Attention-feedback based robust segmentation of online handwritten isolated Tamil words

SURESH SUNDARAM , A G RAMAKRISHNAN, Indian Institute of Science

In this paper, we propose a lexicon-free, script-dependent approach to segment online handwritten isolated Tamil words into its constituent symbols. Our proposed segmentation strategy comprises two modules, namely (1) Dominant overlap criterion segmentation (DOCS) module and (2) Attention feedback Segmentation (AFS) module. Based on a bounding box overlap criterion in the DOCS module, the input word is first segmented into stroke groups. A stroke group may at times correspond to a part of a valid symbol (over-segmentation) or a merger of valid symbols (under-segmentation). Attention on specific features in the AFS module serve in detecting possibly over-segmented or under-segmented stroke groups. Thereafter, feedbacks from the SVM classifier likelihoods and stroke-group based features are considered in modifying the suspected stroke groups to form valid symbols.

The proposed scheme is tested on a set of 10000 isolated handwritten words (containing 53246 Tamil symbols). The results show that the DOCS module achieves a segmentation accuracy of 98.1%, which improves to as high as 99.7% after the AFS strategy. This in turn entails a symbol recognition rate of 83.9% (at the DOCS module) and 88.4% (after the AFS module). The resulting word recognition rates at the DOCS and AFS modules are found to be, 50.9% and 64.9% respectively, without any post processing.

Categories and Subject Descriptors: I.5.4.f [**Handwriting Analysis**]: Pattern Recognition; I.7.5.d [**Optical Character Recognition**]: Document and Text processing; I.5.2.c [**Pattern Analysis**]: Pattern recognition

General Terms: Experimentation, Tamil

Additional Key Words and Phrases: Handwriting recognition, Online Tamil words, stroke group, Dominant overlap criterion segmentation (DOCS) module, Attention feedback segmentation (AFS) module, Support vector machines (SVM).

1. INTRODUCTION

Processing of handwritten documents, in general, considers words as basic units rather than isolated characters. In English texts, there is a well defined separation between words, but the letters within a word are not separated. This is especially evident in the case of cursive handwriting, the recognition of which has been addressed in [Senior and Robinson 1998; Jager et al. 2001; Oliveira et al. 2002; Koerich et al. 2005; Camastra 2007]. In Indic scripts, the constituting words are rarely cursive in nature with the possible exception of Bangla [Bhattacharya et al. 2008]. Characters in a word are written separately from each other with possible overlaps. There is currently limited work addressing the research challenges pertaining to recognizing Indic script words [Bharath and Madhvanath 2007; Bhattacharya et al. 2008; Fink et al. 2010] . Most of the reported techniques deal with the problem of recognizing isolated characters [Joshi et al. 2004; Bhattacharya et al. 2007; Babu et al. 2007; Swethalakshmi et al. 2007].

Word recognition can be categorized into segmentation-free and segmentation-based methods. Segmentation-free approaches [Madhvanath and Govindaraju 2001] treat the word as a single entity and attempt to recognize it as a whole, after appropriate feature extraction. The recognition output necessarily corresponds to a word contained in a lexicon. On the other hand, segmentation-based techniques [Liu et al. 2004b; Nagakawa et al. 2005; Gao et al. 2005] regard a word as a collection of subunits. These techniques have the advantage of not being limited by a lexicon seen in training. These methods segment the word into its constituent units, recognize them and then build a word level interpretation. In general, a suitable set of candidate patterns are generated and concatenated to constitute the word. A classifier trained on the subunits is used to classify each of these patterns. The candidates generated can be represented by a hypothesized network, called the segmentation candidate lattice [Murase 1988; Liu et al. 2004b; Zhou et al. 2007; Cheriet et al. 2008] and the optimal candidate sequence representing the word is traced using dynamic programming techniques [Gao et al. 2005].

Two stage segmentation schemes have been used to segment Chinese characters in [Zhao et al. 2003; Gao et al. 2005]. Apart from recognizing candidate patterns with a classifier, contextual information forms cues in deciding the optimal character sequence in segmentation-based techniques. Geometric features extracted from segments have been used for Japanese online handwriting recognition in [Nagakawa et al. 2005]. The linguistic knowledge obtained from a large corpus of data has been incorporated during recognition in [Marti and Bunke 2002]. Features derived from off-strokes have also been used to segment Japanese characters [Furukawa et al. 2006]. Hypothetical segmentation points are generated in [Zhu and Nakagawa 2006; Zhu et al. 2010] using geometric features (trained with SVM classifier), which are then incorporated into the integrated-segmentation recognition (ISR) framework. Very recently, conditional random fields have been employed for path evaluation in the candidate lattice for word recognition in [Zhou et al. 2009]. Modified path evaluation criteria are proposed in [Zhu et al. 2009] and [Tonouchi 2010] for Japanese text recognition. The extraction of words from an online handwritten text in [Varga and Bunke 2005; Liwicki et al. 2006; Quiniou et al. 2009] is another avenue of research that has been explored in the literature.

The challenges posed with segmenting online handwritten Indic scripts have hardly been investigated [Sundaram and Ramakrishnan 2010]. Among the reported techniques, we came across only one work for the recognition of online Tamil words [Bharath and Madhvanath 2007]. Here, the authors use a HMM framework for modeling the symbols and their relative positions in online Tamil words. However, their work adopts a segmentation-free approach. Apart from a preliminary attempt in Bangla [Bhattacharya et al. 2008], we have not come across any work on segmentation-based methods for recognizing words in online Indic scripts. In [Bhattacharya et al. 2008], based on the positional information of the headline, the online trace is segmented to a set of sub-strokes, which are in turn recognized and concatenated using a look up table into valid characters. However, for off-line handwritten Indic words, segmentation using the reservoir concept has been reported [Tripathy and Pal 2004]. Recursive contour following algorithm and fuzzy-based features have been proposed in [Bishnu and Chaudhuri 1999] and [Basu et al. 2007] respectively for segmenting off-line Bangla text.

In this work, we propose a lexicon free scheme, based on feedbacks from recognition likelihoods and inter-stroke based feature, to segment online handwritten isolated Tamil words. Our strategy comprises two modules (1) Dominant overlap criterion segmentation (DOCS) and (2) Attention feedback segmentation (AFS). The details of the functionality of the modules, as well as their performance evaluation on handwritten Tamil words are presented in this paper.

The remainder of the paper is organized as follows. Section 2 highlights the databases adopted in this work, together with the experimental condition. The first half of Section 3 proposes an initial segmentation step of a Tamil word in the DOCS module into a set of ‘stroke groups’. This is followed by a high level outline of the AFS module, proposed to detect and correct the stroke groups wrongly segmented by the DOCS module. Section 4 distinguishes our methodology from the integrated segmentation recognition technique followed in Oriental scripts. The details of the script specific features, used for selecting the stroke groups, possibly over-segmented or under-segmented by the DOCS module, are described in Sections 5 and 6 respectively. Sections 7 and 8 discuss the AFS strategies for correcting the suspected over-segmented and under-segmented stroke groups. The performance evaluation of the proposed segmentation scheme is presented in Section 9. Section 10 presents an analogy of our approach to concepts in neuroscience. Finally, we summarize the contributions of the present work in Section 11.

2. DATASET DESCRIPTION

In this section, we outline the databases employed for our study. Modern day Tamil alphabet comprises 313 characters. A minimal set of 155 distinct symbols have been derived to recognize these characters [Nethravathi et al. 2010]. The 155 distinct Tamil symbols (comprising 11 vowels, 23 base consonants, 23 pure consonants, 92 CV combinations and 6 additional symbols) are presented in Appendix A. A corpus of isolated Tamil symbols (IWFHR database) is publicly available for research [Madhvanath and Lucas 2006] and is used for learning various statistics about Tamil symbols. In addition, as will be discussed in Sec 9, the choice of using the SVM classifier over other state-of-art classifiers in the present study has been made by measuring its performance on the samples of the test set in the IWFHR database.

Isolated Tamil words are collected using a custom application running on a tablet PC. We have ensured that all the writers who participated in the data collection activity are native Tamil speakers, who currently write in that language, at least irregularly. Accordingly, we came across different popular writing styles for Tamil symbols. Moreover, the participants were provided with a graphics interface with rectangular boxes and were prompted to write Tamil words, one in each box. With this set-up, the experimental data is found to be free from skew or rotation.

One hundred and twenty five high school students from across 6 educational institutions in the Indian state of Tamil Nadu contributed in building the word data-base of 10000 words, referred to as the 'MILE word Database' in this work. A subset of 250 words (denoted as DB1) has been employed for validating the segmentation strategy.

The collected word samples were annotated semi-automatically with a GUI tool. Trained Tamil natives contributed to the annotation of each word sample at the symbol level by labeling one or more strokes that together form a particular symbol. The annotation of the 10000 words in the MILE Word database resulted in 53246 labeled Tamil symbols. We use these symbols as the ground truth for computing the segmentation and recognition accuracies in this work.

The statistics of the features to be used in this work have been derived from the preprocessed symbols [Joshi et al. 2004] of the IWFHR Tamil symbol database. After carrying out similar preprocessing steps on the Tamil symbols in the MILE word database, the derived statistics can be reliably applied to them.

Figures 1(i)-(j) present a few sample Tamil words from our database.

3. PROPOSED METHODOLOGY

Given an online Tamil word, our emphasis in this work is to correctly segment it into its constituent symbols by employing a feedback-based strategy. The structural characteristics of the Tamil script have been exploited in designing the segmentation methodology.

During the collection of online words, the pen-tip movement is detected with pen-up /pen-down states. The set of points captured between successive pen-down to pen-up states is called a stroke. The script being non-cursive in nature, an online word can be represented as a sequence of n strokes $W = \{s_1, s_2....., s_n\}$. It may be noted here that a Tamil symbol alone, at times, may correspond to a word. Typically, the strokes of a Tamil symbol vary from 1 to 5. In the case of multi-stroke Tamil symbols, strokes of the same symbol may significantly overlap in the horizontal direction. This prior knowledge of the script is utilized in proposing an initial segmentation of W as described below.

