Machine Recognition of Online Handwritten Devanagari Characters

Niranjan Joshi, G.Sita, A.G. Ramakrishnan
Indian Institute of Science
Bangalore, India
{joshi, sita, ramkiag}@ee.iisc.ernet.in

Deepu V., Sriganesh Madhvanath
Hewlett-Packard Laboratories
Bangalore, India
{deepuv, srig}@hp.com

Abstract

In this paper, we describe a system for the automatic recognition of isolated handwritten Devanagari characters obtained by linearizing consonant conjuncts. Owing to the large number of characters and resulting demands on data acquisition, we use structural recognition techniques to reduce some characters to others. The residual characters are then classified using the subspace method. Finally the results of structural recognition and feature-based matching are mapped to give final output. The proposed system is evaluated for the writer dependent scenario.

1. Introduction

This paper describes initial efforts at developing a recognition engine for online handwritten Devanagari script. The composition of characters and other modifier symbols in Devanagari script calls for a fresh look at preprocessing and recognition strategies specific to the script and a simple adaptation of algorithms developed for other scripts [3, 4] may not be sufficient. Devanagari script is a logical composition of its constituent symbols in two dimensions. It has eleven vowels and thirty three simple consonants. A horizontal line is drawn on top of all characters which is referred to as the header line or *shirorekha*. A character is usually written such that it is vertically separate from its neighbors. Devanagari script has many multi-stroke characters. The data entry/ recognition mechanisms need to deal with such multi-stroke characters and also conjuncts that are made up by joining two or more characters partially.

In this preliminary study, we assume segmented characters at the data entry level. The database collected for training the system too is acquired in individual boxes for each character with conjuncts being split and written in a linearized fashion. When so linearized, each character is of one of the following forms: C (consonant), V (vowel), N (numerals), CV (consonant modified by vowel sound) and CVM (M is a modifier that modifies the nature of the preceding vowel, e.g. nasalization). Other rare forms were omitted for the purpose of this study. Given |C| = 35, |V| = 11, |N| = 10, |M| = 4, and restricting ourselves to valid combinations, we are left with a total of 1487 characters to recognize. Some examples of each of these categories of characters are shown in Figure 1.

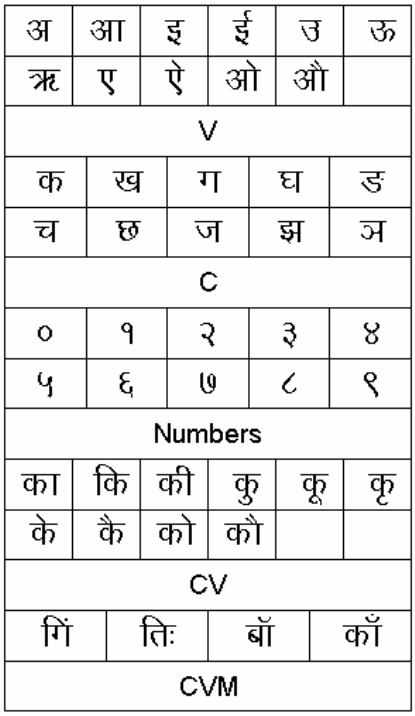

Figure 1. Examples of different forms of Devanagari characters. The vowel modifiers shown in the CVM category are the *anuswara*, the *visarga* the *ardhachandra* and the *chandrabindu.*

The rest of the paper is organized as follows. Section 2 describes some of the constraints imposed both on the handwriting input and the recognition methodology in order to simplify the complexity of data acquisition, and subsequent training and testing. Section 3 briefly describes the process used for acquiring handwriting

data for training and testing the system. Section 4 describes the recognition methodology adopted. Experimental results are presented in Section 5. The final section presents some conclusions and future directions.

2. Simplifying Constraints

In order to reduce the complexity of the problem, it was assumed that the vowel modifiers (M) always occurred as distinct strokes and could be segmented out of the character and recognized using structural techniques alone. This allowed the set of CVM and CCV forms to be reduced to the corresponding CV forms, and the characters to be recognized using statistical means to be reduced to around 440.

It was further assumed that the header line or *shirorekha* would generally be present, but would occur only as a single stroke.

These assumptions were found to hold for majority of writing samples examined as part of an initial empirical study of a small number of native Hindi writers.

