Online Handwriting Recognition for Indic Scripts

Bharath A. and Sriganesh Madhvanath
HP Laboratories, India

HPL-2008-45
May 5, 2008*

online handwriting recognition, indic script recognition

Handwriting recognition refers to the problem of machine recognition of handwritten script. Over the years, a number of algorithms have been proposed and today there are even commercial systems for European and Oriental scripts. Even though handwriting-based text input methods have tremendous potential in the context of Indic scripts, research efforts directed at recognition are at their early stages. The structure of the scripts and a variety of writing styles pose challenges that are different from other scripts and hence require customized techniques for recognition. In this paper, we describe the challenges in recognizing Indic scripts and present an overview of the state of the art approaches developed for isolated character recognition and word recognition. We also provide some pointers to resources such as tools and datasets that are available for online Indic script recognition.

To be published in the book: OCR for Indic Scripts: Document Recognition and Retrieval by Springer, 2008

Online Handwriting Recognition for Indic Scripts

Bharath A. and Sriganesh Madhvanath

Hewlett-Packard Laboratories, Bangalore, India
{bharath.a, srig}@hp.com

1 Introduction

India plays host to 22 official languages and 10 scripts, in addition to a large number of others which do not have official status. The official languages invariably have large numbers of speakers (e.g. approximately 500 million speakers of Hindi, 200 million of Bangla [1]). Many Indic languages have substantial global presence – Tamil for instance is also one of the official languages in countries such as Singapore, Malaysia, and Sri Lanka. The users of these languages and scripts constitute approximately a sixth of the world's population.

In India, Information Technology (IT) is still largely limited to the small fraction of the population that is English-literate. One reason for this has been the complexity of text input in local Indic languages. Over the years a number of QWERTY-overlays and specific keyboard layouts have been devised for different Indic languages, but they remain non-standard and difficult to learn and use. The large alphabet size typically requires multiple keystrokes for entering a character and mandates complex key-character mappings to be remembered, presenting a substantial barrier to use, especially for occasional users. The use of handwriting, on the other hand, is widely entrenched in the home, government and business, and forms the basis for record-keeping and communication.

In this setting, technology for the online recognition of handwriting (Online HWR) in Indic languages and scripts can play a significant role in promoting IT in local languages. As compared to speech, the processing of handwriting input for text entry is less expensive computationally, and potentially more accurate, especially in the presence of ambient noise. Another useful aspect of machine recognition of online handwriting is that it can be easily customized to recognize an individual's own handwriting. Decades of research have made handwriting input a reality for several European and Oriental languages represented respectively using the Latin and the Chinese, Japanese, Korean (CJK) scripts [2–5]. Such methods are now commonly built into mobile devices and Tablet PCs and feature acceptably high recognition accuracies. However similar technology for the Indic languages and scripts is still in its infancy.

In this chapter we provide an overview of the state of the art in Online HWR for Indic scripts. We first describe the structure of Indic scripts, and the challenges posed for online recognition. We then describe past and present research efforts directed at isolated character recognition and word recognition for Indic scripts. A glimpse of handwriting based text input systems developed for Indic scripts is also presented along with some potential applications of Online HWR. An overview of resources such as tools and datasets available for Online HWR of Indic scripts may be found in the last section.

2 The Structure of Indic Scripts

The 10 official Indic scripts - Devanagari, Tamil, Gurmukhi, Telugu, Kannada, Gujarati, Oriya, Bangla, Malayalam and Urdu - differ by varying degrees in their visual characteristics, but share some important similarities. With the exception of the Urdu script which is of Perso-Arabic origin, they have evolved from a single source, the phonographic Brahmi script, first documented extensively in the edicts of Emperor Asoka of the third century BC. They are defined as "syllabic alphabets" or *abugidas* in that the unit of encoding is a syllable of speech, however the corresponding orthographic units show distinctive internal structure and a constituent set of graphemes [6]. A word in these scripts is written as a sequence of these orthographic syllabic units. For simplicity, we will henceforth refer to these units as "characters", a term commonly used to refer to the building blocks of recognition systems (as in "isolated *character* recognition"). Some common rules for formation of orthographic syllables are discussed by Mudur et al. [7]. Some common classes of characters are described below with examples:

- V: An independent vowel */ee/*

  Devanagari: ई   Tamil: ஈ   Telugu: ఈ

- C: An isolated consonant */tha/* with inherent neutral vowel */a/*

  Devanagari: त   Tamil: த   Telugu: త

- CH: An isolated consonant */th/* with the inherent vowel muted by a special diacritic called */halant/*. This form is used rarely; the linear form (CH)(C) of two consecutive consonant sounds is instead generally represented by a single consonant conjunct character (C'C), where C' denotes the half-form of the first consonant. A notable exception is Tamil, wherein the use of the CH form is the norm.

  Devanagari: त्   Tamil: த்   Telugu: త్

- CV: A consonant */tha/* combined with vowel */ee/* to produce */thee/*. The vowel overrides the inherent */a/* and is indicated using a *matra* or diacritic symbol (also see Fig. 1).

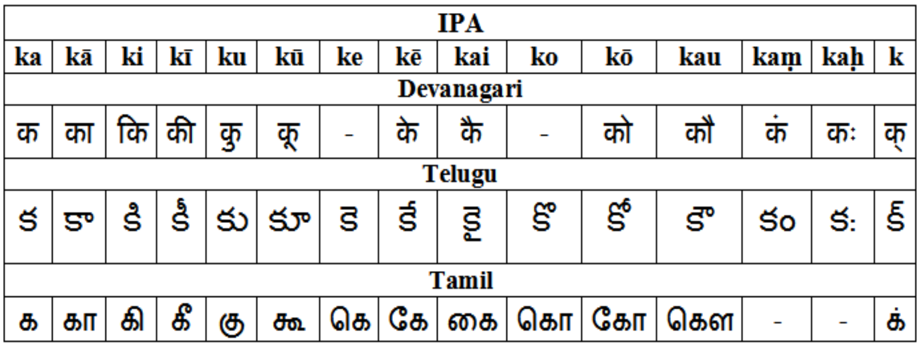
  Devanagari: ती   Tamil: தீ   Telugu: తీ

– CVM: A CV character with a modifier M to indicate nasalization of the vowel V. Devanagari has four different kinds of nasalization, indicated using different kinds of diacritic marks, e.g. तीं */theem/*

– C’C: A conjunct that combines two consonant sounds - */k/* with */tha/* to produce */ktha/*. In Devanagari, the leading consonant */k/* is indicated by a ‘half form’ to suggest that it is missing its inherent vowel. However in Telugu it is just the reverse - the leading consonant is in its full form but */tha/*, indicated by the horizontal diacritic at the bottom, is in its half form. Tamil does not use half forms. Instead, consonant conjuncts are “unraveled” into a linear sequence of (CH) characters followed by the base consonant C.

Devanagari: क्त Telugu: క్త

– CC: A distinct conjunct wherein the constituent consonants are not identifiable e.g. consonant श् */sh/* combines with consonant र */ra/* to produce a distinct shape श्र */shra/* in Devanagari. Many C’C conjuncts have alternative representations as distinct conjuncts



IPA														
ka	**kā**	**ki**	**kī**	**ku**	**kū**	**ke**	**kē**	**kai**	**ko**	**kō**	**kau**	**kaṃ**	**kaḥ**	**k**
Devanagari														
क	का	कि	की	कु	कू	-	के	कै	-	को	कौ	कं	कः	क्
Telugu														
క	కా	కి	కీ	కు	కూ	కె	కే	కై	కొ	కో	కౌ	కం	కః	క్
Tamil														
க	கா	கி	கீ	கு	கூ	கெ	கே	கை	கொ	கோ	கௌ	-	-	க்

Fig. 1. Combinations of the consonant */ka/* with different vowels, and the vowel muting */halanth/*

The above classes represent subsets of the set of all possible characters. In its most complex form a character is a *base consonant* C, optionally surrounded by modifiers for other consonants C’, a vowel unit V and modifier M. For example, क्तीं */ktheem/* in Devangari has the structure C’CVM. This example also shows an alternative representation of */ktha/* as a distinct conjunct. The first two classes of characters C and V are sometimes collectively called “simple characters” and have been the focus of early efforts at recognizing Indic scripts. Most Indic scripts have an order of 600 CV characters and as many as 20,000 C’CV ones in theory, although a much smaller subset of C'CV characters are used in practice. While Devanagari has 35 consonants, 11 vowels and 4 vowel modifiers, Telugu consists of 18 vowels and 36 consonants. The number of characters in both the scripts run into thousands (1500 for Devanagari and around 5000 for Telugu). Tamil has 12 vowels, 18 consonants, six Grantha letters (that can combine with vowels for writing Sanskrit words) and a special symbol, with the number of characters summing up to 325 [8], though a smaller number is in common usage. The much smaller number is largely due to consonant conjunct characters such as C'C being written as linear sequences of separate characters CH and C, as previously mentioned.

