Performance enhancement of online handwritten Tamil symbol recognition with reevaluation techniques

Suresh Sundaram and A G Ramakrishnan

Abstract In this article, we aim at reducing the error rate of the Tamil symbol recognition system by employing multiple experts to reevaluate certain decisions of the primary Support Vector Machine (SVM) classifier. Motivated by the relatively high percentage of occurrence of base consonants in the script, a reevaluation technique has been proposed to correct any ambiguities arising in the base consonants. Secondly, a dynamic time warping method is proposed to automatically extract the discriminative regions for each set of confused characters. Class-specific features derived from these regions aid in reducing the degree of confusions. Thirdly, statistics of specific features are proposed for resolving any confusions in vowel modifiers.

The reevaluation approaches are tested on two databases (a) the isolated Tamil symbols in the IWFHR test set, and (b) the symbols segmented from a set of 10000 Tamil words. The recognition rate of the isolated test symbols of the IWFHR database improves by 1.9%. For the word database, the incorporation of the reevaluation step improves the symbol recognition rate by 3.5% (from 88.4% to 91.9 %). This, in turn boosts the word recognition rate by 11.9% (from 65.0% to 76.9%). The reduction in the word error rate has been achieved using a generic approach, without the incorporation of language models.

Keywords Reevaluation; Experts; Support vector machines, online Tamil symbols, Attention point, Region of attention

Originality and Contributions

1. In the literature, in the context of online Indic handwriting, there is hardly any comprehensive work that addresses the problem of disambiguating confused characters. To the knowledge of the authors, this may be the maiden attempt at reducing the error rate of online handwritten Tamil symbols with reevaluation strategies.
2. A Dynamic time warping approach has been proposed to capture the regions of the trace that discriminate confused Tamil symbols. Thereafter, novel class-specific discriminative features are proposed from the extracted regions to disambiguate these symbols.
3. Dedicated to each confusion set (derived from the confusion matrix), a SVM classifier (referred as expert) has been proposed. The expert classifier operates on the novel discriminative features.
4. A set of novel features have been proposed to reduce the confusions of vowel modifiers in CV combinations.
5. A systematic study of occurrence frequency of linguistically similar Tamil symbols has been performed on a text corpus.

1 Introduction

The Indian sub-continent has as many as 22 official languages and 10 scripts. One such language is Tamil, which is spoken predominantly in the southern region of the country. The language is written using the Tamil script and is written from left to right. Recognizing online handwritten Tamil symbols is a non-trivial pattern recognition problem [1]. The challenges arise primarily due to the presence of a large character set, complex character shapes and different variations of writing styles.

The earliest work on online handwritten Tamil character recognition has been reported in [2]. Here, the authors demonstrated the performance of angle features, Fourier coefficients and wavelet features on a neural network classifier. A combination of time-domain and frequency-domain features has been used to build and test a HMM classifier in [3]. A similar set of feature combinations has been recently tested with a dynamic time warping approach in [4]. For writer dependent recognition of online handwritten isolated Tamil characters, a comparative study of elastic matching schemes using different feature sets is presented in [5]. Three features are considered namely, preprocessed x-y co-ordinates, quantized slope values and dominant point co-ordinates.

The authors in [6] consider a subspace based classification approach to online Tamil symbols. Principal component analysis (PCA) is applied separately to feature vectors extracted from the training samples of each class. The subspace formed by the first few eigenvectors is used to model a class. In the testing phase, a test sample is projected onto each subspace and the class corresponding to the one that gives the minimum reconstruction error is assigned as the recognition result.

Different strategies have been investigated in [7] for selecting prototype samples for recognizing online handwritten characters. In order to model the differences in complexity of different character classes, a prototype set growing algorithm is used

with DTW+NN as the classifier. A set of offline-like features, describing the positional and structural (shape) characteristics of the handwritten unit, have been employed in [8]. The resulting spatio-temporal based features are fed to train a SVM for classification. Stroke based recognition of Tamil symbols has been attempted in [9]. Here, unique strokes in the script are manually identified and each stroke is represented as a string of shape features. By using a string matching algorithm, the test stroke is compared against the database of such strings. The sequence of stroke labels obtained are then concatenated to a character by using a finite state automaton (FSA).

1.1 Problem Definition

A systematic assessment of the classifier performances reported in the literature attribute most of the misclassifications/errors to the presence of symbols that appear visually similar, which confuse the classifier. Each of the classifiers work on features at the global level, and so, at times, fail to capture finer nuances that distinguish these symbols. One way to alleviate this drawback is to incorporate experts that employ class-specific features to reduce the degree of confusion between frequently confused characters. Specifically, the article proposes techniques for reevaluating the recognition output from the first level classifier (hereinafter referred to as 'primary classifier').

Human vision can automatically locate the distinct regions in confused symbol pairs so as to distinguish one from the other. For the handwriting system to mimic this remarkable ability, we propose a dynamic time warping (DTW) approach for learning the finer nuances that discriminate confused symbols. The developed technique aids in extracting the relevant part of strokes for deriving class-specific features.

Literature has many proposals to deal with the problem of reducing the confusions between similar characters in non-Indic scripts. A two stage classification strategy has been adopted [10] for Latin script recognition. At the first level, confusions between characters (referred to as 'conflicts') are detected using an ensemble of classifiers. To resolve the conflicts, two different architectures of support vector classifiers are introduced at the second level as verifiers. Hybrid MLP-SVM structures have been used [11] for recognizing handwritten digits. Specialized SVMs are developed to operate on the two highest MLP outputs at the second level to generate the correct class. This work assumes that the correct class almost consistently occurs within the top two recognized digits from the MLP classifier. A similar approach has been presented by [12], wherein a model based Bayesian classifier is employed at the first stage to generate the two most probable classes for the input character. At the second stage, a discriminative classifier (probabilistic neural network) is used to reduce the confusion between the two ambiguous classes obtained from the first level. For Persian script, fine classification of unconstrained handwritten numerals has been achieved by removing confusions between similar looking classes at the second level [13].

Reverting to the context of online Indic scripts, there is hardly any comprehensive work that addresses the problem of disambiguating confused characters. Most reported techniques deal with the problem of recognizing isolated characters in a

single stage. However, in the area of optical character recognition, post-processing schemes have been successfully attempted for a few scripts. Shape encoding based post-processing methods have been used for improving the Gurmukhi OCR system [14]. In addition, a lexicon look-up strategy based on bigram analysis has been proposed by Lehal [15]. Sub-character level language modeling techniques have been used as a post-processing step to correct Malayalam words [16]. OCR errors in Bangla have been rectified with morphological parsing techniques [17].

The rest of the article is organized as follows. Section 2 presents the details of the script and defines the symbol set used in this work. The design of the first level (primary) classifier is elaborated in Section 3. The motivation/need for reevaluating the output of the primary classifier is elaborated in Section 4. The proposed reevaluation strategies are described with sufficient details in the Sections 5, 6, 7, 8 and 9. Section 10 presents the performance of the strategies on two databases. In Section 11, we summarize our work.

2 Properties of Tamil script

Tamil language consists of 12 pure vowels, 23 pure consonants and two special characters (ஃ /ah/, symbol ஸ்ரீ /sri/). In addition, the vowels and the pure consonants combine to generate $23 \times 12 = 276$ CV combinations. Thus effectively, the complete character set consists of 276 CV combinations, 12 vowels, 23 pure consonants and 2 special characters. These result in a total of 313 characters (listed in Appendix A).

Analysis of the complete set of CV combinations indicates that they may appear in one of the following five forms:

- Pure consonants modified by the inherent vowel அ /a/ are called 'base consonants'. A few examples of base consonants include க /ka/, ச /ca/ and ள /la/.
- In the CV combinations of இ /i/ and ஈ /I/, the vowel modifier (VM) overlaps with the base consonant. Examples of such CV combinations include கி /ki/, சீ /cI/, ழி /zhi/ and ளீ /LI/.
- In the CV combinations of உ /u/ and ஊ /U/, the basic shape of eighteen (out of the 23) base consonants are altered due to the vowel addition. Examples of such CV combinations include பு /pu/, ழு /zhu/, கு /ku/ and சூ /cU/. However, for CV combinations corresponding to the remaining five consonants, the shape of the base consonant is unaltered with the discrete vowel modifier overlapping with it on top. Typical examples are ஸு /su/, க்ஷு /kshu/, ஸூ /sU/ and ஹூ /hU/.
- In the CV combinations of எ /e/, ஏ /E/ and ஐ /ai/, the corresponding vowel modifiers ெ (VM of /e/), ே (VM of /E/) and ை (VM of /ai/) precede the base consonant being modified. Examples of such CV combinations include நெ /ne/, யே /yE/ and னை /kai/.
- In the CV combinations of ஆ /A/, the vowel modifier written as ா, follows the base consonant being modified. Examples include காbe /kA/, தா /tA/ and யா /yA/.

– CV combinations of ஒ /o/, ஓ /O/ and ஔ /au/ consist of two distinct entities with the base consonant sandwiched between them. Examples of such combinations are பொ /po/ , டோ /TO/ and கௌ /kau/.

Unlike other Indic languages, there are no compound characters (wherein two consonants get merged to a single symbol) in Tamil.

2.1 Selection of the Tamil symbol set for recognition

Inspection of the 313 characters indicates redundancy, especially with respect to the way certain CV combinations are written [18]. In this subsection, we present the methodology adopted to reduce the redundancy, with the aim of coming up with a comprehensive set of distinct entities that can be employed in designing the recognition system.

– As an illustration, we consider all the CV combinations of vowel எ /E/. In this case, the vowel modifier ே (VM of /E/) appears as a distinct/separate entity to the left of the base consonant being modified. From the point of view of recognition, it is sufficient to recognize the symbol ே (VM of /E/) separately and then append it to the corresponding base consonant to generate the CV combination, thereby reducing the number of distinct entities for the classifier.
– Similar strategies applied on the vowel modifiers of ஆ /A/, ஏ /e/, ஐ /ai/, ஒ /o/, ஓ /O/ and ஔ /au/ reduce the inherent redundancy in the characters to a substantial extent.
– In addition, the vowel ஔ /au/ consists of 2 distinct entities- ஒ /o/ and ள /La/ that have already been considered as a vowel and a base consonant, respectively. Hence, there arises no necessity for representing it as a separate entity for recognition.

Based on the above analysis, we find that a set of 155 distinct entities (henceforth referred to in this work as 'symbols') is sufficient to form (and hence recognize) all the 313 characters in the Tamil alphabet. The distinct set consists of:

1. 11 pure vowels (excluding ஔ /au/)
2. 23 pure consonants
3. 23 base consonants
4. 23 CV combinations of இ /i/
5. 23 CV combinations of ஈ /I/
6. 23 CV combinations of உ /u/
7. 23 CV combinations of ஊ /U/
8. 6 Additional symbols (ா (VM of /A/) , ி (VM of /e/) , ே (VM of /E/), ை (VM of /ai/), ஃ /ah/ and ஸ்ரீ /sri/.)

