Two Schemas for Online Character Recognition of Telugu script based on Support Vector Machines

Rajkumar.J, Mariraja K., Kanakapriya,K., Nishanthini, S., Chakravarthy, V.S.,

Indian Institute of Technology, Madras
raji.iitmadras@gmail.com, mari.krish@gmail.com, kanakapriya@gmail.com,
nishmailsu@gmail.com, srinivasa.chakravarthy@gmail.com

Abstract

We present two schemas for online recognition of Telugu characters, involving elaborate multi-classifier architectures. Considering the three-tier vertical organization of a typical Telugu character, we divide the stroke set into 4 subclasses primarily based on their vertical position. Stroke level recognition is based on a bank of Support Vector Machines (SVMs), with a separate SVM trained on each of these classes. Character recognition for Schema 1 is based on a Ternary Search Tree (TST), while for Schema 2 it is based on a SVM. The two schemas yielded overall stroke recognition performances of 89.59% and 96.69% respectively surpassing some of the recent online recognition performance results related to Telugu script reported in literature. The schemas yield character-level recognition performances of 90.55% and 96.42% respectively.

1. Introduction

There has been a recent spurt in development and usage of Indic computing technology in India (Govindaraju and Setlur, 2009). Such a technology enables entry of a large number of local language users into the world of computing. In the recent years, there were several efforts to develop online handwritten character recognition (OHCR) engines for Indic scripts. A system for recognition of Devnagari and Telugu scripts using Support Vector Machines (SVMs) was described in (Swethalakshmi et al 2006). A similar SVM-based recognition system for Telugu was presented by Jayaraman et al (2007). A system for recognition of Tamil script based on analysis of shape features was described by Aparna et al (2004). Joshi et al (2004) explored an approach based on elastic matching for Tamil character recognition. A multi-classifier system evolved by genetic algorithms for recognition of Devnagari script was described in (Jitendra and Chakravarthy 2008).A system for OHCR of Tamil and Telugu characters based on elastic matching was proposed in (Prashanth et al 2007). Jagdeesh Babu et al (2007) present a Hidden Markov Model (HMM) based OHCR system for Telugu, at symbol level and not at character level.

In this paper we develop an OHCR system using SVMs for Telugu script. Telugu is a language spoken in the southern part of India. The language consists of 16 vowels and 35 consonants that can combine to form about 10000 composite characters. Defining a stroke as what is written between touch-down and lift-off of the pen, we found 235 unique strokes in Telugu script. However, there is no finality to this list and is likely to grow as more writing styles are included.

The strategy adopted in this paper for recognition of handwritten Telugu characters is as follows. A typical Telugu character is composed of multiple strokes. The script is written from left to right, with major part of a typical character written on top of a baseline, analogous to English script. Considering the large number of stroke classes, it is reasonable to look for efficient ways of preclassification of strokes. A Telugu character can be divided into three horizontal tiers, a feature that serves as a natural basis for stroke preclassification. SVMs are used to classify the preclassified strokes. The strokes thus recognized are combined to identify the character using two different methods – one based on a Ternary Search Tree (TST) and the second based on SVMs. Thus, two schemas for Telugu OHCR are presented in this paper, and the performance levels compared with results reported in literature.

The paper is outlined as follows: Section 2 describes the Telugu stroke data, annotation details, and the steps of preclassification, stroke classification and character recognition. Performance results of the

978-0-7695-4774-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ICFHR.2012.286

two schemas are presented in Section 3 and the work done is discussed in the final section.

2. Components of an OHCR system for Telugu

2.1. Data

Character-level data was collected from 100 writers. The characters were chosen such that they include all the 235 strokes in the script. This character set consisted of 210 characters. The writers wrote one character at a time, using only pre-specified strokes, executed in a fixed order. This fixed order is maintained only for ease of annotation. The order restriction is relaxed for testing. Such constrained acquisition of data rendered the subsequent annotation effort relatively easy. Two thirds of this data is used for training, and the remaining for testing.

2.2 Preprocessing

Online handwriting data is collected and represented as strokes. The data captured includes co-ordinates of the trajectory of the stroke; pen pressure is not recorded. Preprocessing the stroke data before feature extraction reduces some of the variability in handwritten data.