3. Data Acquisition

The data for training and testing was acquired through an HP Tablet PC TC1000, using a GUI developed using Microsoft VC++. Each page of the GUI contains 15 writing blocks in which characters are to be written in an isolated manner.

र	क	न	स	प	के	त
का	ह	अ	ल	म	है	में
ने	व	की	ग	उ	य	से
ता	ए	ब	आ	या	कि	ना
ज	को	इ	रा	औ	वि	पृ
जा	द	मा	हो	सा	वा	भी
नि	लि	ती	ति	ही	ला	ा
ते	ई	हैं	था	री	च	हा
ण	ले	दे	बा	दि	हीं	ख
नी	भा	सं	रि	चा	ट	सी
पा	रे	हु	ये	गा	मि	तो
दा	जि	ली	थ	यों	ध	जी
धि	मु	भ	हि	यो	धा	रू
कु	त्र	खा	पू	ा	जो	छ
वे	मे					

Figure 2. 100 most frequently used characters in Devanagari, based on a study of the TDIL corpus

For the purpose of training, character samples were collected from each of 20 writers corresponding to each of the C, V, N, CV characters, and the M vowel modifiers. Seven instances of each character were collected from each writer, for a total of 7 sets of training data, which we will refer to as CORE1 … CORE7.

However since any of the 1487 characters could be presented to the system for recognition, the test data was collected differently. Two samples apiece of the 100 most common characters (Figure 2) from the 1487 were collected from each writer (FREQ). In addition, a sample each of 250 characters drawn randomly from the remaining set was also collected (RAND).

4. Recognition Methodology

The recognition methodology deployed is described in the flowchart in Figure 3, and can be broadly divided into three modules: Structural Recognition, Feature-based Recognition, and Output Mapping.

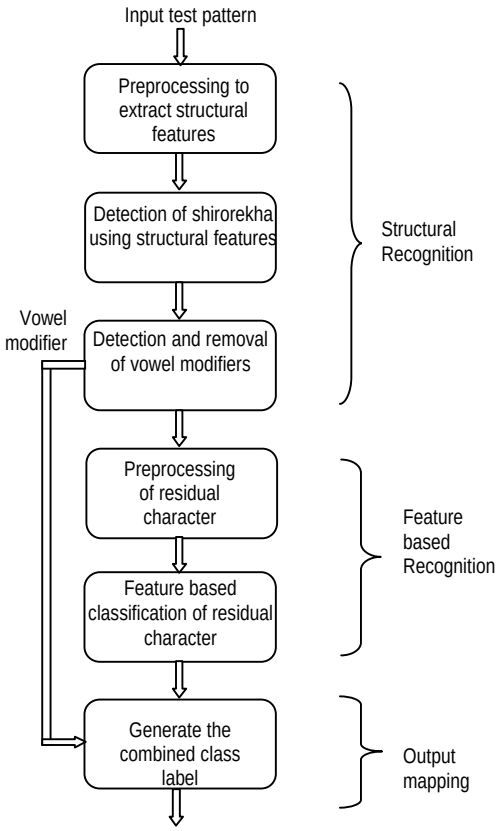

Figure 3. Recognition methodology

Each of the modules is described in the following subsections.

4.1 Structural Recognition

Syntactic and structural approaches have been adopted to recognize special strokes and vowel modifiers (M) such as the *shirorekha*, *anuswara*, *visarga* and *ardha chandra* (Figure 1*)*.

The input test pattern represented by a sequence of x-y coordinates is preprocessed to extract structural features at the stroke level, such as mean (x, y) value, length, offline features, positional cues and directional codes. Using these features, the *shirorekha* is first detected as follows:

For each stroke in the character, stroke statistics such as mean (x, y), length of stroke, discretized slope in 8-directions and frequency of each directional code are computed. If the frequency of either directional code 1 (East) or 5 (West) is more than 60% (an empirically determined threshold), then that stroke is added to the *Shirorekha* Candidate Set (SCS). If the SCS is empty, then *shirorekha* is not present. Otherwise, we compute the x-means of the bottom most strokes of both the entire character C and those in the *SCS.* If the x-mean of bottom most stroke of C equals the x-mean of bottom most stroke of *SCS*, then we remove the corresponding stroke from *SCS*, since *shirorekha* cannot occur at the bottom of a character. Finally, the stroke in the SCS with the maximum length is adjudged the *shirorekha* (Figure 4).