3 Design of the Primary SVM classifier

In this section, we present the details of the primary recognition framework used in our experiments. The recognizer has been designed to classify isolated symbols from online handwritten Tamil data. In the International Workshop on Handwriting Recognition (IWFHR) 2006, HP Labs India released a corpus comprising

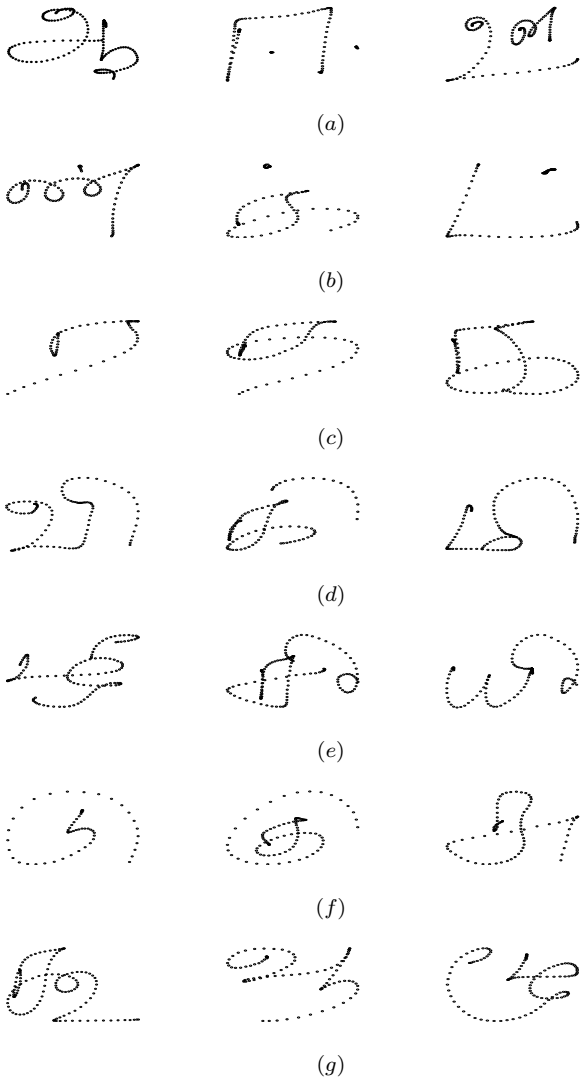


Fig. 1 Online handwritten samples of a few Tamil symbols from the IWFHR database. (a) Vowels. (b) Pure consonants. (c) Base Consonants. (d) CV combinations of `/i/`. (e) CV combinations of `/I/`. (f) CV combinations of `/u/`. (g) CV combinations of `/U/`.

isolated online Tamil symbols for research [19]. This database, (hereinafter referred to as IWFHR Database) comprises 50,385 training samples and 26,926 test samples. The entire training data from this database has been used to train the primary classifier. Figure 1 presents some samples from the IWFHR database.

As far as the choice of the primary classifier is concerned, we prefer the one that gives a good generalization performance on data not seen during training. Amongst the various classifiers discussed in the literature for online Tamil script recognition, the SVM qualifies to be apt in this respect. This classifier works on

features that have a fixed dimension. In the following subsections, we provide an outline of the feature vector used in the training step and the details of the primary classifier.

3.1 Feature description

The online handwritten symbol, captured from the digitizer, is a sequence of x-y coordinates with pen-up and pen-down events. The pre-processing [5,6] step, applied prior to recognition, compensates for variations in time, scale and velocity. It comprises 3 steps : (1) smoothing (2) normalization (3) resampling. The final result of pre-processing is a new sequence of points $\{x_i, y_i\}_{i=1}^{n_P}$ regularly spaced in arc length. A feature vector is constructed from this sequence as

$$\mathbf{x} = (x_1, x_2 .... x_{n_P}, y_1, y_2, ... y_{n_P}) \tag{1}$$

The vector $\mathbf{x}$ is referred as the 'concatenated x-y coordinates' in this work. We experimented with varying number of resampled points and observed that $n_P = 60$ is quite sufficient in capturing the shape of the character including points of high curvature.

3.2 Support Vector Machines

The SVM classifier [20] is a supervised method used for two-class pattern classification problems. Suppose a training data set comprises pairs $\{(\mathbf{x}_i, l_i), 1 \leq i \leq N_{Tr}\}$, where each input vector $\mathbf{x}_i \in \Re^d$ is assigned to l_i. The value of l_i corresponds to one of the binary labels $\{-1, +1\}$. The SVM minimizes the cost function

$$J(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} \tag{2}$$

subject to the constraints

$$l_i(\mathbf{x}_i.\mathbf{w} + b) \geq +1 \tag{3}$$

Here $\mathbf{w}$ is the weight vector and b is the bias term. The above equations apply to the scenario where training samples are linearly separable. Whenever the classes to be recognized are not linearly separable, the cost function is reformulated by introducing slack variables $\xi_i \geq 0 \quad i = 1, 2, ...N_{Tr}$. The SVM now finds $\mathbf{w}$ to minimize

$$J(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N_{Tr}} \xi_i \tag{4}$$

subject to

$$l_i(\mathbf{x}_i.\mathbf{w} + b) \geq +1 - \xi_i \tag{5}$$

The constant C is a regularization parameter. When the decision function is non-linear, the above scheme cannot be used directly. For such cases, the SVM maps the training data from $\Re^d$ to a higher dimensional feature space H, via a mapping function $\phi : \Re^d \rightarrow H$. In this feature space H, the data may be linearly separable. In practice, the so-called 'kernel-trick' is used wherein, a kernel defined by

$K(\mathbf{x}, \mathbf{x}_i) = \phi(\mathbf{x})\phi(\mathbf{x}_i)$ is used to construct the optimal hyperplane in H without considering the mapping function $\phi(\mathbf{x})$ explicitly. For our work, we have used the Radial Basis Function (RBF) kernel defined as

$$K(\mathbf{x}, \mathbf{x}_i) = exp(-\gamma \|\mathbf{x} - \mathbf{x}_i\|^2) \quad \gamma \geq 0 \tag{6}$$

SVMs for multi-class recognition problems are realized by combining several two-class SVMs [21] by the one-versus-one (OVO) technique. Here, for a c-class problem, $c(c-1)/2$ two-class SVMs are constructed. A two-class SVM $C_{ij}, i < j$ is trained using samples from classes i and j, containing positive and negative samples, respectively. Whenever the decision function value for a test sample is positive from C_{ij}, the vote for class i is incremented by one. Otherwise, the vote for class j is increased by one. The sample is assigned to the class with the maximum number of votes. The concatenated x-y features $\mathbf{x}$ (refer Eqn 1) are fed as input to the SVM classifier.

The performance of the SVM classifier is largely dependent on the selection of the parameters. The samples corresponding to the 155 symbols in the IWFHR training set are employed to obtain the model parameters. The RBF function (defined in Equation 6) is used as the kernel in our experimentation. We have employed the LIB-SVM software [22] for learning the SVM model parameters. A recognition performance of 86% is achieved on the IWFHR test set with parameters $C = 5$ and γ=0.2. The kernel and the corresponding parameters are optimally set after performing five-fold cross validation experiments on the IWFHR training data.

4 Need for reevaluation strategies

While considering the need to reevaluate a Tamil symbol, two aspects are taken into account.

- Its frequency of occurrence in a large Tamil text corpus.
- The extent to which it gets confused with another symbol by the primary classifier. The confusions are analyzed from the confusion matrix. This matrix (denoted by $\mathbf{C}$) is constructed by noting the recognition performance of the primary classifier on each of the 155 symbols contained in the testing set of the IWFHR dataset.

An extensive Unicode text corpus, comprising 1.5 million Tamil words (derived from books), was utilized for generating the frequency count of each of the 155 symbols. The corpus essentially is a collection of sentences, wherein each word comprises a sequence of Tamil characters. The Unicode corpus was first transformed to a corpus of Tamil symbols, by inverse mapping from the Unicode sequence of Tamil compound characters to the corresponding symbol sequences. This is essential, since the symbol order and order of the Unicodes are different for different sets of CV combinations as shown in Table 1.

We consider the statistics of the symbols obtained from this corpus to be representative of the script. Based on linguistic similarity, the symbols are divided into 8 groups. Table 2 lists the occurrence frequency of the groups in the corpus.

We observe that base consonants (G_1) alone constitute 33% of the total corpus. In addition, base consonants occur as separate strokes in pure consonants (G_2) ,

Table 1 CV combinations illustrating the differences between the symbol order and the order of the Unicode representations

Vowel forming the CV	Symbol writing order	Unicode order
ஆ /A/	BC+ா	BC+ ா
எ /e/	ெ +BC	BC+ ெ
ஏ /E/	ே + BC	BC+ ே
ஐ /ai/	ை +BC	BC+ ை
ஒ /o/	ெ + BC+ா	BC+ ெ....ா
ஓ /O/	ே + BC+ா	BC+ ே....ா
ஔ /au/	ௌ + BC+ள	BC+ ெ....ள

Table 2 Occurrence statistics of different groups of Tamil symbols, as derived from the MILE text corpus (containing 1.5 million words)

Group	Description	# of symbols	% of symbols
G_1	Base consonants	368387	33.5
G_2	Pure consonants	266525	24.2
G_3	Additional symbols	191282	17.4
G_4	CV combinations of உ /u/	104360	9.6
G_5	CV combinations of இ /i/	99421	9.1
G_6	Pure vowels	57858	5.3
G_7	CV combinations of ி /I/	6252	0.6
G_8	CV combinations of ுள /U/	5105	0.4

CV combinations of இ /i/ (G_5) and ி /I/ vowels (G_7). For multi-stroke handwritten symbols in groups G_2, G_5 and G_7, the base consonant can be extracted by employing spatial cues derived from the strokes. For illustration, consider the CV combinations தி /ti/, தீ /tI/ and the pure consonant த் /t/. From each of these three symbols, we can easily extract the base consonant (BC) த /ta/. Thus, effectively the occurrence of base consonants in the script is much higher than the percentage denoted by G_1 alone. In fact, considering across the groups G_1, G_2, G_5 and G_7, base consonants can be extracted as an independent entity in 67.4% (33.5% +24.2% +9.1%+ 0.6%) of the symbols in the corpus.

Moreover, we analysed the confusion matrix **C** and noted that a few pairs of consonants like (ல /la/, ள /va/) and (ள /La/, ன /Na/) get confused by the primary classifier in 4 to 6.5% of the cases (Table 3). Due to the higher percentage of base consonants and possible confusions, it becomes imperative to reevaluate

- base consonants in CV combinations of இ /i/ and ி /I/.
- base consonants in pure consonants.
- the frequently confused base consonants.

As discussed in Sec 2, the inherent vowel sound of a base consonant is suppressed by the dot, resulting in a pure consonant. Pure consonants (G_2) account for 24% of the symbols in the MILE text corpus. However, the size of the dot varies with the style of writing. From the matrix **C**, we found that the primary classifier at times interprets the dots to be the vowel modifiers (VM) of இ /i/ or ி /I/ and vice versa, thereby resulting in an erroneous symbol. In addition, confusions arise between the VM of இ /i/ and ி /I/ in their corresponding CV

Table 3 Some symbol confusions encountered at the output of the primary SVM classifier and their frequency of occurrence in the IWFHR 2006 Tamil test symbol set.

Symbol pairs	Total # of symbols	# of confusions	Primary classifier accuracy in %
(பூ ,பூ) (/mu/, /zhu/)	349	26	92.6
(னா ,ணா) (/Na/, VM of /ai/)	351	32	90.9
(னி ,ணி) (/Ni/, /Li/)	364	32	91.2
(ளா ,னா) (/La/, /Na/)	353	23	93.5
(கி ,சி) (/ki/, /ci/)	355	17	95.2
(ல ,வ) (la, va)	359	14	96.1

combinations G_5 and G_7 (that account for 9.7% of the symbols in the corpus). Accordingly, we reevaluate

- vowel modifier strokes in test samples assigned to CV combinations of ◌ி /i/ and ◌ீ /I/ by the primary classifier.
- dot strokes in test samples assigned to pure consonants by the primary classifier.

