A Two Stage Recognition Scheme for Handwritten Tamil Characters

U. Bhattacharya, S. K. Ghosh and S. K. Parui
Computer Vision and Pattern Recognition Unit,
Indian Statistical Institute, Kolkata, India
ujjwal@isical.ac.in, suman_t@isical.ac.in, swapan@isical.ac.in

Abstract

India is a multilingual multiscript country with more than 18 languages and 10 different major scripts. Not enough research work towards recognition of handwritten characters of these Indian scripts has been done. Tamil, an official as well as popular script of the southern part of India, Singapore, Malaysia, and Sri Lanka has a large character set which includes many compound characters. Only a few works towards handwriting recognition of this large character set has been reported in the literature. Recently, HP Labs India developed a database of handwritten Tamil characters. In the present paper, we describe an off-line recognition approach based on this database. The proposed method consists of two stages. In the first stage, we apply an unsupervised clustering method to create a smaller number of groups of handwritten Tamil character classes. In the second stage, we consider a supervised classification technique in each of these smaller groups for final recognition. The features considered in the two stages are different. The proposed two-stage recognition scheme provided acceptable classification accuracies on both the training and test sets of the present database.

1 Introduction

Off-line recognition of handwritten characters has been studied well in the literature as far as Latin and a few other scripts of the developed nations are concerned. Surveys of related works are found in [1, 2, 3]. However, there has not been much progress towards recognition of handwritten characters of Indian scripts. On the other hand, such recognition problem for an Indian script is different in nature because of the size of its alphabet and the similarities between different characters of an Indian alphabet. Also, unlike in English script, the alphabet of an Indian script has a large number of compound characters formed by both vowel-consonant and consonant-consonant combinations. Hence, the problem of handwritten character recognition of an Indian script needs more attention.

A few existing studies for off-line recognition of handwritten characters of Indian scripts include [4] for Devanagari, [5, 6] for Bangla, [7] for Telugu, [8] for Tamil and [9, 10] for Oriya. Most of these works are based on small databases collected in laboratory environments. However, a few recent research works on Bangla handwriting recognition are based on large databases of handwritten [11, 12, 13] character samples.

A major obstacle to serious research work on handwriting recognition of an Indian script is the non-availability of standard databases for training and testing purposes. However, recently HP Lab India has developed a database, called hpl-tamil-iso-char, of handwritten samples of 156 different Tamil characters [14]. This database is available freely for research purpose. Although the data was collected using HP Tablet PCs and is in standard UNIPEN format, an off-line version of the same set of samples is also downloadable. In the present report, we have studied an off-line recognition strategy for 156 class handwritten isolated Tamil characters.

In the first stage of the proposed two-stage recognition scheme, we classify an input character into one of a small number of groups. In the second stage, we use a bank of multilayer perceptron (MLP) classifiers, each corresponding to one group of character classes. The groups of character classes used in the first stage are overlapping, that is, samples from one character class usually fall into more than one groups. Certain transition feature values computed from the training samples are used by the unsupervised K-means clustering algorithm to obtain these groups of the first stage. The value of K and the valid character classes in each group are determined empirically.

In the second stage, we use chain code histogram features computed from the contour of the input character. A distinct MLP classifier is trained for each of the above groups. The final recognition accuracy provided by this two stage approach is significantly better than a single stage method using the same chain code features along with an MLP classifier.

The rest of this article is organized as follows. Section 2 presents a brief description of Tamil script and the present database of Tamil handwritten isolated characters. The proposed two-stage recognition methodology is described in Section 3. Experimental results are provided in Section 4.

Figure 1. Basic Tamil characters (a) **Vowels;** (b) **Consonants;** (c) **Grantha;** (d) **Aytam.**

Section 5 concludes the article.