The word W is segmented based on a bounding box overlap criterion, in the 'Dominant overlap criterion segmentation' (DOCS) module to a set of distinct patterns. We refer to the generated patterns as 'stroke groups' in this work. A stroke group is defined as a set of consecutive strokes, which is possibly a valid Tamil symbol. In order to mathematically formulate the operation in the DOCS module, one needs to quantify the degree of horizontal

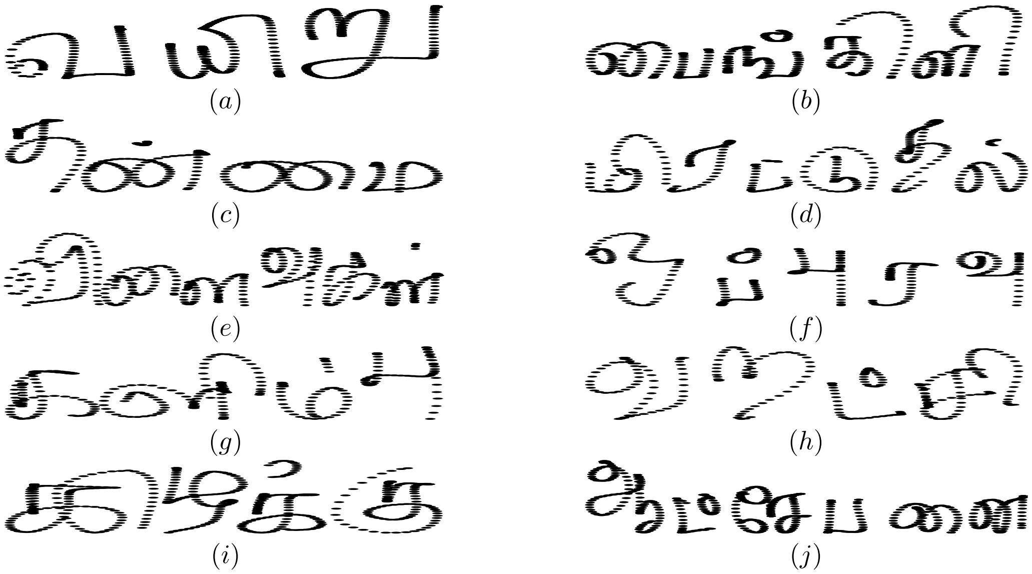


Fig. 1. Sample words from the MILE word database.

overlap. For the k^{th} stroke group S_k under consideration, its successive stroke is taken and checked for overlap, if any. Whenever the degree of overlap exceeds a threshold, the successive stroke is merged with the stroke group S_k. Otherwise, the successive stroke is considered to begin a new stroke group S_{k+1}. The algorithm proceeds till all the strokes of the word are exhausted. The first stroke, s_1 of W, by default, belongs to the first stroke group S_1.

Let the minimum and maximum x-coordinates of the bounding box (BB) of the i^{th} stroke s_i be denoted by (x^i_{min}, x^i_{max}). Given the current stroke s_c, we define the degree of its horizontal overlap O^c_k with the previous stroke group S_k as

$$O^c_k = \max\left(\frac{x^{S_k}_{max} - x^c_{min}}{x^{S_k}_{max} - x^{S_k}_{min}}, \frac{x^{S_k}_{max} - x^c_{min}}{x^c_{max} - x^c_{min}}\right) \tag{1}$$

Here $x^{S_k}_{min}$ and $x^{S_k}_{max}$ denote the minimum and maximum x-coordinates of the BB of the k^{th} stroke group. A threshold T_0 (set to 0.2) applied on O^c_k is used for merging strokes (The choice of the threshold T_0 is discussed in Sec.9.4). Figure 2 depicts the different parameters employed for computing O^c_k. The DOCS module outputs a set of $\tilde{p}$ stroke groups, where $\tilde{p} <= n$.

Figures 3, 4 and 5 present illustrations, wherein the DOCS module combines one or more input raw strokes to generate stroke groups. The resulting stroke groups are valid Tamil symbols மு /mu/, ஊ /U/ and ஈ /I/ respectively.

However, at times, a stroke group generated may correspond to a part of a valid symbol or a merge of valid symbols. These two issues are addressed below with suitable illustrations.

— Splitting of a valid symbol (over-segmentation): The symbol ஈ /I/ in the word ஈசன் (Fig.6(a)) is segmented into 2 stroke groups, as shown by the separate BBs. The DOCS module outputs 4 stroke groups (ஈ, dot ., ச and ன்) for the word instead of 3 (ஈ, ச and ன்).

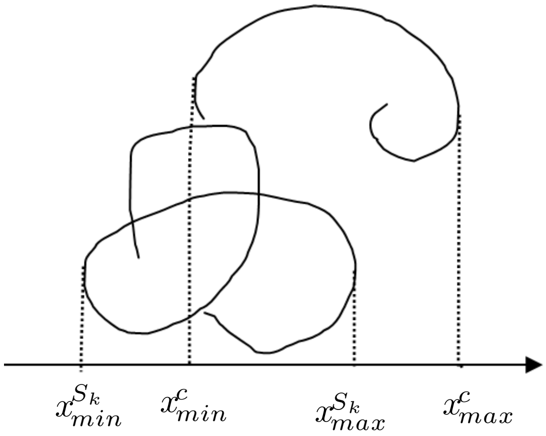


Fig. 2. Parameters employed for computing the overlap O^c_k in the DOCS scheme.

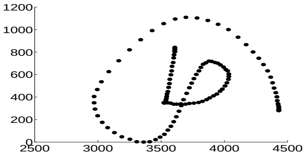
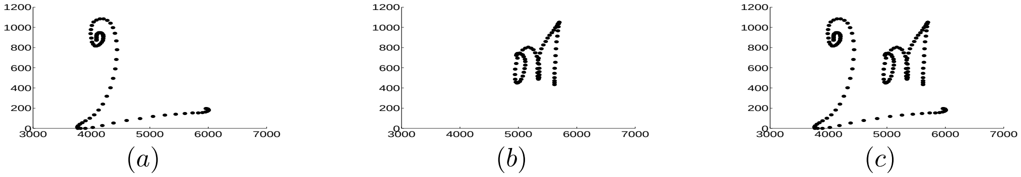


Fig. 3. Generation of a stroke group from the single-stroke Tamil symbol /mu/.



(a) (b) (c)

Fig. 4. Generation of a stroke group for a two-stroke Tamil symbol. (a) and (b): The 2 individual strokes. (c) Stroke group generated by DOCS module. Since the second stroke in (b) completely overlaps with the first stroke in (a) in the horizontal direction, they are merged into a single stroke group (shown in c) by the DOCS module. The resulting stroke group /U/ is a valid symbol.

— Merging of two distinct symbols (under-segmentation): In Fig.6(b), the symbols த் /t/ and தி /ti/ of the word சமுத்திரம் merge to a single stroke group த்தி, as highlighted by a single BB. In this case, DOCS module outputs 5 stroke groups (ச, மு, த்தி, ர and ம்) instead of 6 (ச, மு, த் , தி, ர and ம்).

We propose a strategy to further improve the segmentation performance beyond that given by the DOCS module. Two novel attributes namely, maximum inter-stroke displacement and number of dominant points, have been separately derived to detect under-segmented and over-segmented stroke groups respectively. 'Attention' on these features selects only a subset of the generated stroke groups for subsequent analysis. Upon detection, a stroke group suspected to be incorrectly segmented is fed to a unit, that operates on additional attributes (derived from the statistics of the IWFHR database), to provide 'feedback' on whether or not to proceed in correcting it. Whenever the feedback favors a correction, rearrangement of

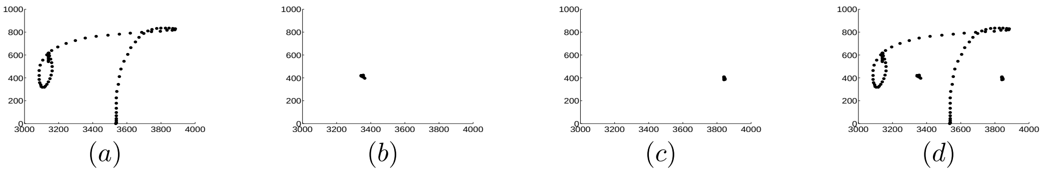


Fig. 5. Generation of a stroke group for a three-stroke Tamil symbol. (a),(b) and (c): The three individual strokes. (d) Generated stroke group. Since the second and third strokes (presented in (b) and (c)) completely overlap in the horizontal direction with the first stroke in (a), the DOCS module combines the 3 strokes to generate a single stroke group (shown in (d)). The resulting stroke group /I/ is a valid symbol.

Fig. 6. Illustration of over-segmented and under-segmented words after the DOCS step. (a) The symbol /I/ gets fragmented (over-segmented) to 2 stroke groups as shown by the separate bounding boxes. (b) The /t/ and /ti/ symbols get merged (under-segmented) to one stroke in this word.

the strokes within or even outside the stroke group under consideration is initiated. It is to be noted that only stroke groups suspected to be broken or under-segmented are fed to the feedback unit. In other words, we focus on the rectification of possible segmentation errors on selected stroke groups. First, we operate on stroke groups likely to contribute to under-segmentation errors, and split them, if necessary. Thereafter, stroke groups suspected to be a part of valid symbol (contributing to over-segmentation errors) are merged with their appropriate neighbors to generate valid symbols. We refer to this proposed segmentation step/ module by ‘attention-feedback segmentation’ (abbreviated as AFS).
Summarizing, the stroke groups resulting from the DOCS module are regarded as tentative candidates for valid Tamil symbols. Based on feedback from various attributes proposed in this work, the AFS module may modify the number of stroke groups output by the DOCS module. In doing so, the AFS module improves the performance of the handwriting system.

For the illustrations ஈசன் and சமுத்திரம் in Fig.6, the refined segmentation (performed by the AFS module), when successful, should output 3 and 6 stroke groups respectively.

Figure 7 (a) presents a high level pictorial summary of our segmentation strategy, for a given input word. We assume that the word W comprises p stroke groups after the AFS step. Each of the p stroke groups obtained after the AFS step are recognized by a classifier (shown with a dotted box) to obtain the symbol labels, that in turn are combined to generate the output word. The details of the AFS module operating on a stroke group generated from the DOCS module is depicted in Fig 7 (b).