3 Challenges for Online HWR

The structure of characters in Indic scripts was introduced in the previous section. In this section we describe some of the challenges for Online HWR of characters and words in Indic scripts, and highlight key differences from Latin and CJK scripts. Some of these stem from the structure of these scripts; others from established writing practice.

3.1 Large alphabet size

As mentioned earlier, most Indic scripts use a large ($>$ 1000) number of characters, as opposed to less than 100 in English. The internal graphemic structure makes a divide and conquer approach to recognition feasible in theory. However many consonant conjuncts are represented by visually distinct conjuncts bearing no resemblance to the constituent consonant shapes (Fig. 2). Similarly, many consonant and vowel combinations give rise to new symbols (e.g. 'து' */thu/* in Tamil) which cannot be segmented into the base consonant and matra. These may need to be dealt with as opaque symbols and exceptions in a divide and conquer recognition strategy. Other challenges for this strategy (discussed in detail later) include cursive styles and stroke and symbol order variations across different writers.

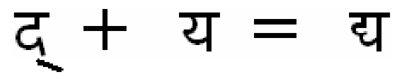
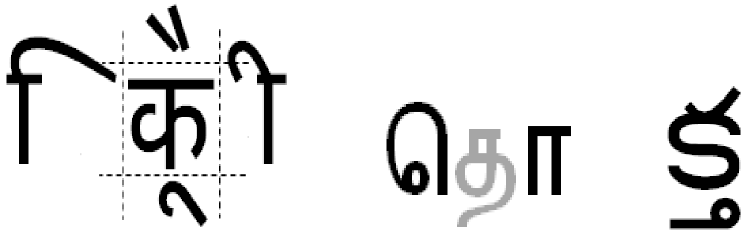


Fig. 2. Combination of consonants resulting in a distinct conjunct character in Devanagari

3.2 Two-dimensional structure

As is evident from Fig. 1, *matras* or vowel diacritics can occur to the left, right, bottom, top, or even as multiple components surrounding the base consonant.

Some possible vowel matras for a consonant symbol in Devanagari are shown in Fig. 3(a). In Fig. 3(b), a two-part matra with components occurring on the left as well as on the right of a base consonant in Tamil is shown. Similarly, half-consonant forms in consonant conjuncts can occur in different positions around the base consonant. In the case of Telugu, they stack vertically as shown in Fig. 3(c). Position is also important to distinguish certain matras and consonant forms that have very similar shapes and differ only in their position relative to the base consonant e.g. half-consonant */n/* and matras */e/* and */oo/* shown in Fig. 4. Thus Indic scripts exhibit a two-dimensional structure much like the CJK scripts. Modeling Indic characters for recognition in terms of the constituent strokes or graphemes requires modeling of their spatial relationships in addition to their shapes. Further, the two dimensional structure results in added variability in symbol and stroke order across writers, unlike the linear left-to-right ordering of the Latin script.



(a) (b) (c)

Fig. 3. Two-dimensional structure: (a) some possible matras for a consonant in Devanagari (b) two-part matra surrounding the consonant in Tamil (c) consonant conjunct in Telugu

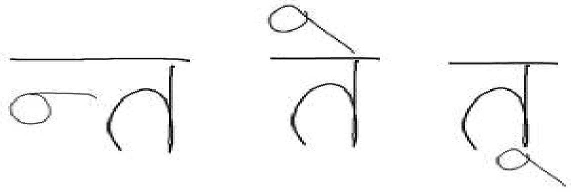
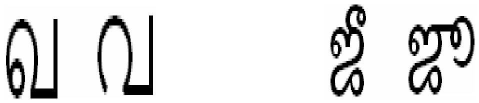


Fig. 4. Similar-looking half-consonant and matras varying only in their position relative to the base consonant */tha/*

3.3 Inter-class similarity

In certain Indic scripts, there is intrinsically high inter-class similarity between some pairs of symbols. Fig. 5(a) shows two characters from Malayalam that look very similar except for the small loop present in the first. Fig. 5(b) shows two Tamil characters with a subtle difference in the shapes of their matras. This calls for reliable, highly distinctive features to describe the shapes of characters and graphemes.



(a) (b)

Fig. 5. Similar-looking characters in (a) Malayalam (b) Tamil

3.4 Issues with writing styles

Indic scripts with the exception of Urdu are written as a left-to-right sequence of characters (syllabic units). As already discussed, the characters themselves have two-dimensional arrangements of graphemes corresponding to consonants, vowels and vowel modifiers. Since characters can vary widely in width, height and complexity, there is no boxed style. This is a key difference from CJK scripts wherein characters are complex but of approximately the same size and may be written in boxes.

In general, character transitions are marked by pen-lifts. Writing an entire word cursively is possible in some Indic scripts (e.g. Bangla), but rare. One might say that the "run-on" style is the most common for Indic scripts. However cursiveness is common within characters, and found generally wherever a pen-up requires additional effort.

While writing a character, users are generally concerned with reconstructing its visual appearance rather than its phonological structure. Various factors such as the relative positions of different strokes, the effort required to move from one stroke to the next given the overall flow of writing, and writing styles taught in school - all have an influence on the stroke order that is eventually used. The consequences for Online HWR are many:

Symbol order variations - The sequence of writing of consonant and vowel units in a character need not correspond to the phonological order of their occurrence in the corresponding syllable. For instance, ‘ि’matra in Devanagari and ‘ெ’ in Tamil are often written before writing the base consonant, since they occur to its left. In contrast, the Unicode representation of CV characters in Indic scripts is based on their phonological structure, and encodes the consonant before the vowel. While modeling characters and the lexicon, the recognition system should take this discrepancy into account.

Stroke order variations spanning multiple symbols - Strokes from different graphemes may be interleaved while writing a character. For example, a two-stroke matra may be written partially and completed only after the base consonant is written. This is loosely related to the phenomenon of delayed strokes in English, wherein some strokes are entered only after the completion of the entire word. However for Indic scripts it happens at the level of individual characters, and the variations are widespread and not limited to a small number of strokes such as t-crossings and i-dots in English. Further, because of the 2-D arrangement, it is often not possible to use heuristics to reorder out-of-order strokes as is common practice for dealing with delayed strokes in English.

Stroke order, number and direction variations within symbols - The ordering of strokes is likely to vary even within a symbol. For example, Fig. 6 shows different styles of writing the consonant */fa/* in Devanagari, where each style (shown along the columns) differs in the number of strokes and/or their ordering.

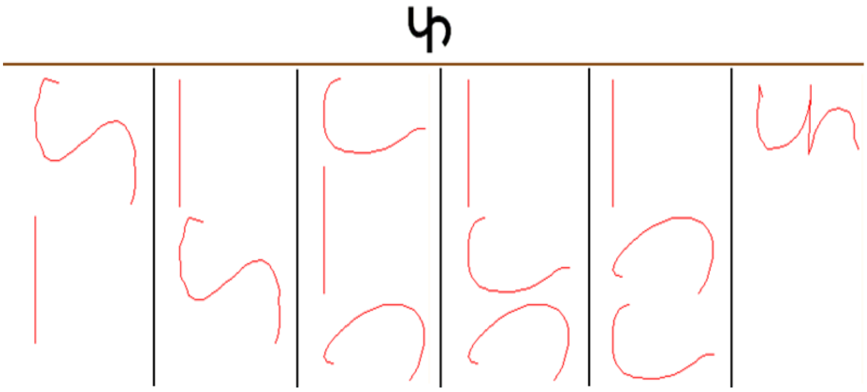


Fig. 6. Writing styles identified for a Devanagari character

In general, stroke order, number and direction variations are quite high in Indic characters and constitute one of the central challenges in online recognition of Indic scripts. These variations may be discovered automatically from the data samples by applying unsupervised learning techniques such as clustering [9].