2 Handwritten database for Tamil characters

2.1 Tamil script

Tamil, a member of the Dravidian language family, is spoken by around 52 million people of the Indian subcontinent. Its script, like other Indic scripts, is known to be evolved from the ancient Brahmi script and written in a left to right fashion. The Tamil script is syllabic but not alphabetic. There are twelve vowels (Fig.1(a)) and eighteen consonants (Fig.1(b)). The modern script of Tamil also consists of another five consonants (Fig.1(c)), called *grantha* letters and these are used to write consonants borrowed from Sanskrit. Thus, this script has 35 basic characters. Also, there is another character, called *aytam* (Fig.1(d)), which is classified in Tamil grammar as being neither a consonant nor a vowel.

In addition to the above isolated characters, often a consonant or a cluster of two or more consonants combines with a vowel causing a modified shape of the vowel, called vowel diacritic. Although this indicates the presence of a large number of Tamil characters, the modern Tamil language does not use many of these combinations and only 156 characters including independent vowels, consonants and their combinations are presently used for writing in Tamil.

2.2 hpl-tamil-iso-char database

The dataset hpl-tamil-iso-char-online-1.0, developed by HP Lab India [14], is used for the present work. This database contains samples of the 156 character classes collected from different writers using a TabletPC application. Each writer contributed 2 to 10 samples per class.

There are approximately 500 samples in most of the classes. However, there are a few classes with fewer samples. In a very few classes the number of samples is approximately 275. The samples were written by native Tamil writers including school children, university graduates, and adults from different cities of South India.

The whole database is divided into training and test sets. The training dataset contains approximately 300 isolated samples in each class save for a few classes only. The rest of the samples form the test dataset.

A simple piecewise linear interpolation with a constant thickening factor has been used on the online data to generate offline image versions of these samples. These are provided as bi-level TIFF images. In the present recognition work, we have used these latter offline samples.

3 Proposed recognition scheme

Usually, a single stage recognition system provides acceptable accuracies in relatively small class problems. However, when the number of underlying classes is very large, the same recognition scheme fails to provide similar accuracies. One common approach to this problem is to use a multistage scheme which considers fewer classes in each of its stages.

To solve the present 156 class recognition problem, we use a two stage recognition scheme. Details of both the stages are given below.

3.1 Preprocessing

First, we apply thickening of the input image. This is accomplished by blackening all the 8-neighbours of each of its object pixels. Next, linear size normalization is applied to convert its row and column numbers into nearest multiples of 7. In Fig. 2 an input character image and the same after its preprocessing are shown.

(a) (b) (c)

Figure 2. Preprocessing (a) **an input character image (**56×87**);** (b) **after its thickening (**58×89**);** (c) **size normalized (**56×91**) thick image.**

3.2 First stage: determination of smaller groups

In the first stage of the proposed scheme, an input character is identified into one of a few smaller groups of characters. These groups are determined by unsupervised clustering of training samples. In the present work, we use K-means clustering technique for the above purpose.

3.2.1 Feature extraction for the first stage

The matrix corresponding to a character image after its preprocessing is divided into 7×7 equal blocks. We scan each of the resulting 49 blocks along both horizontal and vertical directions. In each such scan, we count the number of transitions (white to black and *vice-versa*). Thus, we obtain a $2 \times 49 = 98$ component transition feature vector corresponding to each character sample.

3.2.2 Unsupervised clustering

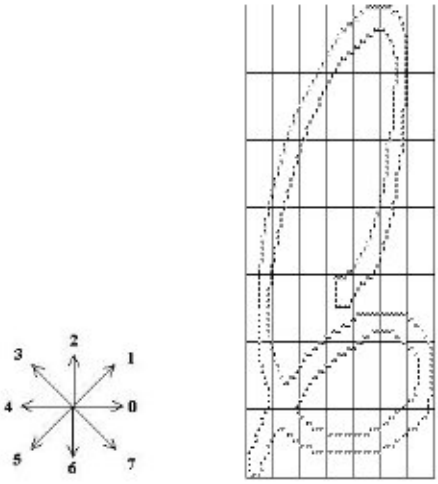
The above transition feature values are computed for the training samples of all of the 156 classes. These feature values are pooled together and standard $K-$means algorithm is applied on the combined set until the error decreases to a preassigned small quantity. Each of the resulting group may consist of samples from one or more character classes. However, if the number of training samples from a character class falling in a particular group is very small (say, less than or equal to 2.5% of the available training samples in that character class), we ignore those training samples during further computation of the groups. The prototype of each such group is obtained by averaging feature values of all the valid training samples falling in the particular group. The groups are identified during further processing by these prototypes.