4. COMPARISON OF THE PROPOSED SEGMENTATION METHODOLOGY WITH THE INTEGRATED SEGMENTATION RECOGNITION (ISR) SCHEME

In this section, we highlight two differences between the proposed segmentation strategy and the integrated-segmentation and recognition (ISR) approach typically followed in recent literature for online Oriental (namely Chinese and Japanese) scripts. The ISR scheme (refer Fig 8) comprises two steps, namely (1) Pre-segmentation (2) Path evaluation [Liu et al. 2004a; Zhu et al. 2010].

— The stroke groups generated by the DOCS module may be regarded to be analogous to the primitive segments obtained with the pre-segmentation strategy of the ISR approach. For Chinese and Japanese scripts, strokes of different characters overlap less frequently, due to which, under-segmentation errors hardly arise [Cheriet et al. 2008]. The primitive

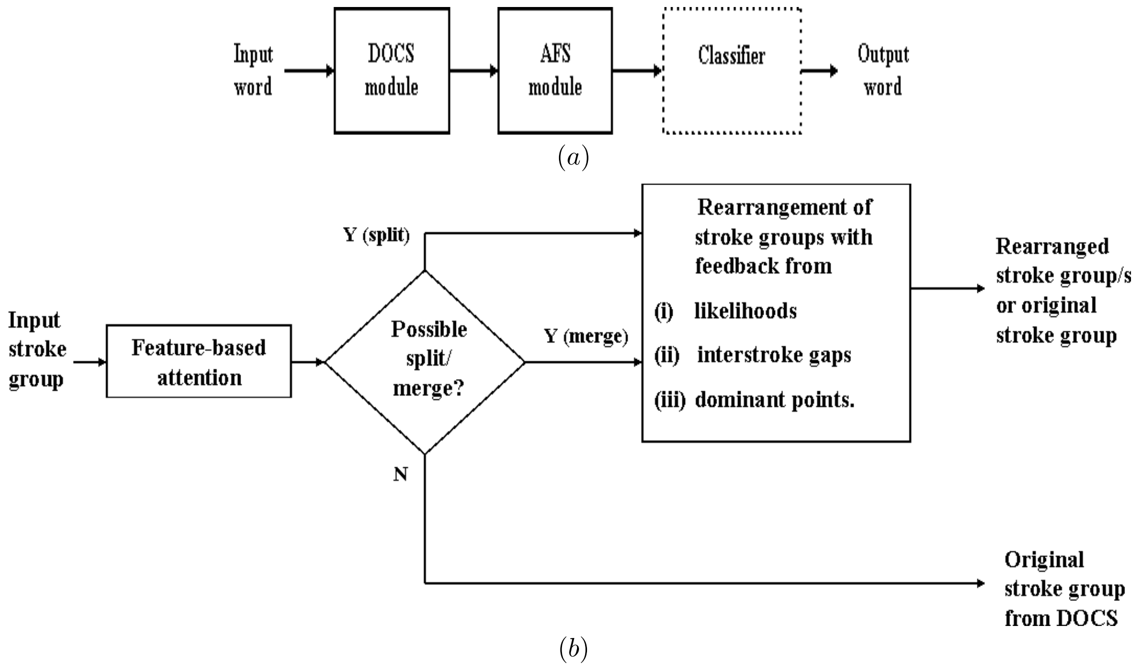


Fig. 7. (a) Pictorial overview of the proposed segmentation methodology. Our segmentation strategy comprises the modules: DOCS and AFS. The stroke groups resulting from the AFS module are fed to a classifier to generate symbol recognition labels, that are concatenated to generate the output word. (b) Pictorial overview of the attention-feedback segmentation approach for a stroke group generated in the DOCS module.

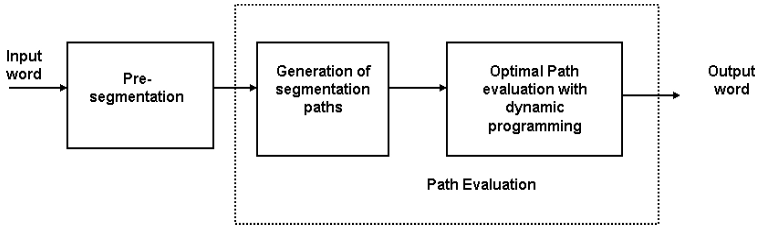

Input word
Pre-segmentation
Generation of segmentation paths
Optimal Path evaluation with dynamic programming
Path Evaluation
Output word

Fig. 8. Pictorial overview of integrated segmentation recognition approach.

segments generated correspond to either a single character or a part of a character. Accordingly, the only type of segmentation error addressed is that of over-segmentation. However, for Tamil, apart from over-segmentation errors (as depicted in Fig 6 (a)), we are likely to encounter a high degree of overlapping of strokes of different symbols in the DOCS step. An example of this is illustrated in Fig 6 (b). Here, the symbols த் /t/ and தி /ti/ of the word சமுத்திரம் merge to a single stroke group த்தி. Thus, there arises a need to rectify such under-segmentation errors, by appropriately splitting stroke groups to valid symbols.

— In the path-evaluation step of the ISR approach, the optimal path is evaluated across all possible segmentation paths in the candidate lattice using dynamic programming. Each segmentation path represents a set of candidate patterns, generated by different com-

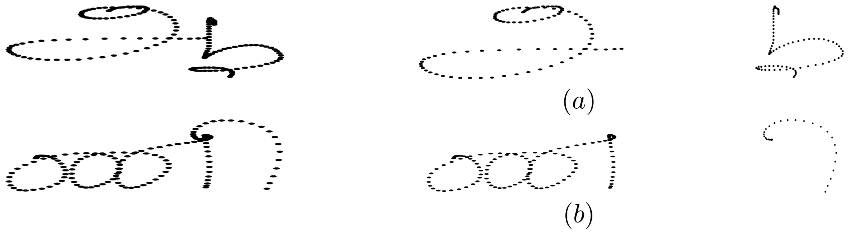


Fig. 9. Illustration of over-segmentation of two samples from the IWFHR database. (a) Sample of /A/ broken to 2 stroke groups. (b) Sample of /nni/ broken to 2 stroke groups.

binations of successive primitive segments obtained from the pre-segmentation step. A classifier is used to generate the likelihood score and recognition label for a candidate pattern. The candidate pattern sequence, corresponding to the optimal path represents the output word.

In contrast to this, the AFS module depicted in Fig 7(b) does not incorporate a dynamic programming step to generate the output word. Instead, we select using feature based attention, only stroke groups from the DOCS step suspected to be wrongly segmented and try correcting them to valid symbols. The feedbacks from the classifier likelihoods and stroke-group based features are employed to correct the segmentation error (if any). In conclusion, our methodology does not have the path evaluation step of the ISR approach. Instead, the main focus is on the rectification of selected stroke groups, detected to contribute to segmentation errors. Each of the stroke groups obtained after the AFS step are recognized to generate the output symbol labels, that in turn are combined to generate the output word.

In the subsequent sections, we outline the proposed attention feedback strategies for detecting and correcting the stroke groups wrongly segmented by the DOCS module. In this context, the following aspects need to be borne in mind.

(1) Out of the stroke groups resulting from the DOCS module, the AFS module operates on selected ones (suspected to be wrongly segmented) and recognizes them.
(2) Owing to its better generalization performance to unseen data (in comparison to the classifiers used in recent works [Joshi et al. 2004; Bharath and Madhvanath 2007; Bhattacharya et al. 2007; Babu et al. 2007]), the Support Vector Machine (SVM) [Burges 1998] is used as the recognizer in this work.
(3) Prior to sending a suspected split or under-segmented stroke group to the SVM classifier for generating the recognition label and likelihoods, we subject it to the preprocessing steps of smoothing, size normalization and resampling [Joshi et al. 2004].
(4) Moreover, since the emphasis here is on improving the segmentation rather than the classifier performance, x-y coordinates of the preprocessed stroke group are used as features. These features are robust for identifying stroke groups wrongly segmented by the DOCS module.

5. DETECTION OF POSSIBLY OVER-SEGMENTED STROKE GROUPS WITH FEATURE-BASED ATTENTION

The training samples of the IWFHR dataset are segmented based on the overlap criterion in the DOCS module. Since this dataset consists of isolated Tamil symbols, the segmentation of any sample into more than one stroke group indicates an over-segmentation. Figures 9 (a) and (b) respectively illustrate a sample of ஆ /A/ and ணி /nni/ that get over-segmented into more than one stroke group by DOCS step. We now explore the utility of a new feature - number of dominant points, to detect possible over-segmentations in the stroke groups. The number of dominant points of a stroke group provides a rich structural description. The

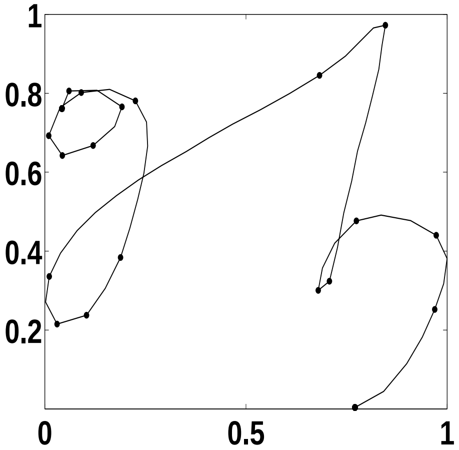


Fig. 10. Representation of the 20 dominant points (marked by dots) for /A/ vowel.

use of dominant points in the area of online handwriting recognition has been documented in the works [Li and Yeung 1997] [Yang and Dai 2002] [Joshi et al. 2004] [Ma et al. 2009].

