mobile devices like phones and tablets make a compelling case for the development of online handwritten recognition systems, especially for Indic scripts which have a large number of alphabets, and hence are more easily entered through handwriting interfaces.

Most of the work done so far in the domain of Tamil online handwriting recognition has been the investigation of different features and classifiers [30], [4], [7], [11], [12], [24], [25], [14] to recognise isolated characters. In their recent works, Ramakrishnan and Urala [20], [31], propose a combination of local and global features to obtain the best reported recognition accuracy on HP Lab's Tamil isolated character database.

For recognition of isolated words, two major approaches can be seen in the literature. In the work by Bharath and Madhvanath [4] and [5], strokes are considered as recognition primitives and a Tamil word is modelled using a hidden Markov model (HMM). In the latter, lexicon-driven recognition is explored with a tiny lexicon of only 85 words, and a 'bag of strokes' model is used to prune the search space in a lexicon. However, in the work by Sundaram and Ramakrishnan [27], [28], recognition primitives are called symbols or stroke groups and correspond to one of the 155 symbols listed in [20]. Here, an input Tamil handwritten word is segmented into symbols, using a two-stage segmentation method, which are subsequently recognised by a classifier. Post-processing methods of using symbol level bigram language model and lexicon to improve the word recognition rate are explored for a limited dataset in [29], [21]. The same strategies have subsequently been extended to Kannada by Ramakrishnan and Shashidhar [19]. Here, a stroke group may be a base character, vowel modifier or a consonant modifier [17].

Line and word segmentation methodologies for online handwritten data have been explored for Latin script in [10], [13], [15] and for Chinese script in [23]. To the best of the knowledge of the authors, this is the first reported work that deals with line and word level segmentation and recognition from online handwritten pages in Tamil, and possibly, in any Indian language.

In this paper, we introduce several modifications to the segmentation approach described in [27], [29] and patent [26]. We describe a complete recognition module for isolated Tamil words, including post-processing based on symbol level bigram models [29]. Further, we propose a line segmentation method based on centroids of strokes and a two-stage word segmentation method, which computes a dynamic threshold based on the mean positive displacement between adjacent strokes.

II. ISOLATED WORD RECOGNITION

The major steps followed in the recognition module 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

(a) Handwritten character (letter /ka/) (b) After pre-processing

Fig. 1. A sample Tamil letter before and after pre-processing (smoothing, normalization and resampling along the trace)

- Extraction of global (Fourier descriptor) and local features from the pre-processed symbols
- Recognition of the symbols by an RBF-SVM 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.
- Generation of unicode sequence using script grammar

A. Initial Segmentation

The first step in the word recognition process is to segment the input word, which is a set of handwritten strokes, into candidate stroke groups. A stroke is a series of (x, y) coordinates, in temporal order, which are generally preceded and succeeded by 'PEN DOWN' and 'PEN UP' signals respectively. A stroke group consists of one or more strokes and corresponds to one of the 155 symbols that partially or wholly describes a distinct Tamil akshara. These symbols are part of the HP Labs dataset [9] and are listed in [20] as well. We have observed that, if multiple strokes form a single stroke group, Tamil writers tend to write them with a high degree of horizontal overlap. On the other hand, writers tend to spatially separate one stroke group from another. We exploit this tendency by measuring extent of horizontal overlap (O_k^c) between a stroke group and the successive stroke using Equation 1. Subsequently a merge or split decision is taken, if the value is above or below an empirically determined threshold (0.2 in our case).

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

where,

- s_c and S_k indicate the current stroke and stroke group, respectively.
- x_M and x_m indicate 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 Equations 2 and 3. If $d_x \geq 0$ or if ($d_x < 0$ and $d_y \leq 0$), then the decision is not to combine the stroke group with the next stroke.

$$d_x = x_1^{s_c} - x_M^{S_k}; \tag{2}$$

$$d_y = y_1^{s_c} - y_{last}^{S_k}; \tag{3}$$

where, x_1, y_1 and y_{last} indicate x coordinate of first point, y coordinates of first and last points respectively.

B. Smoothing and normalisation

The main purpose of pre-processing is to nullify the effect of noise, account for writing size and style variations and remove duplicate points in the stroke [6], [3], [29]. It is carried out in three steps - smoothing, normalisation and resampling. Each stroke group is pre-processed before feature extraction and recognition by the classifier. Figure 1 shows the effect of pre-processing on a sample, handwritten Tamil character.