Amongst the remaining symbols, confusions arise between (பூ /mu/,புூ /zhu/), (ளா /La/, னா /Na/, ணா (VM of /ai/)) and (கூ /ka/,சூ /cu/). Class-specific features derived from the discriminative regions of these symbol sets help in their disambiguation. Table 3 lists a few of the similar looking pairs with their frequencies of confusion and their recognition accuracies from the primary SVM classifier.
Given the confusion matrix $\mathbf{C}$ of size 155×155,

$$\mathbf{C} = \begin{pmatrix} c_{1,1} & c_{1,2} \ldots \ldots & c_{1,155} \\ c_{2,1} & \ldots \ldots & c_{2,155} \\ .. & & \\ .. & & \\ c_{155,1} & \ldots \ldots & c_{155,155} \end{pmatrix}$$

let $c_{i,j}$ represent the number of samples of symbol ω_i getting wrongly classified as ω_j. The number of confusions for a symbol pair (ω_i, ω_j) can be written as $c_T(i,j) = c_{i,j} + c_{j,i}$. For a symbol ω_i, the set of symbols to which it can get frequently confused by the primary classifier is represented by $\Omega_i = \{\omega_j | c_T(i,j) \geq \delta, i \neq j\}$. In this work, we have chosen $\delta = 10$. This implies that we consider only confusion sets having a frequency above 3% in the confusion matrix. We denote the set of all symbols that possibly can get confused, and hence need to be reevaluated as

$$\Omega = \bigcup_i \Omega_i \tag{7}$$

Motivated by the observations outlined above, the present work improves on the recognition accuracy of the primary classifier by proposing reevaluation strate-

gies for resolving any possible ambiguities in base consonants, pure consonants, vowel modifiers and frequently occurring confusion symbol pairs.

5 Overview of proposed reevaluation strategy

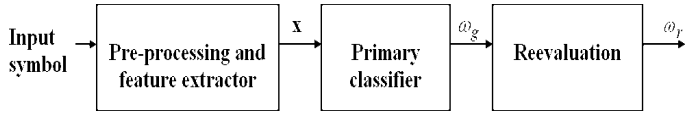


Fig. 2 Block diagram of the recognition strategy for an input Tamil symbol.

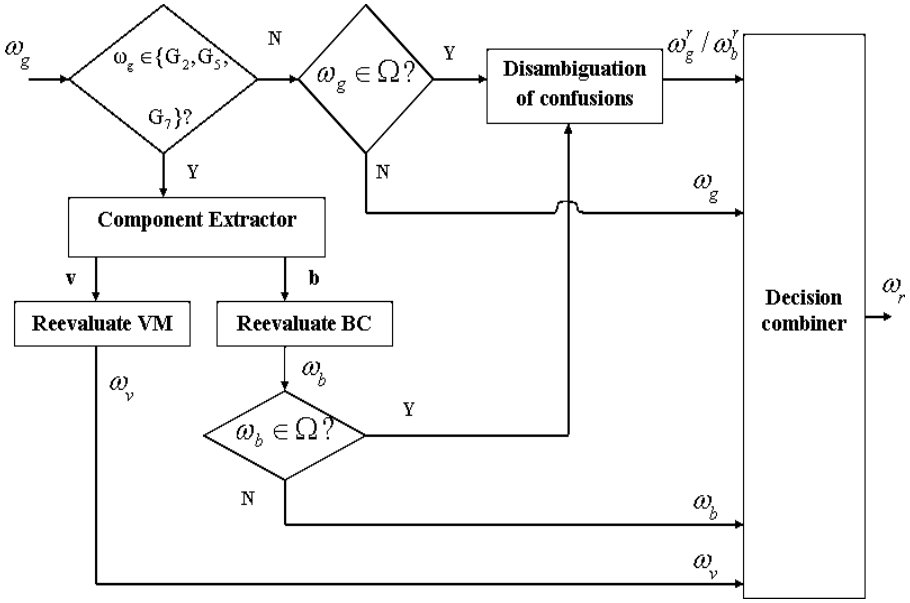


Fig. 3 Details of the proposed reevaluation block. G_2: Pure consonant group; G_5: CV combinations of `/i/`; G_7: CV combinations of `/I/`, Ω: Set of all confused symbols; **b**, **v**: extracted base consonant and vowel modifier/dot stroke part; ω_g: label given by primary classifier; ω_r: label after reevaluation. $\{\omega_b, \omega_v, \omega_b^r, \omega_g^r\}$: refer Table 4.

Figure 2 presents the overall picture of the proposed recognition strategy for a Tamil symbol. The trace of the symbol is preprocessed [6] and the resulting concatenated x-y coordinates $\mathbf{x}$ are fed to the primary SVM classifier. The label of the most probable symbol is denoted by ω_g. Based on ω_g, multiple novel reevaluation strategies are proposed to reduce the chances for the misclassification of the symbol. The reevaluation block in Fig. 2 is expanded in Fig. 3 and discussed below.

1. When the primary classifier outputs a pure consonant or CV combination of ಇ /i/ or ಿ /I/ vowel as its most probable symbol ($\omega_g \in \{G_2, G_5, G_7\}$), we separately extract the base consonant (BC) and vowel modifier (VM)/dot with the component extractor and derive new discriminative features for reevaluating them. Let ω_b and ω_v represent the independently reevaluated labels for the base consonant (BC) and vowel modifier (VM). Furthermore, if the base consonant ω_b is likely to be confused with another base consonant (in other words, $\omega_b \in \Omega$), we subject it to a second round of reevaluation by disambiguating it from its possible confusions.
2. If $\omega_g \in \Omega$, class-specific discriminative features are derived from the preprocessed symbol. The reevaluation strategy is achieved using appropriate expert classifiers, each of which is designed to disambiguate a specific confusion set.

The decision combiner finally combines the various labels to generate the appropriate output symbol ω_r (see Table 4).

Table 4 Logic for generation of the final label ω_r for the recognized symbol in the decision combiner module in Fig. 3.

Logic to get label ω_r	Constraints
ω_g	$\omega_g \notin \{G_2, G_5, G_7\}$, $\omega_g \notin \Omega$
ω_g^r	$\omega_g \notin \{G_2, G_5, G_7\}$, $\omega_g \in \Omega$
CV combination generated by appending ω_v to ω_b	$\omega_g \in \{G_2, G_5, G_7\}$, $\omega_b \notin \Omega$
CV combination generated by appending ω_v to ω_b^r	$\omega_g \in \{G_2, G_5, G_7\}$, $\omega_b \in \Omega$

6 Reevaluation of base consonants

Consider a preprocessed m-stroke ($m > 1$) handwritten symbol recognized as a CV combination of ಇ /i/ (G_5) or ಿ /I/ (G_7). The component extractor module separates the BC from VM by employing the maximum vertical inter-stroke gap h_{max} (derived from the symbol). Let h_{max} correspond to the spacing between the r^{th} and $(r+1)^{th}$ strokes. Accordingly, the first r strokes, assumed to comprise n_B sample points denotes the trace of the BC and is represented by $\mathbf{b}$. The remaining $(m-r)$ strokes represent $\mathbf{v}$, the trace of the VM. As mentioned in Sec 3, the number of resampled points in the preprocessed symbol, $n_P = 60$ in our experiments.

$$\mathbf{b} = \{x_i, y_i\}_{i=1}^{n_B} \tag{8}$$

$$\mathbf{v} = \{x_i, y_i\}_{i=n_B+1}^{n_P} \tag{9}$$

A similar approach is employed to extract the dot and the base consonant from a pure consonant (G_2). For ease of notation, we denote the $(m-r)$ strokes representing the dot in a pure consonant also by $\mathbf{v}$.

The reevaluation module for base consonants (in Fig. 3) is invoked whenever $\omega_g \in \{G_2, G_5, G_7\}$. For illustrating the proposed strategy, assume that the most probable output of the primary classifier ω_g for the input pattern is a CV combination of ಇ /i/ vowel (G_5). The first r strokes of the raw input data, representing

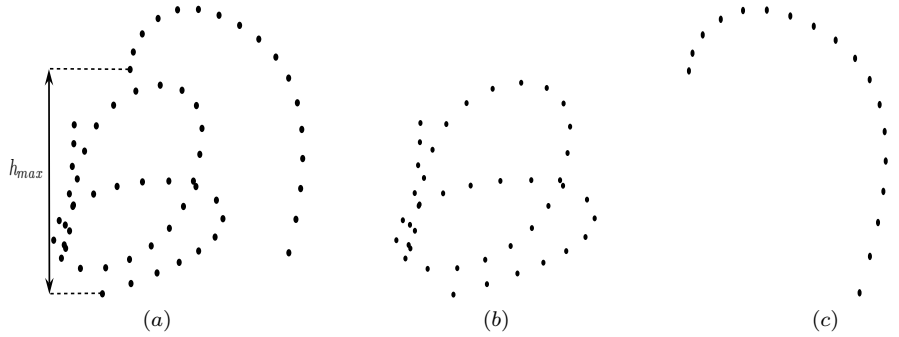


Fig. 4 Extraction of the base consonant and vowel modifier from the CV combination /ki/. (a) CV combination. (b) Base consonant. (c) Vowel modifier.

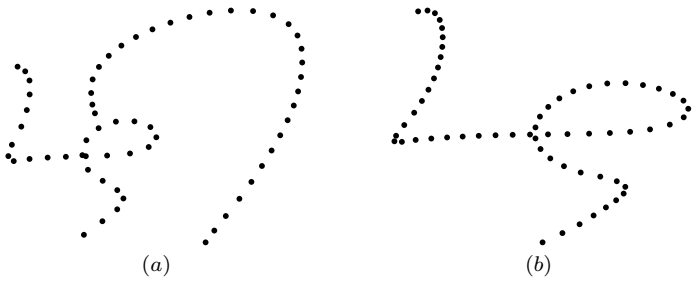


Fig. 5 Illustration of base consonant reevaluation. (a) This symbol, which is `/zhi/`, is wrongly recognized as `/mi/` by the primary classifier. (b) The preprocessed pattern of the extracted base consonant is recognized by classifier C_b as `/zha/`.

the trace of the extracted BC, is sent to the preprocessing module [6]. The resulting feature vector (concatenated x-y features) $\mathbf{x}_b$ is separately fed to the SVM classifier C_b dedicated to recognize only the base consonants. Compared to the primary SVM classifier that is trained across the 155 Tamil symbols of the IWFHR database, classifier C_b is trained using the samples of the 23 base consonants only. Let ω_b be the base consonant label obtained from the reevaluation module. The most probable consonant from the classifier C_b is regarded as the reevaluated label and is assigned to ω_b.

Figure 5 presents the scenario wherein the primary classifier regards the pattern in (a) as மி `/mi/`. However, the classifier C_b assigns the extracted base consonant pattern shown in (b) to ஹ `/zha/` (which happens to be the correct symbol). Hence, the pattern after reevaluation is assigned to ஜி `/zhi/`, provided the reevaluated vowel modifier corresponds to இ `/i/`.

A similar analysis (as described above) is applied to reevaluate the base consonants in CV combinations of vowel ஃ `/I/` and pure consonants.

7 Reevaluation of dots and vowel modifier strokes

In this section, we propose strategies to reevaluate the pattern $\mathbf{v}$ obtained from the component extractor. We adopt a two step process as outlined below

- We first disambiguate the dot stroke from the modifiers of इ /i/ or ि /I/ vowel (Sec 7.1).
- If $\mathbf{v}$ is not a dot stroke, we reevaluate the modifiers of इ /i/ and ि /I/ vowels (Sec 7.2).

Let ω_v correspond to the label of the VM after reevaluation.

7.1 Recognition of dots in pure consonants

In this subsection, we propose strategies to detect the cases of the primary classifier confusing the dot in a pure consonant (G_2) with the vowel modifier in a CV combination (G_5 or G_7). It is assumed here that the primary classifier returns the VM of इ /i/ or ि /I/ vowel for $\mathbf{v}$. Based on a detailed statistical analysis of the dot strokes and vowel modifiers of इ /i/ and ि /I/ in the training set of the IWFHR database, we come up with the two following conditions, one of which the dot stroke definitely satisfies.