Our algorithm for generating the dominant points begins by marking the first pen position as a dominant point. Starting from the current dominant point, we compute the absolute value of the angle between pen directions at successive points and accumulate it along the online trace as long as the cumulative sum is less than a threshold T_θ. The pen position, at which the accumulated angle exceeds T_θ, is marked as the next dominant point and the process continues till the end of the trace. The resulting number of dominant points extracted is used as a feature for attention. Most of the symbols in Tamil comprise curved strokes. We empirically choose threshold T_θ in order to ensure that the shape of the stroke group is approximated with a reduced set of points, without losing any points of high curvature. Very high values of T_θ do not sufficiently capture the shape of the character. On the other hand, for low values of T_θ, the number of dominant points increases with the approximated shape resembling more closer to the original stroke group. By experimentation, we noted that a value of T_θ in the range $[35^o, 55^o]$ works well for shape representation. In the present work, we choose $T_\theta = 45^o$. Figure 10 highlights the 20 dominant points for the stroke group ஆ /A/. The dominant points are extracted from the preprocessed stroke group [Joshi et al. 2004].

We now present a statistical justification towards using the number of dominant points of a stroke group as a cue to detect possible over-segmentation errors. Let us assume that a training sample X from the IWFHR data-set gets split by the DOCS module into $\tilde{p}$ stroke groups. The number of dominant points corresponding to each of the stroke groups is computed and denoted by $\{N^{S_1}, N^{S_2}...N^{S_{\tilde{p}}}\}$. For every sample X, we consider the number of dominant points ($\min_i N^{S_i}$) corresponding to the shortest stroke group in the split. The distribution of the number of dominant points of the shortest stroke group for all the training samples of symbols (in the IWFHR dataset) split by DOCS module is presented in Fig.11. The maximum number of dominant points, as read from Fig.11, is 15. From this, we can infer that a stroke group for which the number of dominant points is less than 16 *may* correspond to a part of a Tamil symbol. In conclusion, shorter stroke groups are more indicative of a broken symbol, than longer ones.

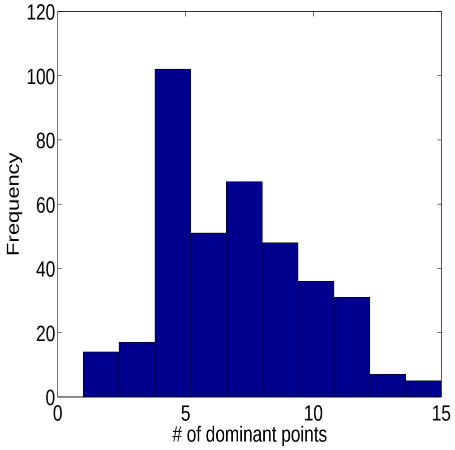


Fig. 11. Distribution of the number of dominant points across the shortest stroke groups of the over segmented symbols in the IWFHR dataset.

6. DETECTION OF POSSIBLY UNDER-SEGMENTED STROKE GROUPS USING FEATURE BASED ATTENTION

For preprocessed stroke groups comprising m strokes ($m > 1$), the horizontal displacement b_i from the bounding box x-maximum of the i^{th} stroke to the first point of the $(i+1)^{th}$ stroke is computed. The maximum of the computed displacements b_{max}, among all successive stroke pairs, is a feature for attention.

$$b_{max} = \max_i \quad b_i; \qquad i = 1, 2, ...m - 1 \tag{2}$$

The inter-stroke displacement b_{max} (interpreted as the maximum 'bounding box to next-stroke displacement' in a stroke group) may be either positive or negative, depending on the relative positions of the strokes under consideration. For the stroke group தி /ti/ shown in Fig.12 (c), written in 3 strokes, we have $b_{max} < 0$.

We now demonstrate the efficacy of this feature in detecting under-segmented stroke groups. An analysis is performed on stroke groups (comprising multiple strokes) obtained from DOCS module on the 250 handwritten words in the data-set DB1. Stroke groups for which $b_{max} > 0$ *may* correspond to Tamil symbols that have been merged. On the other hand, stroke groups satisfying $b_{max} < 0$ rarely produce an under segmentation error. In other words, when two valid Tamil symbols are merged in a stroke group, the inter-stroke displacement between the bounding box of the last stroke of the previous symbol to the first sample point of the next symbol is positive and large, when compared to the inter-stroke displacement in a correctly segmented stroke group. Hence, this feature serves as a cue to detect under-segmented stroke-groups. For the database DB1, as high as 95% of stroke groups contributing to under-segmentation errors satisfy $b_{max} > 0$. Figure 13 depicts the case wherein 2 Tamil symbols ை and ர are merged to a stroke group ைர. It is quite likely for the SVM classifier to regard this stroke group as an outlier pattern by providing a low likelihood to its most probable candidate. On the other hand, Fig. 12(c) presents a correctly segmented sample of தி /ti/ satisfying $b_{max} < 0$.

7. AFS STRATEGY FOR SUSPECTED OVER-SEGMENTED STROKE GROUPS

As explained in Sec 5, a stroke group with less than 16 dominant points may correspond to a part of a Tamil symbol. Figure 14 presents the block diagram of the AFS strategy proposed

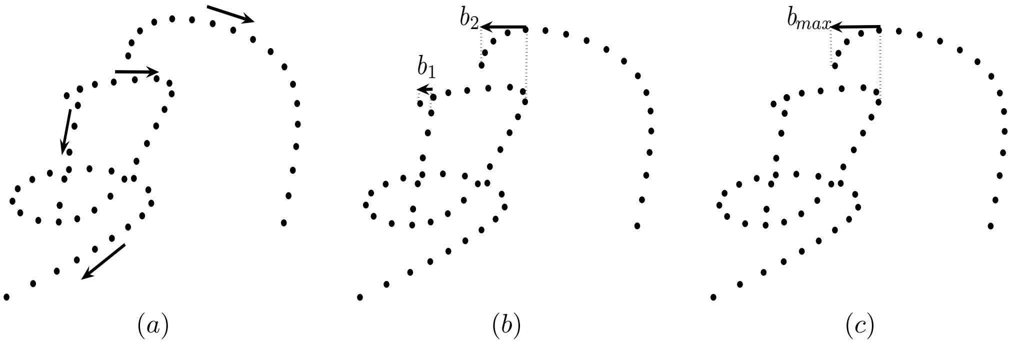


Fig. 12. Computation of b_{max} for /*ti*/ symbol. (a) Stroke group /ti/ with direction of trace marked with arrows. It comprises 3 strokes. (b) Illustration of the two inter-stroke displacements b_1, b_2. (c) Illustration of b_{max}. Note that for this stroke group $b_{max} = b_2$ and $b_{max} < 0$. Attention on the inter-stroke displacement b_{max} indicates that the stroke group is correctly segmented with DOCS module.

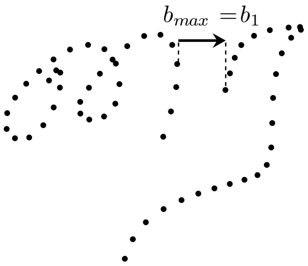


Fig. 13. Inter-stroke displacement b_{max} as an attention feature for detecting possible under-segmentation. The two distinct symbols shown are wrongly merged by DOCS module to a single stroke group. Note that for this stroke group, $b_{max} > 0$, indicating a suspected merger.

for correcting over-segmented stroke groups. Let S_k correspond to a stroke group that is suspected to be a broken symbol. Consider $S_{adj(k)}$ to be the neighboring stroke group whose BB is closest to that of S_k. The feature vector (concatenated x-y coordinates) of the preprocessed S_k and $S_{adj(k)}$ are separately sent to the SVM classifier. Let the likelihoods $P(\omega_{top}^{k})$ and $P(\omega_{top}^{adj(k)})$ correspond to the most probable symbols ω_{top}^{k} and $\omega_{top}^{adj(k)}$ respectively. Let S_M represent the stroke group obtained by merging S_k with $S_{adj(k)}$. For a possible merge, we require the average likelihood of the most probable symbols ω_{top}^{k} and $\omega_{top}^{adj(k)}$ to be less than the likelihood $P(\omega_{top}^{M})$ for S_M. However, for avoiding any unintentional merges, we additionally ensure that the maximum horizontal inter-stroke gap (denoted by d_{max}) in S_M is less than the maximum possible horizontal inter-stroke gap $T_{dmax}(\omega_{top}^{M})$ determined from the IWFHR dataset for the recognized symbol ω_{top}^{M}. In other words,

$$\begin{aligned} \frac{P(\omega_{top}^{k}) + P(\omega_{top}^{adj(k)})}{2} &< P(\omega_{top}^{M}) \\ d_{max}^{S_M} &< T_{dmax}(\omega_{top}^{M}) \end{aligned} \qquad (3)$$

The maximum horizontal inter-stroke gap d_{max} is computed as follows: For a preprocessed stroke group comprising m strokes, the signed horizontal inter stroke gap d_i between the last point of the i^{th} stroke and the first point of the $(i+1)^{th}$ stroke is measured. The maximum of the inter-stroke gaps represents d_{max}.