3.3 Second stage: group wise classification

Different samples from the same character class may fall into different groups obtained during the first stage. On the other hand, a group may consist of one or more character classes. If a group consists of a single class only, then it does not require any processing during the second stage. Otherwise, if a group consists of valid samples from more than one (say, N) classes, then during the second stage of the proposed scheme, we use a distinct MLP classifier with N output nodes for such a group. The MLP classifier uses chain code histogram feature [15] to recognize the character class of each sample identified into the group during the first stage.

3.3.1 Chain code histogram features

During the second stage of the proposed scheme, chain code histogram features of an input character image is computed from its contour representation. The chain code representation of the character contour is obtained by using Freeman's chain code (shown in Fig. 3(a)). This chain code representation of the character contour is divided into 7 equal horizontal and vertical strips resulting into 49 equal rectangular blocks (Fig. 3(b)). In each block, a local histogram of the chain codes is calculated. Since the directions along the skeleton or contour should be effectively



(a) (b)

Figure 3. (a) **Scheme for chaincodes;** (b) **chain code representation of the character shape in Fig. 2**(c)**.**

quantized into one of 4 possible values, *viz.* 0 or 4, 1 or 5, 2 or 6 and 3 or 7, the histogram of each block has four components. Finally, the above 7×7 blocks is down sampled into 4×4 blocks using Gaussian filter. Thus, the feature vector used in the second stage has $4 \times 4 \times 4 = 64$ components. For size normalization, each component of this feature is divided by the product of the height and width of each block.

3.3.2 MLP classifiers

Distinct MLP classifiers have been chosen as the classifiers for each group of the second and final stage. There are 64 input nodes in each such MLP classifier. Number of output nodes of each MLP classifier is equal to the number of classes belonging to the respective group. Since it is difficult to estimate the optimal size(s) of the hidden layer(s) of each classifier, we experimented with several different choices of this size in each case and classification results are reported in the next section corresponding to the best situation. The well known backpropagation (BP) algorithm [16] is used for the training of MLP classifiers. However, in many applications like the present one, the proper training of an MLP largely depends on the choice of the parameter (learning rate and momentum factor) values of BP algorithm and also it often converges too slowly. There exist a number of modified BP algorithms which take care of these problems of the original BP algorithm. In the present classification task, we considered a modified BP algorithm [17] using self-adaptive learning rate values.

Finally, the strategic selection of the point of termination of the iterative learning of BP algorithm is another important issue. Usually, during the initial stages of training of an MLP using BP training algorithm, it gradually decreases the system error [16] on both the training and test sets. However, after a certain amount of training, this error further decreases on the training set while it starts increasing on the test set. The point of time when the error on the test set increases for at least three consecutive sweeps for the first instance is noted and the weight values before the error starts increasing, are considered for reporting classification percentages.

cluster NO	Shape of the Character Class
2	உ ஊ ட ப ம வ டீ ட் ப் ம் ல்
3	ஆ ஊ ஏ ஒ ஓ ஞ த ந ம ர ழ கு ரு ழு ஞ ஙு ஆ பூ யூ ரூ வூ ழூ ஞ ஸ்ரீ ஞ் ந் ர்
4	உ ஊ ட ப ம வ டீ ட் ப் ம்
5	ஏ ணை ல ளை ன ஸ லி ஜீ ண்ணா ணை ஜ ஸு ஜு ஸு ஜ
7	அ ஆ இ ஊ ஜ க ச ழ ஜ சி சீ சு வு து நு று னு தூ ச த் ழ்
10	சி ஞி தி நி ரி ழீ றீ ஷீ ஜீ ழி றி ஷி ஜி சீ ஞீ தீ நீ ரீ ஷு ஜ ஹு ஷு ஜு ஹு று
19	ஈ ஏ ணை ல ள ன ர ஸ ன ர

Figure 4. A few groups of character classes formed during the first stage of the proposed scheme.