(i) **Net distance covered**: In contrast to the vowel modifiers of इ /i/ and ि /I/, the ratio of the Euclidean distance between the first and last points to the arc length is generally small for the dot strokes in pure consonants. This fact is captured by

$$\frac{d^v_{fl}}{l^v_T} \leq T^d_r \tag{10}$$

Here d^v_{fl} is the Euclidean distance between the first and last sample points in $\mathbf{v}$. l^v_T is the total arc length traversed along the trace. The threshold T^d_r is set to the minimum possible ratio of d^v_{fl} to l^v_T across all modifiers of vowels इ /i/ and ि /I/.

(ii) **Relative number of sample points**: In contrast to the vowel modifiers of इ /i/ and ि /I/, the number of sample points representing the dot strokes in pure consonants is usually less.

$$\mathbf{v}_{\#} < T^d_{\#} \tag{11}$$

Here, $\mathbf{v}_{\#}$ corresponds to the number of sample points in the pattern $\mathbf{v}$. From Eqn 9, we have:

$$\mathbf{v}_{\#} = n_P - n_B \tag{12}$$

The value of the threshold $T^d_{\#}$ corresponds to the minimum number of sample points representing the vowel modifiers of इ /i/ and ि /I/ in the IWFHR training data-set.

From the statistics of IWFHR training dataset, we obtain $T^d_{\#} = 7$ and $T^d_r = 0.1$.

Figure 6 illustrates the case wherein the primary classifier wrongly assigns the dot stroke to CV combination of ि /I/. However, on reevaluating the trace of the VM $\mathbf{v}$, we assign it to the dot stroke.

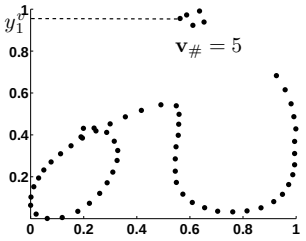


Fig. 6 Identification of a given stroke $\mathbf{v}$ as a dot. The primary classifier interprets the VM stroke as vowel modifier of /I/. However, the pattern $\mathbf{v}$ satisfies $\mathbf{v}_{\#} < 7$. Thus, on reevaluation, $\mathbf{v}$ is assigned the label of dot.

7.2 Reevaluation of /i/ and /I/ vowel modifiers

In this subsection, we propose the strategy for reevaluating the vowel modifiers ി (VM of /i/) and ീ (VM of /I/). Figure 7 illustrates the proposed methodology. Preprocessed x-y coordinates of the samples of vowel modifiers (in the CV combinations of ഇ /i/ and ഈ /I/) are used to train a SVM (denoted by C_m). The trace of the vowel modifier $\mathbf{v}$ (obtained from the component extractor) is assigned to ീ (the vowel modifier of ഈ /I/) whenever at least one of the following two conditions holds good.

C1 : SVM C_m favors it as the most likely vowel modifier

C2 : The relative horizontal distance between the last sample point x_l^v of the trace of the vowel modifier $\mathbf{v}$ to the global x-maximum is greater than a threshold.

$$\frac{x_{M,g}^v - x_l^v}{x_{M,g}^v - x_{y_{Mg}}^v} > T_o^v \tag{13}$$

Here $x_{M,g}^v$ and x_l^v are the global x-maximum and x-coordinate of the last sample point of $\mathbf{v}$, respectively. $x_{y_{Mg}}^v$ represents the x-coordinate corresponding to the global y-maximum of $\mathbf{v}$. Whenever neither of the conditions are satisfied, we favor the vowel modifier of ഇ /i/. From experimental validation, we see that the threshold T_o^v set to 0.2 is quite robust in discriminating ീ (VM of /I/) from ി (VM of /i/).

For the pattern in Fig. 7 (a), recognized as കീ /kI/, the conditions **C1** and **C2** do not hold good for the stroke $\mathbf{v}$ (shown in (b)). Hence, we assign it to കി /ki/ after reevaluation.

8 Disambiguation of confused symbols

Visual inspection of confusions between symbols, arising from the primary classifier, indicates that they share common structures and are just different in some critical parts of the trace. As an example, we observe that the symbols ല /la/ and വ /va/ differ primarily in the middle of the trace. The confusion pair ക /ka/ and ച /cu/ present structural differences at the end of the trace. In this section, we aim to reduce the degree of confusions between such frequently confused characters, thereby improving the overall performance, beyond that given by the primary SVM classifier alone.

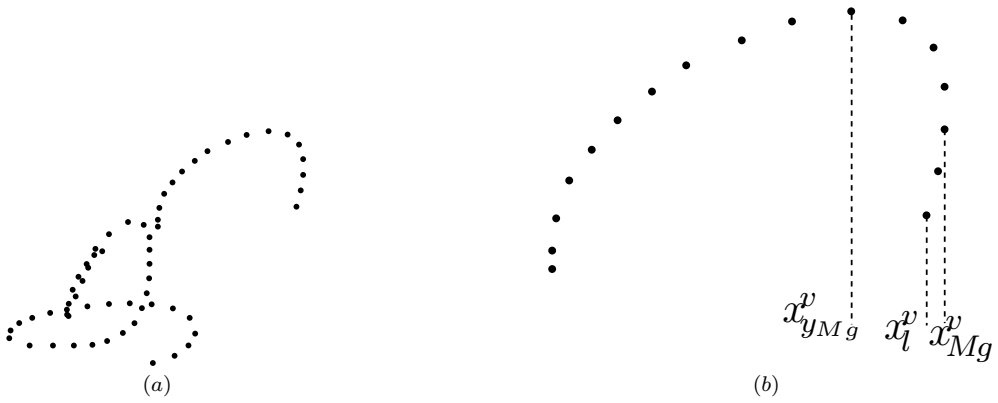


Fig. 7 Illustration of reevaluation of the vowel modifier **v** in CV combinations of `/i/` and `/I/`. (a) This symbol, which is `/ki/`, is wrongly recognized as `/kI/` by the primary classifier. However, it is corrected by reevaluation. (b) Extracted VM stroke with the derived features.

8.1 Use of expert classifiers

Figure 8 presents the block diagram of the strategy proposed to disambiguate the frequently confused symbols. Independent expert networks are designed for each confusion set. Each expert comprises 3 blocks, namely, discriminative region extractor, feature extractor and SVM classifier. For each confusion pair of symbols $(c1, c2)$, the corresponding expert extracts the specific discriminative region (DR) from the input symbol pattern. The discriminative region (denoted by $\Re(c1, c2)$) corresponds to the part of trace containing the finer nuances of structures in $c1$ and $c2$. A set of discriminative features is then derived from the DR $\Re(c1, c2)$ by the feature extractor module. The i^{th} pair-specific feature from $\Re(c1, c2)$ is denoted by $f_i^{(c1,c2)}$. After extracting a set of features for sufficient discrimination of $(c1, c2)$, the SVM classifier is used for the disambiguation.

8.2 Dynamic time warping for automated identification of discriminative regions in confused pairs

The first key step in the proposed methodology is to automatically locate the distinctive parts of strokes in confused pairs. For offline handwriting recognition, techniques have been proposed to extract from images the distinctive regions relevant for classification in the second level [23, 24]. In our work, temporal information of the trace is exploited to propose a dynamic time warping (DTW) approach for learning the finer parts that distinguish the confused symbols.

Let $(c1, c2)$ represent a confusion pair. We employ the training patterns of $c1$ and $c2$ in the IWFHR symbol set to learn the discriminative parts of the online trace. Let N_{Tr}^{c1} and N_{Tr}^{c2} denote the number of training data for $c1$ and $c2$, respectively. Each such pattern is a temporal sequence that can be described (after preprocessing) by $\{(x_1, y_1), (x_2, y_2), ..., (x_{n_P}, y_{n_P})\}$. Dynamic time warping (DTW) is an elastic matching technique used for aligning two temporal sequences. (For more details on the DTW algorithm, we refer the reader to [5]). Our focus here

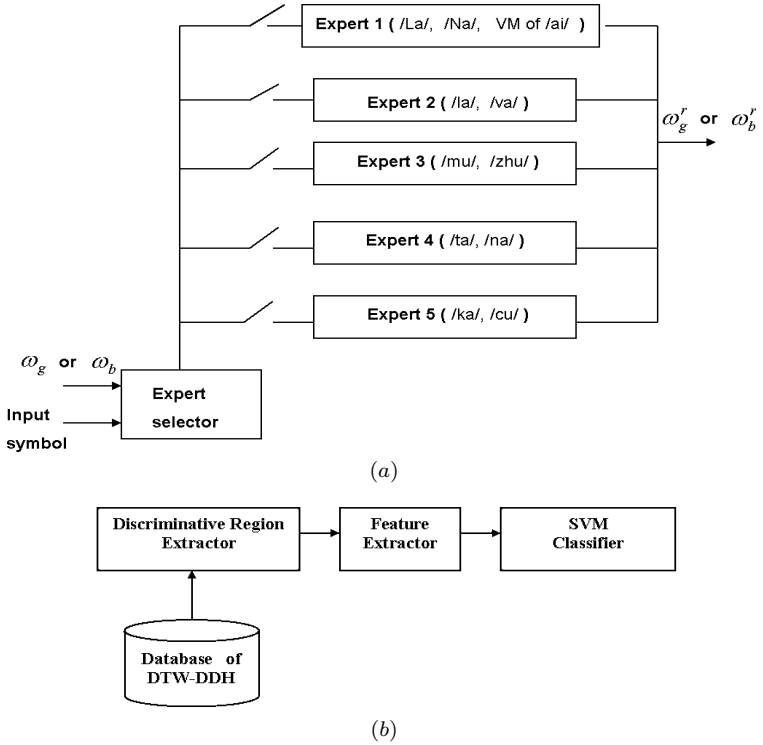


Fig. 8 Resolving confusions by expert classifiers. (a) Block diagram of the proposed disambiguation strategy. Experts 1 to 5 operate on disambiguating the confused sets of (`/La/`, `/Na/`, `/ai/` vowel modifier), (`/la/`, `/va/`), (`/mu/`, `/zhu/`), (`/ta/`,`/na/`) and (`/ka/`, `/cu/`), respectively. (b) Component blocks of an expert.

is to learn the structural differences between the traces of the confused pairs $c1$ and $c2$. For this, we employ a DTW approach to match each temporal sequence in the training set N_{Tr}^{c1} against all the sequences in N_{Tr}^{c2}. This results in a set of $N_{Tr}^{c1} \times N_{Tr}^{c2}$ different cost matrices with corresponding optimal warping paths.

The entries of the cost matrix measure the degree of dissimilarity between the features being considered. Let (x_k^{c1}, y_k^{c1}) and (x_l^{c2}, y_l^{c2}) respectively denote the k^{th} and l^{th} point in the confusion pair $c1$ and $c2$. The (k, l) element in our cost matrix is the Euclidean distance measure $d(k, l)$ between the points (x_k^{c1}, y_k^{c1}) and (x_l^{c2}, y_l^{c2}) and can be calculated as follows:

$$d(k,l) = \sqrt{(x_k^{c1} - x_l^{c2})^2 + (y_k^{c1} - y_l^{c2})^2} \tag{14}$$

We note that the optimal paths $\mathcal{W}^*$ in each of the $N_{Tr}^{c1} \times N_{Tr}^{c2}$ cost matrices is made up of some sections with low values of $d(k, l)$ corresponding to similar regions in the confused pair of symbols and other section or sections with high values of $d(i, j)$ corresponding to the part or regions in the symbol pair that are very distinct. We utilize this property to select the discriminative regions of confused symbol pairs as described below:

We generate a histogram that accumulates the pen positions contributing to

the structural differences in the confused pairs $(c1, c2)$. This is referred to as the 'DTW discriminative distance histogram' (DTW-DDH). We now outline the pseudocode for obtaining the DTW-DDH.