2.2.1 Normalization Normalization reduces the variability in the size of the letters and in writing style of the user. The bounding box of the character of which the stroke is a part is first determined. The x- and y-dimensions of the stroke are divided by the height of this block; this preserves the relative size of strokes.

2.2.2 Smoothening Noise due to erratic hand motion and limited precision of the digitizing process is removed by smoothening. A Gaussian filter, with a σ of 3 and a window size of 21, is used for this purpose.

2.2.3 Interpolation: The number of points in each stroke is adjusted by interpolation such that all strokes have the same number of points. The raw x,y values of the strokes are used to construct fixed-dimensional feature vectors which will be used for further processing of strokes.

2.3 Stroke Pre-classification

As mentioned above, a typical Telugu character can be divided vertically into three tiers: top, middle and bottom. (Even four tiers are possible in extreme cases but such characters are rare.) All tiers may not be populated in a given character; the middle tier however is always present by default. Each character has a baseline stroke and usually one or more attached strokes at the top, bottom and side of the base stroke. Also there can be a lot of size variation in the strokes, with some of the strokes having very few constituent points for proper identification. Preclassification for Telugu seeks to classify the strokes in a character into four categories: 1) main stroke, 2) baseline auxiliary, 3) top stroke and 4) bottom stroke (fig. 1).

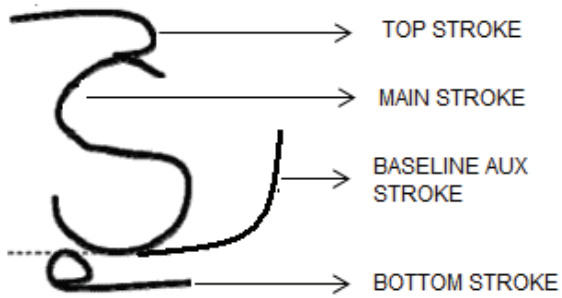

Figure 1. Tier-based Pre-Classification of a Telugu Character

Out of the total number of 235 strokes, there are 149 main strokes, 19 baseline strokes, 26 top strokes and 41 bottom strokes. The first stroke that is written in a character is assumed to be the main stroke, which is almost always true, but baseline auxiliary, bottom and top strokes can be written in any order. So to classify baseline auxiliary, bottom and top strokes, two methods are adopted.

Method 1:
The baseline auxiliary stroke is identified by a histogram of the y-co-ordinates of the stroke. The baseline stroke is the stroke closest to the interval with the least number of points. The top and bottom strokes are identified by considering their average vertical displacement from the horizontal line.

Method 2:
This method is based on a SVM. A feature vector is constructed by concatenating the main stroke and with the stroke that is being preclassified. Both strokes are resampled as in Section 2.2.3, so that either stroke has only 16 points. Such feature vectors are used to train a SVM-based pre-classifier.