After this, vowel modifiers such as *anuswara*, *visarga*, *chandrabindu*, and *ardhachandra* are detected and removed from the original test pattern.

4.2 Feature-based Recognition

The basic feature based recognition procedure used is similar to that used for Tamil online handwritten character recognition [1].

The input character is first pre-processed and then features extracted for classification. Preprocessing includes resampling and low pass filtering. The data is re-sampled to obtain constant number of points in space rather than in time. Uniform resampling results in 60 points for all character samples. To reduce the effect of noise, the x and y coordinate sequences are separately smoothed using a 5-tap Gaussian filter.

Note that in this module preprocessing is done on the original input character after removal of special strokes as part of structural recognition. After removing the special strokes, the residual test pattern can assume any class label from the first 441 classes in the recognition set.

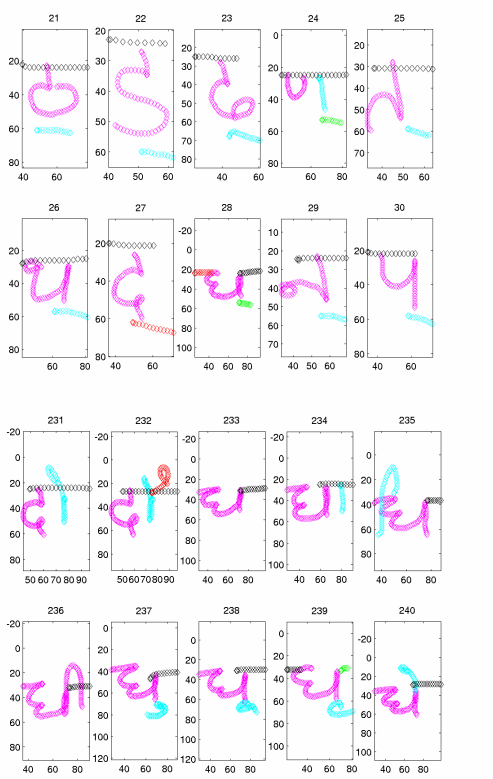

Figure 4. Strokes colored in black correspond to the detected *shirorekha*

The normalized x and y coordinate sequences are classified directly using the subspace method [2, 5]. The basis vectors for each class are computed as a set of N eigenvectors. Each eigenvector is normalized so that the basis is orthonormal. The distance measure of a test vector from a class is its orthogonal distance from the subspace defined by the basis vectors of that class. Since the basis vectors (eigenvectors) are orthogonal, the projection into the subspace is the sum of projections along the basis vectors, which can be computed as the dot product of the corresponding basis vectors with the test vector. Finally, the test vector is assigned to the ``nearest'' class, i.e., the one whose subspace is nearest to the test vector in terms of orthogonal distance.

4.3 Output Mapping

The outputs of structural recognition and feature based recognition are mapped to give the final output class label. The recognition set is ordered in such a manner that only an offset is required to be added to the class

label obtained through the feature-based recognition in order to compute the final class label.

5. Experimental Results

Experimental results are obtained for writer-dependent recognition for two different kinds of training and test setups:

In '*sequential data*' testing, we divide the 7 CORE data sets obtained earlier into 6 training sets and 1 testing set. This setup enables evaluation of performance of the feature based recognition algorithms only, because these sets do not contain any special strokes and hence structural recognition is not required.

User #	Sequential (CORE7)	Frequently Occurring Characters (FREQ)	Randomly Chosen characters (RAND)	Complete Set (FREQ + RAND)
0	89.32%	93.5%	79%	85.33%
1	94.09%	94%	81.62%	87.11%
2	99.09%	99%	92.00%	95.11%
3	95.45%	93.5%	84%	88.22%
4	97.27%	91%	90.4%	90.67%
5	99.09%	99%	84.6%	91.11%
6	96.59%	94%	89.2%	91.33%
7	91.14%	89%	83.2%	85.78%
8	91.14%	88.50%	81.20%	84.44%
9	91.59%	89%	90%	89.56%
Avg	**94.49%**	**93.05%**	**85.52%**	**88.87%**

Table 1a. Recognition results for sequential and complete set testing (shown separated into frequently occurring and randomly chosen) when shirorekha is removed

In '*complete set*' testing, we use all 7 CORE sets for training and the FREQ and RAND sets for testing. To get better insight of how algorithms work on frequently used characters and randomly chosen characters, results are presented separately as well as in combined form.