Let $(c1, c2)$ be a confused symbol pair.

N_{Tr}^{c1}= No. of training samples of $c1$ in the IWFHR Tamil symbol set.

N_{Tr}^{c2}= No. of training samples of $c2$ in the IWFHR Tamil symbol set.

Initialize a histogram meant to capture the pen positions corresponding to the structural differences in the pair $(c1, c2)$. In other words, set the votes for each of the n_P sample indices to zero.

for $i = 1 : N_{Tr}^{c1}$
for $j = 1 : N_{Tr}^{c2}$

Match the temporal sequence of i^{th} training pattern of $c1$ to the j^{th} training pattern of $c2$ by computing the cost matrix. The cost matrix is generated by evaluating the Equation 14 between each pair of sample indices of i^{th} training pattern of c1 and the j^{th} training pattern of c2

Retrace the optimal DTW path.

Record the maximum dissimilarity $d_{max}^{i,j}$ cost along the warping path.

Increment the votes of the histogram for each sample index of the trace, where dissimilarity exceeds a threshold T_d. The threshold is adapted to the maximum dissimilarity $d_{max}^{i,j}$ measure obtained between the i^{th} training pattern of $c1$ and the j^{th} training pattern of $c2$.

End
End

Pseudocode for obtaining the 'DTW discriminative distance histogram'.

The histogram has n_P bins corresponding to the number of resampled points in

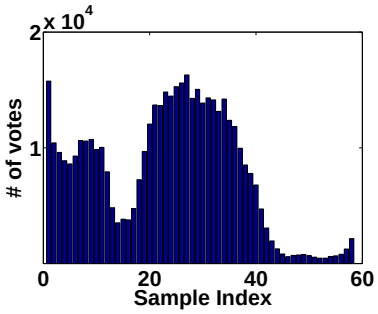


Fig. 9 DTW-DDH corresponding to the symbols /La/ and /Na/ obtained using their samples from IWFHR training set.

any training pattern.

For a given DTW match between any pair of training patterns from the confused symbols $c1$ and $c2$, we first record the maximum dissimilarity cost along the warping path. Thereafter, we compare the remaining costs along the warping path to this maximum dissimilarity cost. The indices (in the warping path) corresponding to costs greater than a threshold T_d are voted in the histogram. This process is repeated for each of the $N_{Tr}^{c1} \times N_{Tr}^{c2}$ DTW matches, thereby resulting in the accumulation of votes for the sample indices. On analyzing the DTW-DDH, we observe that peaks in the histogram correspond to the occurrence of repeated higher costs in the warping paths and hence denote possible regions that could discriminate $c1$ from $c2$.

The threshold T_d is set to 90% of the maximum dissimilarity cost encountered in a particular warping path. We observe that this value is sufficient for identifying the region of finer nuances in the confusion pairs. It is worth clarifying that the value T_d is an adaptive threshold that varies with the maximum dissimilarity cost obtained in each of the $N_{Tr}^{c1} \times N_{Tr}^{c2}$ warping paths. If we denote $d_{max}^{i,j}$ to describe the maximum dissimilarity cost obtained while matching the i^{th} training pattern of $c1$ to the j^{th} training pattern of $c2$, T_d for this pair $= 0.9 \times d_{max}^{i,j}$.

Figure 9 presents the DTW-DDH obtained from the training samples of the confusion set (ಲ /La/, ನ /Na/). The sample index corresponding to the bin having the maximum number of votes gives rise to the maximum peak in the histogram. This is a broad peak. Around this peak, a window of samples is considered to describe the part of trace distinguishing the confusion pair $c1$ and $c2$. This, in turn, forms the discriminative region (DR) $\Re(c1, c2)$.

However, owing to different styles of writing, different transients occur at the start and/or end of the online trace, creating minor spurious peaks at the start and/or end of the DTW-DDH. Such minor peaks, that correspond to the votes of the first and last sample indices in the DTW-DDH, are not included for analysis. From the DTW-DDH of the symbols ಲ /La/ and ನ /Na/, we observe that the main peak occurs in the middle region, thereby indicating that the discriminative region lies in the middle part of the trace. Specific to each $(c1, c2)$, we define an identifiable attention point in $\Re(c1, c2)$, with respect to which the discriminative features are derived. The window of sample points centered around an attention point is referred to as the 'region of attention'.

9 Description of the various experts

We have proposed techniques for disambiguating the following sets of confused symbols on a case-by-case basis [1]. As shown in Fig. 8, each confusion pair is exclusively handled by a dedicated expert. The SVM model file for each expert is learnt by using the training samples from the IWFHR database, corresponding to the confused pair under consideration.

1. (ன /La/, ன /Na/, ன (VM of /ai/))
2. (ல /la/, வ /va/)
3. (மு /mu/, ழ /zhu/)
4. (த /ta/, ந /na/)
5. (க /ka/, ச /cu/)

However,owing to space constraints, we restrict in this article to describe the discriminative features for two confusions namely (ன/La/,ன) /Na/ and (ல/la/,வ/va/).

9.1 Expert 1: Consonants /La/ and /Na/

From Fig. 10(c), the features derived from the middle part of the trace describe the finer nuances in ன /La/ and ன /Na/. The peaks at the start of the trace in DTW-DDH are ignored since they arise due to the variations in writing styles. Accordingly, let

$$\Re(\text{ன},\text{ன}) = \{(x_i, y_i)\}_{i=16}^{45} \tag{15}$$

be the DR selected by the expert 1. From the region of attention around the attention point a_1 in $\Re$(ன/La/,ன/Na/) , corresponding to the first local y-minimum, the following features are defined (see Fig. 10 (d) and (e)).

1. $$f_1^{(\text{ன},\text{ன})} = x_{a_1-1} - x_{a_1+1} \tag{16}$$

   From statistics, we observe that for all samples of ன /Na/, $f_1^{(\text{ன},\text{ன})} > 0$, whereas it is not always true for samples of ன.
2. The angle between successive pen directions at a_1 is used as a feature

   $$f_2^{(\text{ன},\text{ன})} = cos^{-1}\frac{v_1^T v_2}{\|v_1\|\|v_2\|} \tag{17}$$

   where

   $$\begin{aligned} v_1 &= (x_{a_1} - x_{a_1-1}, y_{a_1} - y_{a_1-1}) \\ v_2 &= (x_{a_1+1} - x_{a_1}, y_{a_1+1} - y_{a_1}) \end{aligned} \tag{18}$$

   The values of $f_2^{(\text{ன},\text{ன})}$ are higher for samples of ன than for ன.
3. Consider the region of attention of size 7 centered at a_1. In this region, we compute three distances.

   d_j = dist [(x_{a_1-j}, y_{a_1-j}) (x_{a_1+j}, y_{a_1+j})] for j=1,2,3

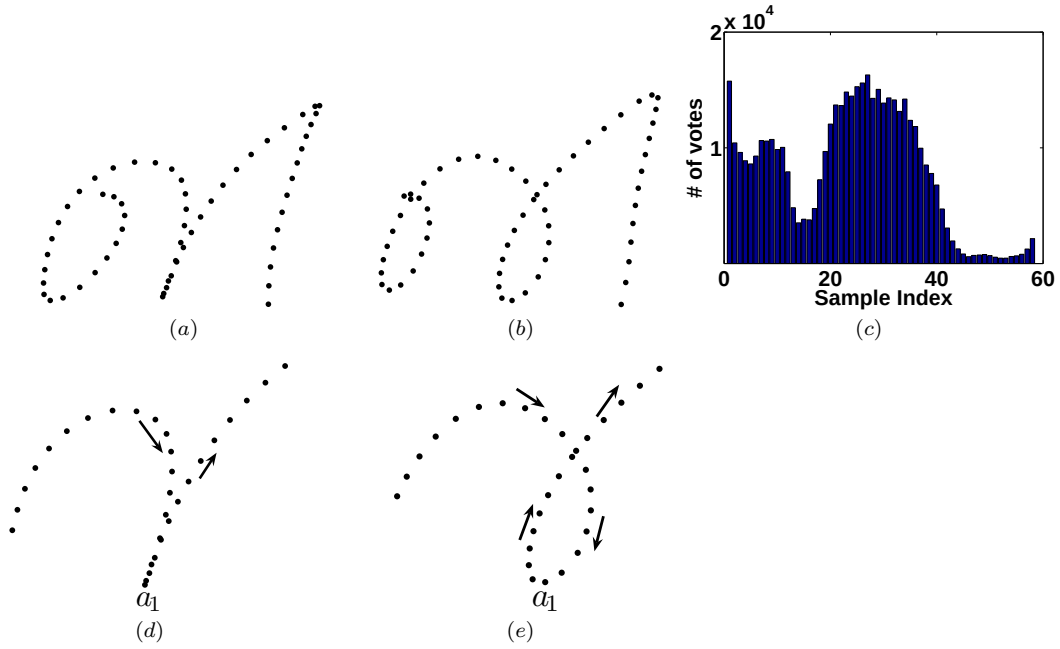


Fig. 10 Disambiguation of consonants /La/ and /Na/. (a) A sample of /La/. (b) A sample of /Na/. (c) DTW-DDH for this pair. (d) $\Re$ for /La/. (e) $\Re$ for /Na/. Features for discriminating these 2 consonants are derived from the region around the attention point a_1. Direction of the trace is indicated with arrows.

Accordingly, we define the feature

$$f_3^{(\text{ന്ന},\text{ന})} = \sum_{j=1}^{3} d_j^2 \tag{19}$$

The values of $f_3^{(\text{ന്ന},\text{ന})}$ are higher for ന than for ന്ന.

9.2 Expert 2: Consonants /la/ and /va/

The DTW-DDH between the consonants ല /la/ and വ /va/ is shown in Fig. 11 (c). We observe that the middle part of the trace primarily discriminates between them. Accordingly, we select the DR as

$$\Re(\text{ല},\text{വ}) = \{(x_i, y_i)\}_{i=16}^{50} \tag{20}$$

The expert 2 is invoked by the selector for the disambiguation. A 4-dimensional feature vector constructed using the region of attention around attention point a_2 corresponding to the first local y-minimum, is robust in disambiguating the symbols (see Fig. 11 (d) and (e)).

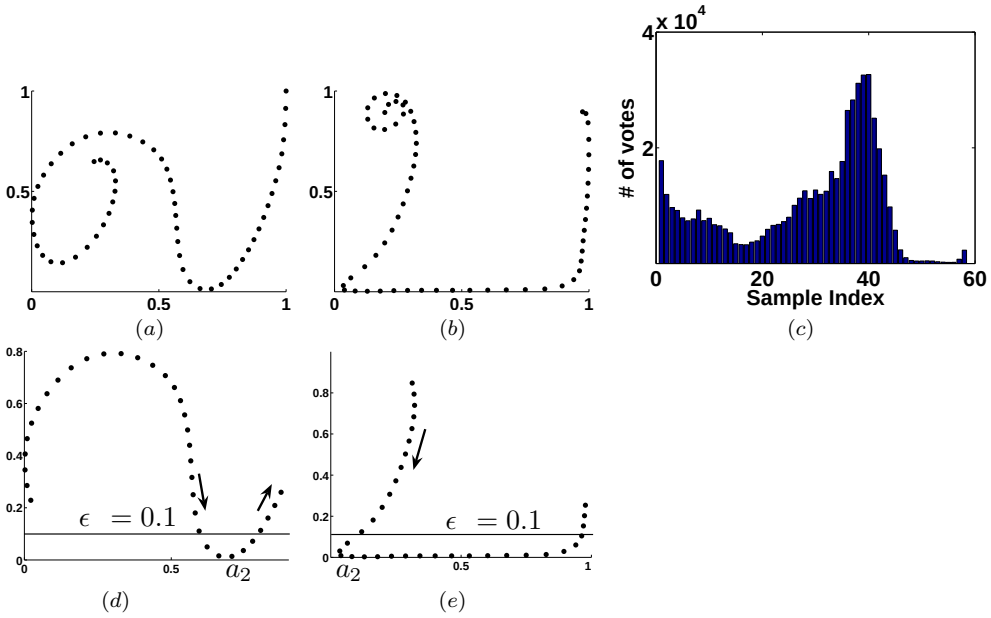


Fig. 11 Disambiguation of consonants /la/ and /va/. (a) A sample of /la/. (b) A sample of /va/. (c) DTW-DDH for this pair. (d) $\Re$ for /la/. (e) $\Re$ for /va/. Features for discriminating these 2 consonants are derived from the region of attention around a_2. Direction of trace are indicated with arrows.