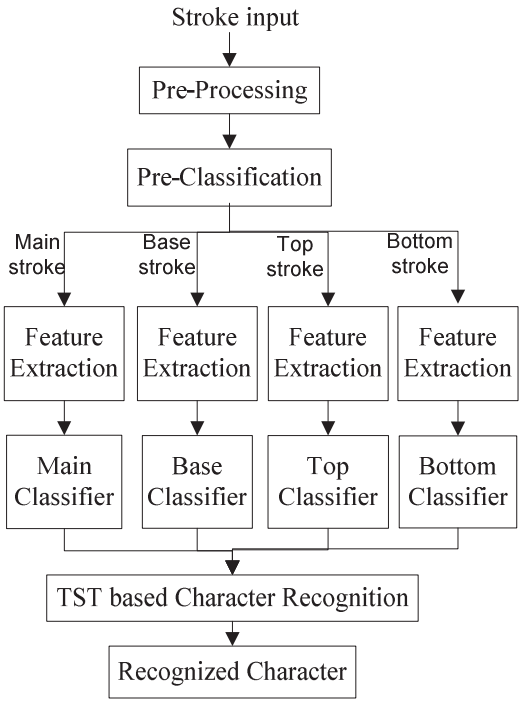

Figure 2. HWR Schema 1

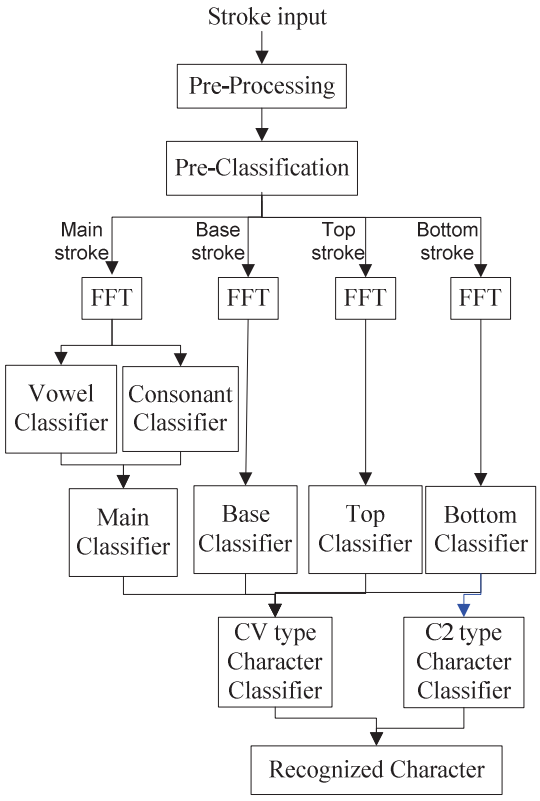

Figure 3. HWR Schema 2

2.3 Feature Extraction

Once the strokes are pre-classified, feature vectors for stroke recognition are constructed. Different types of features are explored for best classification results. The following features are considered: 1) X and Y coordinate points, 2) Fourier transforms, (64 points for X, and 64 points for Y, yielding a 128 dimensional feature vector) 3) Hilbert transform (King, 2009) - logarithm of the spectral density, (64 points for X, and 64 points for Y, yielding a 128 dimensional feature vector) and 4) Wavelet features - the level is fixed to one, varied the families (Daubechies, Haar and Mexican hat) and wavelet type (orthogonal and biorthogonal). Better results are obtained in Daubechies family and orthogonal wavelets, which is reported in this study (32 points for X, and 32 points for Y, yielding a 64 dimensional feature vector) .

2.4 Stroke Recognition using SVMs

The features extracted from baseline auxiliary, bottom and top stroke preclasses are given to SVMs with Gaussian kernel to recognize the specific stroke. Since the main preclass is larger than the other 3, we handle this preclass differently. Two methods are adopted to classify main stroke.

Method 1

The feature vector extracted from main stroke is recognized using a single SVM with Gaussian kernel (fig. 2).

Method 2

This method uses 3 SVMs for classifying the main stroke. The feature vector extracted from the main stroke is passed to two different SVMs: Vowel classifier and Consonant classifier. Output vectors of the Vowel and Consonant SVM classifiers are concatenated (i.e. N classes of from Vowel classifier and M classes of Consonant classifier are combined to form N+M dimensional vector) and passed to third SVM for classifying the main stroke (fig. 3).

2.5. Character Recognition

Method 1: Based on Ternary Search Trees

Ternary Search Tree is used for recognizing characters. Ternary search trees (TST) combine attributes of binary search trees and digital search trees. Like binary search trees, they are space efficient, though each node has three children, rather than two. A search compares the current character in

the search string with the character at the node. If the search character is less, the search goes to the left child; if the search character is greater, the search goes to the right child. When the search character is equal, though, the search goes to the middle child, and proceeds to the next character in the search string.

TST is a simple and efficient data structure for identifying an entity, given a set of constituent units. It natively supports partial match search (PMS) and near-neighbor search (NNS) algorithms. Search has a time complexity of the order of ln(n + k), where n is the number of nodes in the TST and k is the length of the string being searched.

A generalized set of rules, elucidating the possible combinations of strokes that result in a code (ISCII/Unicode) is specified. The set of rules can be grouped in a number of ways:
- One TST from all possible rules for the character stroke mapping of the script. For Telugu Aksharas of type (V, CV & CCV) this is about 19,383 rules.
- One TST from rules corresponding to V and CV for just main, top and bottom strokes. Bottom strokes, which generally represent consonant modifiers (CMs, or C2), have a separate TST and simple mapping would combine the outputs from the two TSTs. This would reduce the Telugu rules to about 944. This method is followed in the present paper.