3.5 Language-specific and regional differences in usage

Marked differences in the use of symbols may be observed in the use of a script like Devanagari across languages such as Hindi, Sanskrit, Marathi and Nepali.

For instance, the *halant* (vowel muting diacritic) and the CH form are used frequently in Sanskrit, but rarely in Hindi. The shapes of symbols show regional variations, influenced by other languages and scripts in use in the region and its surrounding areas. Due to the fact that languages such as Tamil and Bangla span multiple countries, one may also expect country-specific differences in the use of the corresponding scripts. In all these cases, the language models, which are often used in the handwriting recognition system to improve accuracy, need to be substantially unique.

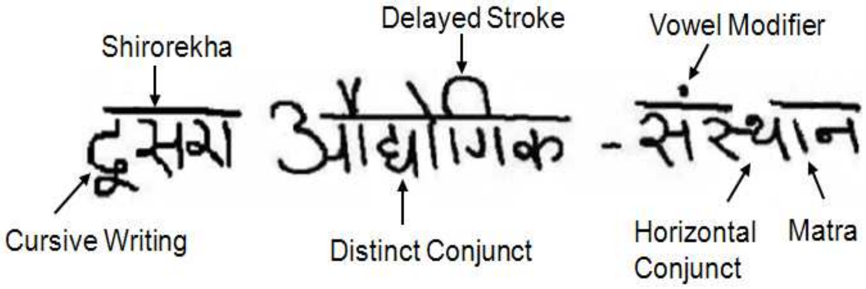


Fig. 7. Some challenges for Online HWR of Devanagari script

The challenges for online recognition of Indic scripts are sufficiently different from those for Latin. Chief among them are the large number of classes, the stroke order/number variation and the two dimensional nature of the script. There are several others. For example, small vowel modifiers may get interpreted as noise in the input, and the *shirorekha* or headline which is often written after completing the word requires special treatment (Fig. 7). Indic script recognition also differs from that of CJK scripts in a few significant ways. In the case of CJK scripts, the shape of each stroke in a character is generally a straight line and hence stroke direction based features are often sufficient. But in the case of Indic scripts, the basic strokes are often nonlinear or curved, and hence features that provide more information than just the directional properties are required. Moreover, in CJK scripts, a word is generally written discretely and hence segmenting it into characters is much easier when compared to Indic scripts where the most common style of writing is run-on. Table 1 summarizes the salient characteristics of Indic, Latin and CJK scripts from the perspective on Online HWR.

4 Recognition of Isolated Characters

We have seen that syllabic units or characters of varying complexity form the basic building blocks of Indic scripts, and are challenging to recognize for a number of reasons. A number of early efforts in the literature have focused on

Table 1. Comparison of Indic, Latin and CJK scripts [6]

Property	**Indic**	**Latin**	**CJK**
writing system	syllabic alphabet	alphabetic	Chinese and Kanji: pictographic-ideographic, Kana: syllabary, Hangul: syllabic alphabet
number of units	> 1500	< 100	Chinese and Kanji: few thousands, Kana: 48, Hangul: few thousands
common style of writing a word	run-on	cursive	discrete
structure of writing	2-D	1-D	2-D
stroke shape complexity	high	high	low
stroke order/number variation	high	low	high

recognition of simple characters such as independent vowels and isolated consonants. The primary challenges in recognizing complex characters as previously outlined come from their number, shape complexity and symbol and stroke order variation.

4.1 Strategies

There are several strategies possible for recognition of isolated (complex) characters:

1. Characters may be viewed as compositions of strokes
2. Characters may be viewed as compositions of C, C', V and M graphemes
3. Characters may be viewed as indivisible units

In the stroke-centric strategy, a set of unique stroke shapes that constitute all characters is found and characters expressed as combinations of these strokes. The number of unique strokes in most Indic scripts is believed to be less than 300. A key problem with this strategy is that these strokes are not known in advance for a given script, and are a function of writing styles. The approach developed at IIT-M [10–13] adopt this strategy. The determination of unique strokes is performed manually by analysis of training data, and are estimated as 123 and 98 for Devanagari and Tamil respectively [13]. However these approaches are rule-based, script-specific and may require significant manual intervention in the training phase.

The second strategy leverages the internal graphemic structure of the character. Recognition of a character is a result of recognizing the (much smaller set of) constituent graphemes. Potential segmentation points are typically at stroke transitions and alignment using Dynamic Programming may be used for segmentation. Hidden Markov Models (HMM) may also be used to solve this problem implicitly. For Telugu script recognition [14], basic graphemes including core characters and ligatures, summing up to 141 when their positions are ignored, are manually identified and then HMMs are used to model each one of them. A major advantage of this strategy is the reduced effort involved in data collection as only samples of the identified graphemes are required. In its simplest form, this strategy does not address stroke order variations across symbols in the character, symbol order variations, co-articulation effects, and the cases where opaque symbols are created by CC and CV combinations. It is highly effective when constraints may be imposed on the writer to aid recognition.

The last strategy of treating complex characters directly as pattern classes has to deal with their large numbers. It also requires large quantities of data for training. Consequently this strategy has been explored chiefly for simple characters such as isolated vowels and consonants in different Indic scripts. It has also been used for Tamil characters, which as mentioned earlier are limited in number due to the linearization of consonant conjuncts [15, 16, 12, 17]. An obvious advantage of this strategy is that standard character recognition techniques in the literature may be used without much knowledge of the structure of the script, as is evident from the results of the Tamil character recognition competition organized in conjunction with IWFHR-10 [18]. Joshi et al. [19] assume that Devanagari can be "linearized" like Tamil by constraining writers to unravel consonant clusters into sequences of vowel-muted consonant characters. The linearization assumption reduces the number of characters to 441 - still a large number - when vowel modifiers are not considered. An accuracy of around 90% is reported for writer-dependent recognition using the subspace based method.

4.2 Preprocessing

Preprocessing techniques for Indic scripts are similar to those used for other scripts. Dehooking, smoothing, resampling, size normalization etc. are commonly performed. In the work by Swethalakshmi [13], different types of size normalization for Devanagari strokes are investigated. Genetic Programming has also been explored as a way of designing an optimal scaling function that reduces classification error [20].

Nonlinear normalization which has been found to be effective for CJK scripts does not appear useful for Indic scripts [13]. In the case of Devanagari, the *shirorekha* is often detected and removed prior to recognition. During word recognition the shirorekha also serves as a valuable cue for detecting reference lines.

4.3 Features

Features used for Latin scripts have been found to be useful for Indic scripts as well. Low-level features such as the normalized x and y coordinates have been widely used. Additional features such as normalized first and second derivatives and curvature have shown promising results for Tamil and Telugu [21, 22]. Structural features such as cusps, bumps, loops and semi-loops have also been explored for Tamil [12, 23] and Devanagari [13] character recognition. In some early work on Tamil character recognition [24], angle features, Fourier coefficients and Wavelet features are compared using a Neural Network classifier. Angle features are shown to be susceptible to noise leading to high intra-class variability. On the other hand, Fourier coefficients do not capture subtle differences between two characters as the change in the values of x and y over a small interval of time gets nullified over the entire frequency domain. Wavelet features are shown to be the most effective for Tamil as they retain both the intra-class similarity and inter-class differences. In general, directional coding approaches popular for CJK scripts are not effective for Indic scripts since the strokes do not have simple shapes.

In the work of Toselli et al. [21], a combination of time-domain and frequency-domain features is shown to improve recognition accuracy on Tamil characters. For Telugu character recognition, Rao and Ajitha [25] propose the use of x and y extrema, direction of pen motion (clockwise/anticlockwise) and relative displacement from the previous point of the same extrema category (x or y).

Offline features which model the input as a raster image rather than a trajectory may also be used to improve recognition accuracy when compared to using online features alone [26]. Being invariant to stroke order, number and direction variations, offline features are very promising in the context of Indic script recognition.