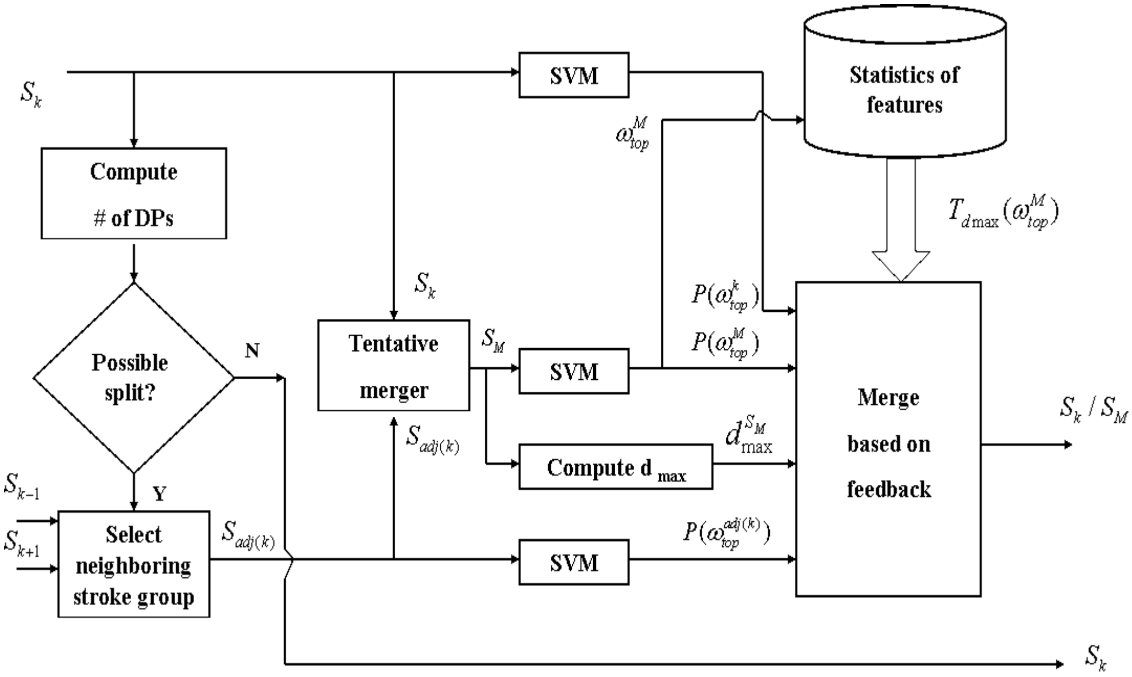


Fig. 14. AFS module for resolving suspected over-segmented stroke groups.

$$d_{max} = \max_{i} \quad d_i \qquad i = 1, 2, ...m - 1 \tag{4}$$

In contrast to b_{max} (defined in Sec 6), d_{max} can be interpreted as the maximum stroke to stroke displacement in a stroke group. We reiterate here that, for a given stroke group, b_{max} is used for detecting possible under segmentation errors (as explained in Sec 6). The maximum horizontal inter-stroke gap d_{max}, on the other hand, is used for correcting stroke groups from the DOCS module, suspected to be wrongly segmented.

Figure 15 presents a suitable illustration, wherein the second stroke group in Fig 15(a), suspected to be broken by the DOCS module, gets properly merged to a valid symbol ங் /ng/ by the AFS strategy. The correctly segmented word பூங்கா after the merge is shown in Fig 15(e).

As an illustration to how the inter-stroke gap d_{max} aids in preventing spurious merges, we consider the last stroke group ா that has 5 dominant points. The number of dominant points being less than 16, we tentatively merge it to the neighboring stroke group க and recognize the resulting pattern S_M (Fig. 16(a)). The SVM favors the symbol தூ /tU/ (the printed sample of which is shown in Fig. 16 (b)). However, we observe that the maximum possible inter-stroke gap for தூ is less than the d_{max} computed for S_M. Accordingly, we do not consider the merge. Instead, the individual stroke groups க and ா are favored.

8. AFS STRATEGY FOR SUSPECTED UNDER-SEGMENTED STROKE GROUPS

As explained in Sec 6, a stroke group satisfying $b_{max} > 0$ may correspond to a merger of valid Tamil symbols. In this section, we outline the proposed AFS strategy for resolving such suspected under-segmented stroke groups. From the block diagram of Fig 17, we observe

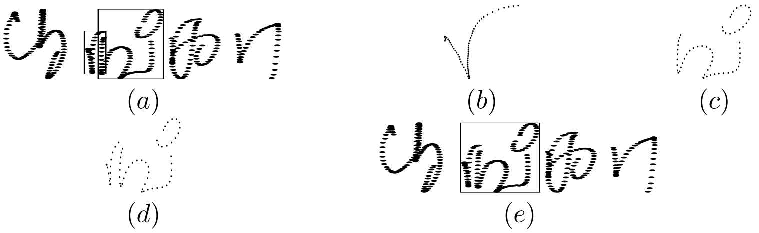


Fig. 15. AFS strategy for resolving suspected over-segmentation error in broken symbols. (a) An example of a word wrongly segmented by the DOCS module. (b) The second stroke group in this word has 8 dominant points and is suspected to be a part of a valid symbol. This stroke group has a low posterior probability. (c) The second split part of the symbol is recognized with a low posterior probability. (d) Merged symbol has a higher likelihood compared to those of the stroke groups in (b) and (c). (e) The correctly segmented word after the merge.

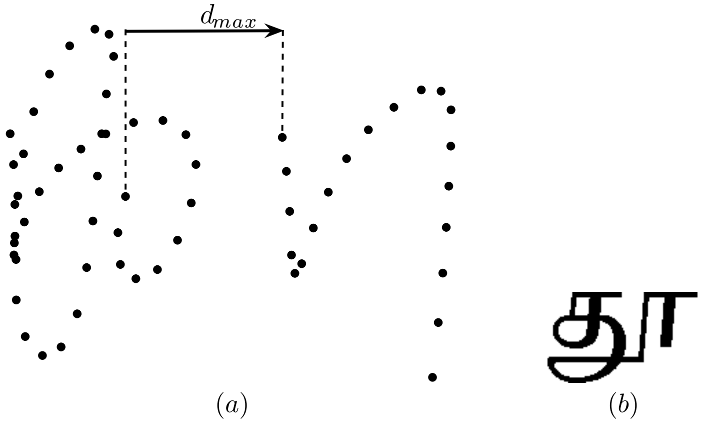


Fig. 16. Inter-stroke gap d_{max} as a feedback attribute to prevent unintentional merges. (a) Computation of d_{max} for the combined stroke group S_M. The SVM favors /tU/ as the most favorable symbol. (b) Printed sample of /tU/. The value of d_{max} computed for S_M is greater than the maximum possible inter-stroke gap for the symbol /tU/, obtained from the IWFHR database. So, the merge is not carried out.

that feedbacks of SVM likelihoods, statistics of number of dominant points and inter-stroke gap d_{max} (defined in Eqn 4) influence our decision to split a stroke group.

Assume that a stroke group S_k, comprising m strokes, satisfies $b_{max} > 0$. If b_{max} corresponds to the inter stroke displacement, occurring between the q^{th} and $(q+1)^{th}$ strokes, then we suspect stroke group S_k to be the merger of two valid symbols S_{k_1} and S_{k_2}, defined by $S_{k_1} = \{s_{k,1}, s_{k,2}, ...s_{k,q}\}$ and $S_{k_2} = \{s_{k,q+1}, s_{k,q+2}, ...s_{k,m}\}$. Here $s_{k,i}$ denotes the i^{th} stroke for stroke group S_k. S_{k_1} and S_{k_2} are in turn preprocessed and subsequently recognized to generate likelihoods

$$P(\omega_{top}^{k_j}) = \max_i \quad P(\omega_i|\mathbf{x}^{S_{k_j}}) \qquad j = 1, 2 \tag{5}$$

We favor splitting the stroke group S_k into S_{k_1} and S_{k_2} whenever

$$\frac{\sum P(\omega_{top}^{k_j})}{2} > P(\omega_{top}^{k}) \tag{6}$$

Here ω_{top}^{k} represents the most probable symbol of the SVM for the stroke group S_k. For the scenario, where the inequality is not satisfied, additional cues (derived from statistics) are employed for resolving the possible under-segmentation error in S_k.

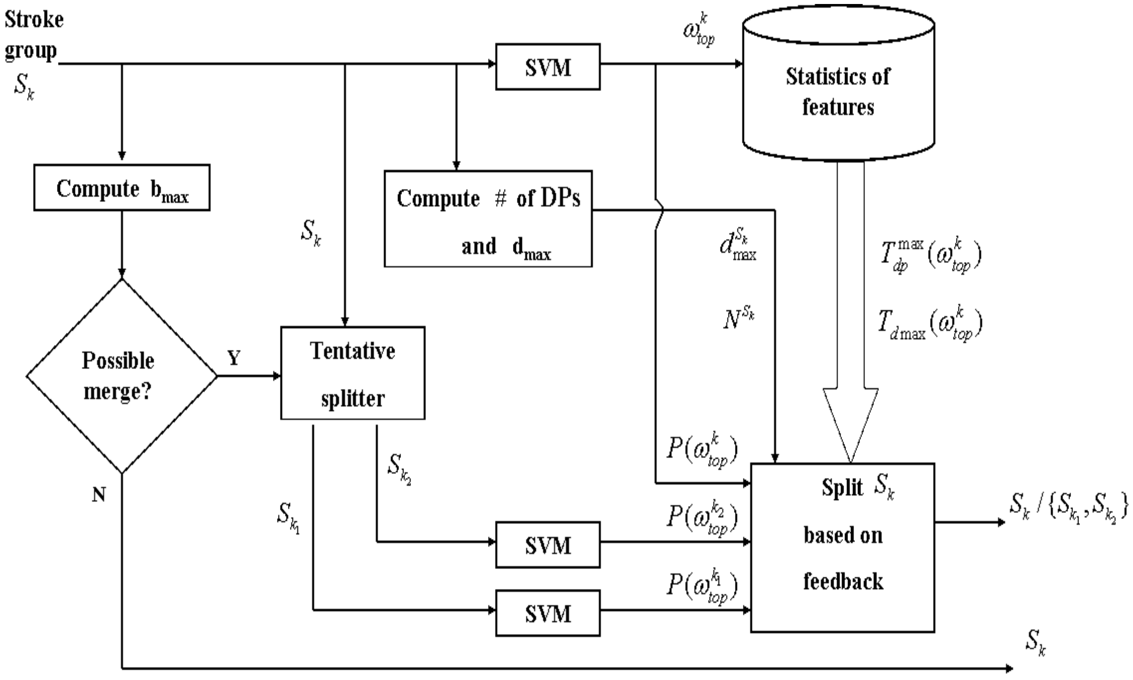


Fig. 17. AFS module for resolving stroke groups suspected to be under-segmented.