1. We define the first two discriminative features as,

$$f_1^{(\text{ல},\text{வ})} = x_{a_2+1} - x_{a_2} \tag{21}$$

$$f_2^{(\text{ல},\text{வ})} = x_{a_2} - x_{a_2-1} \tag{22}$$

   From statistics, $f_1^{(\text{ல},\text{வ})} > 0$ and $f_2^{(\text{ல},\text{வ})} > 0$ applies to a higher percentage of samples of symbol ல.

2. The anti-clockwise angles with respect to the horizontal direction made by the trace between successive pairs in $\{(x_i, y_i)\}_{i=a_2-5}^{a_2}$ are accumulated and used as a feature. Let Θ_i denote the angle made by the segment $(x_{i+1}, y_{i+1}) - (x_i, y_i)$. We define the feature

$$f_3^{(\text{ல},\text{வ})} = \sum_i \Theta_i \tag{23}$$

   where

$$\Theta_i = tan^{-1} \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \tag{24}$$

   The value of Θ_i lies between 0^o to 360^o. We note that $f_3^{(\text{ல},\text{வ})}$ is higher for the symbol ல than for வ.

3. We extract the part of the trace, whose y-coordinates lie in the range $[y_{a_2}, y_{a_2} + \epsilon]$. The variance of the x-coordinates in this range (higher for symbol வ than for ல) is utilized as the feature $f_4^{(\text{ல},\text{வ})}$. In order to adequately capture the discriminability of the variance, the value of ϵ is set to 0.1.

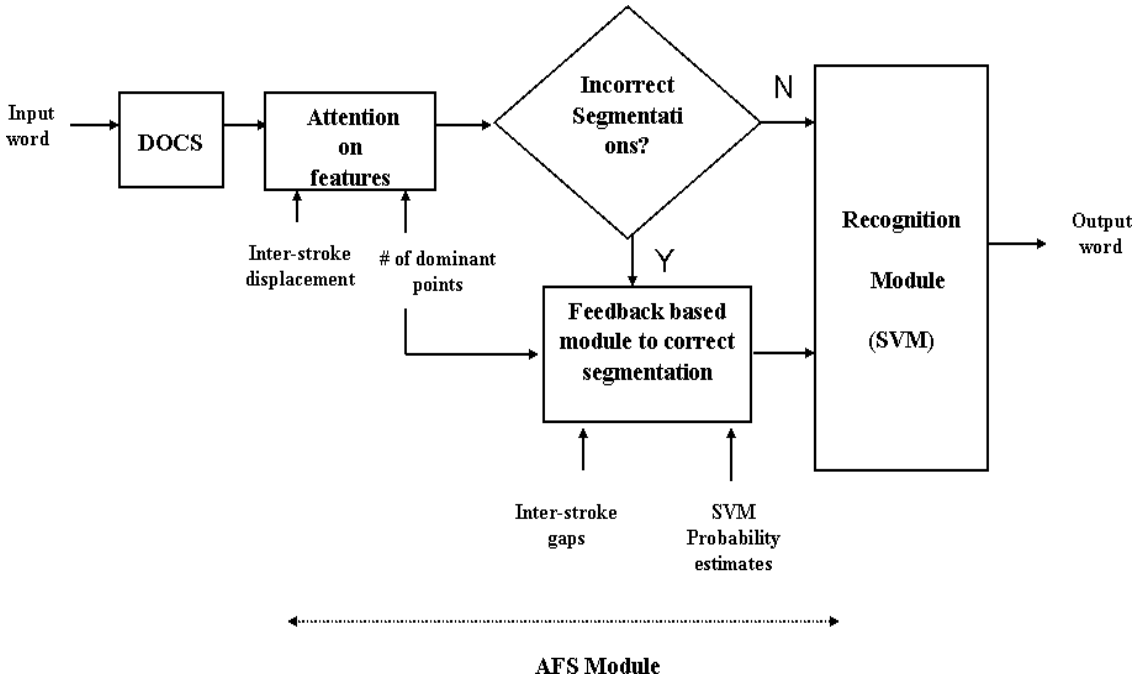


Fig. 12 Block diagram of the feedback based segmentation strategy. The segmentation is accomplished using 2 modules namely : Dominant overlap criterion segmentation (DOCS) followed by Attention feedback segmentation (AFS).

10 Results and discussion

We evaluated the performance of the proposed reevaluation strategies on the IWFHR test dataset and on the symbols segmented from a word database. The Tamil words have been collected using a custom application running on a tablet PC. During the data collection phase, we ensured that all the writers are native Tamil speakers, who currently write in that language, at least irregularly. High school students from across 6 educational institutions in the Indian state of Tamil Nadu contributed in building the word data-base, comprising 10000 words. Prior to the recognition step, we segmented each word to a set of valid symbols using an attention feedback strategy that we proposed earlier in [26].

The following three subsections are oganized as follows: Subsection 10.1 provides a brief overview of the segmentation methodology adopted from [26]. In subsections 10.2 and 10.3, we present the results of the reevaluation strategies on the IWFHR test dataset and segmented symbols of the word database.

10.1 Attention feedback based segmentation

In this section, we present an overall view of the recognition strategy adopted. For more details of the same, we refer the reader to the work [26]. The segmentation technique consists of 2 modules and is depicted in Figure 12. For a multi-stroke

Tamil symbol, there is high degree of horizontal x-overlap between the strokes of the same symbol. In the first module of the segmentation scheme, referred to as the 'Dominant Overlap Criterion Segmentation' (DOCS), we generate a set of stroke groups from the input Tamil word. Ideally, we expect the stroke groups from the DOCS to correspond to valid Tamil symbols. However, at times, based on the overlap threshold, the DOCS module sometimes gives rise to stroke groups, that are either (a) a part of a valid symbol or (b) merger of valid symbols. These stroke groups lead to segmentation errors and affect the performance of the recognition system.

In order to improve the segmentation performance beyond that given by the DOCS module, we use a second module -'Attention feedback segmentation' (AFS). The AFS module consists of 2 units - attention unit and feedback unit. Different sets of features (inter-stroke displacement and number of dominant points) are separately derived to detect under-segmented and over-segmented stroke groups respectively in the attention unit. 'Attention' on these features selects only a subset of the generated stroke groups for subsequent analysis. Only stroke groups, suspected to be incorrectly segmented are fed to the feedback unit. In the feedback unit, a decision is made on whether or not to proceed in correcting the possible segmentation error in the stroke group. Our segmentation procedure is thus unique in a way, that feedbacks are invoked from recognition probability estimates and statistics of spatial features (inter stroke gaps) to correct any wrong segmentation. Whenever the feedback unit favors a correction, the segmentation is refined by rearranging the strokes within or even outside the stroke group under consideration. The AFS module segments the word to a set of valid symbols, that are then subsequently recognized using a SVM classifier, to generate the output word.

The segmentation accuracy (reported as a percentage) is used for the performance evaluation [26]. For a set of t words being tested, we define the following with respect to the i^{th} word,

- N_M^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

Accordingly, the symbol level segmentation accuracy from the DOCS module alone is obtained as $(\sum_{i=1}^{t} N_D^i / \sum_{i=1}^{t} N_M^i) * 100$

In a similar way, the accuracy (at symbol level) after the DOCS+AFS module is computed from $(\sum_{i=1}^{t} N_A^i / \sum_{i=1}^{t} N_M^i) * 100$

For our experiments, we have $t = 10000$ and $\sum_{i=1}^{t} N_M^i = 53246$ (we test the segmentation on a set of 10000 words, comprising 53,246 symbols). The symbol level segmentation accuracy of the DOCS module is 98.1% on these words, which improves to 99.7%, after the application of the AFS module. The improvement in segmentation accuracy , in turn increases the symbol recognition accuracy from 83.9% to 88.4%.

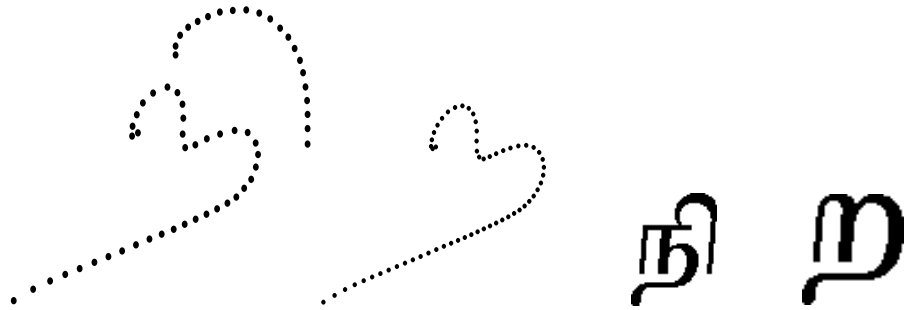
Table 5 Performance evaluation of the base consonant reevaluation strategy on the valid symbols of the test set of IWFHR database.

Group	G_2	G_5	G_7
# of test symbols	3990	3995	3972
# of base consonants incorrectly recognized by primary classifier	194	238	192
# of errors corrected by reevaluation	123	160	122
Error reduction in (%)	63.4	67.3	63.5
% of base consonants correctly recognized by primary classifier	95.1	94	95.2
% of base consonants correctly recognized by reevaluation	98.2	98.0	98.2

10.2 Performance evaluation of the reevaluation strategies on the IWFHR test set

Each of the experiments discussed in this section focuses on demonstrating the improvement in the recognition performance of the primary classifier with a proposed reevaluation technique.

As our first experiment, we reevaluate the base consonants in multi-stroke CV combinations of இ /i/ and ஈ /I/ vowels (G_5, G_7) and in pure consonants (G_2) using the strategy described in Sec 6. We notice that 63.4%, 67.3% and 63.5% of the errors in the base consonants have been corrected in the groups G_2, G_5 and G_7 respectively (Table 5). The errors that remain uncorrected arise mainly due to



(a) (b) (c) (d)

Fig. 13 Illustration of a pattern for which reevaluation of the base consonant fails. (a) This pattern, which is /ni/ (shown in Fig (c)), gets wrongly recognized as /Ri/. (b) Extracted base consonant which is recognized as /Ra/ (shown in Fig (d)). (c) A printed sample of /ni/ for reference. (d) A printed sample of /Ra/ for reference.

samples that appear quite ambiguous. Consider the test sample shown in Fig. 13 (a), that is ground-truthed as the symbol நி /ni/ (displayed in (c)). We observe that the sharp corner of the trace has been smoothed out while writing, making this pattern to appear more like றி /Ri/. The SVM corroborates our intuition by favoring the symbol ற /Ra/ to the extracted base consonant after reevaluation, thereby giving rise to an error (refer sub-figures (b) and (d)).

Table 6 Performance evaluation of the dot recognition strategy on the recognition performance of pure consonants in the IWFHR test set.