4 Experimental results

We simulated the $K-$means algorithm with different choices of K for unsupervised clustering of the first stage. Based on these simulations we observed that $K = 25$ is an acceptable choice in terms of different factors such as number of character classes in each cluster and classification error between clusters. With this choice of K, the maximum number of classes in a group is 28 while the minimum number is 1 (the character ◌ forms a singleton group). There are 10 - 20 character classes in 6 groups and 20 - 25 character classes in 14 groups. Also, the resulting misclassification errors between groups are 3.85% and 5.97% respectively on the training and test sets. In this first stage, we considered only those character classes in a group for which at least 2.5% of the total training samples belong to the group. A few such groups of character classes obtained during the first stage of the present scheme are shown in Fig. 4. Character samples belonging to a group which does not contain the corresponding character classes are considered to be misclassified samples. Percentages of training and test samples misclassified in each of these groups of the first stage are shown in Table 1.

Table 1. Misclassification figures* during the first stage

Cluster No.	Misclassification % Training	Test	Cluster No.	Misclassification % Training	Test
0	0.18	0.19	13	0.18	0.31
1	0	0	14	0.25	0.32
2	0.06	0.06	15	0.18	0.33
3	0.13	0.22	16	0.17	0.17
4	0.08	0.12	17	0.13	0.39
5	0.17	0.33	18	0.18	0.29
6	0.16	0.2	19	0.22	0.28
7	0.21	0.31	20	0.15	0.32
8	0.14	0.19	21	0.21	0.23
9	0.15	0.29	22	0.14	0.19
10	0.13	0.26	23	0.15	0.25
11	0.15	0.19	24	0.19	0.22
12	0.17	0.31			

⋆ The percentages are w.r.t. the entire training and test sets

In the second stage, we consider only 24 groups which include more than one character class. Here, we do not consider those training or test samples which are misclassified during the first stage. Also, in this stage, better recognition accuracies are achieved for smaller groups while the same is comparatively poor for larger groups. In the second stage, the best recognition performance is achieved for group 2 consisting of 11 character classes. The corresponding misclassification percentages for training and test sets are respectively 0.02% and 0.02%. On the other hand, the worst recognition performance is achieved for group 3 consisting of 27 classes. The corresponding misclassification percentages for training and test sets are respectively 4.65% and 6.2%. Compositions of these two groups are shown in Fig. 4. Also, details of the recognition results of the second stage are shown in Table 2.

Finally, the total number of training and test samples misclassified in the first stage are respectively 1951 (out of 50,683) and 1608 (out of 26,926). Thus the corresponding misclassification percentages of the first stage are respectively 3.85% and 5.97%. The numbers of training and test samples used in the second stage are 48,732 and 25,318. Numbers of misclassifications of the second stage are respectively 1711 and 1177. Thus, the respective percentages are 3.51% and 4.65%. The overall recognition accuracies (considering both the stages) are 92.77% and 89.66% on the training and test sets respectively. The best accuracy (off-line) on the same test set reported in the interna-

Table 2. Misclassification figures* during the second stage

Cluster No	Misclassification(%) Training	Test	Cluster No	Misclassification(%) Training	Test
0	0.12	0.14	13	0.19	0.24
1	0.00	0.00	14	0.17	0.18
2	0.02	0.02	15	0.14	0.18
3	0.26	0.27	16	0.13	0.14
4	0.07	0.1	17	0.14	0.29
5	0.2	0.29	18	0.17	0.22
6	0.11	0.12	19	0.16	0.25
7	0.2	0.26	20	0.18	0.31
8	0.11	0.13	21	0.09	0.11
9	0.18	0.28	22	0.13	0.15
10	0.12	0.15	23	0.14	0.19
11	0.14	0.18	24	0.18	0.22
12	0.16	0.23			

⋆ The percentages are w.r.t. the entire training and test sets

tional competition on handwritten Tamil character recognition (IWFHR10) was 87.66%.