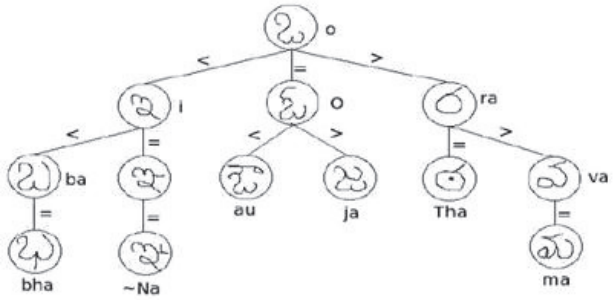

Figure 4. TST representation of Telugu Character

Method 2: Based on SVMs

Information pertaining to C2 is always found in bottom strokes in Telugu characters. Two SVMs - a CV type and a C2 type classifier – are used to recognize the character. Main stroke, base stroke, top stroke and bottom stroke neglecting C2 type strokes (there are 215 of them) are taken as a feature vector to recognize consonant and vowel character. This feature vector is a binary vector of dimension 215: a component is 1 (0) if the corresponding stroke is present (absent). In cases where there are multiple copies of the stroke, the component is set to the number of copies. If a C2 type bottom stroke (there are 18 of them) is present, then the corresponding C2 type bottom stroke-based feature vector is constructed and passed to the C2-type character classifier (fig. 3).

3. Results

The data collected from 100 writers is split into training (roughly 2/3rd) and testing (remaining) data. The exact numbers of strokes of various categories are given in Table 3.1. Pre-classification is performed for analysis of displacement of strokes with respect to the baseline and also SVM based classification. We present results on the classification performance of different features used for four categories of strokes by their respective SVM-based classifiers is given in Tables 3.2 to 3.6. Finally, we present character-level performance on the complete data set character-level performance using the two proposed Schemas (Table 3.7).

Table 3.1. Pre-classification Performance:

Method	No. chars	Classified chars	Accuracy (%)
Stroke displ. w.r.t. baseline	18588	18002	96.84
SVM based Classifier	18435	18353	99.55

Table 3.2 Main stroke classifier Performance:

Feature/ method	Total # Strokes	Classified Strokes	Accuracy (%)
XY coordinates	18588	17323	93.19
FFT only	6793	6434	94.72
Hilbert transform	6793	6225	91.64
Wavelet transform	6793	6222	91.59
FFT with Consonant & vowel classifier	20502	20148	98.27

Table 3.3 Base stroke classifier Performance:

Feature	Total no of Strokes	Classified Strokes	Performance (%)
XY coordinates	3299	3054	92.57
FFT	1279	1262	98.67
Hilbert transform	1279	1254	98.04
Wavelet transform	1279	1254	98.04

Table 3.4 Bottom stroke classifier Performance:

Feature	Total no of Strokes	Classified Strokes	Performance (%)
XY coordinates	3537	3230	91.32
FFT	1408	1237	87.85
Hilbert transform	1408	1194	84.80
Wavelet transform	1408	1191	84.58

Table 3.5 Top stroke classifier Performance:

Feature	Total no of Strokes	Classified Strokes	Performance (%)
XY coordinates	7238	5656	78.14
FFT	2598	2539	97.72
Hilbert transform	2598	2516	96.84
Wavelet transform	2598	2519	96.95

Table 3.6 Overall Stroke Performances:

Feature	Performance (%)
XY coordinates	89.59
FFT	94.74
Hilbert transform	92.83
Wavelet transform	92.79
FFT with CV type classifier in main stroke	96.69

Table 3.7 Overall Character-level Performance:

Method	No of strokes	Classified chars	Char level Perf (%)
Schema 1- TTS based (fig. 2)	18588	16831	90.55
Schema – 2: all SVM based (fig. 3)	14037	13535	96.42