- Smoothing - Every stroke of the stroke group is independently smoothed using a Gaussian filter. The length of the filter is chosen to be approximately one-tenth that of the stroke.
- Normalisation - Range normalisation of the stroke group is carried out by linear mapping of its x and y coordinates to the range $[0, 1]$.
- Resampling - The stroke data obtained from the digitizer is uniformly sampled in time. This results in different number of points for the same stroke group and arc length, when written at different speeds. To account for this variation in writing speed, we uniformly sample the stroke group, in space, 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.

C. Extraction of Fourier descriptors and resampled coordinates

A combination of global and local features are extracted from the pre-processed stroke groups before classification. Discrete Fourier transform (Equation 4) is used to compute global features representing a stroke group. Each point (x_n, y_n) is expressed as a complex number $f_n = x_n + jy_n$ and the vector F is truncated to a 32-point complex vector. Local features are simply the x and y coordinates of the 64-point, pre-processed stroke group. The global and local features are concatenated into a $(64 + 32) * 2 = 192$-length feature vector. This particular method of feature extraction is chosen out of the many different kinds of combinations of global and local features investigated in [20].

$$F(k) = \sum_{n=0}^{N-1} f_n e^{-j2\pi nk/N} \tag{4}$$

D. RBF-SVM classifier

The classifier is a support vector machine (SVM) with radial basis function (RBF) kernel and is trained on the feature vectors extracted from the training dataset of 50, 385 symbols across all the 155 symbols of the HP Labs isolated Tamil character dataset [9]. The RBF-SVM is implemented using the LIBSVM open source package [1]. For every test feature vector, the classifier returns the most likely class label (1 to 155) and the confidence values [1] for each class label. The parameters for the RBF-SVM are chosen via grid-search with 5-fold cross validation. Table I shows the accuracies of the classifier, for different combination of global and local features, when tested on the HP labs test dataset as well as the cross validation accuracies.

TABLE I
CROSS VALIDATION AND TEST ACCURACIES FOR DIFFERENT FEATURES ON THE COMPLETE TEST SET OF HP LABS ONLINE TAMIL HANDWRITTEN CHARACTER DATABASE

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
Truncated DFT	64	91.5	95.69
(x,y)+ truncated DFT	192	91.3	95.85

E. 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 and we continue along the word. An additional parameter called the maximum displacement (d_M) is also used in the merge-split decision and is computed from the pre-processed stroke group using Equation 5. The value of d_M^{symbol} is computed using the training set and stored for each of the 155 symbols. The two stroke groups are merged only if the computed d_M for the merged stroke group is less than the d_M^{symbol} for that symbol from training set.

$$d_M = \max_{i=1}^{N} (x_1^{s_i} - x_{last}^{s_{i-1}}) \tag{5}$$

where, s_i is the i^{th} stroke and N is the number of strokes in the stroke group.

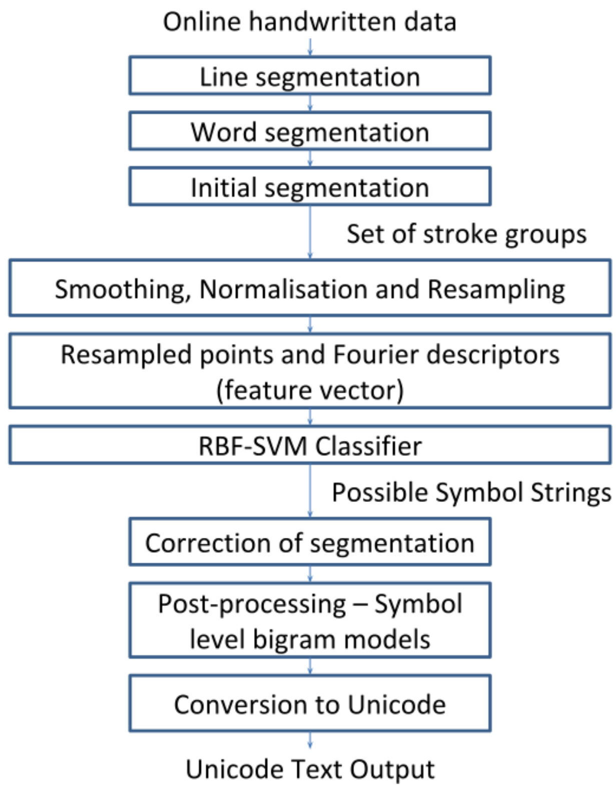

Fig. 2. Overall flow of the page recognition system

F. Symbol level bigram models

A large corpus of Tamil text data - derived from combining EMILLE corpus with the text from Project Madurai [2] is converted from standard UNICODE representation to class label (1 to 155) representation. Bigram probabilities are obtained for every possible pair of symbols using the method detailed in [29]. Probabilities for each symbol to be the starting or ending symbol of a word are also computed from this corpus, because, in Tamil script, some symbols can appear only in starting positions in a word. Further, some Tamil symbols are not allowed to occupy the last position in a word.