Group	G_2
# of test symbols	3990
# of dot strokes incorrectly recognized by primary classifier	35
# of errors corrected by reevaluation	28
Error reduction in (%)	80
% of dot strokes correctly recognized by primary classifier	99.1
% of dot strokes correctly recognized after reevaluation	99.8

Table 7 Performance evaluation of the reevaluation strategy on the recognition accuracy for vowel modifiers of /i/ and /I/ in the IWFHR test set.

Group	G_5	G_7
# of test symbols	3995	3972
# of vowel modifiers incorrectly recognized by primary classifier	105	44
# of errors corrected by reevaluation	95	33
Error reduction in (%)	90.5	75
% of vowel modifiers correctly recognized by primary classifier	97.3	98.9
% of vowel modifiers correctly recognized after reevaluation	99.7	99.8

The second experiment demonstrates the robustness of the techniques proposed for reevaluating the stroke **v** (extracted by the component extractor). We observe from Table 6 that 80% of the dot strokes in pure consonants wrongly recognized by the primary SVM as the vowel modifier of ಇ /i/ and ಋ /I/ have been corrected by the criteria in Sec 7.1. This takes the correct dot recognition performance in pure consonants from 99.1% to 99.8%. On reevaluating the vowel modifiers of ಇ /i/ and ಋ /I/ for a given base consonant (refer Sec 7.2), an average of 86% of vowel modifiers wrongly recognized by the primary SVM get corrected (Table 7). This incidentally raises the /i/ and /I/ vowel modifier recognition rate from 98.1% to 99.7%.

As discussed in Sec 8, for a given confusion pair, a particular expert is selected to work on the class-specific features defined in the DR $\Re$. We now proceed in demonstrating the efficacy of these features. For each of the frequently confused pairs $(c1, c2)$, two different feature sets are separately explored for the reevaluation by the selected expert. The first feature vector comprises the concatenated x-y coordinates of the DR $\Re(c1, c2)$. The other feature vector is derived using the localized features for the confusion pair, as described in Sec 9. From the recognition accuracies in the third and fourth columns of Table 8, we observe that, for each confusion pair, the proposed localized features perform better than the x-y features, except for the pair (ಕಿ/ki/, ಕಿ/ci/), where the performance remains the same.

Table 8 Illustration of the reduction in error rate on some of the confused pairs of the IWFHR test set with reevaluation. The numbers presented are in %.

Confusion Pair	Primary classifier recognition rate	Disambiguation with x-y features over $\Re$	Disambiguation with proposed local features over $\Re$	Error reduction with local features (in %) over primary classifier
(ல ,வ) (/la/, /va/)	96.1	97.2	98.6	64
(ள ,ன) (/La/, /Na/)	93.5	94.9	98	69
(மு ,ழு) (/mu/, /zhu/)	92.6	95.1	98	73
(னி,ளி) (/Ni/, /Li/)	91.2	95.3	97.6	73
(கி,சி) (/ki/, /ci/)	95.2	98.9	98.9	77

The increase in the recognition performance is significant for the symbols

$$\begin{gathered}(\text{ள /La/}, \text{ன /Na/}) \quad 3.1\%, \\ (\text{மு /mu/}, \text{ழு /zhu/}) \quad 2.9\%, \\ (\text{ன /Na/}, \text{ண (VM of /ai/)}) \quad 2.3\% \\ (\text{ல /la/}, \text{வ /va/}) \quad 1.4\%\end{gathered}$$

For each of the above symbols, we compare the dimensionality of the proposed features to that of the concatenated x-y features. As an illustration, consider the DR $\Re(\text{ள}, \text{ன})$ employed for the confusion pair ள /La/ and ன /Na/. When the x-y coordinates of the 30 sample points in $\Re(\text{ள}, \text{ன}) = \{(x_i, y_i)\}_{i=16}^{45}$ (refer Sec 9.1) are employed, we obtain a 60-dimensional feature vector. However, extraction of the robust localized features from $\Re(\text{ள}, \text{ன})$ leads to a 3-dimensional feature vector - a 20 fold reduction in dimensionality. Moreover, this advantage is coupled with the fact that the recognition performance is improved with a lower dimensional feature vector. On similar lines, one can observe that the confusions in (மு/mu/, ழு/zhu/) and (ல/la/, வ/va/) are resolved to a greater extent by employing lower dimensional feature vectors. With the proposed localized features, the experts achieve a significant disambiguating performance on the confusion sets from the primary classifier. This fact can be observed from the recognition rates in the second and fourth columns. From the fifth column, we note that more than 60% of the errors in each confusion pair have been rectified.

Table 9 presents the improvement in recognition of a few symbols after reevaluation. For nearly all the symbols listed, we observe an increase of more than 4%. Across the 26926 samples in the testing set, an accuracy of 87.9% is reported with the reevaluation strategies. Compared to the primary SVM classifier that reported an accuracy of 86% (without reevaluation), this corresponds to a 1.9% increase in recognition performance.

Figure 14 presents a few of the samples that were wrongly recognized by the experts. The samples in (a) and (b) represent the symbol ழு /zhu/. However, the SVM trained with the proposed features in the reevaluation step favors மு /mu/ in both the cases. The part of the trace enclosed by a circle in Figs (a) and (b) (that describe ழு /zhu/) are not captured by the proposed features, thereby leading to

Table 9 Illustration of the error reduction rate of a few symbols from the IWFHR test set with reevaluation strategies. The numbers listed are in %

Symbol	Primary classifier performance	Primary classifier+ reevaluation	Error reduction in (%) over primary classifier
ல /la/	98.4	99.0	33
வ /va/	94.9	98.3	66
ள /La/	94.9	97.2	44.4
ன /Na/	82.8	94.3	66.6
ன (VM of /ai/)	93.8	97.7	63.6
க /ka/	96.3	98.7	65
த /ta/	96.8	98.4	50
மு /mu/	90.1	98.3	83
ழு /zhu/	95.2	97.6	50
ள் /L/	84.2	95.4	71.5
ன் /N/	84.6	97.8	85.7
கி /ki/	91.0	98.8	71
சி /ci/	85.7	96.7	76.9

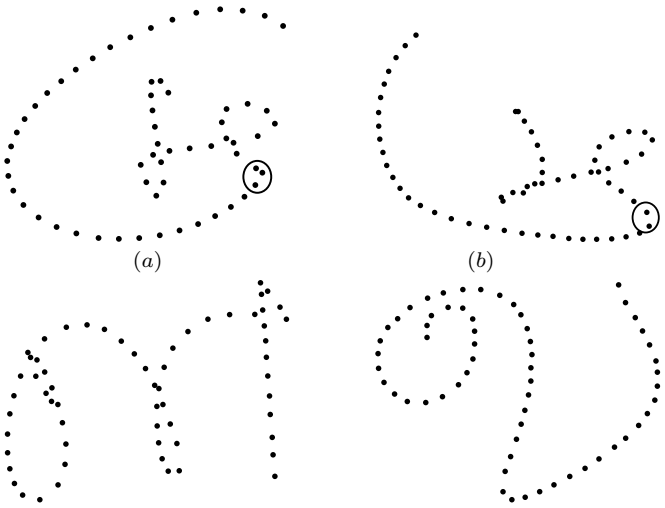
the error. The pattern in Fig (c), which is supposed to be ள /La/ gets reevaluated to ன /Na/. This error clearly arises due to the visual ambiguity of the handwritten character. Finally, the pattern in Fig (d), which is வ /va/ gets recognized as ல /la/. This error is attributed to the lesser value of the x-variance of sample points in the region around the attention point.

10.3 Performance enhancement on the Word Database

Across the 10,000 words (comprising 53246 symbols), an improvement of 3.5% is observed over the performance of the primary classifier (from 88.4% to 91.9%) by incorporating the various strategies (Table 11). This improvement in turn reflects in the word recognition rate, as shown in the second row. The word recognition rate of 65.0% corresponds to that given by the primary SVM classifier (without reevaluation).

A few sample words that have been wrongly classified by the primary classifier but have been corrected by reevaluation are shown in Table 10. The erroneous symbols output from the primary classifier are highlighted with a rectangle in the third column. Appropriate strategies are invoked to correct them as described above. The dot in the last symbol of the first word is wrongly recognized by the SVM as a vowel modifier of ◌ /I/. However, it gets corrected by the reevaluation strategy in Sec 7.1. Reevaluation of base consonants (Sec 6) aids in rectifying the erroneous symbols in the second, third and fourth words. For the last word, the disambiguation algorithm for the confusion pair ல /la/ and வ /va/ (Sec 9.2) is invoked to resolve the error in the third symbol. As far as the fourth symbol is concerned, reevaluation of base consonants described in Sec 6 together with disambiguation of (ள /La/ , ன /Na/) (Sec 9.1) ensure that the error is corrected.

We also analyzed the time complexity of the proposed reevaluation techniques. Given a word comprising L segments, its recognition by the primary classifier alone



Pattern	(a)	(b)	(c)	(d)
Actual Symbol	ழ	ழ	ள	வ
Reevaluated symbol	ழ	ழ	ன	ல

Fig. 14 Examples of patterns that fail to get corrected by the proposed reevaluation techniques.

(without any reevaluation strategy), involves a time complexity of $O(L|V|)$. Here, $|V|$ represents the number of symbols being compared at each segment. In our case we have $|V| = 155$. The incorporation of the reevaluation techniques adds to the complexity of the overall system. Let C_T^i correspond to the number of confusions for the symbol in the i^{th} segment/ position of the word. Then one needs to make C_T^i comparisons in the reevaluation step for that segment. Accordingly, with the incorporation of reevaluation techniques to each segment in the word, we get the overall complexity of the system as $O(L|V| + \sum_{i=1}^{L} C_T^i)$.

The primary classifier may, at times, wrongly recognize symbols, written with a style infrequently encountered in the script. As an illustration, consider the word in Fig. 15 (a), in which the first and fifth symbols, (பி /pi/ and வி /li/) are written in an unconventional style. From the output, we observe that the first symbol பீ /pI/ from the primary classifier is corrected to பி /pi/ by employing the strategy for the vowel modifiers described in Sec 7.2. However, the fifth symbol வி /vi/ is wrongly recognized as வ /va/ by the primary SVM classifier. The disambiguation strategy for the pair (ல /la/, வ /va/) is invoked and the output remains unchanged after this step. The reason behind this recognition error not getting corrected to வி /vi/ is attributed to the fact that the symbols (வ /va/, வி /vi/) rarely get confused by the primary classifier, and hence are not a confusion set in this work. Accordingly, there is no expert dedicated to the disambiguation of வ /va/ from வி /vi/) (refer Sec 8).

For the word in (b), the first symbol அ /a/ is wrongly recognized as சு /cu/

Table 10 Illustration of a few word samples, that have been wrongly recognized by the primary SVM classifier but corrected with reevaluation.

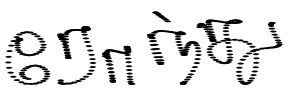
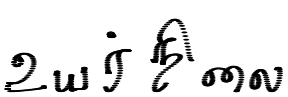
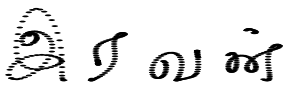
Sl.No	Input word	primary classifier output	primary classifier+ reevaluation output
1		வீரமீ /vIramI/	வீரம் /vIram/
2		குழஞ்தை /kuzhanjtai/	குழந்தை /kuzhantai/
3		ரோர்து /rOrtu/	ரோந்து /rOntu/
4		உயர்திலை /uyartilai/	உயர்நிலை /uyarnilai/
5		இரலள் /iralaL/	இரவன் /iravaN/

Table 11 Illustration of the error reduction rate on the word database after the reeevaluation strategies. Number of words =10000. Number of symbols=53246.