4. Discussion:

We present two schemas for online recognition of Telugu characters. A few crucial differences between the two schemas seem to make a difference in the final performance. In schema 1, X-Y features are used, whereas FFT features are used in schema 2. In schema 1, character recognition is done using a TST, while schema 2 uses an SVM. In Schema 1, pre-classification is based on horizontal and vertical position of the strokes, whereas in schema 2, pre-classification is done using a SVM. The main stroke recognition in schema 2 uses a more elaborate method than that used in schema 1, and achieves higher performance. The proposed schemas are compared to similar systems developed for online Telugu recognition in the past. The system proposed by Jayaraman et al (2007), which is also based on tier-based preclassification of Telugu strokes, reports an overall stroke recognition accuracy of 86.9%. In comparison, in the proposed system, in schema 1, the overall stroke recognition accuracy is 89.59% (Table 3.6) depending on the type of features used. Schema 2, with its more elaborate processing of main stroke, and use of FFT features, gives an overall stroke recognition accuracy of 96.69%. Our approach compares favorably with an even older SVM-based system for online Telugu stroke recognition which reports a best performance of 82.96% at stroke level (Swethalakshmi et al 2006). Character level recognition accuracy of Schema 1 is 90.55% while that of Schema 2 is 96.42%. Thus a modular strategy seems to be critical to design efficient systems for Telugu online character recognition. Typically character recognition performance is considerably lower than stroke recognition performance. But in the proposed methods (Schemas 1 and 2), character-level performance is only slightly higher than stroke-level performance for two reasons: 1) rules corresponding

to imperfect combinations of strokes that uniquely match with a character are incorporated in the TST, and robustness against small deviations is built into the SVM-based classifier, 2) the character-level performance is evaluated using edit distance between the Unicode strings of the actual and desired characters. A shortcoming of the proposed technique is that it assumes perfect segmentation of words into characters. Our future efforts will be directed towards overcoming this constraint.

5. References:

1. V. Govindaraju, S. Setlur (Eds), Guide to OCR for Indic scripts, Advances in Pattern Recognition series, Springer, 2009.
2. H. Swethalakshmi, A. Jayaraman, V. S. Chakravarthy, C. Chandra Sekhar, "Online Handwritten Character Recognition of Devanagari and Telugu Characters using Support Vector Machines", 10th International Workshop on Frontiers in Handwriting Recognition, La Baule, France, October 23-26, 2006.
3. H. Swethalakshmi, C. Chandra Sekhar, V. Srinivasa Chakravarthy: Spatiostructural Features for Recognition of Online Handwritten Characters in Devanagari and Tamil Scripts, ICANN(2)2007: 230-239.
4. Jayaraman, A., Chandra Sekhar, C., Chakravarthy, V. S., "Modular approach to recognition of strokes in Telugu script", Proceedings of the Ninth International Conference on Document Analysis and Recognition, September 23-26, 2007, pp 501-505.
5. Prasanth L, Jagadeesh Babu V, Raghunath Sharma R, Prabhakara Rao G.V., Dinesh M., Elastic Matching of Online Handwritten Tamil and Telugu Scripts Using Local Features, ICDAR'2007, 23-26 September 2007, Curitiba, Brazil.
6. Jagadeesh Babu V, Prasanth L, Raghunath Sharma R, Prabhakara Rao G.V., Bharath A., HMM-based Online Handwriting Recognition System for Telugu Symbols, ICDAR'2007, 23-26 September 2007, Curitiba, Brazil.
7. K.H. Aparna, Vidhya Subramanian, M. Kasirajan, G. Vijay Prakash, V.S. Chakravarthy, Sriganesh Madhvanath, "Online Handwriting Recognition for Tamil," Ninth International Workshop on Frontiers in Handwriting Recognition (IWFHR-9 2004), Kokubunji, Tokyo, Japan, October, 2004.
8. N. Joshi, G. Sita, A. G. Ramakrishnan, and S. Madhvanath, "Comparison of elastic matching algorithms for online Tamil handwritten character recognition," in International Workshop on Frontiers in Handwriting Recognition, (Tokyo), pp. 444– 449, October 26-29 2004.
9. Jitendra Kumar, V.S. Chakravarthy, Designing an optimal Classifier Ensemble for online character recognition using Genetic Algorithms, Proc. of 11th International conference on Frontiers in Handwriting Recognition, Montreal, Quebec, Canada, Aug. 19-21, 2008.
10. King, Frederick W. (2009), Hilbert Transforms, 2, Cambridge: Cambridge University Press, pp. 453, ISBN 978-0-521-51720-1.