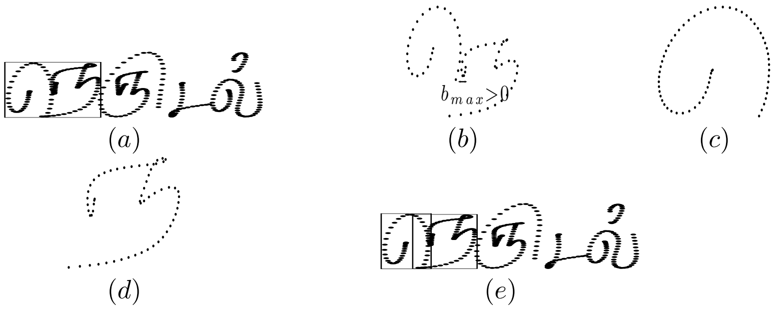


Fig. 18. AFS scheme for resolving under-segmentation errors in Tamil words. (a) An example of a word wrongly segmented by the DOCS module. (b) The first stroke group in the word satisfies $b_{max} > 0$ and is suspected to comprise 2 merged valid symbols. (c)(d) The extracted symbols are recognized separately. The stroke group is split if the mean likelihood of the extracted symbols exceeds the likelihood for the combined symbol shown in (b). (e) The correctly segmented word after the split.

(1) If the number of dominant points N^{S_k} in S_k is greater than the maximum number ($T_{dp}^{max}(\omega_{top}^{k})$) determined for the most probable symbol ω_{top}^{k} determined from the IWFHR data-set, we proceed ahead in segmenting it to 2 valid symbols S_{k_1} and S_{k_2}.
(2) If d_{max} obtained for the stroke group S_k (denoted by $d_{max}^{S_k}$) is greater than maximum horizontal inter stroke gap ($T_{dmax}(\omega_{top}^{k})$) for ω_{top}^{k}, we segment it.

Figure 18 illustrates the case, wherein the wrongly segmented stroke group நெ at the start of the word நெருடல் is segmented correctly to 2 valid symbols ெ and ந, respectively.

Table I. Performance evaluation of classifiers on the test data of the IWFHR Tamil symbol database.

Classifier	Recognition Accuracy (in %)
k-NN (k=3)	76
DTW+NN	77.6
HMM	83.3
SVM	86.5

9. RESULTS AND DISCUSSION

The SVM is chosen as the classifier in the AFS module, based on a comprehensive evaluation of different state of art classifiers on the test data used in the IWFHR 2006 Tamil symbol competition data-set. All the classifiers (listed in Table I) are trained with the concatenated x and y coordinates of the preprocessed Tamil symbols from the IWFHR training set. From the recognition rates, we note that the SVM provides a performance superior to the classifiers k-NN, DTW+NN and HMMs.

The SVM classifier has been trained using a RBF kernel with the parameters $C = 5$ and $\gamma = 0.2$. The kernel and their corresponding parameters have been optimally set after performing five-fold cross validation on the IWFHR training data. The LIB-SVM toolbox [Chang and Lin 2001] has been used for the experiments.

In addition, for each symbol ω_i, the following statistics are generated.

(1) Maximum number of dominant points ($T_{dp}^{max}(\omega_i)$) across all samples of ω_i.
(2) $T_{dmax}(\omega_i)$ - Maximum horizontal inter stroke gap (as defined in Eqn.4) over all samples.

The segmentation accuracy is used as a measure of performance evaluation for our experiments. It is computed as follows : For a set of t words (being tested), we define the following attributes with respect to the i^{th} word,
N_C^i - Actual number of valid symbols/stroke groups obtained with manual segmentation
N_D^i - Number of stroke groups corresponding to valid symbols, obtained from the DOCS module alone
N_A^i - Number of stroke groups corresponding to valid symbols, obtained from the AFS module
The segmentation accuracy (at symbol level) from the DOCS module alone = $\sum_{i=1}^{t} N_D^i / \sum_{i=1}^{t} N_C^i$
The accuracy (at symbol level) after the AFS module = $\sum_{i=1}^{t} N_A^i / \sum_{i=1}^{t} N_C^i$
For our experiments, $t = 10000$ and $\sum_{i=1}^{t} N_C^i = 53246$ (as discussed in Sec 2, we test our segmentation methodology on the 10000 words of the MILE word data-set comprising 53246 symbols).

9.1. Segmentation results on the IWFHR Tamil database

Though the primary focus is on segmenting Tamil words, as a first experiment, we evaluate the performance of the proposed approach on the symbols in the IWFHR training dataset. For the isolated symbols in this dataset, the errors can arise only due to over-segmentation. For ease of analysis, we manually divide the 155 symbols (listed in Appendix A) into 8 groups.

G_1	Base consonants
G_2	Pure consonants
G_3	Additional symbols
G_4	CV combinations of vowels உ
G_5	CV combinations of vowels இ

Table II. Performance evaluation of the proposed segmentation strategy on the symbols of the IWFHR database. (Trial experiment performed on the training data).

Group	# of samples	# of errors from DOCS module	# of errors after AFS module	% Error red- -uction by (AFS module)	Segmentation performance of DOCS module	Segmentation performance after AFS module
G_1	7457	46	8	82.6	99.4	99.9
G_2	7523	108	8	92.6	98.5	99.9
G_3	1998	337	13	96.1	83.1	99.3
G_4	7351	481	34	92.9	93.4	99.5
G_5	7534	201	15	92.5	97.4	99.8
G_6	3714	277	6	97.8	92.5	99.8
G_7	7525	195	14	92.8	97.4	99.8
G_8	7237	432	151	65.0	94.0	97.9
Total	50339	2077	249	88.0	95.9	99.5

Table III. Results of the proposed segmentation strategy on one set of words from the MILE word database (DB1). Total # of words=250. Total # of symbols=1210.

	DOCS module	AFS module	% error reduction
# of correctly segmented symbols	1107	1198	
# of merged symbols	89	9	89.9
# of broken symbols	14	3	78.6
Correctly segmented symbols (in %)	91.5	99.0	88.3
# of wrongly segmented words	67	7	89.5

G_6	Vowels
G_7	CV combinations of vowels ஈ
G_8	CV combinations of vowels ஔ

Table II outlines the results of the proposed segmentation strategies on each of these groups. It can be noted that the DOCS module alone gives a reasonable accuracy of 95.9%. Incorporation of the AFS module increases the segmentation accuracy to 99.5%. We also observe that the over-segmentation errors by the DOCS module are drastically reduced by the AFS module (by around 88.0%) across the entire database.

9.2. Segmentation results on the MILE word database

The proposed techniques are tested on the entire word database. However, to start with, we evaluate the performance on the validation set DB1. Table III outlines the segmentation statistics. The proposed DOCS module correctly segments 1107 out of the 1210 symbols, accounting for 91.5% segmentation accuracy. Of the 103 errors, 86.4% (89 errors) corresponds to the merging of valid symbols. The AFS module described in Sec.8 aids in properly detecting and correcting 89.9% of these errors. In addition, the method proposed in Sec.7 effectively merges 78.6% of the over-segmented stroke groups to valid symbols. The improvement in symbol segmentation rate with the incorporation of the AFS module in turn reduces the number of wrongly segmented words. It can be observed from the last row of the table that 60 additional words have been properly segmented.

Table IV presents the the performance of the segmentation methodology across the entire MILE word database (comprising 10000 words). The results show that the DOCS module achieves a segmentation accuracy of 98.1% at the symbol level, which improves to as high as 99.7% after the AFS strategy. This correspond to a 86.2% reduction in symbol segmentation errors.

9.3. Recognition results on the MILE word database

Tables V and VI respectively present the recognition results, for sample Tamil words, from the DOCS and AFS modules. Application of the DOCS module on each word in Table V leads to a merge of a few valid symbols. On the other hand, at least one valid symbol in

Table IV. Performance evaluation of the proposed DOCS and AFS schemes on the symbols in the entire MILE word database. Total # of words=10000. Total # of symbols=53246.

	DOCS	AFS	% error reduction
# of correctly segmented symbols	52245	53107	
# of segmentation errors	1001	139	86.2
Segmentation rate in (%)	98.1	**99.7**	1.6
Symbol recognition rate in (%)	83.9	88.4	4.5

Table V. Improvement in recognition due to correct segmentation. The valid symbols merged by the DOCS module is shown within a box in the first column. The symbols contained within the boxes in the second column indicate the symbol recognition errors. The third column displays the recognition output after the segmentation is refined by the AFS module.

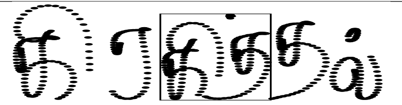
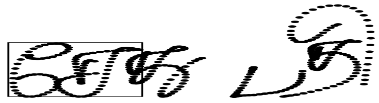
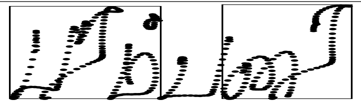
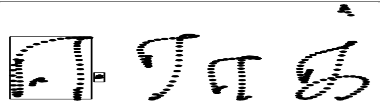
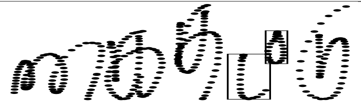
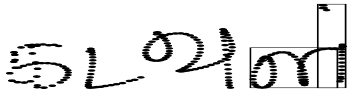
Input word	Word recognized from DOCS module	Word recognized from AFS module
	கிரஓதல்	கிரகித்தல்
	ச்ஷதுபதி	சேதுபதி
	ஹூபங்	பரம்பரை

Table VI. Improvement in recognition due to correct segmentation. The split parts of valid symbols broken in the DOCS module are highlighted with boxes in the first column. The symbols contained within the boxes in the second column indicate the symbol recognition errors. The third column displays the recognition output after the segmentation is refined by the AFS module.

Input word	Word recognized from DOCS module	Word recognized from AFS module
	ஈஃராக்	ஈராக்
	கைதடபடு	கைதட்டு
	கடவுணகூ	கடவுள்

each word in Table VI appears as more than one stroke group due to over-segmentation. The incorrect segmentation in turn increases the symbol recognition errors, as shown in the second column of the two tables. From the third columns, we observe that all the constituent symbols of these words are recognized correctly after the segmentation is refined by the incorporation of the AFS module.