4.4 Classification

A number of different classification techniques have been applied to the problem of Indic script character recognition. While some approaches [13] are aimed at recognizing all the characters in the script, others [26, 14] only address specific subsets. These classification approaches may broadly be categorized as follows:

- Template matching
- Rule-based approaches
- Neural Networks
- Hidden Markov Models
- Subspace-based approach

Template matching - In this approach, the features extracted from the test character are compared with those of stored prototypes or templates. The test character is assigned the label of the template that is most similar to it. The templates could either be samples selected from the training set or a categorical representation [27]. In the context of Indic script recognition, efforts based on template matching are aplenty and a majority of them have reported encouraging results. For example, a two stage classification scheme [17] using Nearest Neighbor classifiers is described for *writer-dependent* Tamil character recognition. The first stage filters the templates based on the Euclidean distance from the test sample, and the second stage computes the more expensive Dynamic Time Warping (DTW) distance (Fig. 8) from the shortlisted templates. The label of the nearest template is assigned to the test sample.

The same scheme has also been applied for *writer-independent* Telugu and Tamil character recognition using a different set of features [22]. For the problem of Telugu character recognition, Rao and Ajitha [25] perform a coarse matching with the templates using the number of X-Y extrema points in the test sample and a fine matching using Dynamic Programming.

In another effort on Tamil character recognition [28], templates are identified from the training set using Agglomerative Hierarchical Clustering and Learning Vector Quantization (LVQ) with DTW as the distance measure. A DTW-based Nearest Neighbor classifier is then employed for matching the test sample.

Rule-based approaches - These approaches do not have an explicit training phase; instead they exploit human knowledge about the problem. The task of classification now becomes a deterministic verification procedure. Although the approach has the advantage of requiring minimal training data, it suffers from being labor-intensive and highly script-specific. Another disadvantage is that the approach typically does not provide alternate recognition choices. In a

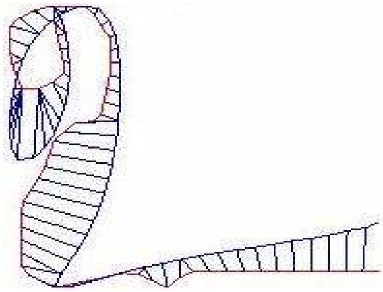


Fig. 8. Dynamic Time Warping (DTW) for matching two samples of a Tamil character

recent effort on Devanagari character recognition [13], strokes are first classified using Support Vector Machines (SVM) and predefined rules are then used for grouping the stroke labels into characters.

In the work of Ranade [27] for the Devanagari script, a set of stroke templates is derived from analysis of common writing styles of different Devanagari characters, and each character represented by a set of combinations of these stroke templates.

In another effort [12], Tamil strokes are represented as strings of shape features. In order to recognize an unknown stroke, its equivalent feature string is computed. The test stroke is then identified by searching the database using a flexible string matching algorithm. Once all the strokes in the input are known, the character is determined using a Finite State Automaton.

Prior knowledge about popular writing styles has also been exploited to design a first stage classifier for Tamil characters [23]. The authors observe that the start of any Tamil character is either a line, semi-loop or a loop. Accordingly, the candidate choices are pruned during recognition.

Neural Networks - In the work of Kunte and Samuel [29], Feed-forward Neural Networks with a single hidden layer are used for the recognition of handwritten Kannada characters. The authors use approximation coefficients derived from Wavelet decomposition on the preprocessed (x,y) as features for representing characters. The input character is initially classified into its consonant group (defined as a consonant and any of its vowel combinations), and separate neural networks are used for further classification within the consonant group.

The neural network based approach is also adopted by Sundaresan and Keerthi [24] for the recognition of online Tamil characters. Their work compares the performance of Time Delay Neural Network (TDNN) and a single hidden layer network for the classification task. Due to the presence of similar-looking characters and high dimensionality of the input, TDNN exhibits poor performance when compared to the single hidden layer network. The work also studies the relevance of different features such as (x,y) coordinates, sequence of directions, curvature, sequence of cosine angles, and wavelet features.

Another example is the use of Multi-layer Perceptrons (MLP) trained on eight-direction code histogram features for the problem of Bangla character recognition [30].

Hidden Markov Models - While HMMs have been used widely for English character recognition, they have seen limited application for Indic scripts. Connell et al. [26] use a combination of two HMM classifiers trained with online features and three Nearest Neighbor classifiers each trained on different sets of offline features for Devanagari character recognition. The combination of online and offline classifiers is shown to improve the accuracy from 69.2% (online, HMM alone) to 86.5%.

HMMs have also been used for Telugu [14] and Tamil [21] character recognition. Each character is modeled as a left-to-right HMM and a combination of time-domain and frequency-domain features is shown to result in accuracies higher than using either of them alone.

Subspace based approach - In this approach, 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 considered to represent the model for that class. During recognition, the test sample is projected onto each subspace and the class corresponding to the one that is closest is declared as the recognition result (Fig. 9). Joshi et al. [19] apply the subspace method to writer-dependent Devanagari character recognition. The shirorekha and vowel modifiers are preclassified using heuristics before recognizing the core character using the subspace method. The approach has also been employed for Tamil character recognition [15]. In these efforts, the feature vectors used for eigenanalysis are composed of the normalized (x,y) coordinates extracted following equi-spaced resampling of the character trajectory. Consequently different feature dimensions may not have stable interpretations (e.g. as salient points along the trajectory), and the method as used is outperformed by DTW-based template matching [16].

Recognition accuracies of some of the systems described above are shown in Table 2. The accuracies in the case of Devanagari are not comparable as the datasets used and the classes defined differ widely.

5 Word Recognition

Word recognition for Indic scripts is a nascent area of research, and published literature on the topic is limited. The general strategies for word recognition for Indic scripts may be classified broadly along the lines of those available for Latin scripts [32–35] into analytic approaches based on explicit segmentation, those based on implicit segmentation, and holistic approaches.

Table 2. Reported accuracies for isolated character recognition

System	**Number of Classes**	**Features**	**Classification**	**Accuracy**
		Devanagari		
Swethalakshmi [13]	123 strokes	X-Y coordinates, Fourier descriptors and structural features	SVM and rule-based stroke regrouping	89.88%
Connell et al. [26]	40 simple characters	Online and offline	Combination of HMM and Nearest Neighbor classifiers	86.5%
		Tamil		
Vision Objects [18]	156 characters	Online and offline	Neural Networks	93.53%
Bulacu M. [18]	156 characters	X-Y coordinates	Nearest Neighbor Classifier using Dynamic Time Warping (DTW) distance	91.20%
Toselli et al. [18, 21]	156 characters	Time-domain and frequency-domain	HMM	90.72%
		Telugu		
Prashanth et al. [22]	141 core characters and ligatures	X-Y coordinates, normalized first & second derivatives and curvature	Nearest Neighbor Classifier using Euclidean and DTW distance	89.77%
Babu et al. [14]	141 core characters and ligatures	Time-domain and frequency-domain	HMM	91.6%
		Bangla		
Bhattacharya et al. [30]	50 simple characters	8-direction code histogram	MLP	83.61%

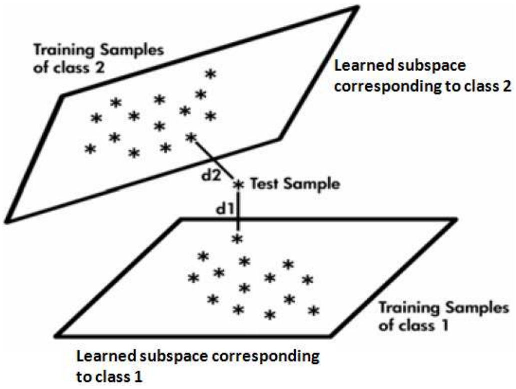

Training Samples of class 2
Learned subspace corresponding to class 2
d2
Test Sample
d1
Training Samples of class 1
Learned subspace corresponding to class 1

Fig. 9. Subspace based approach for character recognition [31]

5.1 Preprocessing

When part of larger systems [36], line and word segmentation are needed in order to isolate words. There has not been much work in the area of layout analysis or page segmentation specially directed at Indic scripts. In general since writing is structured into lines and lines into words, strategies used for English may be adapted for Indic scripts. In the case of Devanagari, the shirorekha may be used to simplify segmentation of lines into words.