We also tried experimenting with recognizing the character with the *shirorekha*, as opposed to segmenting out the *shirorekha*.

All these results are presented for *writer dependent* case – the training and test data are all from the same writer.

The results from our experiments are as above. Each row of each table presents recognition results for a particular writer. The last row of each result table provides average accuracy over all writers.

For each writer, accuracy is given in terms of percentage of correctly recognized test patterns out of total test patterns from the appropriate test set. The CORE7 test set used for sequential testing contain 441 characters. The Frequently occurring character set (FREQ) has 200 characters. Similarly the randomly selected character set (RAND) has 250 test characters.

User #	Sequential	Frequently Occurring Characters	Randomly Chosen characters	Complete Set
0	93.65%	93.5%	80%	85.78%
1	93.56%	94.50%	78.80%	85.78%
2	97.73%	96%	86%	90.44%
3	94.77%	93%	80%	85.78%
4	95.68%	90%	90.8%	90.22%
5	99.32%	99%	84.6%	91.11%
6	95.45%	93%	84.4%	88.22%
7	87.5%	88%	75.2%	80.67%
8	94.09%	88%	78.8%	82.89%
9	93.18%	94%	87.2%	90.22%
Avg	**94.51%**	**92.80%**	**82.56%**	**87.11%**

Table 1b. Recognition results for sequential and complete set testing (shown separated into frequently occurring and randomly chosen) when shirorekha is not removed

Note that the recognition results reflect the sequential combination of structural recognition of *shirorekha* and vowel modifiers, with feature based recognition of the residuals.

The mean recognition accuracy on the core set of 441 characters for which six samples were used for training and one for testing was of the order of 95%. Total set testing is in comparison, a 1475-class problem and combines structural techniques for interpretation of vowel modifiers, with the feature-based statistical classification of the residuals. Here the average accuracy on the FREQ subset was of the order of 93%. The accuracy on the RAND set varied between 82% and 85% depending on whether the *shirorekha* was removed or retained in the character. This can be explained in the light of the fact that these characters tend to be more complex and have more

strokes in them. Hence the presence of the *shirorekha* as an additional stroke tends to increase the net variability in stroke order, and reduce accuracy of classification.

6. Conclusions

A scheme for recognition of online handwritten Devanagari characters is described wherein consonant conjuncts are broken down into individual consonant symbols. When linearized in this fashion, the Devanagari character set contains 1487 characters in all.

In order to reduce the search space to 441, a structural feature based algorithm is proposed. This module detects and removes special strokes and vowel modifiers such as *anuswara*, *visarga*, *chandrabindu*, *nukta* and *shirorekha*, and precedes the actual character recognition module.

The actual character recognition module uses resampled and normalized stroke coordinates directly as features and the subspace method for classification.

Performance of the recognition scheme on the partial set of 441 characters, the 100 most frequently used characters; randomly chosen "rare" characters and the complete set of characters are presented.

Some of the specific research directions being currently investigated are the use of other feature-based classification algorithms such as DTW and RBFNN, and higher-level structural features in addition to sample points. Efforts are also underway to verify the assumptions and techniques described for larger datasets, and extend them for writer-independent recognition.

8. References

[1] Niranjan Joshi, G. Sita, A. G. Ramakrishnan and Sriganesh Madhvanath. Comparison of Elastic Matching Algorithms for Online Tamil Handwritten Character Recognition, *Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition (IWFHR-9),* , 444.449 October 2004

[2] Deepu Vijayasenan and Sriganesh Madhvanath. Principal Component Analysis for Online Handwritten Character Recognition, *Proceedings of the 17th Intl Conf. Pattern Recognition (ICPR 2004),* 2:327-330August 2004

[3] C. C.Tappert, C. Y.Suen, and T. Wakahara. The state of art in online handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(8):787.807, August1990

[4] S. Jaeger, C. L. Liu, and M. Nakagawa. The State of Art in Japanese Online Handwriting Recognition Compared to Techniques in Western Handwriting Recognition. *International Journal on Document Analysis and Recognition*, 6:75.88, July 2003

[5] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons Inc, New York, 2000