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 confidence values 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. In our case, a value of $N = 3$ is found to be satisfactory.

Tamil is a morphologically very rich language and is also agglutinative and hence the size of the lexicon, i.e. number of unique words, grows with the size of text corpus analyzed [18]. Therefore, we have chosen post-processing using n-gram $(n = 2)$ models over usage of lexicon.

Figure 2 shows the block diagram of the process by which online handwritten page data is converted to unicode text.

III. LINE AND WORD LEVEL 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.

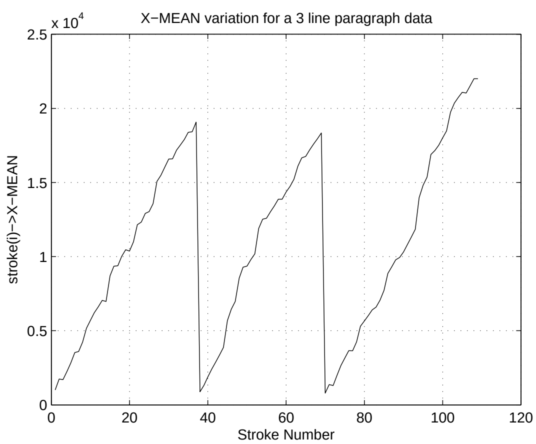

Fig. 3. Sawtooth-like variation of x-coordinates of centroids of successive strokes in a handwritten paragraph data.

A. Segmentation of text into lines

As seen in Figure 3, x_c of successive strokes varies in a sawtooth pattern for page data containing multiple lines. To detect each line, we compare x_c values of consecutive strokes and check if they are in increasing order. A decrease in x_c value of the current stroke from previous value may indicate a line break or temporary left movement of the pen to write certain vowel modifiers. To eliminate vowel modifiers from being detected as start of new lines, 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.

B. Segmentation of line into words

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,

$$T_w = (1.25/N) \sum_{i=1}^{N} w^i \tag{6}$$

Now, any stroke (say, k^{th}) in the line is marked as the first stroke of a new word, if $b_x^k \geq T_w$. 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.

For these errors, all 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. Figures 4 and 5 show a sample handwritten page and the recognized text, respectively.