Given an isolated word sample, common preprocessing steps performed at the word level include the determination of reference lines, baseline skew correction, and character slant correction. For a script such as Devanagari, reference lines generally delineate the upper matra zone, the middle zone and the lower matra zone (Fig. 10). Shirorekhas where present may be used for determination of reference lines and are also frequently removed before further processing.

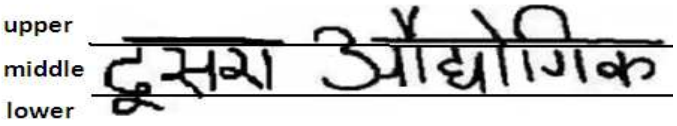


Fig. 10. Reference lines identified during Devanagari word preprocessing

5.2 Analytic approaches based on explicit segmentation

Explicit segmentation strategies are based on explicitly segmenting the handwritten word into simpler units, recognizing those units and thereby deriving the graphemes or character identities, typically by imposing some constraints on the writer. Explicit segmentation is often error prone because of Sayres paradox

[32]: “To recognize a letter, one must know where it starts and where it ends; to isolate a letter, one must recognize it first”. Some efforts that adopt this approach [19, 37] presuppose that Indic supports may be written as a sequence of space separated or boxed characters. As already observed, this is not natural given that syllabic units can vary greatly in size and complexity. Further, there may be multiple ways of decomposing syllabic units into simpler symbols, and different possible symbol orders adopted while writing in boxed form (Fig. 11). However preliminary studies suggest that “standard” ways of decomposing complex characters and ordering symbols can be successfully learned as part of a training phase [37].

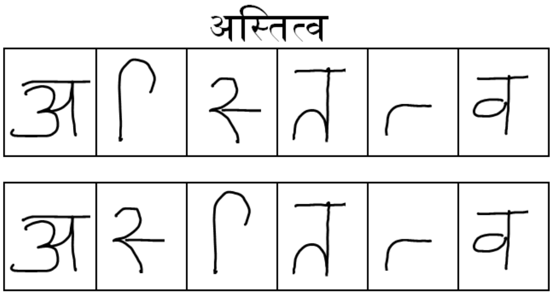


Fig. 11. Alternate ways of writing the Devanagari word */asthithva/* in boxed form

The stroke-based approach adopted by Swethalakshmi [13] does not require boxed input, but performs explicit segmentation based on proximity analysis for grouping strokes into characters. Proximity analysis is conventionally carried out based on the spatial information between consecutive strokes in a character. For scripts such as Devanagari where the most proximal strokes need not be consecutively written (e.g. /ko/, ‘को’), a two-stage proximity analysis is proposed. In the first stage, consecutive strokes are grouped together based on the spacing between their bounding rectangles. In a second stage, character-like units whose bounding box overlap with each other are grouped together. Any residual oversegmentation or undersegmentation of characters is resolved in a postprocessing step when the stroke labels are used to classify the characters. Given the interpretation of individual syllabic units, the best word level interpretation is determined using a lexicon or other language model.

5.3 Analytic approaches based on implicit segmentation

Approaches based on implicit segmentation address the segmentation problem implicitly by adopting a strategy of “over-segmentation”. The word input is segmented into a number of small segments with the only criterion that none of the segments should contain parts of more than one symbol. In other words, the intention is that the actual segmentation points are always covered as a subset of the segmentation boundaries obtained by over-segmenting the word. Analytic approaches based on implicit segmentation typically use models of an underlying set of symbols to analyze the word sample to recognize the word. Segmentation is a byproduct of recognition, and for this reason, this approach is also known as ‘recognition-based segmentation’. The two most popular techniques employed for implicit segmentation are Dynamic Programming (DP) (Fig. 12) and Hidden Markov Models (HMM). Segmentation is performed implicitly by aligning the input with a dictionary word model composed from the symbol models.

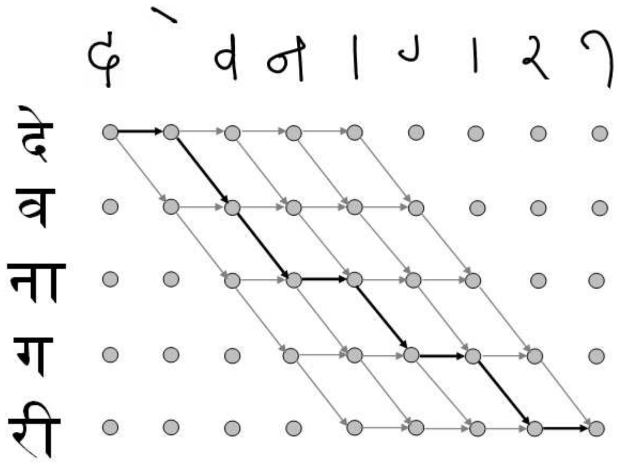


Fig. 12. Implicit segmentation at stroke boundaries based on Dynamic Programming. The arrows illustrate all the possible paths for matching the Devanagari word */devanagari/*. The black arrows show the minimum-cost path.

An example of this class of approaches applied to Tamil word recognition is that of the authors [38] which uses a set of 83 C, V and “opaque”’ CV characters in Tamil as the underlying symbol set. Characters are not modeled explicitly; the lexicon is represented as a network of HMMs corresponding to these symbols. The system uses normalized y-value, normalized first and second derivatives, writing angle and pen-up/down information as features. Stroke order variation within a symbol is captured implicitly by the symbol model; however symbol order variation is not captured; instead a standard symbol order is assumed. This assumption is broadly true for Tamil which is more linear than two-dimensional, but may not hold for other scripts such as Devanagari. This approach also assumes that one symbol is completed before another is initiated; in practice, strokes of different symbols may be temporally intermingled in creating the character (e.g. the delayed stroke in Fig. 7). Under the constraints mentioned, recognition accuracy of 98% is obtained for lexicons of size 1000, decreasing to 92% on lexicons of size 20000.

5.4 Holistic approaches

Techniques that follow the segmentation-free or holistic approach circumvent the problem of segmentation completely. Feature vectors are extracted at the word level and the most similar lexicon word is considered the recognition result. The training of holistic recognition algorithms is inseparable from the lexicon because adding a new word to the lexicon typically requires retraining the recognizer with samples of that word. As the lexicon size increases, similarities between words increase and hence the recognition accuracy deteriorates. However, the approach has its own advantages. As it uses the global word shape, it is less sensitive to local shape variations that are common in unconstrained writing. To the best of our knowledge, there is no Indic script recognition system that adopts this approach. However there has been some research on synthesis of words in Indic languages [39]. A collection of handwritten documents may be searched by aligning handwriting synthesized from search keywords, with handwritten words in the documents, using a DTW-like algorithm.

5.5 Language models

Lexicons and other language models are an important aspect of achieving acceptable accuracy for Online HWR. Barring a few reported methods that use lexicons, the use of language models has not been explored substantially for Online HWR of Indic scripts. In general these may be derived from available text corpora for Indic languages [40, 41], or from increasingly available web content especially in the form of online versions of newspapers and magazines. In addition, a number of such resources are being created by research communities working on Natural Language Processing, Machine Translation, Optical Character Recognition, Offline Handwriting Recognition and Automatic Speech Recognition for Indic languages.

Unicode [42] is the most popular standard for representing strings in Indic languages, although ISCII [43] is still available. One of the issues with Unicode is that it imposes the phonological symbol ordering - for example a CV is expressed as a C followed by a V. This can be an issue when the V is actually written before the C.

A second issue especially with languages such as Sanskrit is that of compounds words, which makes it difficult to enumerate all possible words. Also in the case of lexicon-driven and stroke-based approaches, where each grapheme has different stroke order representations, the number of possible definitions of a lexicon word in terms of stroke labels tends to increase exponentially. This could be an important problem to address in implementing such an approach.

6 Applications

The principal application of Online HWR is to support input of text into devices of various form factors, by providing an alternative to hard and soft keyboards.