5 Conclusions

In the present work, we considered a two stage recognition scheme for handwritten Tamil characters. We obtained simulation results of the proposed scheme on a database of handwritten Tamil characters developed by HP Labs India and freely available on the WWW. The recognition accuracies obtained are 92.77% and 89.66% respectively on the training and test sets.

We also simulated a single stage method using the same chain code histogram feature and an MLP classifier towards this recognition problem. However, we obtained only 74.65% and 69.48% recognition accuracies respectively on the training and test sets.

Acknowledgment

The authors would like to acknowledge the partial financial support of the Department of Information Technology, Govt. of India towards this research work.

References

[1] O. D. Trier, A. K. Jain, and T. Taxt, Feature extraction methods for character recognition - a survey, *Pattern Recognition*, Vol. 29, pp. 641-662, 1996.

[2] R. Plamondon, S. N. Srihari, On-line and off-line handwriting recognition: a comprehensive survey, *IEEE Trans. PAMI*, Vol. 22, pp. 63-84, 2000.

[3] N. Arica, F. Yarman-Vural, An overview of character recognition focused on off-line handwriting, *IEEE Trans, SMC, Part C: Applications and Reviews*, Vol. 31, pp. 216-233, 2001.

[4] K. R. Ramakrishnan, S. H. Srinivasan, S. Bhagavathy, The independent components of characters are 'Strokes', *Proc. of the* 5^{th} *ICDAR*, pp. 414-417, 1999.

[5] F. R. Rahman, R. Rahman, M. C. Fairhurst, Recognition of handwritten Bengali characters: a novel multistage approach, *Pattern Recognition*, Vol. 35, pp. 997-1006, 2002.

[6] A. Datta, S. Chaudhury, Bengali alpha-numeric character recognition using curvature features, *Pattern Recognition*, Vol. 26, pp. 1757-1770, 1993.

[7] M. B. Sukhaswami, P. Seetharamulu, A. K. Pujari, Recognition of Telugu characters using neural networks, *Int. J. Neural Syst.*, Vol. 6, pp. 317-357, 1995.

[8] R. M. Suresh, L. Ganesan, Recognition of printed and handwritten Tamil characters using fuzzy approach, *Proc. of* 6^{th} *ICCIMA*, pp. 286-291, 2005.

[9] S. Mohanti, Pattern recognition in alphabets of Oriya Language using Kohonen Neural Network, IJPRAI, Vol. 12, pp. 1007-1015, 1998.

[10] T. K. Bhowmick, S. K. Parui, U. Bhattacharya, B. Shaw, An HMM based recognition scheme for handwritten Oriya numerals, *Proc. of the* 9^{th} *Int. Conf. on Information Technology*, IEEE Computer Society Press, pp. 105-110, 2006.

[11] U. Pal, A. Belaid and B. B. Chaudhuri, A complete system for Bangla handwritten numeral recognition, *IETE Journal of Research*, Vol.52, pp.27-34, 2006.

[12] U. Bhattacharya, M. Shridhar and S. K. Parui, On recognition of handwritten Bangla characters, *Proc. of the* 5^{th} *Indian Conf. on Computer Vision, Graphics and Image Processing*, pp. 817-828, 2006.

[13] S. K. Parui, U. Bhattacharya, S. K. Ghosh, Recognition of handwritten Bangla vowel modifiers, *Proc. of the* 6^{th} *International Conference on Advances in Pattern Recognition*, pp. 129-134, 2006.

[14] Isolated Handwritten Tamil Character Dataset, hpl-tamil-iso-char *http://www.hpl.hp.com/india/research/penhw/resources/tamil-iso-char.html*

[15] F. Kimura, Y. Miyake, M. Sridhar, Handwritten ZIP code recognition using lexicon free word recognition algorithm, *Proc. Int. Conf. Document Analysis and Recognition*, Vol. II, pp. 906-910, 1995.

[16] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning internal representations by error propagation, *Institute for Cognitive Science Report 8506*, San Diego: University of California, 1985.

[17] U. Bhattacharya, S. K. Parui, Self-adaptive learning rates in backpropagation algorithm improve its function approximation performance, *Proc. of the IEEE Int. Conf. on Neural Networks*, pp. 2784-2788, 1995.