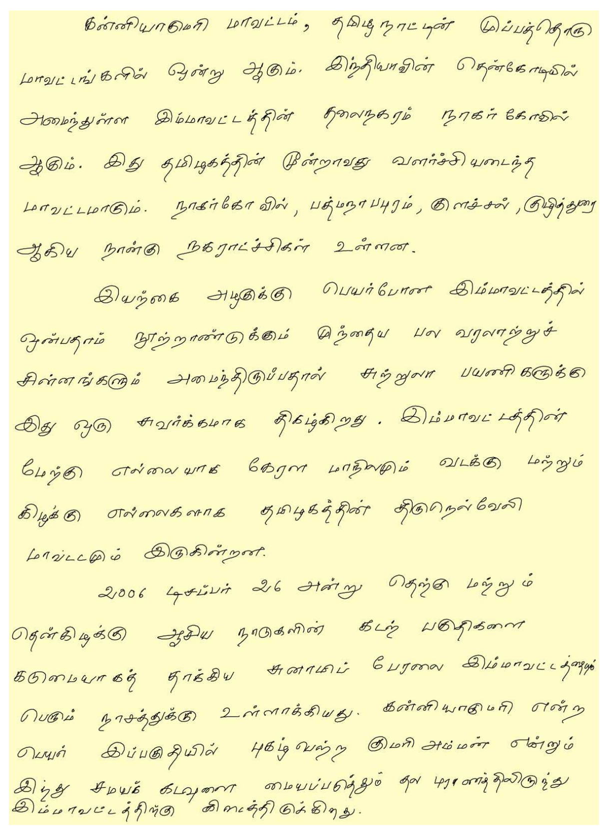

Fig. 4. Image of a sample page of handwritten Tamil data

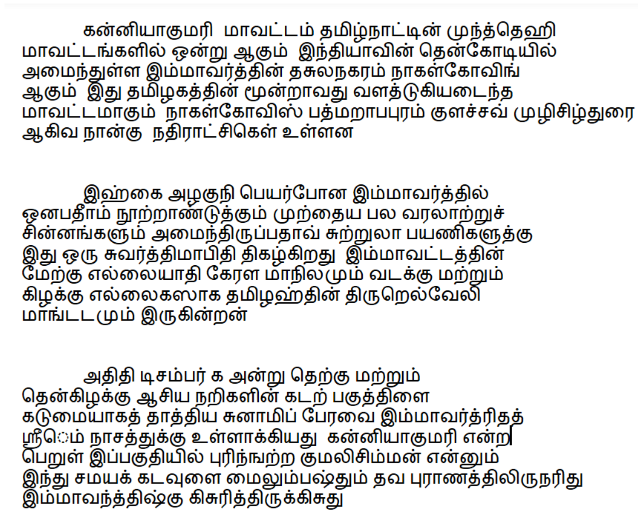

Fig. 5. Recognition results of handwritten Tamil data shown in Figure 4

IV. Results and discussion

The isolated word recognition module was tested on a database of 45,405 words, consisting of a total of 2,53,095 symbols. These words were collected on a TabletPC with WACOM digitizer and were written by 181 different regular Tamil writers [16]. 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. The results of running the word recognition module with and without language models can be seen in Table II. The number of errors at the symbol level is the Levenshtein distance [8] between the obtained class label sequence and the ground truth sequence. Word recognition accuracy is computed by matching entire word as a singular entity.

We can see that the usage of bigram models resulted in an increase of symbol level accuracy from 78.5% to 83.2%. If we consider the top 3 choices obtained after using bigram models, we get a further increase of 2%. Use of bigram models increased the word recognition accuracy from 40.0% by 14.2% and 19.6% for first choice and first 3 choices, respectively.

TABLE II
Symbol and word level accuracies for isolated word recognition on Tablet PC data

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

The word recognition module was also run on a separate set of 1897 words consisting of 6627 symbols collected from GNote data capturing device. Table III shows the obtained symbol and word level accuracies for this data.

TABLE III
Symbol and word level accuracies for isolated word recognition on GNote data

SA: Symbol level accuracy, WA: Word level accuracy
Total No. of test words: 1,897
Total No. of symbols in the test data: 6,627

Recognizer	SA (%)	WA (%)
SVM + bigram	89.2	74.5
SVM + bigram (Top 3 choices)	92.5	83.3

The line and word recognition algorithms have been tested on six handwritten A4 size pages written by different regular Tamil writers. A Hitech digitizer was used to capture the online data. The text totally contained 127 lines and 732 words. We have obtained an accuracy of 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 word recognition accuracy of 56.8% and a symbol level accuracy of 86.6% are obtained. Errors are contributed by wrong segmentation, confusion between closely resembling symbol pairs which result in unreliable classifier performance and lack of framework to simultaneously handle Indo-Arabic numerals [20] and Tamil symbols.

V. Conclusions

For the casual observer, the approach may appear heuristic and one may have question its scalability (for different writers) and robustness (for different devices). In this context, we would like to mention that the training data [9] and the test data (word data collected from over 100 school and college students on Tablet PC) are from totally distinct writers. Further, the same word recognition engine has given consistent performance on word test data collected from a different set of writers using G-note device which has a different spatial resolution as well as different sampling rate. We have also obtained consistent recognition results on page level handwritten data written by other writers on HiTech device which has lower sampling rate and spatial resolution. Thus, the processing, features and the classifiers have proved to be rather robust for different writers, devices and resolutions.

For future work, we would like to address the problem of multiple script recognition of page data considering that most meaningful handwritten pages may contain numerals, special symbols and occasionally Latin script characters. We would like to improve the segmentation of word into stroke groups by investigating the integration of segmentation, recognition and post-processing methodologies. It is also planned to explore a combination of the classifier with another operating on the image data obtained by converting stroke information to an offline image as explored for Kannada script in [22]. 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.

Acknowledgment

We thank all the research students and the project staff at MILE lab, IISc. We would like to specially thank Technology Development for Indian Languages (TDIL), Department of Information Technology, Government of India for funding this work and HiTech Solutions Pvt. Ltd. for providing the page level test data.

References

[1] *LIBSVM - A Library for Support Vector Machines*. http://www.csie.ntu.edu.tw/cjlin/libsvm/.

[2] *Project Madurai - Database of Tamil Unicode text*. http://www.projectmadurai.org/index.html.

[3] A. Arora and A. M. Namboodiri. A hybrid model for recognition of online handwriting in Indian scripts. In *Proc. ICFHR*, pages 433–438, 2010.

[4] A. Bharath and S. Madhvanath. Hidden Markov Models for online handwritten Tamil word recognition. In *Proc. IWFHR*, pages 506–510, 2007.