Online HWR may also be used (in conjunction with ink parsing and layout analysis techniques) for subsequent interpretation of handwritten notes captured as digital ink - either to support transcription, or indexing and keyword-search. Online HWR may also be used to complement or supplement speech input as a part of multimodal interfaces and systems [44, 45].

While the recognition of unconstrained handwriting is clearly the end-objective of research, a number of practical Indic text input solutions (also called Input Method Editors or IMEs) can be obtained by imposing constraints on the writer. Further, since text input is interactive, partial recognition may be performed periodically during the course of writing a word or character rather than at the end.

One such constraint is symbol order, imposing the phonological order of the occurrence of consonant and vowels in the syllable: for example, constraining the writer to complete the consonant (C) before the vowel diacritic (V) while writing a CV character (even when the vowel diacritic appears to the left of the consonant, and is typically written before). This allows a two-step IME in which the consonant is first written and recognized (and manually corrected by selecting from an n-best list), and the vowel diacritic written and recognized in a second step. A second constraint could be to ask the user to unravel consonant clusters or conjuncts into sequences of CH characters using the vowel muting diacritic, rather than by using the half forms or consonant conjuncts. When used together, these constraints allow recognition of complex characters such as C'C'CV using only isolated symbol recognizers for C and V. Such strategies are used by IMEs such as the Gesture Keyboard [31], Guided Handwriting [46], Compact IME [47] and IndicDasher [48]. The Gesture Keyboard uses a combination of coarse pen position in a grid of base consonants, and recognition of the written matra, to interpret the user's CV input (Fig. 13). Guided Handwriting uses a set of basic stroke and substroke shapes together with rules to predict a consonant even as it is being written. In contrast to these approaches which only recognize isolated consonants or matras, FreePad [49] is an IME based on word recognition which assumes phonological symbol order and pen lifts between symbols, but does not require any explicit indication of symbol transition. Despite supporting "eyes-free" input using a stylus or finger, recognition accuracy is reported to be on par with conventional handwriting input, at least for Tamil.

Another specialized application of online handwriting recognition is pen-based form filling. These forms may be electronic in nature or printed on paper for use with a device capable of capturing electronic ink. In either case, they are filled by hand, and the handwriting captured as digital ink and recognized. As mentioned earlier, boxed styles are not natural for Indic scripts since characters may be of different sizes, but users may be trained to adopt a standard decomposition and ordering [37]. In the general case and to support walk-up use, form filling applications would need to support recognition of words and phrases with all of the attendant variations and issues. Electronic forms may alternatively be filled using an appropriate IME.

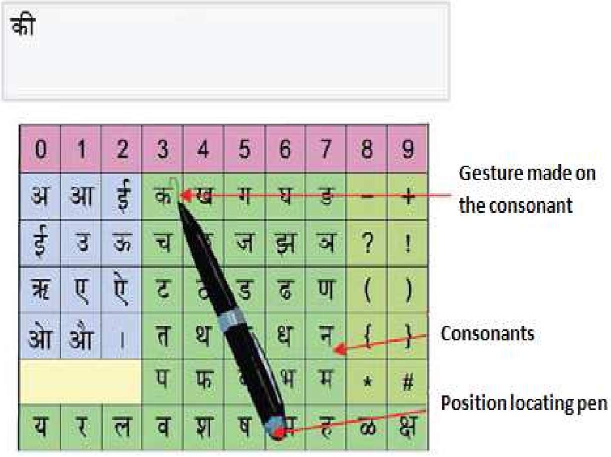


Fig. 13. Gesture keyboard for entering Devanagari characters [31]

The PATRAM system [36] is an interactive handwritten notes capture application for Indic languages. It allows a handwritten document to be created, edited, and exported into standard formats such as InkML. The system also integrates layout analysis and character recognition algorithms to interpret the document and a spell checker to correct mis-recognized words.

7 Resources

One of the major stumbling blocks for language technology research in the Indian context has been the lack of significant publicly available linguistic resources, and this is especially true for HWR. It is imperative that tools and data formats be standardized and validated datasets be created and made available to change the status quo. It should be mentioned that such datasets would also benefit research in handwritten document analysis, writer identification, script identification, handwritten document indexing and retrieval, and so forth.

Data collection is a challenge for Indic scripts due to the large number of characters. It is generally easy to collect symbols of the basic C and V characters, since they are a small number. However data collection should ideally cover all possible characters in order to capture different coarticulation effects and stroke and symbol order variations.

Different strategies have been used to address this challenge in practice. For CV characters, one common strategy is to collect samples of isolated matras under the assumption that the shape does not change as a function of the consonant being modified. As seen earlier, there are clearly several exceptions that this strategy does not address. Even where this assumption holds true, it is important to capture the relative position of the matra with respect to the base consonant. Therefore another possible strategy is to pre-print or render the base consonants and instruct writers to mark only the matras over them [14].

Data collection to support word recognition is often optimized to suit the specific recognition approach, for example, by identifying a minimal set of words that covers all the basic graphemes used. A Set Cover algorithm may be employed to discover the optimal set of words from a text corpus [50]. The word data thus collected still needs to be annotated at the grapheme level in order to provide training samples for the grapheme models. In order to avoid the labor-intensive task of labeling the collected word samples at the character level, Dynamic Programming has been explored to propagate the word truth to the constituent characters [51]. Propagation of manual character or grapheme-level annotation between samples of the same word written by the same writer has also been explored [52]. Some of the practical challenges encountered in creating a handwriting corpus for an Indic script are described by Agrawal et al. [53].

7.1 Dataset standards

Standards are important for the creation of handwriting datasets, in order to ensure that resources created can be used by others. UNIPEN 1.0 [54] is still the de facto standard for encoding of handwriting data because of its simplicity and widespread use. However it suffers from some shortcomings [55] and hence new standards such as hwDataset [52] and UNIPEN XML (UPX) [55] have been proposed. UPX in particular is being seen as an XML-based successor to the (ASCII-based) UNIPEN standard. UPX provides a hierarchical and structured solution for the representation of online handwriting data and uses W3C Digital Ink Markup Language (InkML) [56] as the underlying representation of digital ink. However the use of UPX has been hampered by the lack of implementations and tools; this is expected to change in 2008 with InkML itself expected to become a standard with available implementations.

A second aspect of standardization is the development of conventions for the annotation hierarchy and labels to be used for annotating handwriting in Indic scripts. It is clear that a standard should minimally allow markup of words, syllabic units, the core graphemes and perhaps strokes. While there is agreement at the word and syllabic unit level because of their linguistic significance, no standards are as yet available for the core graphemes or for strokes. There is no consistent definition of these units, and also a clear dependence on the recognition strategy, since for example certain CV characters or CC conjuncts may be treated as opaque graphemes for the purposes of recognition. Hence a requirement of the standard may be the ability to support customizations of a standard annotation hierarchy, especially below a certain level.

Since Unicode for Indic scripts [57] is an encoding of the sound of the word, it may at best be used for labeling at the level of words and syllabic units. Unicode labels are clearly insufficient to encode lower level visual structures that a recognition strategy may adopt. For example, Unicode does not have any labels for half-consonant forms, since these do not occur in isolation in the phonological sense.

7.2 Tools

The UNIPEN Project [54] created tools for visualization and annotation of English handwriting data in the UNIPEN 1.0 format, as well as a library of preprocessing functions. While the original project is no longer active, the tools are still available [58] and usable for Indic scripts.

The Lipi Toolkit Project [59, 60] provides open source tools and recognizers for Online HWR of Indic scripts. The focus at the present time is recognition of isolated characters. The tools and recognizers are in fact generic and may be used for arbitrary character/symbol sets. The project provides handwriting data collection tools for different devices such as TabletPCs and ACECAD DigiMemo [61], an inexpensive paper-based electronic clipboard device which allows handwriting capture simultaneously in the form of digital ink and paper. The toolkit uses the UNIPEN format for storage of ink data. The "Core Toolkit" provides common preprocessing operations and feature extractors, and a k-NN classifier based on DTW distance. The project also provides an interactive recognizer development (IDE) tool which allows creation of a simple recognizer using a few training samples, as well as "pre-built" recognizers for different scripts. An annotation tool for Indic (and other) scripts that supports UNIPEN, W3C InkML and hwDataset/UPX [52] is likely to be released as part of the Lipi Toolkit Project once InkML and UPX are standardized.