Table VII gives the recognition accuracy for the set DB1, obtained with DOCS and AFS modules. Since a significant percentage of DOCS errors are corrected by the AFS strategy, a drastic improvement of 16.6% (from 70.5% to 87.1%) in symbol recognition is observed. In computing the symbol recognition rate, apart from the substitution errors, we take into account the insertion and deletion errors, caused by over-segmentation and under-segmentation, respectively. The edit distance [Duda et al. 1995] is used for matching the recognized symbols with the ground truth data. Moreover, 11.6% of the words, (29 additional words) wrongly recognized after DOCS step, have been corrected by the AFS

Table VII. Results of the proposed DOCS and AFS schemes on the recognition of words in database DB1. Total # of words=250. Total # of symbols=1210.

	DOCS module	AFS module	% error reduction
# of correctly recognized symbols	853	1054	56.3
Symbol recognition rate (in %)	70.5	87.1	16.6
# of correctly recognized words	85	114	11.6

Table VIII. Results obtained with the proposed DOCS and AFS schemes on the word recognition rate. Total # of words=10000.

	DOCS module	AFS module	Improvement	% error reduction
# of words correctly recognized	5089	6493	1404	
Word recognition rate (in %)	50.9	64.9	14	28.5

technique. Across the 10000 words in the MILE word database, an improvement of 4.5% (from 83.9% to 88.4%) in symbol recognition rate was obtained (refer Table IV).

Table VIII presents the word recognition rates obtained on the MILE word database with our method. We observe word recognition rates of 50.9% and 64.9% for the DOCS and AFS module respectively. In summary, the incorporation of the AFS module lowers the word error rate by 28.5%.

9.4. Choice of the threshold for the DOCS module

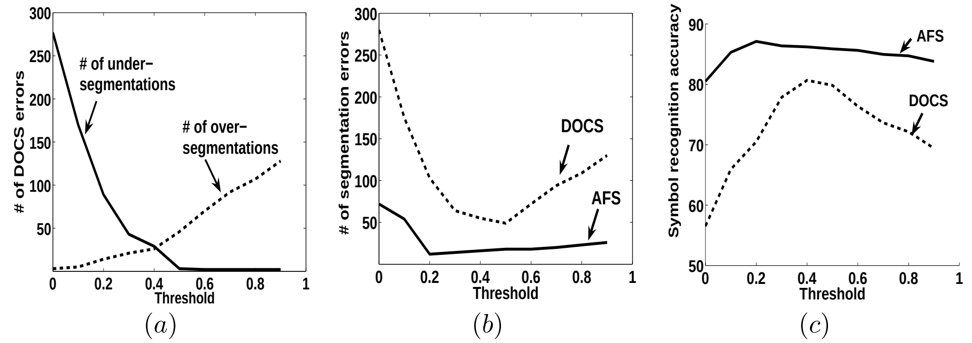
In all of the preceding experiments and discussions, sets of consecutive strokes of the word are merged into stroke groups in the DOCS module by comparing their degree of overlap O_k^c (defined in Eqn.1) to a threshold $T_0 = 0.2$. The number of properly segmented stroke groups generated by DOCS module depends on the value of T_0. Figure 19(a) quantifies the frequency of errors due to symbol merges and splits as a function of the threshold. We vary T_0 from 0 to 0.9 in steps of 0.1 and demonstrate the effectiveness of the proposed attention feedback segmentation method on DB1, irrespective of the threshold selected. $T_0 = 0$ leads to the maximum number of unintentional merges, especially when symbols are written close enough to each other that their bounding boxes are adjacent. For higher values of T_0, a significant number of valid stroke groups get over segmented (refer Fig. 19 (a)). Irrespective of the threshold set, the AFS scheme is able to correct at least 75% of the segmentation errors encountered (Fig.19 (b)). The corresponding improvement in symbol recognition accuracy of the handwriting system for the different threshold values is presented in Fig.19(c). We observe from Fig.19(b) that $T_0 = 0.2$ gives the minimum segmentation error rate after the AFS step. Hence, we chose this threshold for our experiments and illustrations in this work.

Further, Fig.19 (b) shows that while the performance of the DOCS module is sensitive to the choice of T_0, the overall segmentation performance, after the AFS strategy, is fairly insensitive to the choice of the threshold value between 0.2 and 0.9.

9.5. Error Analysis

Despite a segmentation accuracy of 99.7% with the proposed approach, we analyzed the wrongly segmented words to determine the causes for segmentation errors after the AFS step. These have been enumerated below:

(1) It was observed that segmentation errors arise in cases where symbols are written as a different temporal sequence rarely encountered in Tamil script.
(2) Cursive writing in Tamil is rare. Therefore, Tamil words, in which two or more symbols are joined by a single stroke, were not correctly segmented.
(3) Errors were noted in incorrectly written symbols, comprising large horizontal inter-stroke gaps, that are comparable to the horizontal inter-character gaps. For an



(a) (b) (c)

Fig. 19. Sensitivity of DOCS and AFS modules to the choice of overalp threshold T_0 in the DOCS module. The threshold is varied from 0 to 0.9 in steps of 0.1. The DB1 dataset is used for this experiment. (a) Variation of number of over-segmentations and under-segmentations in the DOCS module with T_0. (b) Variation of number of incorrect segmentations in the DOCS and AFS modules with T_0. (c) Symbol recognition rates on outputs from the DOCS and AFS modules as a function of T_0.

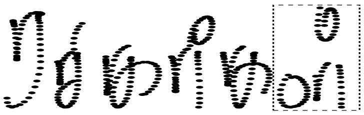


Fig. 20. Illustration of a word that does not get properly segmented by the proposed segmentation strategy. The broken stroke groups contained within the dotted box fail to merge to the valid symbol /L/.

illustration, we refer the reader to Fig.20. Here, the otherwise double stroke symbol ள் /L/ in the word ரசிகர்கள் is so badly written with four strokes that their horizontal inter-stroke gap is comparable to the inter-character gaps. Accordingly, the stroke groups (highlighted in the bounding box) do not get properly merged to the symbol ள்.

From Table IV, we observe that SVM recognition on the segmented stroke groups after the AFS step across the 10000 words yields an accuracy of 88.4%. The relatively low recognition rate (compared to the segmentation accuracy) is attributed to the fact that there are many symbols in Tamil that appear visually similar. With post processing techniques, (that aim at describing discriminative features between similar looking symbols), one can possible achieve a higher symbol recognition accuracy on the segmented stroke groups using a SVM classifier.

10. ANALOGY OF OUR TECHNIQUE TO NEUROSCIENCE CONCEPTS

For justifying the proposed term ‘attention-feedback’, we present an analogy to concepts in the area of neuroscience. Studies on visual perception in primates demonstrate the effect of attention on the response of the visual neurons. Feature based attention [Boynton 2005] biases the neuronal responses as though the attended stimulus was presented alone. Also, shifting spatial attention from outside to the inside of the receptive field increases the neuronal responses. Further, studies on visual pathways [Sillito and Jones 2002] show extensive feedback from the cortex to the lateral geniculate nucleus (LGN), which have both inhibitory and facilitatory effects on the responses of LGN relay cells. As evident from the previous sections, in the proposed work, we incorporate local feature based attention in the AFS module to correct and improve the segmentation of the DOCS module. In addition, feedback based on stroke based features as well as the classifier likelihoods are employed to rectify any incorrect segmentations by regrouping the strokes.

11. CONCLUSION

This paper presents a novel, script-dependent, lexicon-free, segmentation approach for online handwritten isolated Tamil words. Initial segmentation of the given word is performed by the DOCS module into a set of stroke groups. Attention on certain spatial and temporal features, derived from the characteristics of the script, detect likely split and under-segmented stroke groups, if any. The likelihoods fed back by the SVM as well as known statistics of stroke-group based features corrects the wrongly segmented stroke groups in the AFS module. The results of our proposed methodology have been presented on a challenging data-set of 10000 words.

It is to be noted that the statistics of features and likelihood have been derived from the IWFHR database of isolated Tamil symbols. That they work reliably for segmentation of word level data written by a completely different set of people shows the promise of the general applicability of our approach to any database of handwritten isolated Tamil words.

Given that there is no prior work done in segmenting online Tamil words, it is difficult to compare our method to a benchmark. The segmentation scheme proposed for cursive Bangla words in [Bhattacharya et al. 2008; Fink et al. 2010] cannot be extended to Tamil, owing to major structural differences in the scripts. To the best of our knowledge, this is the first work directed at addressing the practical issues in Tamil online handwriting segmentation.

Acknowledgements

We would like to thank the anonymous reviewers of this paper for their critical review and constructive suggestions, which helped us to restructure the paper and enhance the effectiveness of its presentation. The support received from Prof. P S Sastry, Department of Electrical Engineering, IISc, in reviewing the manuscript is gratefully acknowledged. We also thank Dr. Arun Sripathy, Centre for Neuroscience, IISc for the discussion we had with him. The high school students and project staff (in our laboratory) are acknowledged for their contribution in building and annotating the word database respectively.

REFERENCES

Babu, J., Prashanth, L., Sharma, R. R., Rao, G. V. P., and Bharath, A. 2007. HMM-based online handwriting recognition system for telugu symbols. In *Proc. Int'l Conf. Document Analysis and Recognition.* 1028–1032.

Basu, S., Sarkar, R., Das, N., Kundu, M., Nasipuri, M., and Basu, D. K. 2007. A fuzzy technique for segmentation of handwritten Bangla word images. In *Proc. Int'l Conf. Computing: Theory and Applications.* 427–433.

Bharath, A. and Madhvanath, S. 2007. Hidden markov models for online handwritten Tamil word recognition. In *Proc. Int'l Conf. Document Analysis and Recognition.* 506–510.

Bhattacharya, U., Gupta, B. K., and Parui, S. 2007. Direction code based features for recognition of online handwritten characters of Bangla. In *Proc. Int'l Conf. Document Analysis and Recognition.* 58–62.

Bhattacharya, U., Nigam, A., Rawat, Y. S., and Parui, S. K. 2008. An analytic scheme for online handwritten Bangla cursive word recognition. In *Proc. Int'l Workshop. Frontiers in Handwriting Recognition.* 320–325.

Bishnu, A. and Chaudhuri, B. B. 1999. Segmentation of Bangla handwritten text into characters by recursive contour following. In *Proc. Int'l Conf. Document Analysis and Recognition.* 402–405.

Boynton, G. M. 2005. Attention and visual perception. *Current Opinion in Neurobiology 15*, 465–469.