	Primary classifier performance	Primary classifier+ reevaluation	Error reduction (in %) over primary classifier
Symbol recognition rate	88.4	91.9	30.2
Word recognition rate	65.0	76.9	34

due to the specific writing style being infrequently encountered. Owing to the fact that the symbol pair (அ /a/, சு /cu/) is not one of the confusion sets, there is no expert proposed to disambiguate them (refer Sec 8). However, சு /cu/ is part of the (கு /ka/, சு /cu/) confusion set and hence this input symbol is wrongly sent to the expert dealing with கு /ka/ and சு /cu/. Thus, in a very small number of cases, due to some of the less frequent confusions of the primary classifier, the input symbols land at the wrong expert. Hence, the recognition error can not get corrected.

Note that, for both the words in (a) and (b), the misclassifications encountered are not covered by the confusion sets considered.

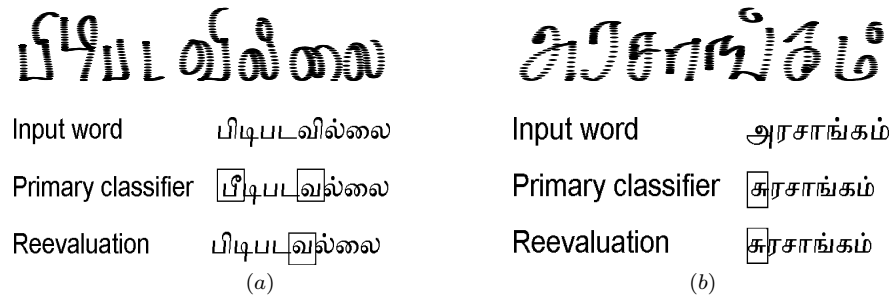


Fig. 15 Illustration of recognition errors not handled by current reevaluation strategies. (a) The first and fifth symbols in this word are written with an unconventional style. The first symbol, belonging to `/pi/` (in group G_5), is assigned to `/pI/` (in group G_7) by the primary classifier. Since the vowel modifiers of `/i/` and `/I/` of the CV combinations G_5 and G_7 get frequently confused, this error is corrected with reevaluation by employing the strategy in Sec 7.2. However, the fifth symbol `/vi/` (also of group G_5) is assigned to the base consonant `/va/` in G_1. Since the symbols `/vi/` and `/va/` rarely get confused with each other, they are not considered for disambiguation and hence this error is not corrected. (b)The first symbol (which is supposed to be vowel `/a/`, is badly written, almost resembling the CV combination `/cu/`. Hence, it is recognized as `/cu/` only. Owing to the fact that these 2 symbols rarely get confused with each other, this pair is not part of the confusion sets considered for reevaluation. In other words, the misclassified symbols in the two words are not covered by the confusion sets considered in this work.

11 Summary

In this article, various reevaluation strategies are proposed to reduce the Tamil symbol error rate of the primary recognition system. In particular, with these techniques, ambiguities arising in the base consonants, pure consonants and vowel modifiers are resolved to a considerable extent. Secondly, to deal with confused pairs, a DTW approach is proposed to automatically extract their discriminative regions. Novel localized cues derived from these regions are fed to an appropriate expert for subsequent disambiguation. The proposed features are shown to be quite promising in improving the recognition of the confusion sets of Tamil symbols. It is to be noted that we adopt a generic approach for recognizing words, without involving the use of language models. Our main objective was to explore as to how far we can go ahead in improving the recognition rate of the primary classifier, by reevaluating symbols based on class-specific features.

It is important to remember that the training of the primary classifier, the identification of the confusion sets and the training of the experts, have all been achieved using the isolated symbols of the IWFHR dataset. However, all of them work well in recognizing and disambiguating the symbols segmented from the handwritten word database, which are from completely different individuals and recorded many years later than that of the IWFHR dataset. This shows the scalability of the proposed strategies to unseen data-sets.

Acknowledgements

The authors thank Technology Development for Indian Languages (TDIL), Department of Information Technology, Govt of India for funding this work. The

help rendered by staff of Medical Intelligence and Language Engineering (MILE) Laboratory in data collection and truthing is acknowledged.

References

1. S Sundaram, Lexicon-free recognition strategies for online handwritten Tamil words, PhD Thesis, Indian Institute of Science, (2011).
2. C S Sundaresan, S S Keerthi, A study of representations for pen based handwriting recognition of Tamil characters, Proc. International Conference on Document Analysis and Recognition, pp.422-425, (1999).
3. A H Toselli, M Pastor, E Vidal, On-line handwriting recognition system for Tamil handwritten characters, Proc. Pattern Recognition Image Analysis, pp.370-377, (2007).
4. L Prasanth, J Babu, R Sharma, P Rao, M Dinesh, Elastic Matching of Online Handwritten Tamil and Telugu Scripts Using Local Features, Proc. International Conference on Document Analysis and Recognition, pp.1028-1032, (2007).
5. N Joshi, G Sita, A G Ramakrishnan, S Madhvanath, Comparison of Elastic Matching Algorithms for Online Tamil Handwritten Character Recognition, Proc. International Workshop Frontiers Handwriting Recognition, pp.444-449, (2004).
6. V Deepu, S Madhvanath, A G Ramakrishnan, Principal Component Analysis for Online Handwritten Character Recognition, Proc. International Conference Pattern Recognition, pp.327-330, (2004).
7. B S Raghavendra, C K Narayanan, G Sita, A G Ramakrishnan, M Sriganesh, Prototype Learning Methods for Online Handwriting Recognition, Proc. International Conference on Document Analysis and Recognition, pp.287-291, (2005).
8. H Swethalakshmi, C Chandra Sekhar, V S Chakravarthy, Spatiostructural Features for Recognition of Online Handwritten Characters in Devanagari and Tamil Scripts, Proc. International Conference Artificial Neural Networks, vol 2, pp.230-239, (2007).
9. K H Aparna, V Subramanian, M Kasirajan, G V Prakash, V S Chakravarthy, S Madhvanath, Online Handwriting Recognition for Tamil, Proc. International Worshop Frontiers Handwriting Recognition, pp.438-443, (2004).
10. L Vuurpijl, L Schomaker, M Van Erp, Architectures for Detecting and Solving Conflicts: Two-Stage Classification and Support Vector Classifiers, International Journal Document Aanalysis Recognition, vol 5, Issue 4, pp.213-223, (2003).
11. A Bellili, M Gilloux, P Gallinari, An MLP-SVM combination architecture for offline handwritten digit recognition, International Journal Document Aanalysis Recognition, vol 5, Issue 4, 244-252, (2003).
12. L Prevost, L Oudot, A Moises, C Michel-Sendis, M Milgram, Hybrid generative/discriminative classifier for unconstrained character recognition, Pattern Recognition Letters, vol 26, Issue 12, pp.1840-1848, (2005).
13. A Alaei, P Nagabhushan, U Pal, Fine Classification of Unconstrained Handwritten Persian/Arabic Numerals by Removing Confusion amongst Similar Classes, Proc. International Conference on Document Analysis and Recognition, pp.601-605, (2009).
14. D V Sharma, G S Lehal, S Mehta, Shape Encoded Post Processing of Gurmukhi OCR, Proc. International Conference on Document Analysis and Recognition , pp. 788-792, (2009).
15. G S Lehal, C Singh, A Post Processor for Gurmukhi OCR, SADHANA, vol 27, Issue 1, pp.99-112, (2002).
16. K Nair, C V Jawahar, A Post-Processing Scheme for Malayalam using Statistical Subcharacter Language Models, Proc. Document Analysis System, pp.363-370, (2010) .
17. B B Chaudhuri, U Pal, OCR error detection and correction of an inflectional Indian language script, Proc. International Conference Pattern Recognition, vol 3, pp.245-249, (1996).
18. B Nethravathi, C P Archana, K Shashikiran, A G Ramakrishnan, V Kumar, Creation of a huge annotated database for Tamil and Kannada OHR, Proc. International Workshop Frontiers Handwriting Recognition, pp.415-420, (2010).
19. Isolated IWFHR 2006 Tamil Handwritten Character Dataset www.hpl.hp.com/india/research/penhw-interfaces-1linguistics.html
20. J C Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, vol 2, pp.121-167, (1998).

21. Duda, Hart, Stork, Pattern Classification, Springer Wiley, (1995).
22. C C Chang and C J Lin, LIBSVM : a library for support vector machines, ACM Transactions on Intelligent Systems and Technology, Vol 2, Issue 3, (2011).
23. A F R Rahman and M C Fairhurst, Selective partition algorithm for finding regions of maximum pairwise dissimilarity among statistical class models, Pattern Recognition Letters, vol 18, Issue 7, pp.605-611, (1997).
24. K C Leung and C H Leung Recognition of handwritten Chinese characters by critical region analysis, Pattern Recognition, Vol 43, Issue 3, pp.949-961, (2010).
25. S Sundaram, A G Ramakrishnan, Lexicon-free, novel segmentation of online handwritten Indic words, Proc. International Conference on Document Analysis and Recognition, pp.1175-1179 (2011).
26. Suresh, S. and Ramakrishnan, A. G. 2013. Attention-feedback based robust segmentation of online handwritten isolated Tamil words. ACM Trans. Asian Lang. Inform. Process. vol 12, Issue 1, Article 4, (March 2013).

Appendix A: The complete list of Tamil characters

- **Vowels**

அ ஆ இ ஈ உ ஊ எ ஏ ஐ ஒ ஓ ஔ

- **Base Consonants**

க ங ச ஞ ட ண த ந ப ம ய ர ல

வ ழ ள ற ன ஸ ஷ ஹ ஜ க்ஷ

- **Pure consonants**

க் ங் ச் ஞ் ட் ண் த் ந் ப் ம் ய் ர் ல்

வ் ழ் ள் ற் ன் ஸ் ஷ் ஹ் ஜ் க்ஷ்

- **CV combinations of ஆ vowel**

கா ஙா சா ஞா டா ணா தா நா பா மா

யா ரா லா வா ழா ளா றா னா ஸா ஷா

ஹா ஜா க்ஷா

- **CV combinations of இ vowel**

கி ஙி சி ஞி டி ணி தி நி பி மி யி ரி

லி வி ழி ளி றி னி ஸி ஷி ஹி ஜி க்ஷி

– **CV combinations of ◌ீ vowel**

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

– **CV combinations of ◌ூ vowel**

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

– **CV combinations of ◌ூ vowel**

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

– **CV combinations of ெ vowel**

கெ ஙெ செ ஞெ டெ ணெ தெ நெ பெ
மெ யெ ரெ லெ வெ ழெ ளெ றெ னெ
ஸெ ஷெ ஹெ ஜெ க்ஷெ

– **CV combinations of ே vowel**

கே ஙே சே ஞே டே ணே தே நே பே
மே யே ரே லே வே ழே ளே றே னே
ஸே ஷே ஹே ஜே க்ஷே

– **CV combinations of ஐ vowel**

கை ஙை சை ஞை டை ணை தை நை

பை மை யை ரை லை வை ழை ளை

றை னை ஸை ஷை ஹை ஜை க்ஷை

– **CV combinations of ஓ vowel**

கொ ஙொ சொ ஞொ டொ ணொ தொ
நொ பொ மொ யொ ரொ லொ வொ
ழொ ளொ றொ னொ ஸொ ஷொ ஹொ
ஜொ க்ஷொ

– **CV combinations of ஔ vowel**

கௌ ஙௌ சௌ ஞௌ டௌ ணௌ தௌ
நௌ பௌ மௌ யௌ ரௌ லௌ வௌ
ழௌ ளௌ றௌ னௌ ஸௌ ஷௌ ஹௌ
ஜௌ க்ஷௌ

– **CV combinations of ஒள vowel**

கெள ஙெள செள ஞெள டெள ணெள
தெள நெள பெள மெள யெள ரெள
லெள வெள ழெள ளெள றெள னெள
ஸெள ஷெள ஹெள ஜெள க்ஷெள

– **Additional characters**

ஃஃ ஸ்ரீ