7.3 Datasets

Publicly available datasets of handwriting at the time of writing are limited to those used for the IWFHR-10 Tamil character recognition competition [62] and some datasets of simple characters such as core consonants [63]. A partial list of available resources for Indic (and other) languages is hosted at the Lipi Toolkit project website [64].

The Technology Development for Indian Languages (TDIL) program of the Ministry of Information Technology of the Government of India has recently funded a consortium of universities to create resources as well as technology for Online HWR of Indic scripts [65]. The consortium is collecting word data for several Indian languages and scripts. It is hoped that many of the tools and datasets created in the process will become publicly available for research purposes.

8 Summary

Online Handwriting Recognition for Indic scripts is still in its infancy. Given the large number of scripts and millions of users, two-dimensional layout, and stroke and symbol order variations, there are a number of aspects that make the problem different from Latin and CJK scripts, and worthy of serious investigation. The benefits of developing such a technology are also very significant, given the large user population both within and outside the Indian subcontinent, and the lack of standardized or convenient keyboard-based text input solutions.

Much research is needed on recognition of complex characters and complete words in the different Indic scripts. Given the issues of stroke order, direction and number variations in Indic scripts, offline features and approaches stand a good chance of improving recognition accuracy when used in combination with online features and approaches. For continuous handwriting input to become a reality, research on language models is imperative as well. There is also a clear need that approaches to recognition be as data-driven as possible, given the challenges of covering a large number of languages and scripts. There is an urgent need for standardized datasets using well-defined sets of characters and graphemes to be created and made available for all Indic scripts, in order to allow meaningful comparison of different published approaches. Resources such as tools and datasets for a few scripts are gradually becoming available in the public domain, and should provide a significant fillip to research. The next five to ten years are likely to produce interesting new approaches and new levels of performance, hopefully leading to widespread adoption of Online HWR technology, especially for mobile devices.

9 Acknowledgements

We gratefully acknowledge the contributions and influence over the years, of various present and former colleagues at HP Labs India - notably Deepu Vijayasenan, M. Dinesh, Ajay Bhaskarabhatla, Mudit Agrawal, Kalika Bali and K.S.R. Anjaneyulu, collaborators Prof. A. G. Ramakrishnan at the Indian Institute of Science (IISc), Prof. Srinivasa Chakravarthy at Indian Institute of Technology-Madras (IIT-M), and Prof. C. V. Jawahar at International Institute of Information Technology-Hyderabad (IIIT-H), their students, and various interns who have worked with us at some time or the other.

References

1. List of Languages by Number of Native Speakers, `http://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers`
2. Tappert, C., Suen, C., Wakahara, T.: State of the Art in On-line Handwriting Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **12**(8) (1990) 787–808
3. Plamondon, R., Srihari, S.N.: Online and Off-line Handwriting Recognition: A Comprehensive Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **22**(1) (2000) 63–84
4. Liu, C.L., Jaeger, S., Nakagawa, M.: Online Recognition of Chinese Characters: The State-of-the-Art. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **26**(2) (2004) 1489–1500
5. Jaeger, S., Liu, Nakagawa, M.: The State of the Art in Japanese On-Line Handwriting Recognition Compared to Techniques in Western Handwriting Recognition. International Journal on Document Analysis and Recognition (IJDAR) **6**(2) (2003) 75–88

6. Coulmas, F.: The Blackwell Encyclopedia of Writing Systems. Blackwell, Oxford (1996)
7. Mudur, S.P., Nayak, N., Shanbhag, S., Joshi, R.K.: An Architecture for the Shaping of Indic Texts. Computers & Graphics **23**(1) (1999) 7–24
8. Tamil Script, `http://en.wikipedia.org/wiki/Tamil_script`
9. Bharath, A., Deepu, V., Madhvanath, S.: An Approach to Identify Unique Styles in Online Handwriting Recognition. In: 8th International Conference on Document Analysis and Recognition (ICDAR 2005), Seoul, Korea (Aug-Sep 2005)
10. Swethalakshmi, H., Jayaraman, A., Chakravarthy, V.S., Sekhar, C.C.: Online Handwritten Character Recognition of Devanagari and Telugu Characters using Support Vector Machines. In: 10th International Workshop on Frontiers in Handwriting Recognition (IWFHR 2006), La Baule, France (October 2006)
11. Jayaraman, A., Sekhar, C.C., Chakravarthy, V.S.: Modular Approach to Recognition of Strokes in Telugu Script. In: 9th International Conference on Document Analysis and Recognition (ICDAR 2007), Curitiba, Brazil (September 2007)
12. Aparna, K.H., Subramanian, V., Kasirajan, M., Prakash, G.V., Chakravarthy, V.S., Madhvanath, S.: Online Handwriting Recognition for Tamil. In: 9th International Workshop on Frontiers in Handwriting Recognition (IWFHR 2004), Tokyo, Japan (October 2004)
13. Swethalakshmi, H.: Online Handwritten Character recognition for Devanagari and Tamil Scripts using Support Vector Machines. Master's thesis, Indian Institute of Technology, Madras, India (October 2007)
14. Babu, V.J., Prasanth, L., Sharma, R.R., Rao, G.V.P., Bharath, A.: HMM-Based Online Handwriting Recognition System for Telugu Symbols. In: 9th International Conference on Document Analysis and Recognition (ICDAR 2007), Curitiba, Brazil (September 2007)
15. Deepu, V., Madhvanath, S., Ramakrishnan, A.G.: Principal Component Analysis for Online Handwritten Character Recognition. In: 17th International Conference on Pattern Recognition (ICPR 2004), Cambridge, United Kingdom, (August 2004)
16. Joshi, N., Sita, G., Ramakrishnan, A.G., Madhvanath, S.: Tamil Handwriting Recognition Using Subspace and DTW Based Classifiers. In: 11th International Conference on Neural Information Processing (ICONIP 2004), Calcutta, India (November 2004)
17. Joshi, N., Sita, G., Ramakrishnan, A.G., Madhvanath, S.: Comparison of Elastic Matching Algorithms for Online Tamil Handwritten Character Recognition. In: 9th International Workshop on Frontiers in Handwriting Recognition (IWFHR 2004), Tokyo, Japan (October 2004)
18. Madhvanath, S., Lucas, S.M.: IWFHR 2006 Online Tamil Handwritten Character Recognition Competition. In: 10th International Workshop on Frontiers in Handwriting Recognition (IWFHR 2006), La Baule, France (October 2006)
19. Joshi, N., Sita, G., Ramakrishnan, A.G., Deepu, V., Madhvanath, S.: Machine Recognition of Online Handwritten Devanagari Characters. In: 8th International Conference on Document Analysis and Recognition (ICDAR 2005), Seoul, Korea (Aug-Sep 2005)
20. Deepu, V., Madhvanath, S.: Genetically Evolved Transformations for Rescaling Online Handwritten Characters. In: IEEE India Annual Conference (INDICON 2004), Kharagpur, India (December 2004)
21. Toselli, A.H., Pastor, M., Vidal, E.: On-Line Handwriting Recognition System for Tamil Handwritten Characters. In: Pattern Recognition and Image Analysis. Springer Berlin / Heidelberg (July 2007) 370–377