Burges, C. 1998. A tutorial on support vector machines for pattern recognition. *Data Mining Knowledge Dicovery 2*, 1–47.

Camastra, F. 2007. A SVM-based cursive character recognizer. *Pattern Recognition 40,* 12, 3721–3727.

Chang, C. C. and Lin, C. J. 2001. LibSVM – a library for support vector machines:. *http://www.csie.ntu.edu.tw/ cjlin/libSVM/.*

CHERIET, M., KHARMA, N., LIU, C. L., AND SUEN, C. 2008. *Character Recognition Systems: A Guide for Students and Practitioners*. Wiley.

DUDA, HART, AND STORK. 1995. *Pattern Classsification*. Springer Wiley.

FINK, G. A., VAJDA, S., BHATTACHARYA, U., PARUI, S. K., AND CHAUDHURI, B. B. 2010. Online Bangla word recognition using sub-stroke level features and hidden markov models. In *Proc. Int'l Conf. Frontiers in Handwriting Recognition*. 393–398.

FURUKAWA, N., TOKUNO, J., AND IKEDA, H. 2006. Online character segmentation method for unconstrained handwritten strings using off-stroke features. In *Proc. Int'l Workshop. Frontiers in Handwriting Recognition*. 361–366.

GAO, X., LALLICAN, P. M., AND VIARD-GAUDIN, C. 2005. A two-stage online handwritten chinese character segmentation algorithm based on dynamic programming. In *Proc. Int'l Conf. Document Analysis and Recognition*. 735–739.

JAGER, S., MANKE, S., REICHERT, J., AND WAIBEL, A. 2001. Online handwriting recognition: the npen++ recognizer. *Int'l J. Document Analysis and Recognition 3,* 3, 169–180.

JOSHI, N., SITA, G., RAMAKRISHNAN, A. G., AND MADHAVANATH, S. 2004. Comparison of elastic matching algorithms for online Tamil handwritten character recognition. In *Proc. Int'l Workshop. Frontiers in Handwriting Recognition*. 444–449.

KOERICH, A. L., SABOURIN, R., AND SUEN, C. Y. 2005. Recognition and verification of unconstrained handwritten words. *IEEE Trans. Pattern Analysis and Machine Intelligence 27,* 10, 1509–1522.

LI, X. AND YEUNG, D. Y. 1997. On-line handwritten alphanumeric character recognition using dominant points in strokes. *Pattern Recognition 30*, 31–44.

LIU, C. L., JAEGER, S., AND NAKAGAWA, M. 2004a. Online recognition of chinese characters: The state-of-the-art. *IEEE Trans. Pattern Analysis and Machine Intelligence 24,* 2, 198–213.

LIU, C. L., SAKO, H., AND FUJISAWA, H. 2004b. Effects of classifier structures and training regimes on integrated segmentation and recognition of handwritten numeral strings. *IEEE Trans. Pattern Analysis and Machine Intelligence 26,* 11, 1395–1407.

LIWICKI, M., SCHERZ, M., AND BUNKE, H. 2006. Word extraction from on-line handwritten text lines. In *Proc. Int' Conf. on Pattern Recognition*. 929–933.

MA, L., HUO, Q., AND SHI, Y. 2009. A study of feature design for online handwritten chinese character recognition based on continuous-density hidden markov models. In *Proc. Int'l Conf on Document Analysis and Recognition*. 526–530.

MADHVANATH, S. AND GOVINDARAJU, V. 2001. The role of holistic paradigms in handwritten word recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence 23,* 2, 149–164.

MADHVANATH, S. AND LUCAS, S. M. 2006. Isolated Tamil handwritten character database:. In *www.hpl.hp.com/india/research/penhw-interfaces-1linguistics.html*.

MARTI, U. V. AND BUNKE, H. 2002. Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system. *Int'l J. Pattern Recognition and Artificial Intelligence 15,* 1, 65–90.

MURASE, H. 1988. Online recognition of free-format Japanese handwritings. In *Proc. Int'l Conf. Pattern Recognition*. 1143–1147.

NAGAKAWA, M., ZHU, B., AND ONUMA, M. 2005. A model of online handwritten Japanese text recognition free from line direction and writing format constraints. *IECIE Trans. Information and Systems*, 1815–1822.

NETHRAVATHI, B., ARCHANA, C. P., SHASHIKIRAN, K., RAMAKRISHNAN, A. G., AND KUMAR, V. 2010. Creation of a huge annotated database for Tamil and kannada ohr. In *Proc. Int'l Conf. Frontiers in Handwriting Recognition*. 415–420.

OLIVEIRA, L. E. S., SABOURIN, R., BORTOLOZZI, F., AND SUEN, C. Y. 2002. Automatic recognition of handwritten numerical strings: A recognition and verification strategy. *IEEE Trans. Pattern Analysis and Machine Intelligence 24,* 11, 1438–1454.

QUINIOU, S., BOUTERUCHE, F., AND ANQUETIL, E. 2009. Word extraction associated with a confidence index for online handwritten sentence recognition. *Int'l J. Pattern Recognition and Artificial Intelligence 23*, 945–966.

SENIOR, A. W. AND ROBINSON, A. J. 1998. An off-line cursive handwriting recognition system. *IEEE Trans. Pattern Analysis and Machine Intelligence 20,* 3, 309–321.

SILLITO, A. M. AND JONES, H. E. 2002. Corticothalamic interactions in the transfer of visual information. *Philos Trans R Soc Lond B Biol Sci*, 1739–1752.

SUNDARAM, S. AND RAMAKRISHNAN, A. G. 2010. Attention feedback based robust segmentation of online handwritten words. In *Indian Patent Office Reference*. No: 03974/CHE.

SWETHALAKSHMI, H., SEKHAR, C. C., AND CHAKRAVARTHY, V. S. 2007. Spatiostructural features for recognition of online handwritten characters in Devanagari and Tamil scripts. In *Proc. Int'l Conf. Artificial neural networks*. 230–239.

TONOUCHI, Y. 2010. Path evaluation and character classifier training on integrated segmentation and recognition of online handwritten Japanese character string. In *Proc. Int'l Conf. Frontiers in Handwriting Recognition*. 513–517.

TRIPATHY, N. AND PAL, U. 2004. Handwriting segmentation of unconstrained oriya text. In *Proc. Int'l Workshop. Frontiers in Handwriting Recognition*. 306–311.

VARGA, T. AND BUNKE, H. 2005. Tree structure for word extraction from handwritten text lines. In *Proc. Int'l Conf on Document Analysis and Recognition*. 352–356.

YANG, S. AND DAI, G. 2002. Detecting dominant points on online scripts with a simple approach. In *Proc. Int'l Workshop. Frontiers in Handwriting Recognition*. 351–356.

ZHAO, S. Y., CHI, Z. R., AND SHI, P. F. 2003. Two-stage segmentation of unconstrained handwritten chinese characters. *Pattern Recognition 36*, 145–156.

ZHOU, X. D., LIU, C. L., AND NAKAGAWA, M. 2009. Online handwritten Japanese character string recognition using conditional random fields. In *Proc. Int'l Conf. Document Analysis and Recognition*. 521–525.

ZHOU, X. D., YU, J. L., LIU, C. L., NAGASAKI, T., AND MARUKAWA, K. 2007. Online handwritten Japanese character string recognition incorporating geometric context. In *Proc. Int'l Conf. Document Analysis and Recognition*. 48–52.

ZHU, B. AND NAKAGAWA, M. 2006. Segmentation of on-line freely written Japanese text using SVM for improving text recognition. In *IECIE Trans.Information and Systems*. 105–113.

ZHU, B., ZHOU, X. D., LIU, C. L., AND NAGAKAWA, M. 2010. A robust model for on-line handwritten Japanese text recognition. *Int'l J. Document Analysis and Recognition 13,* 2, 121–131.

ZHU, B., ZHOU, X. D., LIU, C. L., AND NAKAGAWA, M. 2009. Effect of improved path evaluation for on-line handwritten Japanese text recognition. In *Proc. Int'l Conf. Frontiers in Handwriting Recognition*. 516–521.

Online Appendix to: Attention-feedback based robust segmentation of online handwritten isolated Tamil words

SURESH SUNDARAM , A G RAMAKRISHNAN, Indian Institute of Science

The list of 155 Tamil symbols used in this work

— **Vowels**[1]

அ ஆ இ ஈ உ ஊ எ ஏ ஐ ஒ ஓ

— **Base Consonants**

க ங ச ஞ ட ண த ந ப ம

ய ர ல வ ழ ன ற ள ஸ ஷ

ஜ ஹ க்ஷ

— **Pure Consonants**

க் ங் ச் ஞ் ட் ண் த் ந் ப் ம் ய்

ர் ல் வ் ழ் ன் ற் ள் ஸ் ஷ் ஜ்

ஹ் க்ஷ்

— **CV combinations of இ vowel**

கி ஙி சி ஞி டி ணி தி நி பி மி

யி ரி லி வி ழி னி றி ளி ஸி

ஷி ஜி ஹி க்ஷி

[1] The vowel ஔ is represented with 2 symbols, namely ஒ and ள. These symbols have been already included in the list of vowels and base consonants respectively.

© ACM 0000-0000//00-ARTA $10.00
DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

- **CV combinations of ஈ vowel**

  கீ ஙீ சீ ஞீ டீ ணீ தீ நீ பீ மீ
  யீ ரீ லீ வீ ழீ ளீ றீ னீ ஸீ
  ஷீ ஜீ ஹீ க்ஷீ

- **CV combinations of உ vowel**

  கு ஙு சு ஞு டு ணு து நு பு மு
  யு ரு லு வு ழு ளு று னு ஸு
  ஷு ஜு ஹு க்ஷு

- **CV combinations of ஊ vowel**

  கூ ஙூ சூ ஞூ டூ ணூ தூ
  நூ பூ மூ யூ ரூ லூ வூ
  ழூ ளூ றூ னூ ஸூ ஷூ
  ஜூ ஹூ க்ஷூ

- **Additional symbols**

  ஃ ா ெ ே ை ஸ்ரீ