[5] A. Bharath and S. Madhvanath. HMM-based lexicon-driven and lexicon-free word recognition for online handwritten Indic scripts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):670–682, 2012.

[6] 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.

[7] M. Dinesh and M. K. Sridhar. A feature based on encoding of relative position of a point in the character for online handwritten character recognition. In *Proc. ICDAR*, pages 1014–1017, 2007.

[8] Duda, Hart, and Stork. *Pattern Classification*. Springer Wiley, 1995.

[9] HP Labs India. *Isolated Tamil Handwritten Character Dataset*. http://lipitk.sourceforge.net/hpl-datasets.htm.

[10] A. K. Jain and A. M. Namboodiri. Indexing and retrieval of on-line handwritten documents. In *ICDAR*, page 655, 2003.

[11] N. Joshi, G. Sita, A. G. Ramakrishnan, and S. Madhvanath. Comparison of elastic matching algorithms for online Tamil handwritten character recognition. In *Frontiers in Handwriting Recognition, 2004. IWFHR-9 2004. Ninth International Workshop on*, pages 444–449. IEEE, 2004.

[12] 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.

[13] M. Liwicki, M. Scherz, and H. Bunke. Word extraction from on-line handwritten text lines. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 2, pages 929–933. IEEE, 2006.

[14] V. N. Murthy and A. G. Ramakrishnan. Choice of classifiers in hierarchical recognition of online handwritten Kannada and Tamil aksharas. *J. UCS*, 17(1):94–106, 2011.

[15] A. M. Namboodiri and A. K. Jain. Robust segmentation of unconstrained online handwritten documents. In *ICVGIP*, pages 165–170, 2004.

[16] B. Nethravathi, C. Archana, K. Shashikiran, A. G. Ramakrishnan, and V. Kumar. Creation of a huge annotated database for Tamil and Kannada OHR. In *Frontiers in Handwriting Recognition (ICFHR), 2010 International Conference on*, pages 415–420. IEEE, 2010.

[17] M. M. Prasad, M. Sukumar, and A. G. Ramakrishnan. Divide and conquer technique in online handwritten Kannada character recognition. In *Proc. of the International Workshop on Multilingual OCR*, pages 1–7, 2009.

[18] A. G. Ramakrishnan and Arulmozhi. Language models in recognition applications: a new approach. In *Proc. of the Tamil Internet Conf., Kuala Lumpur, Malaysia*, 2013.

[19] A. G. Ramakrishnan and J. Shashidhar. Development of online handwriting recognition system for Kannada. *VishwaBharat@TDIL*, 2013.

[20] 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.

[21] A. G. Ramakrishnan, B. Urala, S. Sundaram, and P. V. Harshitha. Development of online handwriting recognition system for Tamil. *VishwaBharat@TDIL*, 2013.

[22] R. Rampalli and A. G. Ramakrishnan. Fusion of complementary online and offline strategies for recognition of handwritten Kannada characters. *Journal of Universal Computer Science*, 17(1):81–93, 2011.

[23] G. Rui and J. LianWen. Segmentation of on-line cursive handwritten Chinese word based on stroke speed feature and stroke vector feature. In *Automation and Logistics, 2007 IEEE International Conference on*, pages 1576–1579. IEEE, 2007.

[24] A. Sen, G. Ananthakrishnan, S. Sundaram, and A. G. Ramakrishnan. Dynamic space warping of strokes for recognition of online handwritten characters. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(05):925–943, 2009.

[25] K. Shashikiran, K. S. Prasad, R. Kunwar, and A. G. Ramakrishnan. Comparison of HMM and SDTW for Tamil handwritten character recognition. In *Signal Processing and Communications (SPCOM), 2010 International Conference on*, pages 1–4. IEEE, 2010.

[26] S. Sundaram and A. G. Ramakrishnan. Attention feedback based robust segmentation of online handwritten words. *Indian Patent Office Reference No.*, (03974/CHE/2010).

[27] 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.

[28] S. Sundaram and A. G. Ramakrishnan. Performance enhancement of online handwritten Tamil symbol recognition with reevaluation techniques. *Pattern Analysis and Applications*, pages 1–23, 2013.

[29] 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.

[30] C. S. Sundaresan and S. S. Keerthi. A study of representations for pen based handwriting recognition of Tamil characters. In *Proc. ICDAR*, pages 422–425, 1999.

[31] B. Urala and A. G. Ramakrishnan. Identification of Tamizh script on Tablet PC. In *Proc. of the Tamil Internet Conf., Kuala Lumpur, Malaysia*, 2013.