22. Prasanth, L., Babu, V.J., Sharma, R.R., Rao, G.V.P., Dinesh, M.: Elastic Matching of Online Handwritten Tamil and Telugu Scripts using Local Features. In: 9th International Conference on Document Analysis and Recognition (ICDAR 2007), Curitiba, Brazil (September 2007)
23. Sundaram, S., Ramakrishnan, A.G.: A Novel Hierarchical Classification Scheme for Online Tamil Character Recognition. In: 9th International Conference on Document Analysis and Recognition (ICDAR 2007), Curitiba, Brazil (September 2007)
24. Sundaresan, C.S., Keerthi, S.S.: A Study of Representations for Pen based Handwriting Recognition of Tamil Characters. In: 5th International Conference on Document Analysis and Recognition (ICDAR 1999), Bangalore, India (September 1999)
25. Rao, P.V.S., Ajitha, T.M.: Telugu Script Recognition - A Feature based Approach. In: 3rd International Conference on Document Analysis and Recognition (ICDAR 1995), Montreal, Canada (August 1995)
26. Connell, S.D., Sinha, R.M.K., Jain, A.K.: Recognition of Unconstrained On-Line Devanagari Characters. In: 15th International Conference on Pattern Recognition (ICPR 2000), Barcelona, Spain (September 2000)
27. Ranade, A., Ranade, M.: Devanagari Pen-written Character Recognition. In: 9th International Conference on Advanced Computing and Communications (ADCOM 2001), Bhubaneshwar, India (December 2001)
28. Niels, R., Vuurpijl, L.: Dynamic Time Warping Applied to Tamil Character Recognition. In: 8th International Conference on Document Analysis and Recognition (ICDAR 2005), Seoul, Korea (Aug-Sep 2005)
29. Kunte, R.S.R., Samuel, R.D.S.: On-line Character Recognition System for Handwritten Characters/Script with Bilingual Facility Employing Neural Classifiers and Wavelet Features. In: International Conference on Knowledge based Computer Systems (KBCS 2000), Mumbai, India (December 2000)
30. Bhattacharya, U., Gupta, B.K., Parui, S.K.: Direction Code based Features for Recognition of Online Handwritten Characters of Bangla. In: 9th International Conference on Document Analysis and Recognition (ICDAR 2007), Curitiba, Brazil (September 2007)
31. Balaji, R., Deepu, V., Madhvanath, S., Prabhakaran, J.: Handwritten Gesture Recognition for Gesture Keyboard. In: 10th International Workshop on Frontiers in Handwriting Recognition (IWFHR 2006), La Baule, France (October 2006)
32. Vinciarelli, A.: A survey on Off-line Cursive Word Recognition. Pattern Recognition **35**(7) (2002) 1433–1446
33. Lecolinet, E., Baret, O.: Cursive Word Recognition: Methods and Strategies. In: Fundamentals in Handwriting Recognition. Springer-Verlag, New York (1994) 235–263
34. Steinherz, T., Rivlin, E., Intrator, N.: Offline Cursive Script Word Recognition - A Survey. International Journal on Document Analysis and Recognition (IJDAR) **2**(2-3) (1999) 90–110
35. Madhvanath, S., Govindaraju, V.: The Role of Holistic Paradigms in Handwritten Word Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **23**(2) (2001) 149–164
36. Madduri, K., Aparna, K.H., Chakravarthy, V.S.: PATRAM - A Handwritten Word Processor for Indian Languages. In: 9th International Workshop on Frontiers in Handwriting Recognition (IWFHR 2004), Tokyo, Japan (October 2004)
37. Krishna, A., Prabhu, G., Bali, K., Madhvanath, S.: Indic Scripts based Online Form Filling - A Usability Exploration. In: 11th International Conference on Human-Computer Interaction (HCII 2005), Las Vegas, USA (July 2005)

38. Bharath, A., Madhvanath, S.: Hidden Markov Models for Online Handwritten Tamil Word Recognition. In: 9th International Conference on Document Analysis and Recognition (ICDAR 2007), Curitiba, Brazil (September 2007)
39. Balasubramanian, A.: Document Annotation and Retrieval Systems. Master's thesis, International Institute of Information Technology, Hyderabad, India (2006)
40. Indian Languages Corpora, `http://www.ciilcorpora.net/`
41. Baker, P., Hardie, A., McEnery, T., Cunningham, H., Gaizauskas, R.: EMILLE, A 67-Million Word Corpus of Indic Languages: Data Collection, Mark-up and Harmonisation. In: 3rd International Conference on Language Resources and Evaluation (LREC 2002), Canary Islands, Spain (May 2002)
42. The Unicode Consortium: The Unicode Standard Version 5.0. Addison Professional, MA (2006)
43. Indian Script Code for Information Interchange (ISCII), Ministry of Information Technology, Government of India, `http://tdil.mit.gov.in/standards.htm` (April 2003)
44. Manohar, P.: The Multimodal Interaction for the Computer: An Application-independent Approach. Master's thesis, Indian Institute of Technology, Madras, India (2006)
45. Barthelmess, P., Kaiser, E., McGee, D.: Toward Content-aware Multimodal Tagging of Personal Photo Collections. In: 9th International Conference on Multimodal Interfaces (ICMI 2007), Nagoya, Japan (November 2007)
46. Prasad, A., Prashant, A., Borgaonkar, S.: Guided Handwriting: Predictive Writing Input Method Environment. Internal technical report, HP Labs India (December 2005)
47. Manish Kumar: Compact Stylus-based Input Method for Indic Scripts. Diploma Thesis, National Institute of Design, Ahmedabad, India (2007)
48. Srinivas, N.K., Varghese, N., Raman, R.K.V.S.: IndicDasher: A Stroke and Gesture based Input Mechanism for Indic Scripts. In: Workshop on Intelligent User Interfaces for Developing Regions (IUI4DR 2008), Canary Islands, Spain (January 2008)
49. Bharath, A., Madhvanath, S.: Recognition of Eyes-free Handwriting Input for Pen and Touch Interfaces. In: To be published in 11th International Conference on Frontiers in Handwriting Recognition (ICFHR 2008), Montreal, Canada (August 2008)
50. Bhaskarabhatla, A.S., Madhvanath, S.: Experiences in Collection of Handwriting Data for Online Handwriting Recognition in Indic Scripts. In: 4th International Conference on Language Resources and Evaluation (LREC 2004), Lisbon, Portugal (May 2004)
51. Kumar, A., Balasubramanian, A., Namboodiri, A., Jawahar, C.V.: Model-based Annotation of Online Handwritten Datasets. In: 10th International Workshop on Frontiers in Handwriting Recognition (IWFHR 2006), La Baule, France (October 2006)
52. Bhaskarabhatla, A.S., Madhvanath, S., Kumar, M.N.S.S.K.P., Balasubramanian, A., Jawahar, C.V.: Representation and Annotation of Online Handwritten Data. In: 9th International Workshop on Frontiers in Handwriting Recognition (IWFHR 2004), Tokyo, Japan (October 2004)
53. Agrawal, M., Bhaskarabhatla, A.S., Madhvanath, S.: Data Collection for Handwriting Corpus Creation in Indic Scripts. In: International Conference on Speech and Language Technology and Oriental COCOSDA (ICSLT-COCOSDA 2004), New Delhi, India (November 2004)

54. Guyon, I., Schomaker, L., Plamondon, R., Liberman, M., Janet, S.: UNIPEN Project of Online Data Exchange and Recognizer Benchmarks. In: International Conference on Pattern Recognition (ICPR 1994), Jerusalem, Israel (October 1994)
55. Agrawal, M., Bali, K., Madhvanath, S., Vuurpijl, L.: UPX - A New XML Representation for Annotated Datasets of Online Handwriting Data. In: 8th International Conference on Document Analysis and Recognition (ICDAR 2005), Seoul, Korea (October 2005)
56. W3C Working Group: Ink Markup Language, `http://www.w3.org/2002/mmi/ink` (2003)
57. Richard Ishida: Unicode Technical Note #10: An Introduction to Indic Scripts, `http://unicode.org/notes/tn10/` (August 2003)
58. International Unipen Foundation: The Unipen Project, `http://www.unipen.org` (1994)
59. Madhvanath, S., Deepu, V., Kadiresan, T.M.: Lipitk: A Generic Toolkit for Online Handwriting Recognition. In: 10th International Workshop on Frontiers in Handwriting Recognition (IWFHR 2006), La Baule, France (October 2006)
60. The Lipi Toolkit, `http://lipitk.sourceforge.net`
61. ACECAD DigiMemo, `http://www.acecad.com.tw/products.html`
62. HP Labs Isolated Handwritten Tamil Character Dataset hpl-tamil-iso-char, `http://www.hpl.hp.com/india/research/penhw/resources/tamil-iso-char.html`
63. Handwritten Character Databases of Indic Scripts, `http://www.isical.ac.in/~ujjwal/download/database.html`
64. Online HWR Resources, `http://lipitk.sourceforge.net/resources.htm`
65. Online Handwriting Recognition System for Indian Languages (OHWR), `http://ragashri.ee.iisc.ernet.in/ohwr/`